

Predicting Cervical Spine Compression
and Shear in Helicopter Helmeted
Conditions Using Artificial Neural
Networks

by

Christopher A. B. Moore

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Master of Science

in

Kinesiology

Waterloo, Ontario, Canada, 2021

© Christopher A.B. Moore 2021

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Introduction: Military helicopter pilots around the globe experience a high prevalence of neck pain. The requirement for pilots to use night vision goggles (NVGs) has been linked to increases in neck pain and injury prevalence. As a result, next generation helmet designs aim to offset or mitigate NVG-related consequences on cervical spine loading. However, *in vivo* human-participant experiments are currently required to collect necessary data (e.g., electromyography) to estimate joint contact forces on the cervical spine associated with unique helmet designs. This is costly, and inefficient. Thus, a more time and resource-efficient approach is required. A digital human modelling approach wherein multi-body dynamics (MBD) models, which provide inverse dynamics, are combined with artificial neural networks (ANNs) can provide a surrogate for more costly musculoskeletal joint modeling to predict joint contact forces.

Objective: To develop ANNs to predict cervical spine compression and shear, given inputs available through MBD modelling, with enough sensitivity to differentiate between compression and shear exposures associated with different helicopter helmet designs.

Methods: ANNs with systematically varied inputs and parameters were developed to predict cervical spine compression and shear given head-trunk kinematics and C6-C7 neck joint moments, data readily available from digital human models. ANN development was driven by a previously collected and processed dataset. Motion capture and electromyography data were collected from 26 participants who performed flight-relevant reciprocal head

movements about pitch and yaw axes while donning one of four helmet configurations.

These data were input into an electromyography-driven musculoskeletal model of the neck to generate time series C6-C7 compression and shear outputs. ANNs were trained to predict the electromyography-driven model compression and shear outputs given only the head-trunk kinematics and C6-C7 moments as inputs.

Results: Rotation-specific (i.e., yaw and pitch) ANNs yielded stronger predictive performance than ANNs that generalized to both pitch and yaw axes of rotation. ANNs for pitch rotations accurately predicted peak and cumulative compression and shear outputs with an absolute error that was lower than absolute differences in joint contact forces between relevant helmet conditions. ANNs for yaw rotations were similarly successful in predicting cumulative C6-C7 compression and shear where absolute error was lower than corresponding differences between relevant helmet conditions. However, they were unable to do so for peak C6-C7 compression and shear.

Conclusions: When combined with biomechanical data readily available from digital human modeling software, use of an ANN surrogate for joint musculoskeletal modeling can permit evaluation of joint contact forces in the cervical spine associated with novel helmet design concepts during upstream design. Improved consideration of joint contact forces during a computer-aided helmet design process will assist in identifying helmet designs that reduce the biomechanical exposures of the cervical spine during helicopter flight.

Acknowledgements

First, thank you to my supervisor and mentor, Dr. Steven Fischer. You truly go above and beyond to support your students in all aspects of their graduate education. You have been instrumental in helping me grow as both a researcher and person. I am very fortunate to have had your guidance and mentorship over the years.

I am grateful for my thesis committee members, Dr. Jack Callaghan and Dr. Brian Horslen. Your insights and questions regarding my thesis project pushed me to learn and grow as a scientist and have been invaluable in strengthening this project. A second thank you to Dr. Callaghan who provided me with my first research opportunity as a co-op student in his lab several years ago.

I'd also like to thank Jeff Barrett for graciously generating data that were crucial for the completion of this project.

I have had the privilege of working alongside countless inspirational lab mates within the OBEL team. I'd especially like to thank Dan Armstrong, Aleks Budarick, Amanda Calford, Justin Davidson, Sheldon Hawley, Laura Healey, Alexander Malone, Nathalie Oomen, Sarah Remedios and Sanjay Veerasammy. Thank you for your advice, encouragement and friendship. My studies would not have been nearly as enjoyable or rewarding without you.

A huge thank you to all the people who I am fortunate to call friends. In particular, Devon, Ellen, James, Nat, Robert, Shane, TJ and Turk. Thank you for countless great memories. I look forward to many future adventures with you.

To Gabi, thank you for your unwavering support and encouragement. At the start of my university career I never imagined myself reaching this point and would not have done so without you.

To my parents, thank you for everything you have done for me. Thank you for always being there for me, for knowing when I needed a push forward and for encouraging me to do what makes me happy.

Table of Contents

AUTHOR'S DECLARATION	ii
Abstract	iii
Acknowledgements	v
List of Figures	xi
List of Tables	xix
Chapter 1 : Introduction	1
1.1 Neck Pain Among Military Helicopter Aircrew	1
1.2 Designing Interventions to Prevent Neck Pain in Helicopter Aircrew	1
1.3 Artificial Neural Networks as a Potential Tool for Virtual Helmet Design.....	5
1.4 Objective	7
1.5 Research Question.....	7
1.6 Hypothesis.....	7
Chapter 2 : Literature Review.....	8
2.1 Overview of Artificial Neural Networks.....	8
2.2 How do Artificial Neural Networks Learn?.....	10
2.2.1 Mini-Batch Gradient Descent.....	13
2.3 The Bias-Variance Dilemma (Geman et al., 1992).....	14
2.4 Over/Underfitting.....	17
2.5 Overview of the Vanishing (Hochreiter, 1998) and Exploding (Hochreiter, 1991) Gradient Problems.....	18
2.6 Hyperparameter Tuning	19

2.6.1 Tuning Architectural Parameters.....	20
2.6.2 Tuning the Learning Rate.....	21
2.7 Overview of the Adam Optimization Algorithm	23
2.8 Regularization Techniques.....	24
2.8.1 Early Stopping.....	25
2.8.2 Dropout Regularization	26
2.9 Recurrent Neural Networks.....	27
2.9.1 Long Short-Term Memory (Hochreiter & Schmidhuber, 1997a)	29
2.10 Best Practices for Machine Learning in the Field of Biomechanics (Halilaj et al., 2018).....	33
2.10.1 Feature Selection and Scaling.....	33
2.10.2 Assessing ANN Generalizability.....	34
2.11 Combining Multi-Body Dynamics Models and ANNs as a Surrogate for More Computationally Expensive Models	34
2.12 Applying an ANN to Predict Lumbar Muscle Activity (Nussbaum, Chaffin & Martin, 1995).....	36
2.13 Previous Applications of Artificial Neural Networks to Predict Intervertebral Contact Forces Within the Lumbar Spine	38
2.13.1 Hou et al., 2004.....	40
2.13.2 Hou et al., 2007.....	42
2.13.3 Parkinson & Callaghan, 2009.....	46
2.13.4 Arjmand et al., 2013	49
2.13.5 Mousavi, Sayyaadi & Arjmand, 2020	52

2.14 Summary	54
Chapter 3 : Methods.....	56
3.1 Participants	56
3.2 Motion Capture and EMG Data Collection.....	56
3.2.1 Visual Target Acquisition System.....	56
3.2.2 Experimental Design	57
3.2.3 Instrumentation.....	60
3.3 Estimating C6-C7 Compression and Shear with an EMG-Driven Model	61
3.4 Artificial Neural Network Development.....	65
3.5 Hyperparameter Tuning Stage One: Selecting ANN Inputs, Type of ANN and ANN Architecture.....	66
3.5.1 Hyperparameter Tuning Stage One Methods	66
3.5.2 Hyperparameter Tuning Stage One Results	72
3.6 Hyperparameter Tuning Stage Two: Selecting Initial Learning Rate, Dropout rate and Mini-Batch Size.....	76
3.6.1 Hyperparameter Tuning Stage Two Methods	76
3.6.2 Hyperparameter Tuning Stage Two Results.....	77
3.7 Hypothesis Testing.....	80
3.7.1 Generating Variables of Interest.....	80
Hypothesis Testing	82
3.7.2 Secondary Analysis: Mean Time-Series Error	83

3.7.3 Secondary Analysis: Temporal Error in ANN-Predicted Peak Compression and Shear	84
Chapter 4 : Results	85
4.1.1 Statistical Testing	85
4.1.2 Mean Time-Series Error	87
4.1.3 Temporal Error in Timing of ANN-Predicted Peak Compression and Shear	89
Chapter 5 : Discussion	95
5.1 ANN Performance.....	95
5.2 Comparison to Previous ANN Models	96
5.3 ANN Input Features	99
5.4 Future Directions.....	101
5.5 Limitations	102
5.6 Appropriate Application of ANNs Within a Military Helicopter Helmet Design Process	104
5.7 Novel Contributions	105
Chapter 6 : Conclusion.....	108
References.....	110
Appendix A : Results of Hyperparameter Tuning Stage Two.....	122
Appendix B : Results of Hyperparameter Tuning Stage Two	126
Appendix C : Scatterplots of ANN-Predicted vs. EMG-Driven Model C6-C7 Compression and Shear Outputs	131
Appendix D : ANN Performance Across Multiple Cycles Within a Trial	135

List of Figures

Figure 1: Exemplar helmet-NVG system including depictions of COM locations.	2
Figure 2: Schematic of a 3-layer, densely connected FF ANN with one hidden layer from Haykin (2004). The term “densely connected” refers to each node being connected to all nodes of a previous layer. Arrows represent the unidirectional flow of data from one node to another.....	9
Figure 3: Diagram of a node from Haykin (2004). Inputs, x , of a given node, k , are multiplied by weights, w , before being summed. m represents the number of input features. After the weighted inputs are summed, a bias, b , is added. The resultant value, v , is then passed through the activation function, ϕ , to generate the final output of the node, y	10
Figure 4: Visual representation of forward and backward propagation adapted from Haykin (2004).	12
Figure 5: Visual representation of bias and variance from Fortmann-Roe (2012). Blue dots represent ANN outputs relative to corresponding target values represented by the red bulls-eye.	16
Figure 6: Experimental findings from Geman, Bienenstock & Doursat, 1992 demonstrating a decrease in bias error and increase in variance error as the number of hidden nodes in an ANN is increased.	16
Figure 7: Visual representation of over/underfitting of a relationship between inputs (x) and outputs (y). The orange line represents the input-output relationship that is “learned” by the ANN. Blue circles represent input/output pairs from a training set.	17
Figure 8: Visual representation of training becoming “stuck” in a local minimum/plateau. Each arrow represents an update to network weights. In practice, the relationship between the loss function and network weights would be in a much higher dimensional space where the weight and bias of each node corresponds to one dimension within the optimization space.	22

Figure 9: 2D conceptual representation of the effects of different relative learning rates on the loss function of an ANN during training. Each arrow represents an update to weights within a network. Similar to Figure 8, this 2D figure is a simplification of what is a higher dimensional space in practice. 22

Figure 10: Figure from Kingma & Ba (2015) demonstrating the performance of ANNs employing one of five different optimization algorithms to learn a handwriting recognition problem. Adam optimization yielded the best performance with fewer training iterations and lower computational cost relative to other optimization algorithms. 23

Figure 11: Validation error curve of an ANN’s validation error over 425 training epochs adapted from Prechelt (2013). The red arrow highlights a local minimum early in the training process. The green arrow indicates the global minimum. In this example, ANN training would ideally be stopped near epoch 200. 26

Figure 12: Visual representation of dropout from Srivastava et al. (2014) of a FF neural network before and after applying dropout. Circles and arrows represent nodes and internode connections respectively. Crossed circles represent nodes that have been “dropped”. 27

Figure 13: Model of a simple RNN with one hidden layer consisting of two nodes. Blue arrows represent the flow of data from timepoint, t , while yellow arrows represent hidden node activations originating from timepoint $t - 1$ 29

Figure 14: Graphical representation of an LSTM cell from Ng, Katanforoosh & Mourri (n.d.). 30

Figure 15: Figure from Nussbaum et al. (1995) comparing the performance of a committee of ANNs to optimization models that minimized muscle activation intensity and lumbar compression (MIC) and sum of cubed muscle activation intensity (SCI). Included muscles are the right and left latissimus dorsi (R/LLD), erector spinae (R/LES), external obliques (R/LEO) and rectus abdominus (R/LRA). 37

Figure 16: mRNN diagram from Hou et al. (2004). EMG outputs were rescaled such that they had similar numerical magnitudes as the other mRNN inputs. After being rescaled, they were looped back as an input to the mRNN at one and two timesteps in the future. 41

Figure 17: Figure from Hou et al., 2004 showing mean absolute error (N) of mRNN predictions of low back lateral shear (LSF), AP shear (ASF) and compression (CMP) during symmetric and asymmetric lifting tasks. 42

Figure 18: RFNN diagram from Hou et al., 2007. Layer 1 received kinematic inputs as well as recurrent EMG feedback. Layer 2 fuzzified these inputs i.e. converted the numerical inputs in to linguistic sets. Nodes in Layer 3 calculated the products of their inputs and output the firing strength of various fuzzy rules that aimed to model the relationship between kinematics and EMG and lumbar forces. Layer 4 performed defuzzification in order to output numerical values. 43

Figure 19: Graphical representation of the fuzzification of temperature values from https://commons.wikimedia.org/wiki/File:Fuzzy_logic_temperature_en.svg. 44

Figure 20: Mean absolute error (in N) and percentage error of RFNN vs. EMG-driven model mediolateral shear (LSF), AP shear (ASF) and compression (CMP) outputs during sagittal and asymmetrical lifting tasks. Figure from Hou et al. (2007). 45

Figure 21: Comparison of RFNN (Predict) and EMG-driven model (Actual) mediolateral shear (LSF), AP shear (ASF) and compression (CMP) outputs during asymmetrical lifting. 46

Figure 22: Comparison of ANN (simulated) predictions of L4/L5 compression to EMG-assisted (Original) model outputs from Parkinson & Callaghan, 2009. The authors note that the ANN was unable to replicate peak values. 48

Figure 23: Cumulative compression as calculated by an EMG-assisted model (original) vs. FF ANN (simulated) over several lifts performed by one participant. Figure from Parkinson & Callaghan (2009). 48

Figure 24: The kinematics-driven FE model of Arjmand & Shirazi-Adl (2006) that was applied by Arjmand et al. (2013). Modelled muscles included iliocostalis lumborum pars thoracic (ICPT), longissimus thoracis pars thoracic (LGPT), multifidus (MFL), longissimus thoracis pars lumborum (LGPL), iliocostalis lumborum pars lumborum (ICPL), quadratus lumborum (QL), iliopsoas (IP), rectus abdominus (RA), external oblique (EO) and internal oblique (IO)..... 50

Figure 25: Scatter plot from Arjmand et al. (2013) showing ANN-predicted (y-axis) and kinematics-driven FE model (x-axis) outputs..... 51

Figure 26: Schematic adapted from Mousavi et al. (2020) illustrating the structure of their neuro-fuzzy approach. Aspects pertaining to prediction of lumbar compression and shear are highlighted in blue. Inputs to the first RNN were participant standing height, H, and the anteroposterior, mediolateral and inferosuperior distances (x , y , and z) of the load relative to a point on the floor on which participants were standing..... 53

Figure 27: Compression and shear outputs of the RNNs (Neuro-Fuzzy) developed by Mousavi et al. (2020), the outputs of a FF ANN (FFNN) developed in a previous study (Gholipour & Arjmand, 2016) and corresponding FE model-derived values (Target) when lifting a load. Figure from Mousavi et al (2020). 54

Figure 28: Figure, adapted from Healey (2019), demonstrating the changes in colour of VTAS LEDs when the laser is not in contact with it (A), when target is initially exposed to the laser pointer (B) and when the laser pointer has been in contact with the target for greater than 300ms (C)..... 57

Figure 29: VTAS target locations adapted from Healey (2019). “A” and “a” denote targets used to elicit yaw head movements of 70° and 35° respectively. Similarly, “B” and “b” denote targets used to elicit pitch head movements of 70° and 35°. 58

Figure 30: Reflective marker (circle) and rigid body marker cluster (square) placement on the trunk. Figure from Healey (2019)..... 61

Figure 31: Reflective marker placement on the helmet. Figure from Healey (2019). 61

Figure 32: EMG-driven model of the neck adapted from the OpenSim model of Barrett et al. (2021). (A) the model in a neutral posture with muscle elements represented in red, and ligaments in green. (B) demonstration of the pitch rotation kinematics, which involved reciprocal flexion-extension (flexion shown here). (C) the axial-twisting yaw rotation. The left image in (C) is the static portion of the trial, whereas the image on the right is taken during the accelerating phase of the movement, hence the notable activation in the sternocleidomastoid. 63

Figure 33: Overview of the groups and sub-groups of ANNs developed in the first stage of hyperparameter tuning. 69

Figure 34: Performance of LSTM-FF_{ALL} ANNs with a variety of number of hidden layers and nodes per hidden layer. 76

Figure 35: Head-trunk flexion angle (upper figure) and C6-C7 time-series compression (lower figure) during a pitch trial. The blue curve represents ANN-predicted C6-C7 compression while the yellow curve represents EMGD model C6-C7 compression outputs. Vertical dotted lines approximate where head movement cycles would be defined as beginning and ending. 82

Figure 36: Exemplar compression (bottom) and shear (top) LSTM-FF_{ALL} ANN outputs for a single pitch trial selected from the test set. These specific waveforms were selected as they are representative of mean error values calculated across all pitch trials within the test set. . 88

Figure 37: Exemplar compression (bottom) and shear (top) LSTM-FF_{ALL} ANN outputs for a single yaw trial selected from the test set. These specific waveforms were selected as they are representative of mean error values calculated across all yaw trials within the test set. 89

Figure 38: Histogram indicating proportion of head movement cycles with various magnitudes of absolute error in timing of predicted peak compression loading during pitch trials within the test set. 91

Figure 39: Histogram indicating proportion of head movement cycles with various magnitudes of absolute error in timing of predicted peak shear loading during pitch trials within the test set. 92

Figure 40: Histogram indicating proportion of head movement cycles with various levels of error in timing of predicted peak compression loading during yaw trials within the test set. 93

Figure 41: Histogram indicating proportion of head movement cycles with various levels of error in timing of predicted peak shear loading during yaw trials within the test set. 93

Figure 42: Composite error scores of dual-rotation ANNs of various types with a range of number of layers and nodes per layer. 122

Figure 43: Composite error scores of labelled-trial ANNs of various types with a range of number of layers and nodes per layer. 123

Figure 44 Composite error scores of pitch-specific ANNs of various types with a range of number of layers and nodes per layer. 124

Figure 45: Composite error scores of yaw-specific ANNs of various types with a range of number of layers and nodes per layer. 125

Figure 46: Composite error score of dual-rotation LSTM-FF ANNs with 3 hidden layers and a variety of learning rates, mini-batch sizes and dropout rates when applied to the validation set. 126

Figure 47: Composite error score of labelled-trial LSTM-FF ANNs with 3 hidden layers and a variety of learning rates, mini-batch sizes and dropout rates when applied to the validation set. Although ANNs with a learning rate of 0.1 and 0.01 were developed, they yielded composite error scores that were consistently 2 to 5 times higher than ANNs with other learning rates and were therefore excluded from the figure. 127

Figure 48: Composite error score of pitch-specific LSTM-FF ANNs with 3 hidden layers and a variety of learning rates, mini-batch sizes and dropout rates when applied to the validation

set. Although ANNs with a learning rate of 0.1 were developed, they yielded composite error scores that were consistently 2 to 5 times higher than ANNs with other learning rates and were therefore excluded from the figure. 128

Figure 49: Composite error score of yaw-specific LSTM-FF ANNs with 3 hidden layers and a variety of learning rates, mini-batch sizes and dropout rates when applied to the validation set. Although ANNs with a learning rate of 0.1 were developed, they yielded composite error scores that were consistently 2 to 5 times higher than ANNs with other learning rates and were subsequently excluded from the figure. 129

Figure 50: Scatterplot comparing C6-C7 compression outputs of the best performing LSTM-FF_{ALL} ANN and corresponding EMGD model-derived C6-C7 compression during pitch rotation trials. Each point represents data from one frame from the test dataset. 131

Figure 51: Scatterplot comparing C6-C7 compression outputs of the best performing LSTM-FF_{ALL} ANN and corresponding EMGD model-derived C6-C7 compression during yaw rotation trials. Each point represents data from one frame from the test dataset. 132

Figure 52: Scatterplot comparing C6-C7 shear outputs of the best performing LSTM-FF_{ALL} ANN and corresponding EMGD model-derived C6-C7 shear during pitch rotation trials. Each point represents data from one frame from the test dataset. 133

Figure 53: Scatterplot comparing C6-C7 shear outputs of the best performing LSTM-FF_{ALL} ANN and corresponding EMGD model-derived C6-C7 shear during yaw rotation trials. Each point represents data from one frame from the test dataset. 134

Figure 54: Absolute error of ANN-predicted C6-C7 cumulative compression vs. EMGD model-derived values across the first four head movement cycles of yaw trials within the test set. 136

Figure 55: Absolute error of ANN-predicted C6-C7 cumulative shear vs. EMGD model-derived values across the first four head movement cycles of yaw trials within the test set. Asterisks denote significant pairwise differences ($p < 0.05$). 137

Figure 56: Absolute error of ANN-predicted C6-C7 peak compression vs. EMGD model-derived values across the first four head movement cycles of yaw trials within the test set. Asterisks denote significant pairwise differences ($p < 0.05$)..... 138

Figure 57: Absolute error of ANN-predicted C6-C7 peak compression vs. EMGD model-derived values across the first four head movement cycles of yaw trials within the test set. Asterisks denote significant pairwise differences ($p < 0.05$)..... 139

List of Tables

Table 1: ANN parameters in studies involving predictions of spinal contact forces with reduced data or modelling requirements.	39
Table 2: Summary of the relative inertial properties of each helmet configuration adapted from Healey et al. 2021.	59
Table 3: Description of each of the five ANN sub-groups that were developed. “ALL” subscripts denote ANNs who received head-trunk angular displacement, velocity, acceleration and moment inputs. “DM” subscripts denote ANNs that received head-trunk angular displacement and moment inputs.	68
Table 4: Default hyperparameters applied to all ANNs during first hyperparameter tuning stage.	70
Table 5: Summary statistics of composite error scores within each ANN sub-group. The best performing ANNs within each group are highlighted in bold text.	73
Table 6: Mean composite error scores (N) of ANNs grouped by learning rate. Except in the dual-rotation group, ANNs with a learning rate of 0.1 often yielded values that were undefinable/not a number. Models that yielded NaN values were excluded from calculation of mean values.	78
Table 7: Overview of the hyperparameters that yielded the lowest composite error score within each ANN group.	79
Table 8: Mean absolute differences in compression and shear measures during hNVG vs. hNVGCW conditions.	85

List of Equations

Equation 1: Mean squared error loss function.	13
Equation 2: General expression from Nielsen (2015) demonstrating the linear relationship between updates of weights, w and biases, b within layer j and changes in layer output, y ...	13
Equation 3: Output of the forget gate layer within an LSTM cell	31
Equation 4: Outputs of the update gate layer within an LSTM cell	32
Equation 5: Output of the tanh layer accompanying the update gate within an LSTM cell...	32
Equation 6: Output of the update gate layer within an LSTM cell	32
Equation 7: Cell state output of an LSTM cell	33
Equation 8: Hidden activation output of an LSTM cell.....	33

Chapter 1: Introduction

1.1 Neck Pain Among Military Helicopter Aircrew

Neck pain is a substantial issue among military helicopter pilots around the world. In the Royal Canadian Airforce, 80% of Griffon helicopter aircrew report chronic neck pain (Chafé & Farrell, 2016). In the Netherlands, the one-year prevalence of neck pain among military helicopter aircrew is reported to be 43% (van den Oord et al., 2010) while in Sweden the 3-month prevalence is 57% (Äng & Harms-Ringdahl, 2006). Research suggests that the risk of neck pain among helicopter pilots is exacerbated by the use of night vision goggles (NVGs) (Farrell et al., 2020; Chafé & Farrell., 2016; Manoogian et al., 2006; Adam, 2004). This is highlighted by reports indicating that aircrew who use NVGs are 45% more likely to experience head and neck injury than those who do not (Shannon & Mason, 1997). While the etiology of neck pain is multifactorial (Dibblee et al., 2015; Croft et al. 2001; Côté et al., 2004), increases in compression and shear loading of the cervical spine during NVG use are considered to be an important factor (Forde et al., 2011).

1.2 Designing Interventions to Prevent Neck Pain in Helicopter Aircrew

NVGs, which are mounted to the front of aircrew helmets (Figure 1), are generally considered to increase the risk of neck pain by shifting the centre of mass (COM) of the head-helmet system anteriorly (Chafé & Farrell, 2014; Manoogian et al., 2006; Adam, 2004). In response, posteriorly mounted counter-weights (CWs) have been proposed as an ergonomic intervention to prevent neck pain in helicopter aircrew (Harrison et al., 2007). CWs have been considered to mitigate the injurious effects of NVGs by returning the head-helmet COM closer to the neck

to reduce associated external flexion moments about the spine (Harrison et al., 2007). While this may be efficacious when the head is static in a neutral posture, aircrew are required to perform dynamic head movements through a wide range of motion to scan their environment during flight (Forde et al., 2011). As a result, CWs have been reported to not reduce compression and shear loading of the cervical spine when performing flight-relevant head movement (Barrett, 2016). This is likely related to the associated increases in the mass and moment of inertia of the helmet system.

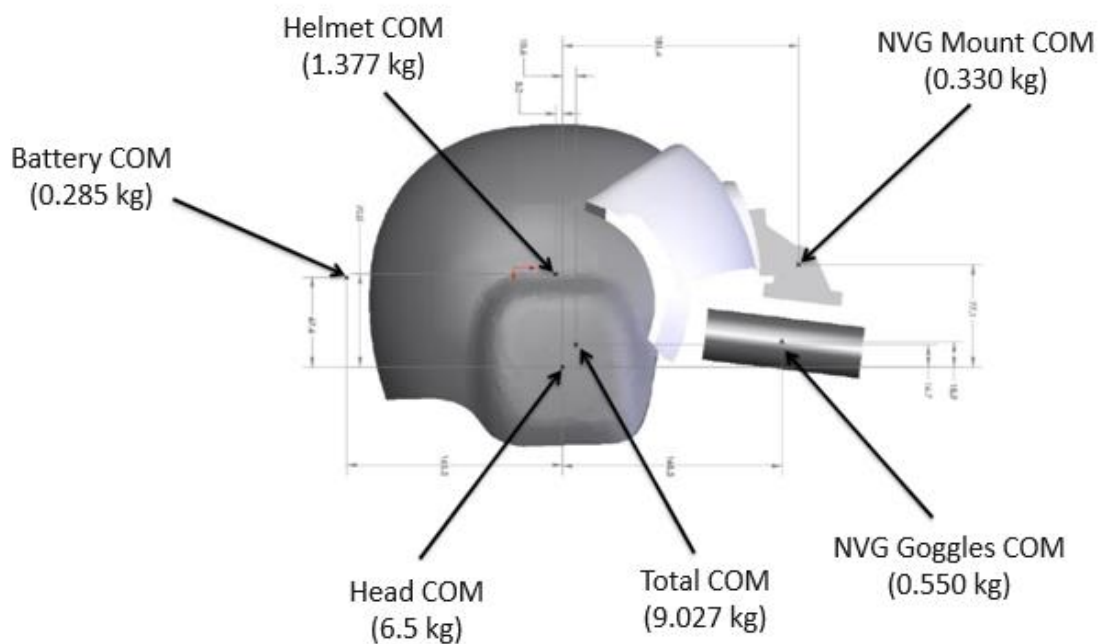


Figure 1: Exemplar helmet-NVG system including depictions of COM locations.

The multi-faceted relationship between helmet configurations and biomechanical exposures of the cervical spine creates a unique challenge for helmet designers. Several

parameters, such as the mass, distribution of mass, rate of change in posture and vibration exposure, may all independently or interdependently affect joint contact forces within the cervical spine during flight. As a result, the ability to assess the biomechanical implications of many candidate helmet-NVG system configurations would be ideal in order to identify candidate designs that minimize the associated joint contact forces within the cervical spine. Furthermore, because this biomechanical assessment is only one of many steps in the helmet design process, it is important that it can be performed in a time and resource-efficient manner to support rapid iterative computer-aided design (CAD) processes.

At present, the ability to estimate joint contact forces during virtual design is limited. Existing digital human modeling tools can be used to solve multi-body dynamics problems to yield joint torques about a candidate joint, perhaps based on different head supported mass configurations, but few robustly consider joint contact forces (E.g. AnyBody Technology, Aalborg, Denmark). Open-source alternatives include OpenSim (Delp et al., 2007), however use of OpenSim requires additional programming and modelling expertise, where such software may not be as easily deployed within a business's information technology infrastructure. Many engineering firms currently rely on commercially available enterprise-based multi-body dynamics (MBD) solutions, such as Ansys Motion (Ansys, Pennsylvania, USA), or Adams Multibody Dynamics Simulation (MSC Software, Surrey, UK) to inform mechanical design and may not have time or expertise to learn additional software packages to estimate the underlying contact forces. MBD modelling represents a relatively time and resource-efficient approach to estimate inter-body moments and reaction forces, or

intersegmental moments and reaction forces when applied in the digital human context. Specifically, MBD modelling allows helmet designers to apply different helmet “bodies” to a model and perform inverse dynamics computations to understand how different helmets might influence joint moments under specific load cases (i.e., pitch or yaw head rotation). While these exposure metrics may provide some indication of injury risk, they do not account for the significant (Reeves & Cholewicki, 2003) contributions of muscles to intervertebral joint contact forces as they do not consider EMG data. In contrast, modelling approaches capable of accounting for the contributions of muscles and other soft tissues to joint contact forces provide a more comprehensive indication of injury risk.

To date, the joint contact forces associated with helmet-NVG designs have only been evaluated via *in vivo* human participant studies in which participants don physical helmet prototypes while performing flight-relevant actions or postures (Healey et al., 2021; McKinnon et al., 2016; Dibble et al., 2015). Electromyography (EMG) and motion capture data can then be recorded during these actions and input into EMG-driven (EMGD) musculoskeletal models to estimate joint contact forces (Barrett, 2016). However, *in vivo* human participant research and the production of physical helmet prototypes is time and resource intensive. This limits the ability of a helmet designer to feasibly assess the joint contact force consequences of multiple helmet designs in a rapid, iterative, process prior to physical helmet model creation. Ideally, designers could quickly assess the compression and shear joint contact forces associated with a given helmet design using a CAD-based approach, without necessitating the use of physical prototypes or laboratory-based collection of EMG and other *in vivo* data.

1.3 Artificial Neural Networks as a Potential Tool for Virtual Helmet Design

Artificial neural networks (ANNs) represent an especially promising approach to predict joint contact forces given reduced data inputs. An ANN is a computational model inspired by the functional structure of the human brain, consisting of simple processing units known as nodes. ANNs can “learn” an input-output mapping by iteratively adjusting the weights and biases of internodal connections such that the model generates desired outputs for a given set of inputs. ANNs are known to be efficacious in mapping complex, nonlinear relationships (Haykin, 2004; Agatonovic-Kustrin & Beresford, 2000). Additionally, because information processing in an ANN is distributed across a range of parallel computational units, ANN outputs are generally robust to error/noise in inputs (Gurney, 1997). In the field of biomechanics, ANNs have previously been applied to estimate intervertebral compression and shear within the lumbar spine with reduced data inputs relative to traditional biomechanical models (Mousavi et al., 2020; Arjmand et al., 2013; Parkinson & Callaghan, 2009; Hou et al., 2007; Hou et al., 2004). However, similar ANNs have not been developed for the cervical spine.

ANNs, in conjunction with MBD/digital human modelling may be a useful *in-silico* method of predicting cervical spine compression and shear within a virtual helmet design process. For example, flight-relevant head-trunk kinematics could be fed in to an MBD model and the resultant moments, as well as head-trunk kinematics, could be input to an ANN to generate compression and shear predictions. Importantly, this approach would not require *in situ* data collection while providing a more comprehensive indication of injury risk than current engineering design methods.

While a general ANN model for application to a variety of head movement trajectories would be ideal, rotation-specific ANNs (pitch and yaw) may have improved performance relative to a more general ANN. This is due to biomechanical differences between trajectory types that may be difficult to account for with the ANN input features that are available through MBD modelling. For example, muscle activation profiles, which influence intervertebral contact forces (Granata & Marras, 1993), are likely to be consequentially different between pitch and yaw head movements. An ANN would likely have difficulty in accounting for these between-rotation type differences without EMG input. Therefore, both general and rotation-specific ANNs should be developed and assessed.

In summary, ANNs may be a useful tool in a CAD process of helicopter aircrew helmets. Currently, helmet designers can leverage multi-body dynamics modelling to calculate intervertebral moments during simulated head movement as a function of the inertial properties of the head-helmet system. This allows the designer to understand how different helmet-NVG configurations influence associated joint moments but does not allow for estimation of joint contact forces. An ANN capable of predicting joint contact forces based on joint moments and head-trunk kinematics would facilitate the helmet design process by providing a more comprehensive indication of injury risk, without requiring *in situ* data collection.

1.4 Objective

The aim of this thesis was to develop ANNs to predict C6-C7 compression and anteroposterior (AP) shear with adequate sensitivity to be a useful tool in a virtual helmet design process. The decision to focus on the C6-C7 joint was made because, while intervertebral joint contact forces generally increase cranially to caudally, they plateau between C5-C6 and C7-T1 (Barrett et al., 2020). Therefore, compression and shear measures at C6-C7 may be useful as proxy measures of compression and shear at both C5-C6 and C7-T1 joints as well.

1.5 Research Question

Can an ANN predict C6-C7 compression and AP shear, given C6-C7 joint moments and head-trunk kinematics, with sufficient sensitivity to be able to differentiate between C6-C7 compression and AP shear exposures associated with a helmet equipped with NVGs and exposures associated with a helmet equipped with a prevalent ergonomic counter-weight intervention during a reciprocal scanning task?

1.6 Hypothesis

An ANN can predict C6-C7 compression and AP shear where the absolute error, relative to a criterion standard EMGD model, is less than the absolute difference in EMGD model-computed C6-C7 compression and AP shear during flight-relevant head movement when donning a helmet with NVGs relative to donning a helmet with NVGs and a prevalent counter-weight intervention.

Chapter 2: Literature Review

2.1 Overview of Artificial Neural Networks

Traditional feedforward (FF) ANNs typically consist of an input layer, hidden layer(s) and an output layer (Figure 2). The input layer receives inputs in the form of a tensor. Tensors can be defined as an array of values with a variable number of dimensions. Hidden layers perform calculations to process inputs before transferring this processed information to the output layer. Finally, the output layer generates the overall output of the ANN. Each of these layers consists of simple processing units known as nodes. These nodes receive inputs, which are each multiplied by an associated weighting. Next, these weighted inputs are summed arithmetically. Each node also has a bias which increases or decreases the sum of the weighted inputs. The resultant value is then passed to an activation function which limits or “squashes” values to within a given range. Common examples of activation functions include hyperbolic tangent or sigmoid functions. Finally, the result of the activation function computation is transferred to nodes of the following layer or, in the output layer, returned as the final output of the ANN. A graphical representation of the process by which a node processes information in a simple ANN is presented in Figure 3.

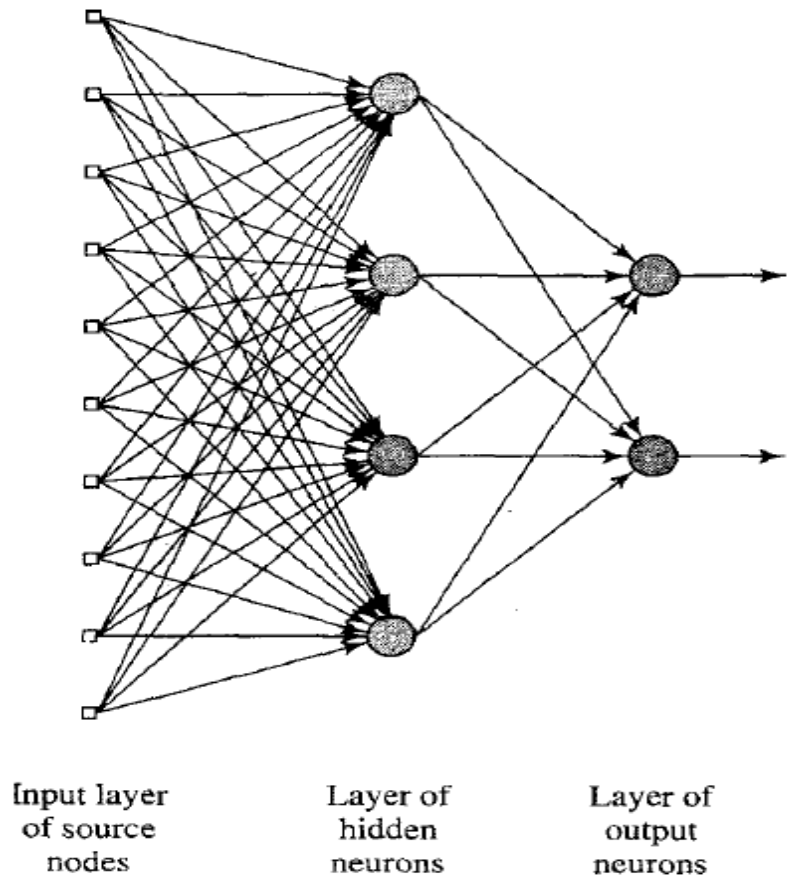


Figure 2: Schematic of a 3-layer, densely connected FF ANN with one hidden layer from Haykin (2004). The term “densely connected” refers to each node being connected to all nodes of a previous layer. Arrows represent the unidirectional flow of data from one node to another.

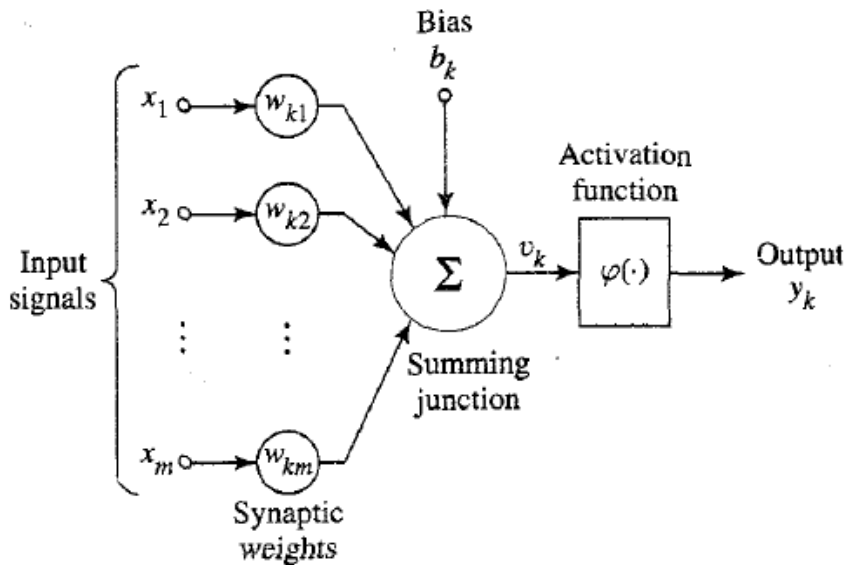


Figure 3: Diagram of a node from Haykin (2004). Inputs, x , of a given node, k , are multiplied by weights, w , before being summed. m represents the number of input features. After the weighted inputs are summed, a bias, b , is added. The resultant value, v , is then passed through the activation function, ϕ , to generate the final output of the node, y .

2.2 How do Artificial Neural Networks Learn?

During supervised learning, ANNs “learn” by being presented with examples of input and target output pairs and iteratively adjusting the biases and weights of connections between nodes such that ANN outputs for a given input approximate target outputs. These input-target output pairs are collectively known as the training set. Best practice (Halilaj et al., 2018) in ANN development also requires the use of validation and test sets. Briefly, the validation set contains target input-output pairs that are used to assess ANN performance throughout the ANN development process. The test set contains target input-output pairs that are used to

assess the generalizability of the final ANN by evaluating ANN performance when provided data to which it has no prior exposure.

The adjustment of weights and biases can be considered an optimization problem where the loss function is the error of the outputs of the network relative to desired target values. There are many training methods available to optimize the weights and biases such as gradient descent-based methods (E.g. Kingma & Ba, 2015), the Newton method (Fischer, 1992), conjugate gradient method (Gilbert & Nocedal, 1992) and Levenberg-Marquardt algorithm (Moré, 1978). However, gradient descent is the most popular (Shrestha & Mahmood, 2019; Ruder, 2016) and will thus be the focus of this chapter. Gradient descent involves minimizing a loss function over a number of training iterations by iteratively adjusting weights and biases in the manner that results in the steepest decrease of the loss function.

The gradient descent process requires both forward and backward propagation (Rumelhart et al., 1985) of information through the ANN (Figure 4). When a sample input from the training set is provided to the ANN, this information flows through the nodes of the network as described in Chapter 2.1 and outputs are generated. These outputs are compared to corresponding target values and performance is assessed through a loss function. The loss function (often mean squared error [LeCun, Bengio & Hinton, 2015]) provides a measure of how closely the outputs associated with a given set of weights and biases match the target values (Equation 1). During each training iteration, the gradient, or first derivative, of the calculated loss function is calculated, with respect to ANN weights and biases, over a number

of training samples and propagated backward, layer-by-layer, through the network. This information flows from the output layer, through the network such that each node receives a partial derivative that is the loss gradient normalized to the weight and bias of that particular node (Equation 2). Based on these partial derivatives, the weights and biases of each node are adjusted with the goal of converging on a configuration that results in the loss function ideally reaching its global minimum. Generally, the weights and biases of a node that receives a larger partial derivative will be adjusted to a greater degree than those of a node that receives a smaller partial derivative. Once the weight of each internodal connection is updated, this forward and backward propagation training cycle is repeated to provide the partial derivative information to drive another training iteration.

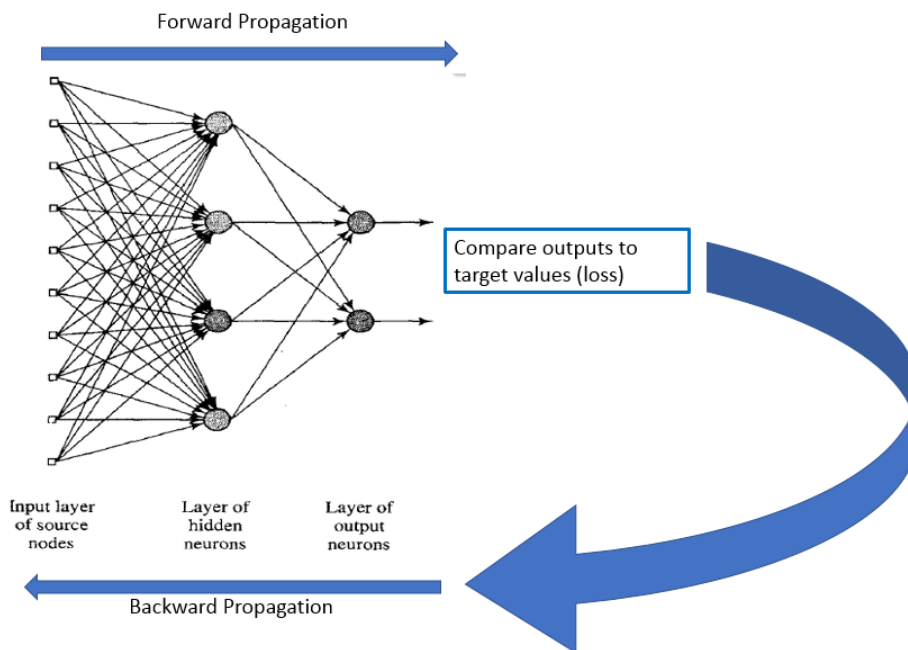


Figure 4: Visual representation of forward and backward propagation adapted from Haykin (2004).

Equation 1: Mean squared error loss function.

$$L(w, b) = \frac{1}{2n} \sum_x \|y(x) - a\|^2$$

Where w represents the weights of nodes within the ANN, b denotes node biases, n is the number of training input-output pairs, y denotes the target outputs, x represents input training samples and a denotes the ANN outputs. ANN performance is improving when a approaches $y(x)$, yielding smaller magnitudes of loss, L .

Equation 2: General expression from Nielsen (2015) demonstrating the linear relationship between updates of weights, w and biases, b within layer j and changes in layer output, y .

$$\Delta y \approx \sum_j \frac{\partial y}{\partial w_j} \Delta w_j + \frac{\partial y}{\partial b_j} \Delta b_j$$

2.2.1 Mini-Batch Gradient Descent

Each iteration of the ANN training process can be performed while considering all available training samples, a subset of samples or a single sample. Calculating the derivative of a loss function over all available training samples involves computing the average gradient over all samples. This is likely to be very computationally expensive. Therefore, a common (Lipton et al., 2015) approach is to only consider a subset of samples at each training iteration in a process known as mini-batch gradient descent. During each training iteration, a portion known as a “mini-batch”, of the training data is processed to generate a loss function gradient. Each mini-batch contains a practitioner-defined number of samples where each

sample is a matching set of inputs and target outputs. The use of smaller mini-batch sizes tends to accelerate ANN training (Bengio, 2012) and may assist in ANN generalizability, by introducing noise in to the gradient descent process (Ruder, 2017). However, there is no optimal mini-batch size for all ANN use cases. Nonetheless, mini-batch sizes of less than 32 samples are the recommended default (Bengio, 2012).

During ANN training, mini-batches continue to be passed through the network until a given number of epochs have passed. An epoch represents one complete pass through the training set where the ANN has been exposed to all available training samples. For example, a training set of 100 samples could be split in to four batches of 25 samples where one epoch would be completed when all four batches have been passed through the network.

2.3 The Bias-Variance Dilemma (Geman et al., 1992)

During training, weights and biases are optimized such that the overall network is able to map a relationship between a set of input and output features based on training data. Given adequate network complexity and training epochs, an ANN can theoretically map any function to any desired accuracy (Hornik, Stinchcombe & White, 1989). However, to be useful, the ANN must be able to generalize i.e. adequately map input-output relationships within novel datasets.

When applying an ANN to novel data, the associated prediction error is composed of bias, variance and noise. Noise is intrinsic to the underlying data on which the ANN is trained and is due to factors that ANNs are unable to model directly. Bias (Figure 5) is the

average difference of model predictions and the corresponding target values. Bias is associated with the error present when an ANN is approximating the relationship between input and output features within a training dataset. An ANN with high bias oversimplifies its mapping of the input-output relationship and has high error on both training data and novel data. Variance (Figure 5) is the amount that ANN outputs for a given set of inputs would change if the ANN had been trained on a different set of training examples. A model with high variance has learned the specific relationships within training data very well but performs poorly when applied to novel data. The bias-variance dilemma stems from the fact that bias and variance are inversely related (Geman, Bienenstock & Doursat, 1992). Therefore, it is up to the ANN developer to achieve an appropriate balance between the two types of error. For example, reducing model complexity (e.g. decreasing the number of hidden nodes and/or input features) typically decreases variance but increases bias (Hastie, Tibshirani & Friedman, 2009; Geman, Bienenstock & Doursat, 1992) (Figure 6).

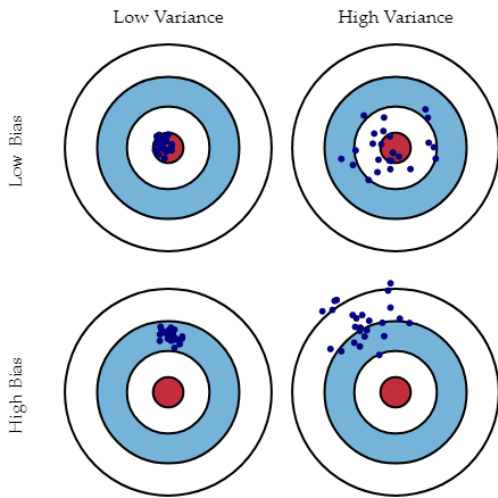


Figure 5: Visual representation of bias and variance from Fortmann-Roe (2012). Blue dots represent ANN outputs relative to corresponding target values represented by the red bulls-eye.

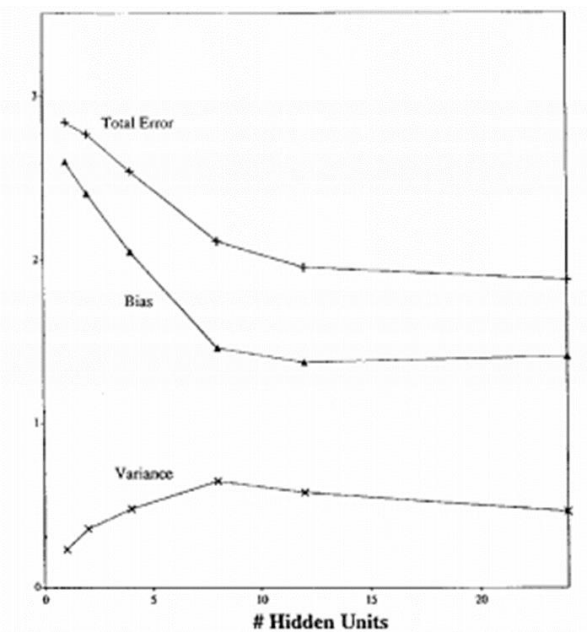


Figure 6: Experimental findings from Geman, Bienenstock & Doursat, 1992 demonstrating a decrease in bias error and increase in variance error as the number of hidden nodes in an ANN is increased.

2.4 Over/Underfitting

The goal of an ANN is to learn a function that maps the underlying relationship between sets of inputs and outputs. An ANN's ability to map functions of increasing complexity is termed capacity (Goodfellow, Bengio & Courville, 2016). An ANN with insufficient capacity will be unable to learn input-output relationships within the training set and will exhibit high bias and low variance. This is referred to as underfitting (Figure 7). In contrast, an ANN with excessive capacity will “memorize” the input-output relationship within the training set but have poor generalizability to novel data. This ANN would have low bias and high variance and would exhibit overfitting (Figure 7). ANN developers can modulate a network's capacity by changing the number of layers and hidden units per layer (Goodfellow, Bengio & Courville, 2016). However, arriving at the set of ANN parameters that appropriately balance the risk of overfitting and underfitting typically requires trial and error during a process called hyperparameter tuning. This is discussed in Chapter 2.6.

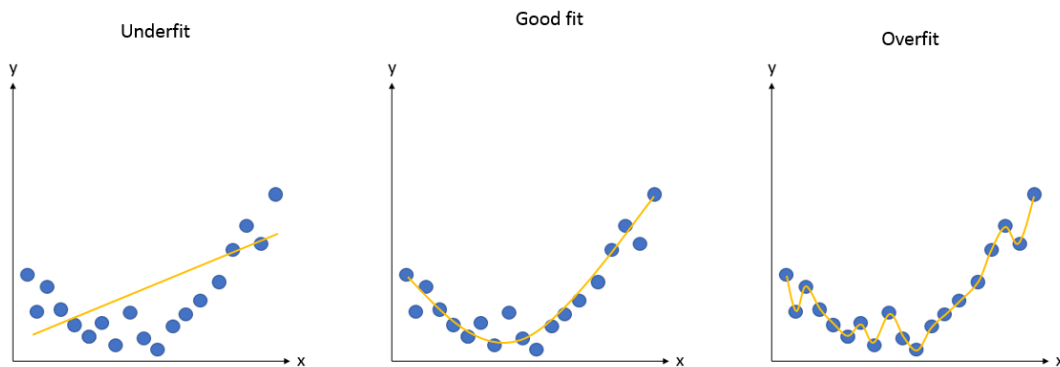


Figure 7: Visual representation of over/underfitting of a relationship between inputs (x) and outputs (y). The orange line represents the input-output relationship that is “learned” by the ANN. Blue circles represent input/output pairs from a training set.

2.5 Overview of the Vanishing (Hochreiter, 1998) and Exploding (Hochreiter, 1991) Gradient Problems

The vanishing and exploding gradient problems become increasingly important to consider when training deeper ANNs. Recall that weights and biases are updated during training based on the gradient of the loss function and associated partial derivatives. During backpropagation, the partial derivatives that drive weight and bias updates are calculated layer-by-layer, starting from the output layer. These calculations are performed via the chain rule which is used to calculate the portions of the change in a loss function that are due to adjustments of the values of the weights and biases of individual nodes. During backpropagation, the partial derivatives of nodes within a layer are multiplied by the partial derivatives of nodes within the previous layer (in this case, the previous layer is the layer closer to the output layer). Therefore, in an ANN with n hidden layers, $n - 1$ sets of derivatives have been multiplied together by the time information is backpropagated to the first hidden layer. The outcome of this is that gradients can increase or decrease exponentially from layer to layer (Philipp, Song & Carbonell, 2018). As a result, these gradients can “vanish” or “explode” and result in ineffective training. For example, a very small gradient will prevent weights and biases from being updated effectively after each training iteration. In this scenario, training will not be able to converge on an appropriate weight and bias configuration. In contrast, when gradients “explode”, weight and bias updates are excessively large in magnitude, leading to unstable training. As a result, selection

of the number of hidden layers within an ANN should consider the increased risk of vanishing and exploding gradients with deeper networks.

2.6 Hyperparameter Tuning

In the context of ANNs, hyperparameters are parameters that influence the ANN learning process but are not directly selected by a learning algorithm. Hyperparameters include architectural variables such as the number of hidden layers and number of nodes per layer. They also include the learning rate and mini-batch size variables that were introduced in Chapter 2.2. The process of hyperparameter tuning involves training multiple ANNs with a variety of hyperparameter values and then assessing the performance of the trained ANNs on novel data. This novel data is referred to as the validation set. It is data that was not input to the ANN during training, and which will not be considered during final evaluation of ANN performance (i.e. hypothesis testing). It is important that validation data remains separate from training and testing datasets to ensure valid evaluations of model generalizability.

To select the set of hyperparameters that yield the best performance, trained ANNs are applied to the validation set and their performance is assessed through a pre-defined performance metric. When applying ANNs in sequence regression problems, mean squared error (MSE) is the recommended error metric (Halilaj et al., 2018) where MSE is calculated between ANN outputs and corresponding target values. Upon calculation of the MSE associated with each of the trained ANNs, the developer can select the ANN with the lowest MSE for final evaluation on the test set.

2.6.1 Tuning Architectural Parameters

As mentioned previously, an ANN with a single hidden layer and a sufficient number of hidden nodes can learn to map a function of any level of complexity (Hornik, Stinchcombe & White, 1989). In practice, it is difficult to identify the “sufficient” number of hidden nodes. Furthermore, it is generally more computationally efficient to develop an ANN with more hidden layers than an ANN with a single hidden layer consisting of a very large number of nodes (Goodfellow, Bengio & Courville, 2016). Unfortunately, increasing the number of hidden layers poses other challenges such as increased risk of vanishing/exploding gradients (Goh, Hodas & Vishnu, 2017) (Chapter 2.5).

Typically, each node within the input layer of an ANN corresponds to a single input feature. Therefore, the number of nodes in the input layer is determined by the number of input features being provided to the ANN. Similarly, when applying an ANN to a regression problem, the number of nodes within the output layer will be equal to the number of output variables. Identifying an appropriate number of hidden layers and nodes within hidden layers is less straightforward. Generally, including a greater number of hidden nodes will allow an ANN to map increasingly complex relationships at the cost of increased computational demands as well as an increased risk of overfitting (Sheela & Deepa, 2013). While advanced methods can be used to automate the selection of ANN architectural parameters (Cortes et al. 2017), they are beyond the scope of this thesis. Traditionally, selecting ANN architectures has involved trial and error where developers train and evaluate ANNs with a range of number of hidden layers and nodes per layer.

2.6.2 Tuning the Learning Rate

During ANN training, the magnitudes of weight and bias updates are scaled by a value called the learning rate. A larger learning rate will result in larger adjustments of weights and biases after each training iteration. Generally, a smaller learning rate will result in slower and more computationally expensive training and can cause training to get “stuck” in suboptimal, local minima and plateaus (Goodfellow, Bengio & Courville, 2016) (Figure 8). In contrast, a larger learning rate can lead to unstable training (Figure 9) as well as oscillation around minima which lead to difficulties in converging on these minima (Goodfellow, Bengio & Courville, 2016). Large learning rates also increase the risk of weight values becoming extremely large (Goodfellow, Bengio & Courville, 2016), leading to the exploding gradient problem (Hochreiter, 1991) (Chapter 2.5). Unfortunately, it is generally not feasible to identify the optimal learning rate *a priori* (Reed & Marks, 1999). Therefore, ANNs with a variety of learning rates should be evaluated. Traditional default learning rates are 0.01 or 0.1 (Bengio, 2012).

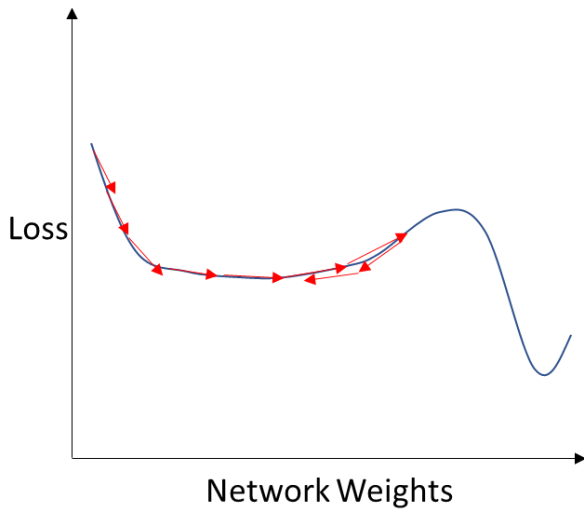


Figure 8: Visual representation of training becoming “stuck” in a local minimum/plateau. Each arrow represents an update to network weights. In practice, the relationship between the loss function and network weights would be in a much higher dimensional space where the weight and bias of each node corresponds to one dimension within the optimization space.

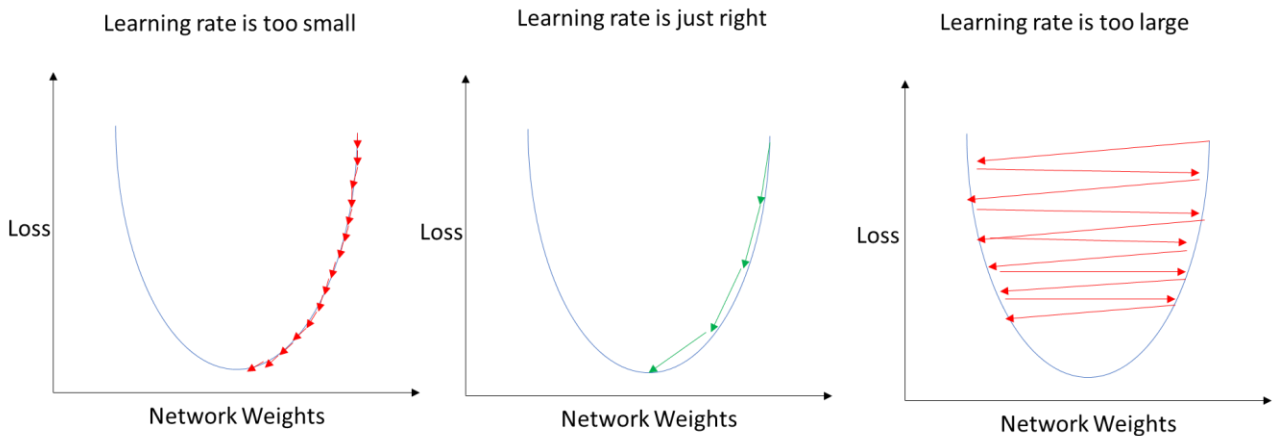


Figure 9: 2D conceptual representation of the effects of different relative learning rates on the loss function of an ANN during training. Each arrow represents an update to weights within a network. Similar to Figure 8, this 2D figure is a simplification of what is a higher dimensional space in practice.

2.7 Overview of the Adam Optimization Algorithm

Gradient descent provides the basis for weight and bias optimization in ANNs. However, several gradient-based optimization algorithms, such as stochastic gradient descent with momentum (Sutskever et al. 2013), Adagrad (Duchi, Hazan & Singer, 2011) RMSprop (Hinton, Srivastava & Swersky, 2012) and Adam (Kingma & Ba, 2014) exist. This chapter will introduce the Adam algorithm as it is the recommended default optimization algorithm for practical applications due to a combination of strong performance and computational efficiency (Ruder, 2016; Kingma & Ba, 2015) (Figure 10).

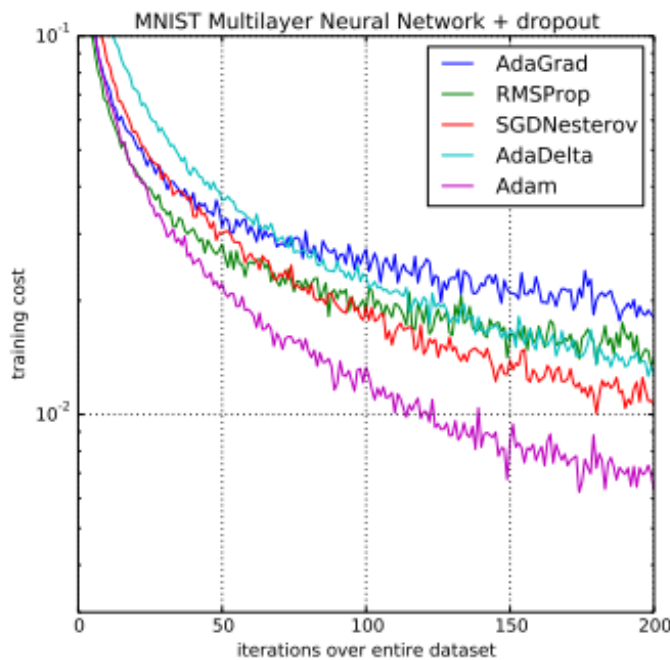


Figure 10: Figure from Kingma & Ba (2015) demonstrating the performance of ANNs employing one of five different optimization algorithms to learn a handwriting recognition problem. Adam optimization yielded the best performance with fewer training iterations and lower computational cost relative to other optimization algorithms.

Traditional gradient descent (Chapter 2.2) employs a constant, user-defined, learning rate across all training iterations. However, it can be beneficial to apply a larger learning rate during early stages of training and a smaller learning rate as training progresses. Doing so provides the advantages of a larger learning rate (computational efficiency, lower risk of getting stuck in local minima) without the same difficulties in converging on minima. To achieve these benefits, learning rate schedules can be implemented to adjust learning rate according to a pre-defined schedule. However, this requires the practitioner to perform time-consuming experiments to select a learning rate schedule that is appropriate for a given problem. The Adam optimization algorithm provides the benefits of adaptive learning rates without requiring a user-defined learning rate schedule (Kingma & Ba, 2014). Specifically, Adam calculates learning rates based on exponential weighted moving averages of the gradient. The exact formulae underlying Adam are complex and outside the scope of this thesis. However, the general outcome is that learning rates are adapted throughout training to decrease computational cost as well as improve convergence on optimal weight and bias configurations.

2.8 Regularization Techniques

Regularization involves modifying the ANN training process to reduce ANN variance/overfitting. Common regularization techniques include early stopping (Prechelt, 2012) and dropout (Srivastava et al., 2014).

2.8.1 Early Stopping

Training an ANN over a greater number of epochs will generally improve the performance of the ANN on training data at the cost of increasing the risk of overfitting (Tetko et al., 1995). Early stopping (Prechelt, 2012) is a regularization technique used to stop ANN training at the training epoch where the ANN begins to become overfit. Early stopping involves three basic steps. First, all available training data is split into a training set and a validation set. Next, the ANN is trained on the training set while performance is repetitively evaluated by applying the ANN to the validation set after a given number of training iterations. Finally, training is stopped when the performance of the ANN on the validation set begins to worsen. Training should generally not be stopped after just one epoch of increased validation error because validation error curves almost always exhibit several local minima (Figure 11). Prechelt (2012) suggests that the optimal trade-off between performance and training time was achieved in a variety of problems when training was stopped after 6 successive epochs of validation error increases. In this example, the early stopping algorithm would be referred to as having a “patience” of 6 epochs.

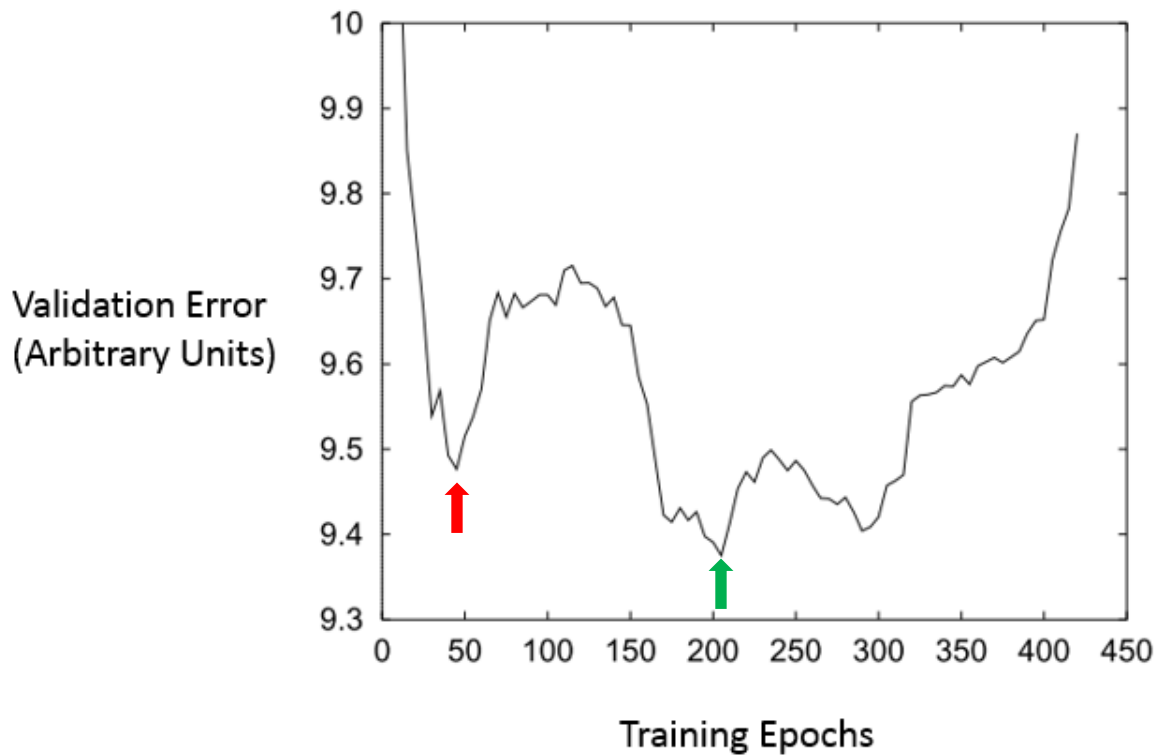


Figure 11: Validation error curve of an ANN’s validation error over 425 training epochs adapted from Prechelt (2013). The red arrow highlights a local minimum early in the training process. The green arrow indicates the global minimum. In this example, ANN training would ideally be stopped near epoch 200.

2.8.2 Dropout Regularization

Dropout regularization (Srivastava et al., 2014) involves temporarily eliminating or “dropping out” a portion of nodes from the neural network during each training iteration (Figure 12). When a node is dropped, its connections are removed, and its weights are not updated during backward propagation. By randomly dropping a portion of nodes during each training cycle, excessive co-adaptation between nodes is prevented. Co-adaptation refers to a

node becoming overly dependent on another node (Hinton et al., 2012). For example, a given node may adapt such that it compensates for errors made by another node. These co-adaptations are unlikely to generalize to novel data, resulting in overfitting. Dropout can be thought of as an approach to force nodes to work with a random subset of other nodes. In doing so, the network becomes more robust and typically demonstrates improved performance on novel datasets (Srivastava et al., 2014).

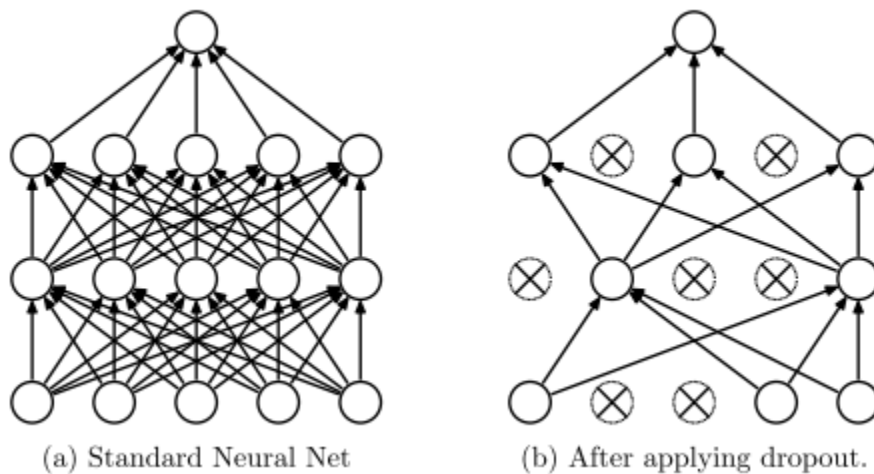


Figure 12: Visual representation of dropout from Srivastava et al. (2014) of a FF neural network before and after applying dropout. Circles and arrows represent nodes and internode connections respectively. Crossed circles represent nodes that have been “dropped”.

2.9 Recurrent Neural Networks

When applying FF ANNs in sequence-to-sequence modelling, each frame of data is processed independently. When processing sequences of time-series data, which is required

to achieve the objective of this thesis project, recurrent neural networks (RNNs) are likely advantageous relative to traditional ANNs (Lipton et al., 2015). RNNs can process inputs at a given time step while recalling information from the previous time step and are typically applied to problems involving time-series data and/or forecasting (Shrestha & Mahmood, 2019). RNNs are able to “recall” information from the previous time step by providing the hidden node activations at timepoint $t - 1$ as an additional input to the RNN at timepoint t (Figure 13). This feature allows RNNs to account for interdependencies between subsequent data points within a given sequence (Elman, 1990). Unfortunately, traditional RNNs are not a viable approach when there are long time lags between relevant input signals and the outputs at a given time step (Bengio et al., 1994). Furthermore, it is difficult to train deep RNNs (RNNs with several layers) as they are very susceptible to the exploding gradient problem. (Hochreiter et al., 2001)

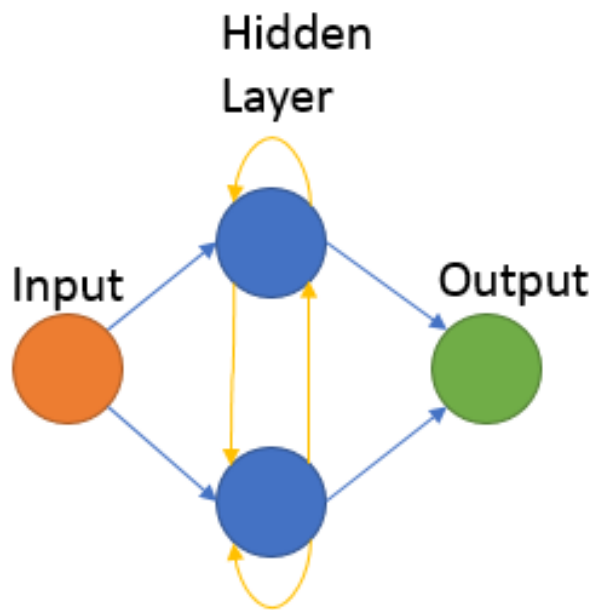


Figure 13: Model of a simple RNN with one hidden layer consisting of two nodes. Blue arrows represent the flow of data from timepoint, t , while yellow arrows represent hidden node activations originating from timepoint $t - 1$.

2.9.1 Long Short-Term Memory (Hochreiter & Schmidhuber, 1997a)

Long short-term memory (LSTM) (Hochreiter & Schmidhuber, 1997a) neural networks are a type of RNN that are well-suited to predict time-series outputs given lags of unknown duration between relevant input signals and outputs at a given time step (Hochreiter & Schmidhuber, 1997b). This separates them from traditional RNNs which are only capable of recalling information over one time step. As a result, LSTM networks are the current state of the art in sequence modelling contexts such as voice recognition (Chiu et al., 2017) and language translation (Sidorov, 2018).

An LSTM network includes one or more LSTM cells which have four defining components: a cell state, a forget gate, an update gate and an output gate (Figure 14). The cell state allows for the flow of information over multiple time steps. It is an array of numerical values that can be thought of as the “memory” of the LSTM cell. In theory, information from the first timestep in a given sequence could be carried forward within the cell state to any future timestep in the sequence. The collective role of the gates is to control what information is added or removed from the cell state at each step within a sequence. The gates are composed of individual neural network layers that learn to perform their specific role by updating their weights and biases through training.

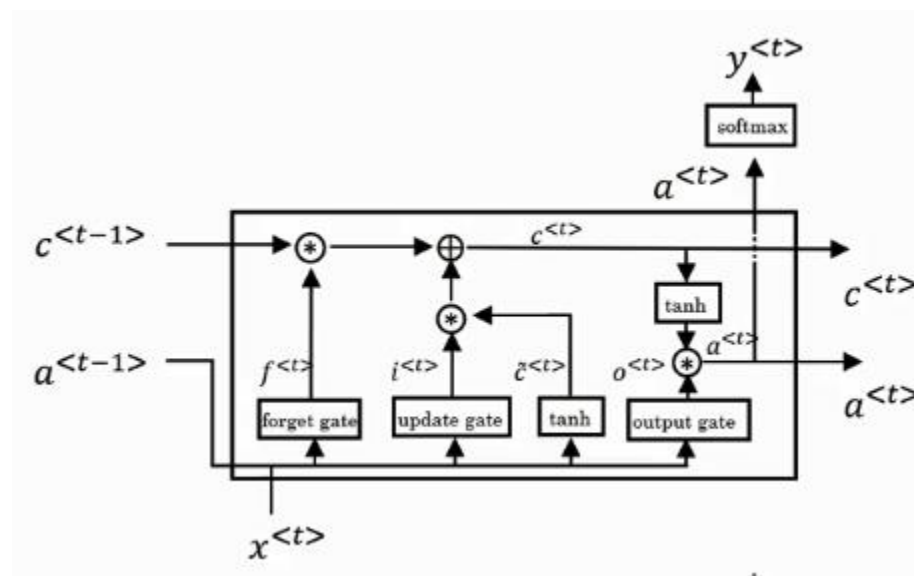


Figure 14: Graphical representation of an LSTM cell from Ng, Katanforoosh & Mourri (n.d.).

At a given timestep (t), the forget gate is the first to regulate the cell state. The inputs of the forget gate layer are the inputs to the LSTM cell (x_t) at the current timestep as well as

the hidden activation of the LSTM cell from the previous timestep (a_{t-1}). The forget gate layer applies a sigmoid activation function that receives x_t and a_{t-1} as inputs. It outputs values between zero and one, each value corresponding to a value within the cell state. Zero indicates “this information should be completely forgotten” and one indicates “this information should be completely retained”. The overall output (Equation 3) of the forget gate is a matrix of values between zero and one indicating to what degree cell states values should be forgotten.

Equation 3: Output of the forget gate layer within an LSTM cell

$$f_t = \sigma(W_f[a_{t-1}, x_t] + b_f)$$

Where f_t is the output of the LSTM forget layer, and W_f and b_f , represents matrices of weights and biases of nodes respectively within the forget gate layer.

Next, the update gate (also known as input gate) determines what information should be added to the cell state. The update gate (Equation 4) layer works alongside a hyperbolic tangent (tanh) activation layer (Equation 5). The update gate layer identifies which, and to what degree, cell state values should be updated while the tanh layer generates candidate values to add to the cell state. Similar to the forget gate, information regarding the degree to which candidate values should be added to the cell state are represented by a matrix containing values between zero and one. Greater values indicate that the candidate value should be considered to a greater degree within the cell state.

Equation 4: Outputs of the update gate layer within an LSTM cell

$$\mathbf{u}_t = \sigma(W_u[\mathbf{a}_{t-1}, \mathbf{x}_t] + \mathbf{b}_u)$$

Where u_t is the output of the update gate layer, and W_u and b_u , represent associated weights and biases of the nodes within the update gate layer.

Equation 5: Output of the tanh layer accompanying the update gate within an LSTM cell

$$\tilde{c}_t = \tanh(W_c[\mathbf{a}_{t-1}, \mathbf{x}_t] + b_c)$$

Where \tilde{c}_t represents candidate values with which to potentially update the cell state and W_c and b_c , represent corresponding weight and bias matrices.

The output gate consists of a sigmoid layer that determines what cell state information will be output and passed to the next time step (Equation 6). It receives the hidden activation from $t - 1$ and the input from the current timestep. Similar to the forget and update gates, the output of this gate is a matrix containing values between 0 and 1 that scale corresponding output values. The output values being scaled are the cell state, after it has been passed through a tanh function which “squashes” these values to the range -1 to 1.

Equation 6: Output of the update gate layer within an LSTM cell

$$o_t = \sigma(W_o[\mathbf{a}_{t-1}, \mathbf{x}_t] + b_o)$$

Where o_t is the output of the update gate layer, and W_o and b_o , represent associated weight and bias matrices.

The outputs of the whole LSTM cell are the cell state and hidden activation which are computed based on Equations 7 and 8 respectively. The cell state information is passed forward in time to be an input to the LSTM cell at time $t + 1$. The hidden activation, a^t , can be passed to the next layer within an ANN or, if there is no subsequent layer, returned as the final output of the ANN.

Equation 7: Cell state output of an LSTM cell

$$c_t = u_t \cdot \tilde{c}_t + f_t \cdot c_{t-1}$$

Where c_t represents the cell state at time t .

Equation 8: Hidden activation output of an LSTM cell

$$a^t = o_t \cdot c_t$$

2.10 Best Practices for Machine Learning in the Field of Biomechanics (Halilaj et al., 2018)

Halilaj et al (2018) outlined best practices in machine learning applications in biomechanics. This chapter will summarize these concepts as they apply to ANNs in this thesis.

2.10.1 Feature Selection and Scaling

Feature engineering is the process of selecting and pre-processing input and output features prior to ANN training. It is important to select appropriate input features as reducing data dimensionality will reduce the risk of overfitting (Halilaj et al., 2018). Halilaj et al (2018) recommend testing a combination of automated feature extraction techniques, such as principal component analysis, as well as features selected by a researcher based on domain

knowledge. In this thesis, input features were selected based on domain knowledge. Once features are selected, they should be rescaled such that the magnitudes of values within each feature vector are similar across features. This ensures that each feature has equal opportunity to contribute to weight and bias updates and allows optimization algorithms to converge on minima more quickly (Bishop, 1995). Recommended rescaling techniques are Z-score normalization and min-max scaling (Halilaj et al., 2018). In this thesis, features were rescaled using min-max scaling.

2.10.2 Assessing ANN Generalizability

When assessing ANN performance, it is important that computed performance metrics represent the ANN's ability to generalize to novel data. As discussed in Chapter 2.6, the data used to assess ANN generalizability must remain separate from the training data. It is recommended to split data into the training, validation and test sets such that all data from a given subject is included in only one of the aforementioned datasets. When testing the generalizability of a regression model on the test set, MSE or R^2 values are recommended performance metrics.

2.11 Combining Multi-Body Dynamics Models and ANNs as a Surrogate for More Computationally Expensive Models

Lu et al. (2013) applied a combined MBD and ANN method as a surrogate for a finite element (FE) model to reduce computational demands in predicting cartilage stress within the knee joint. MBD models had previously been successfully applied to model joint kinematics

and reaction forces. However, they do not allow for the prediction of stress and/or strain distributions which influence cartilage degeneration within the knee. To predict these variables of interest, FE modelling was required. However, FE modelling is highly complex and computationally expensive. Therefore, ANNs were developed to predict cartilage tissue stress (outputs of FE model) given rigid body reaction forces (outputs of MBD model).

Lu et al. (2013) developed FF ANNs as well as focused time delay neural networks (FTDNN). FTDNNs are similar to FF ANN but their input layer simultaneously receives inputs originating from multiple time steps within a sequence. The ANNs in this study had three layers (one hidden layer) and a variable number of hidden nodes. The selection of three-layer networks was motivated by a desire to minimize computational demands. Aligned with the dimensionality of ANN inputs and outputs, the input layer contained 1200 nodes and the output layer contained 400 nodes. The number of hidden nodes was increased in increments of ten and evaluated based on associated r and MSE values between ANN and FE modelling outputs. Specifically, hidden layer size was iterated from 10 to 200 in FF ANNs and 130 to 320 in FTDNNs. All hidden layers had sigmoid activation functions while output layers had linear activation functions. Committees of ANNs were applied to generate outputs. This involved applying three ANNs simultaneously and calculating the mean output across the three models. A committee of 3 FF ANNs with 80, 100 and 160 hidden nodes as well as a committee of 5 FTDNNs with 160, 230, 240, 280 and 320 hidden nodes were evaluated. Both committees performed well, yielding outputs with strong agreement ($r \approx 0.9$) with FE model outputs. The success of Lu et al. in using ANNs as a surrogate for highly complex FE

models supports the use of ANNs as a surrogate for EMGD musculoskeletal models in this thesis.

2.12 Applying an ANN to Predict Lumbar Muscle Activity (Nussbaum, Chaffin & Martin, 1995)

The success of relatively low capacity ANNs in mapping the relationship between moments and nEMG suggest that more complex ANNs may be capable of mapping a relationship between moments and joint contact forces while accounting for active muscle force contributions. Nussbaum, Chaffin & Martin (1995) used FF ANNs to predict bilateral EMG normalized (nEMG) as a percentage of maximal voluntary contraction (%MVC) from erector spinae, rectus abdominus, external oblique and latissimus dorsi while participants resisted static moment loads that were applied in a variety of directions using a shoulder harness while standing upright. Inputs to the ANN were the applied flexion, extension, right lateral bend and left lateral bend external moments. ANNs were developed with one hidden layer and all nodes within the ANN were assigned sigmoid activation functions. This led to outputs being constrained to the range of zero to one which aligned with the %MVC target outputs. Hyperparameter tuning was performed in two stages. First, ANNs were developed with eight hidden nodes and learning rate was iterated from 0.1 to 0.9 in increments of 0.1. Learning rate did not substantially influence ANN performance. In the next tuning stage, the learning rate was set to 0.7 while the number of nodes in the hidden layer was iterated from two to twenty. Four ANNs were developed with each

number of hidden nodes and each ANN was assessed by calculating R^2 values between ANN-predicted nEMG and measured nEMG.

ANNs with four to twenty hidden nodes performed similarly. However, as the number of hidden nodes decreased below four, R^2 values began to decrease. Therefore, the authors averaged the outputs of all ANNs with three or more hidden nodes and assessed the correlation of these mean outputs to the measured nEMG. This average ANN output method yielded a mean R^2 of 0.83 across all muscles and task conditions and outperformed two optimization-based approaches (mean $R^2 = 0.65$) (Figure 15).

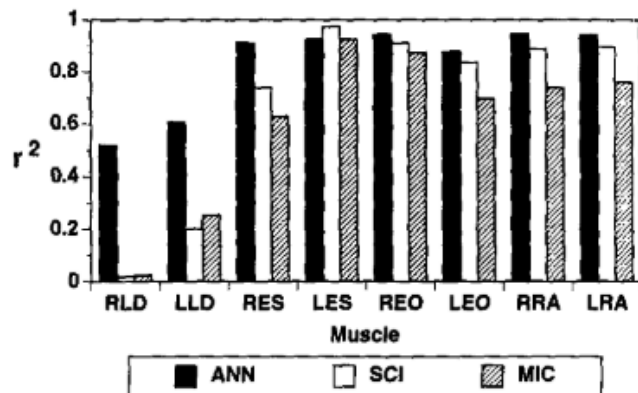


Figure 15: Figure from Nussbaum et al. (1995) comparing the performance of a committee of ANNs to optimization models that minimized muscle activation intensity and lumbar compression (MIC) and sum of cubed muscle activation intensity (SCI). Included muscles are the right and left latissimus dorsi (R/LLD), erector spinae (R/LES), external obliques (R/LEO) and rectus abdominus (R/LRA).

2.13 Previous Applications of Artificial Neural Networks to Predict Intervertebral Contact Forces Within the Lumbar Spine

The literature does not include studies of the application of ANNs to predict cervical spine contact forces. However, ANNs have been applied to predict lumbar spine contact forces. While these ANNs were developed in the context of lifting/grasping tasks, they provide valuable information regarding the range of network hyperparameters that should be evaluated in the context of this thesis. Specifically, the choice of ANN activation functions, regularization techniques, number of hidden layers and number of nodes/hidden layer in the studies described in this chapter were used to inform ANN development in this thesis. Studies in which ANNs were used to predict intervertebral moments (e.g. Aghazadeh, Arjman & Nasrabadi, 2020; Kingma et al., 2001) are not included in this summary as they are mapping inherently different input-output relationships.

Table 1: ANN parameters in studies involving predictions of spinal contact forces with reduced data or modelling requirements.

Author, Year	Type of ANN	Learning Rate	Number of Layers	Nodes per Hidden Layer	Regularization Technique(s)	Activation Functions
Hou et al., 2004	Recurrent	0.2	3	35	Did not specify	Hidden layer: Tanh Output layer: Sigmoid
Hou et al., 2007	Recurrent fuzzy	0.01	4	N/A	Did not specify	Gaussian membership function
Parkinson & Callaghan, 2009	Densely connected feedforward	Did not specify	3	5	Early stopping. Patience was not stated.	Hidden and output layers: Linear
Arjmand et al., 2013	Densely connected feedforward	Did not specify	6	20	Early stopping with a patience of 6 iterations	Hidden layer: Tanh Output layer: Linear
Mousavi et al. 2020	Two RNNs, fuzzy logic algorithm	Did not specify	3	5	Did not specify	Hidden layer: sigmoid Input layer: Linear Output layer: Not discussed

2.13.1 Hou et al., 2004

Hou et al. (2004) demonstrated that modified RNNs (mRNNs) can predict lumbar spine compression as well as AP and mediolateral shear during a lifting task without direct consideration of EMG data. The mRNN input features included several anthropometric measures, load mass, lifting height, participant handedness and lifting style (stoop or squat), triaxial low back moments as well as angular displacement, velocity and acceleration of the trunk relative to the pelvis. At each time step, the mRNN produced intermediate EMG outputs. These EMG outputs were looped back to the mRNN at future timesteps as an additional input for generation of the lumbar spine loading outputs (Figure 16). The number of hidden nodes was set to 35 after experimentation with hidden layers of eight to fifty nodes. Similarly, a trial and error approach was applied to identify an appropriate learning rate which was 0.2.

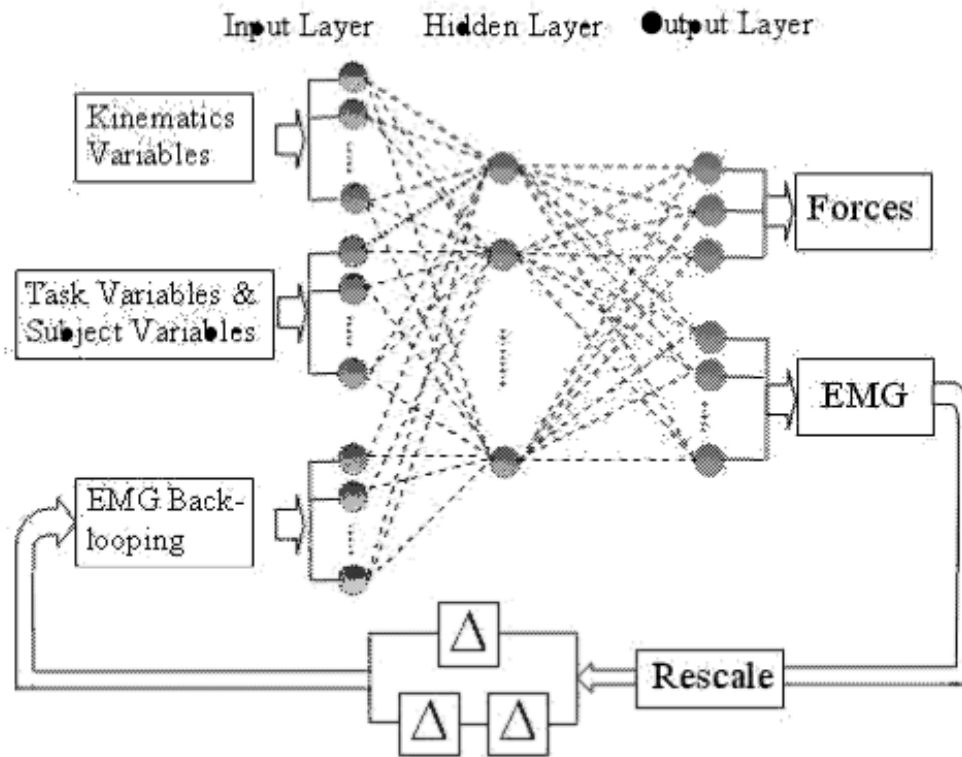


Figure 16: mRNN diagram from Hou et al. (2004). EMG outputs were rescaled such that they had similar numerical magnitudes as the other mRNN inputs. After being rescaled, they were looped back as an input to the mRNN at one and two timesteps in the future.

When the mRNNs were applied to a sagittal lifting task, mean absolute error (MAE) was 152 N, 60.3 N and 14.5 N for low back compression, AP shear and lateral shear joint contact forces respectively. MAE values were greater when the mRNN was applied to an asymmetric lifting task (Figure 17). It should be noted that these mRNNs were trained and tested on a sample of 2 male participants. The ability of these models to generalize to female and/or novel male participants was not assessed.

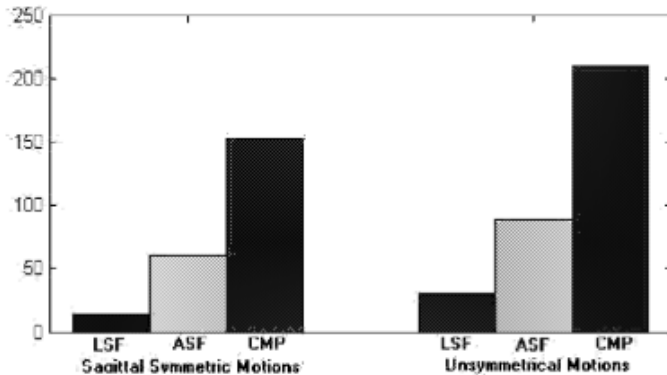


Figure 17: Figure from Hou et al., 2004 showing mean absolute error (N) of mRNN predictions of low back lateral shear (LSF), AP shear (ASF) and compression (CMP) during symmetric and asymmetric lifting tasks.

2.13.2 Hou et al., 2007

In a follow-up study, Hou et al. (2007) leveraged recurrent fuzzy neural networks (RFNN) (Figure 18) to predict lumbar compression, AP shear and mediolateral shear. Target compression and shear values were generated by an EMG-assisted model. The use of fuzzy logic was motivated by the noisy relationships between kinematics, EMG and contact forces. Briefly, fuzzy logic (Zadeh, 1966) allows models to mathematically incorporate the uncertainty inherent to real world phenomena. For example, when describing the temperature outside a human may define it as cold, warm or hot. However, some temperatures, for example 25°C could belong to two of these categories and be defined as “very warm” or “somewhat hot”. To model this, fuzzy logic would use a membership function to define how much a given numerical temperature value belongs to one of the three temperature categories (Figure 19). The process of determining the degree to which an input belongs to each of the

fuzzy sets (in this example, fuzzy sets are “cold”, “warm” and “hot”) is known as fuzzification.

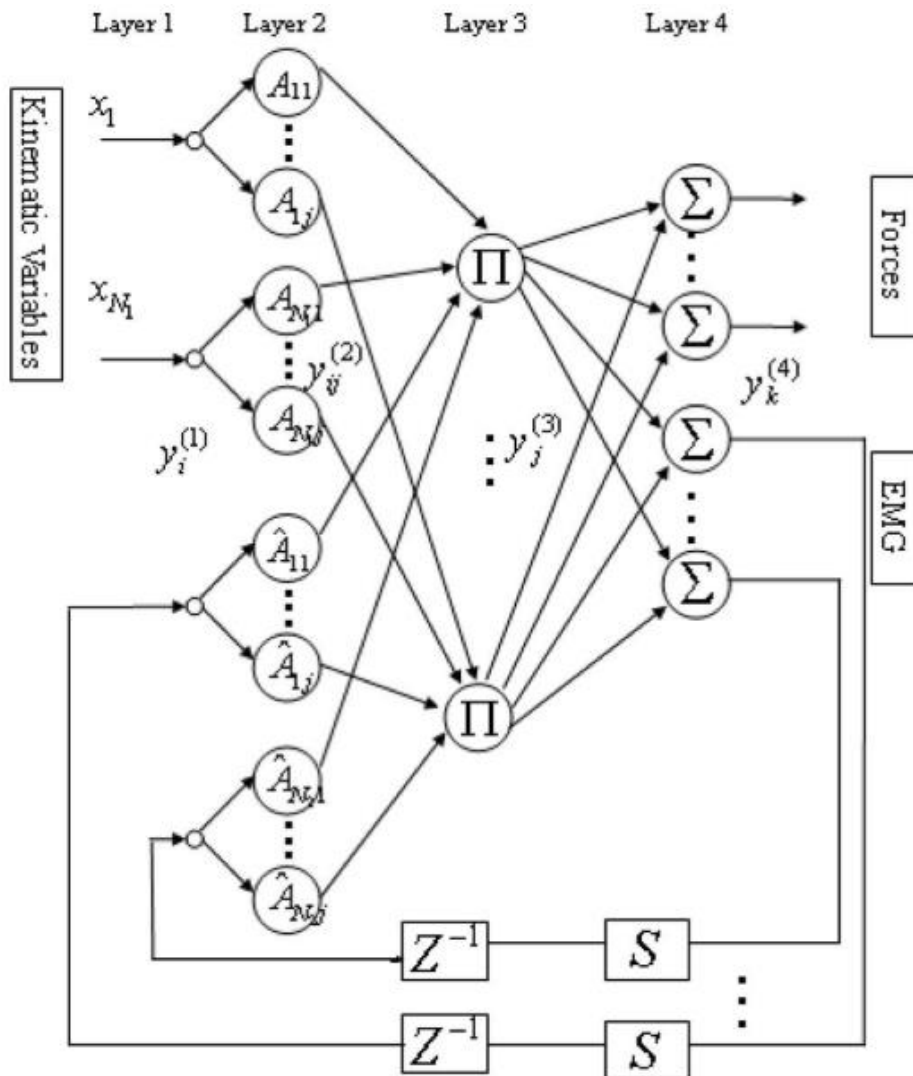


Figure 18: RFNN diagram from Hou et al., 2007. Layer 1 received kinematic inputs as well as recurrent EMG feedback. Layer 2 fuzzified these inputs i.e. converted the numerical inputs in to linguistic sets. Nodes in Layer 3 calculated the products of their inputs and output the firing strength of various fuzzy rules that aimed to model the

relationship between kinematics and EMG and lumbar forces. Layer 4 performed defuzzification in order to output numerical values.

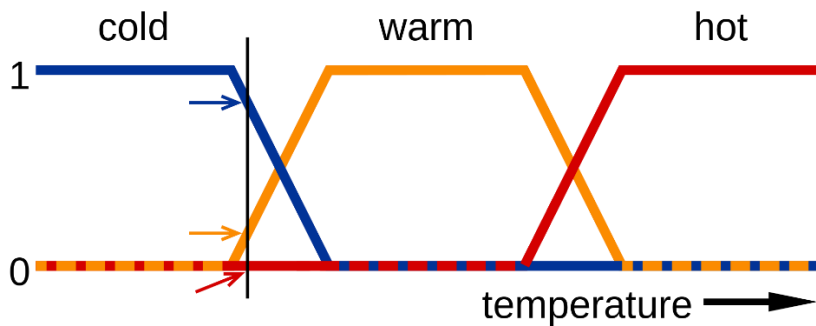


Figure 19: Graphical representation of the fuzzification of temperature values from https://commons.wikimedia.org/wiki/File:Fuzzy_logic_temperature_en.svg.

During training, the RFNN received triaxial angular trunk displacement, velocity, acceleration, moments as well as EMG signals from six trunk muscles. These data were used to learn kinematics-EMG-force relationships such that, when applying the trained model, EMG input was not required. Specifically, the trained model generated EMG predictions based on kinematic and moment inputs at time $t - 1$ then fed these EMG outputs to itself as an additional input to the model at time t . The interval between timepoints was approximately one ms and aimed to represent electromechanical delay. However, this is not aligned with the much longer electromechanical delays reported in trunk musculature (Stokes, 2005). RFNNs were trained for specific task conditions and subject groups. Subjects were grouped by standing height, forearm length and spine length. The method of measuring spine length was not discussed. Task conditions included combinations of different levels of

load mass, lift height and lift asymmetry. RFNNs were successful in generating similar lumbar compression, anteroposterior shear and mediolateral shear as an EMG-assisted model across sagittal and asymmetric lifts of various load masses and lift height. Results indicate that mean absolute error in compression and shear outputs was less than 10% in both lift types (Figure 20). Visual examination of RFNN outputs suggest that they had difficulties in accurately predicting peak joint contact forces (Figure 21).

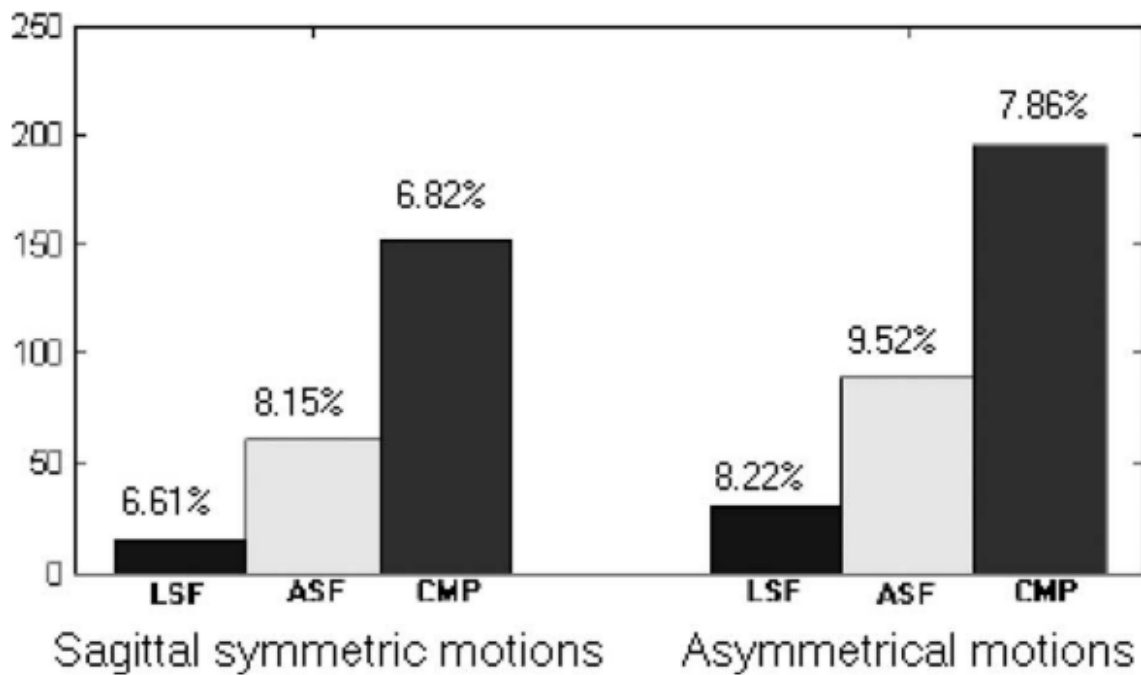


Figure 20: Mean absolute error (in N) and percentage error of RFNN vs. EMG-driven model mediolateral shear (LSF), AP shear (ASF) and compression (CMP) outputs during sagittal and asymmetrical lifting tasks. Figure from Hou et al. (2007).

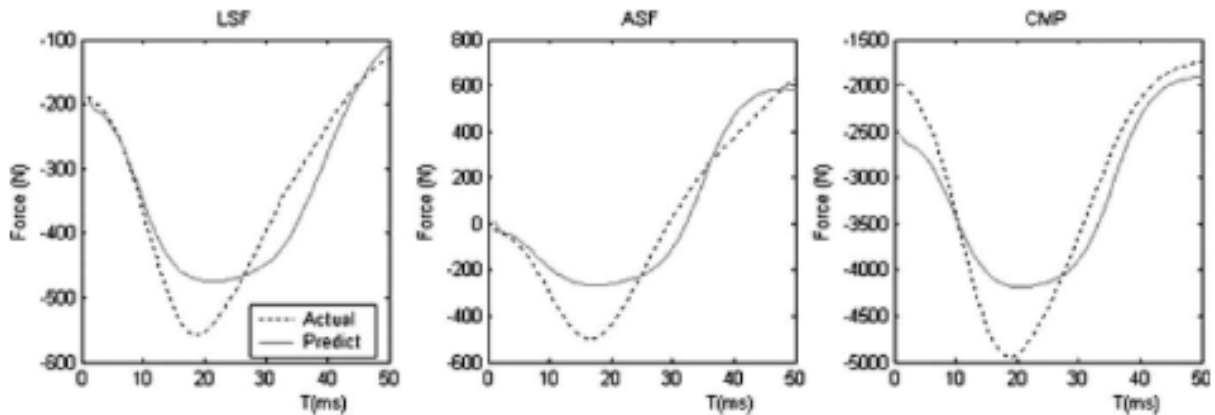


Figure 21: Comparison of RFNN (Predict) and EMG-driven model (Actual) mediolateral shear (LSF), AP shear (ASF) and compression (CMP) outputs during asymmetrical lifting.

2.13.3 Parkinson & Callaghan, 2009

Parkinson & Callaghan (2009) aimed to reduce the data requirements for estimation of lumbar spine loads during lifting and lowering trials with two load conditions and three lift asymmetry conditions (right, centre/sagittal, left). Two FF ANNs were developed, one to predict L5-S1 moments and the other to predict L5-S1 contact forces. This chapter will focus on the joint contact force ANN as that is more relevant to this thesis. Data was collected on two separate days from 20 participants and included ground reaction forces, motion capture and trunk musculature EMG. These data were used to drive an EMG-assisted model to generate compression and shear estimates at L5-S1.

A data reduction approach was performed in which the utility of candidate ANN input variables was assessed by calculating coefficients of determination between inputs and joint contact force outputs. This procedure yielded bilateral internal oblique and erector

spinae EMG, trunk flexion angle, participant height, mass, trunk depth and width at the level of the umbilicus and gender as inputs for the ANN. Three-layer ANNs were developed where the number of nodes in the hidden layer was selected through a trial and error process. This process began by developing ANNs with the maximum number (25) of hidden nodes that would allow for training. The number of hidden nodes was pared down until additional removal of nodes resulted in increased training error. At each step in this process, three networks were trained to ensure stable estimates of ANN performance. Nodes of both the hidden and output layers contained linear activation functions. The hidden layer activation functions were not reported.

Data from five participants were withheld to create a novel test set with which to test the generalizability of the ANN. The outputs of the ANN on this dataset were compared to corresponding EMG-assisted model outputs. Similar to the findings of Hou et al. (2007), statistical testing indicated that the ANN significantly underestimated peak joint contact forces (Figure 22). This was attributed to the use of a root mean squared difference (RMSD) loss function where alternative loss functions were theorized to improve peak error metrics at the cost of worsening cumulative error metrics. The ANN was successful in generating similar cumulative loads as the EMG-assisted model (Figure 23) on a group level. However, the authors note that the ANNs should only be applied to large scale studies, not estimates of exposure on an individual level as the ANN over and under-estimated contact forces at the single participant level.

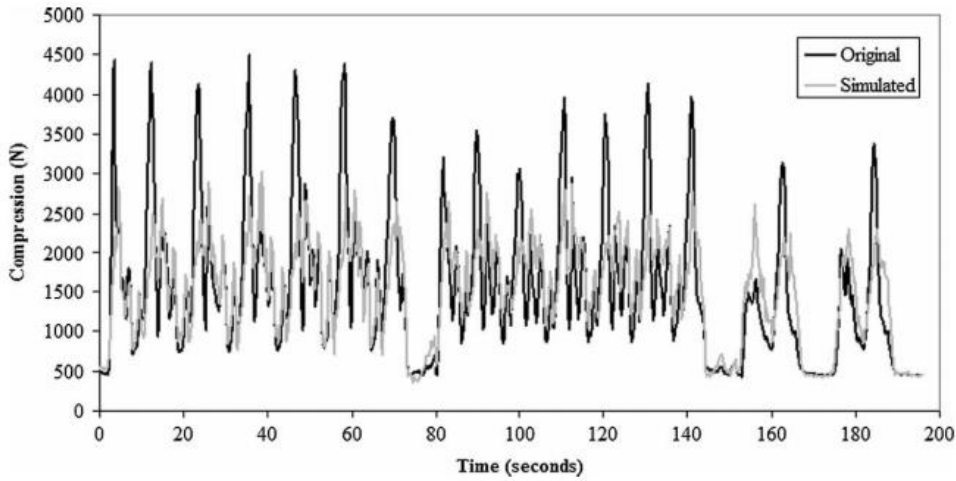


Figure 22: Comparison of ANN (simulated) predictions of L4/L5 compression to EMG-assisted (Original) model outputs from Parkinson & Callaghan, 2009. The authors note that the ANN was unable to replicate peak values.

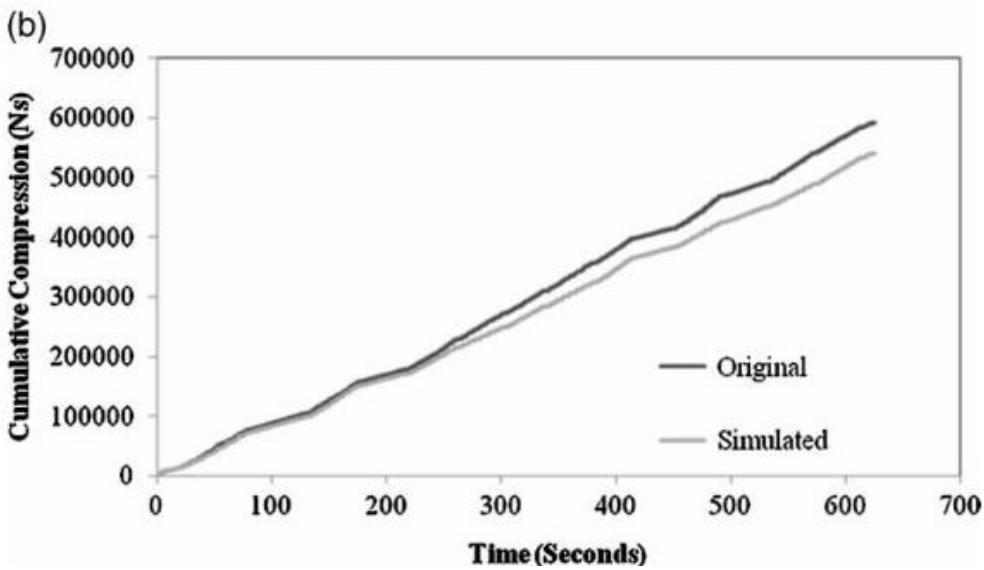


Figure 23: Cumulative compression as calculated by an EMG-assisted model (original) vs. FF ANN (simulated) over several lifts performed by one participant. Figure from Parkinson & Callaghan (2009).

2.13.4 Arjmand et al., 2013

Arjmand et al. (2013) aimed to compare the performance of ANNs and quadratic regression equations in matching the outputs of a kinematics-driven finite element (FE) model of the trunk (Figure 24). The outputs of the FE model included L4-L5 and L5-S1 compression and AP shear and considered 76 trunk muscle forces during a simulated sagittal plane lifting task. The inputs of the FE model were the mass of the load, the anterior and lateral distance of the load to the L5-S1 joint centre, whether the lift was performed with one or two hands as well as sagittal plane trunk and pelvis kinematics that were calculated from markers tracked by an active motion capture system. The FE model calculated trunk muscle forces through an optimization algorithm whose cost function was the sum of squared muscle stresses.

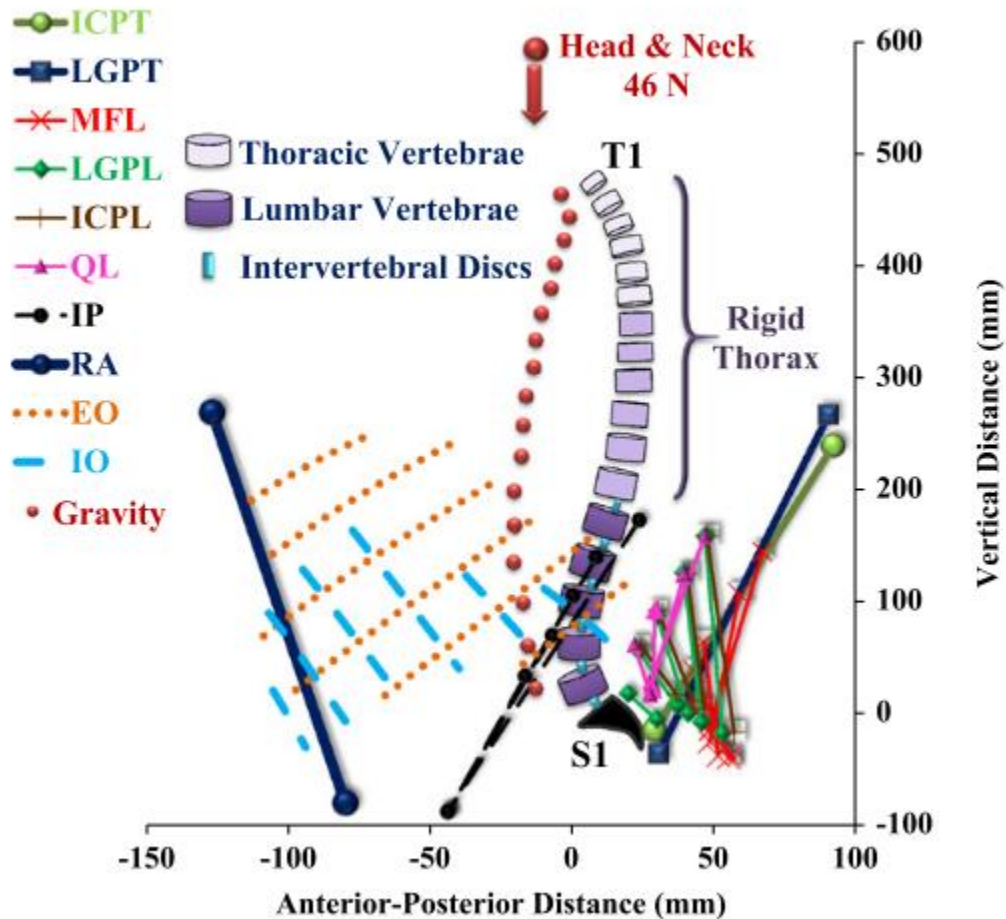


Figure 24: The kinematics-driven FE model of Arjmand & Shirazi-Adl (2006) that was applied by Arjmand et al. (2013). Modelled muscles included iliocostalis lumborum pars thoracic (ICPT), longissimus thoracis pars thoracic (LGPT), multifidus (MFL), longissimus thoracis pars lumborum (LGPL), iliocostalis lumborum pars lumborum (ICPL), quadratus lumborum (QL), iliopsoas (IP), rectus abdominus (RA), external oblique (EO) and internal oblique (IO).

Two ANNs were developed, one to predict FE model-derived compression and shear joint contact forces and the other to predict muscle forces. This summary will focus on the joint contact force ANN as that is more relevant to this thesis. The inputs of the joint contact

force ANN were thorax flexion angle, load mass and the anterior and lateral location of the mass relative to the L5-S1 joint centre. A “trial and error” approach was used to select ANN hyperparameters to minimize RMSE of compression and shear outputs relative to the FE model outputs. This process led to an ANN with 6 layers and 20 nodes per hidden layer. This ANN outperformed the corresponding quadratic regression equation and achieved an R^2 of 0.99 (Figure 25).

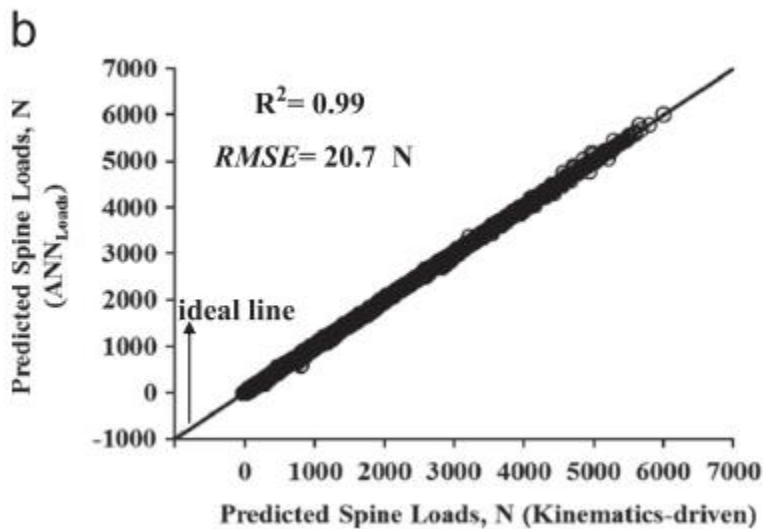


Figure 25: Scatter plot from Arjmand et al. (2013) showing ANN-predicted (y-axis) and kinematics-driven FE model (x-axis) outputs.

A limitation of this study is that the trunk was assumed to only move in the sagittal plane. Additionally, the optimization approach that was used to calculate muscle forces within the FE model resulted in zero force production by antagonistic muscles. This does not reflect EMG-based reports of co-contraction during trunk flexion/extension (Granata, Lee &

Franklin, 2005). Overall, this study demonstrated that ANNs are effective in mapping complex relationships between relevant biomechanical inputs and outputs.

2.13.5 Mousavi, Sayyaadi & Arjmand, 2020

Mousavi, Sayyaadi & Arjmand (2020) applied a neuro-fuzzy approach consisting of two RNNs and a fuzzy logic algorithm (Figure 26) to predict 3D orientations of the thorax and pelvis as well as L5-S1 compression and shear. This summary will focus on their application of an RNN to predict compression and shear. Twenty male participants performed a series of reaching tasks in which participants picked up weights that were located at a variety of heights and horizontal distances. Thorax and pelvis orientations were tracked with inertial sensors and L5-S1 compression and AP shear were calculated via a kinematics-driven FE model of the thoraco-lumbar spine (Arjmand & Shirazi-Adl, 2006). These data were split such that 70% of data were used for RNN training, 15% was used for RNN validation and 15% was used for testing.

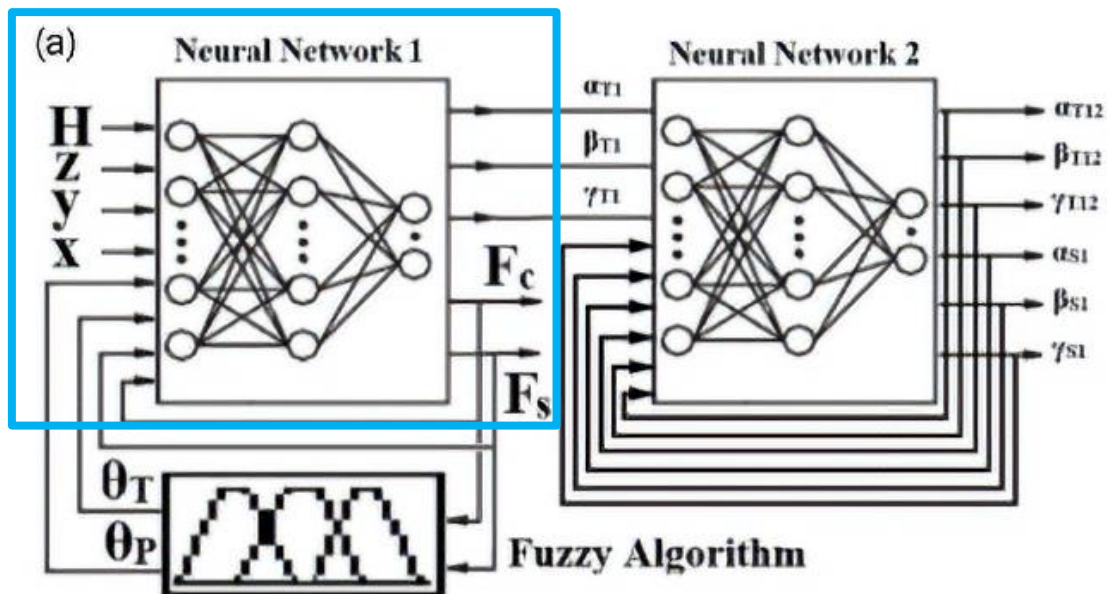


Figure 26: Schematic adapted from Mousavi et al. (2020) illustrating the structure of their neuro-fuzzy approach. Aspects pertaining to prediction of lumbar compression and shear are highlighted in blue. Inputs to the first RNN were participant standing height, H , and the anteroposterior, mediolateral and inferosuperior distances (x , y , and z) of the load relative to a point on the floor on which participants were standing.

An RNN with one hidden layer of 5 nodes, an input layer with 6 nodes and an output layer consisting of 2 nodes was developed to predict L5-S1 prediction and shear. Network inputs were participant height, the starting height of the load and it's horizontal anterior and mediolateral distances from the participant as well as pelvis and thorax flexion angles. Nodes in the hidden layer were assigned sigmoid activation functions while the input layer was assigned linear activation functions. The activation functions of the output layer were not discussed. The network was trained using the Levenberg-Marquardt algorithm (Moré, 1978). Comparison (Figure 27) of the RNNs developed in this study to FF ANNs developed by

Gholipour & Arjmand (2016) to predict lifting postures suggests that RNNs may be advantageous when estimating time-series intervertebral compression and shear.

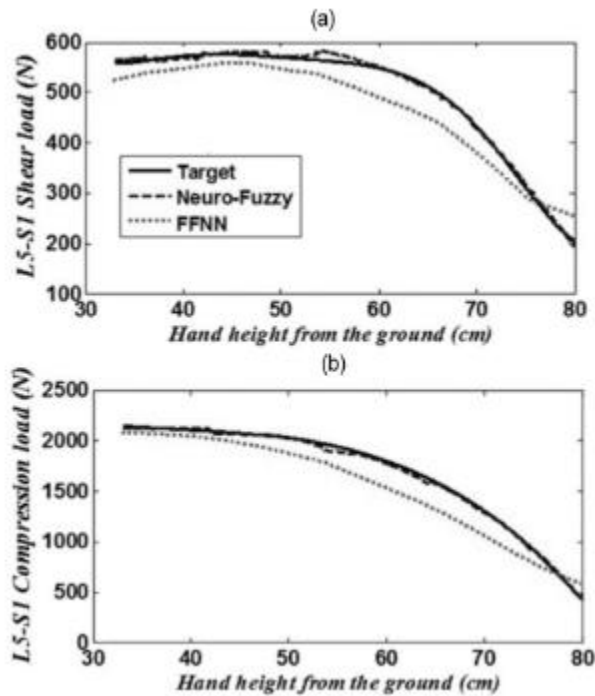


Figure 27: Compression and shear outputs of the RNNs (Neuro-Fuzzy) developed by Mousavi et al. (2020), the outputs of a FF ANN (FFNN) developed in a previous study (Gholipour & Arjmand, 2016) and corresponding FE model-derived values (Target) when lifting a load. Figure from Mousavi et al (2020).

2.14 Summary

This literature review provided several key considerations that inform the methodology of this thesis. First, while FF ANNs have been successful in predicting cumulative loading of the lumbar spine, RNNs are likely more well-suited to accounting for interdependencies

between timepoints within a sequence to predict time-series intervertebral compression and shear forces. Second, ANN development should include evaluation of ANNs with a wide range of hyperparameters to identify an appropriate balance between bias and variance. To mitigate the risk of overfitting, early stopping and dropout regularization are appropriate regularization techniques. Third, developing ANNs for specific motions (e.g. symmetric vs. asymmetric lifting) is likely to provide stronger ANN performance. Fourth, to appropriately assess ANN generalizability, it is important that ANNs are tested on data from participants whose data were not included in either training or validation sets. Additionally, MSE or r values are appropriate measures of model performance in this context. Each of these points played an important role in shaping the ANN development process described in Chapters 3.4-3.6.

Chapter 3: Methods

A variety of ANNs were systematically developed with different types of layers (FF and LSTM), inputs and hyperparameters. These ANNs produced time-series C6-C7 compression and shear prediction outputs. The motion capture and EMG data underlying the ANNs developed in this thesis were collected and processed by Laura Healey (Healey et al., 2021; Healey, 2019). These data were input to an EMGD musculoskeletal model of the neck (Barrett et al., 2021; Barrett, 2016) which output time-series compression and shear joint contact forces. These outputs were generated by Jeffery Barrett and used as target values for the ANNs in this thesis.

3.1 Participants

Participants ($n = 26$) included 14 females and 12 males (mean \pm standard deviation; age: 24.7 ± 4.0 years, height: 173.7 ± 9.4 cm, and body mass: 77.9 ± 18.1 kg) from a university population. All participants provided informed written consent and collection of this data was approved by the University of Waterloo office of research ethics (ORE # 40080).

3.2 Motion Capture and EMG Data Collection

3.2.1 Visual Target Acquisition System

Participants performed flight-relevant visual scanning trials using the 3D Visual Target Acquisition System (VTAS) (Derouin & Fischer., 2019). The VTAS is a validated system to elicit consistent head movement similar to that of a helicopter pilot during flight. It consists of

solar panels (6V 100mA, 100mm diameter, Sundance Solar, Hopkinton NH) that interact with helmet-mounted laser pointers. These solar panels are encased in a plastic frame with a 20 mm aperture which represent targets to be acquired. Mounted on the frame, are LEDs capable of emitting red, blue and green light. Participants don helmets to which a laser pointer is affixed such that the laser aligned with the centre of vision of the participant. When the laser beam interacts with a solar panel, the surrounding LEDs change colour from red to blue. When the laser has maintained contact for 300 ms, the LEDs turns green indicating that this visual target has been acquired (Figure 28).

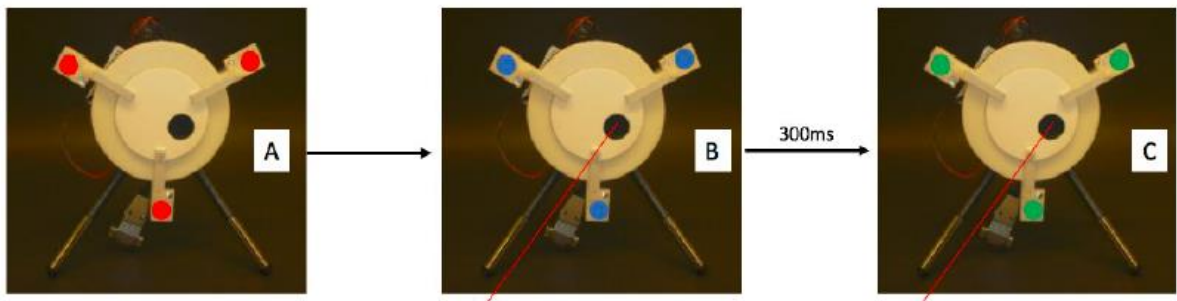


Figure 28: Figure, adapted from Healey (2019), demonstrating the changes in colour of VTAS LEDs when the laser is not in contact with it (A), when target is initially exposed to the laser pointer (B) and when the laser pointer has been in contact with the target for greater than 300ms (C).

3.2.2 Experimental Design

Flight-relevant visual scanning was elicited by asking participants to perform 30-second scanning trials consisting of rapidly and reciprocally moving between pairs of targets. Participants performed three scanning trials under each combination of two rotation axis conditions (pitch and yaw), two magnitude conditions (35° and 70°) (Figure 29) and four

helmet conditions: helmet (Gentex HGU-56/P [Gentex, Carbondale, Pennsylvania]) only (hOnly), helmet with NVG (hNVG), helmet with NVG and counter-weight (hNVGCW) and helmet with NVG and counter-weight liner (hCWL) (Table 2). The order in which conditions were presented was block randomized. Appropriate rest was provided to prevent participants from fatiguing throughout the protocol. During all trials, participants sat in an automobile seat with a four-point harness to mimic seating in a helicopter.

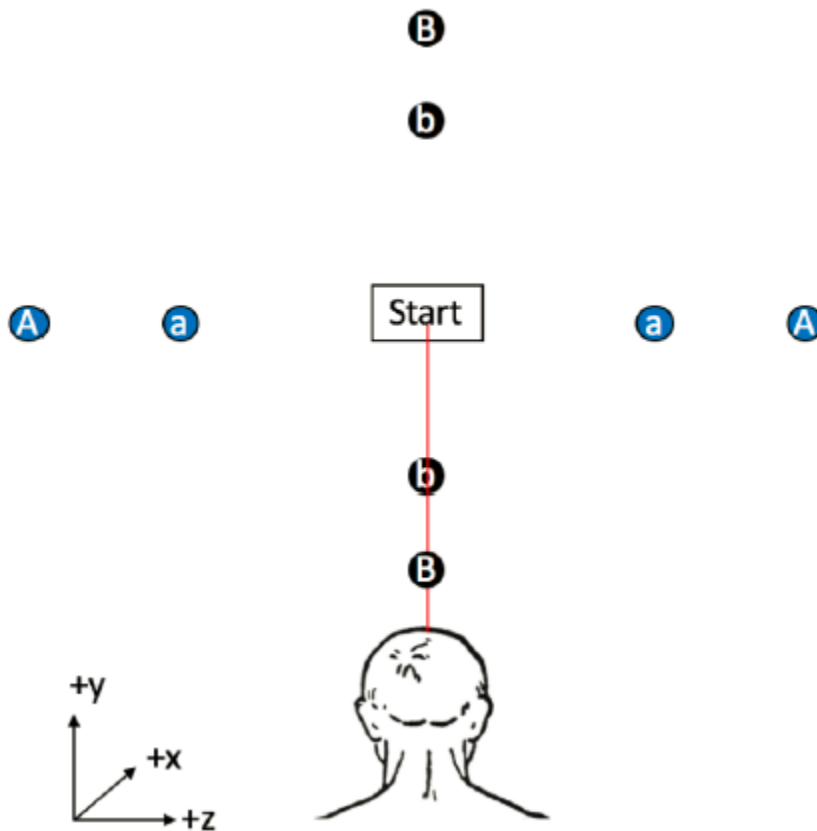
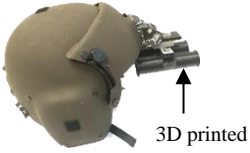
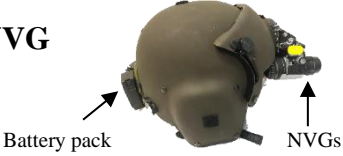
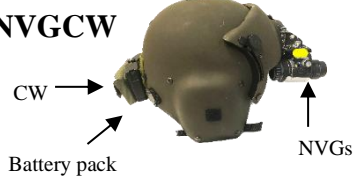
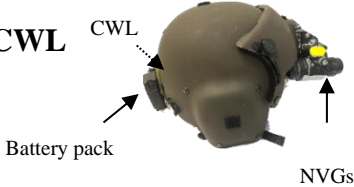


Figure 29: VTAS target locations adapted from Healey (2019). “A” and “a” denote targets used to elicit yaw head movements of 70° and 35° respectively. Similarly, “B” and “b” denote targets used to elicit pitch head movements of 70° and 35°.

Table 2: Summary of the relative inertial properties of each helmet configuration adapted from Healey et al. 2021.

Helmet Condition	Relative Increase in Total Mass (%)	Relative Increase in Moment of Inertia (%)
hOnly  <p>3D printed</p>	Baseline	Baseline
hNVG  <p>Battery pack</p> <p>NVGs</p>	54.5	Lateral bend = 12.2 Rotation = 56.1 Flex/ext = 63.1
hNVGCW  <p>CW</p> <p>Battery pack</p> <p>NVGs</p>	96.5	Lateral bend = 18.2 Rotation = 71.5 Flex/ext = 77.9
hCWL  <p>CWL</p> <p>Battery pack</p> <p>NVGs</p>	96.5	Lateral bend = 15.0 Rotation = 65.3 Flex/ext = 71.7

3.2.3 Instrumentation

EMG was sampled bilaterally at a frequency of 2000 Hz from sternocleidomastoid, upper trapezius and upper neck extensors using wireless Trigno mini sensors (Delsys, Natick MA). EMG data were linear enveloped where filter cut-off frequencies were specific to each muscle in consideration of their electromechanical delays (Almosnino et al., 2009). Participants were also instrumented with passive-reflective markers on the trunk (Figure 30) and helmet (Figure 31). The 3D positions of these markers were sampled at a frequency of 80 Hz using a 12-camera Vicon optical motion capture system (Vicon, Oxford, UK). To mitigate potential learning effects, only data from the third (and final) trial within each combination of helmet, rotation axis and magnitude conditions was processed further. First, marker position data was labelled and gap filled in Nexus 2.0 (Vicon, Oxford, UK). These data were then imported into MATLAB R2018a (Mathworks Inc., USA) where head and trunk segments were defined in accordance with International Society of Biomechanics recommendations (Wu et al., 2002). Euler angles were calculated to yield angular displacement of the head relative to the trunk. Angular velocity was calculated using the “gradient” function in MATLAB 2019b which applies finite differentiation to the first and last points in a sequence and central difference differentiation to all other data points. Angular velocity was differentiated in the same manner to yield angular acceleration.

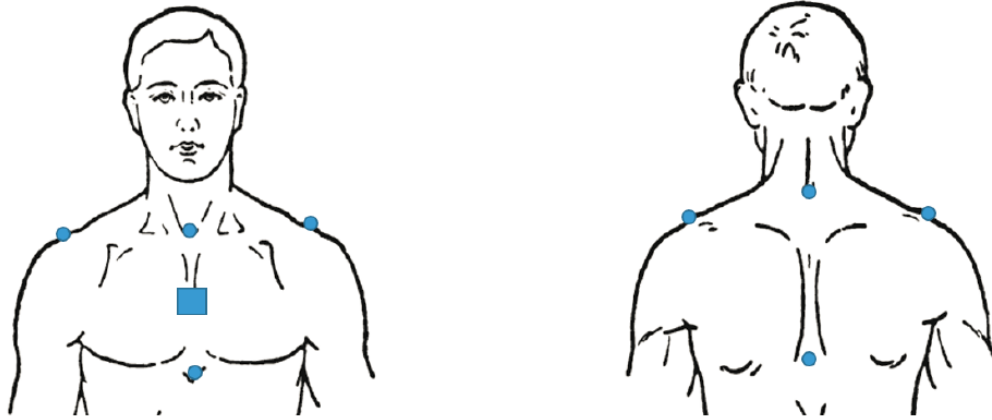


Figure 30: Reflective marker (circle) and rigid body marker cluster (square) placement on the trunk. Figure from Healey (2019).



Figure 31: Reflective marker placement on the helmet. Figure from Healey (2019).

3.3 Estimating C6-C7 Compression and Shear with an EMG-Driven Model

Conditioned EMG and head-trunk kinematics data were down sampled to 10 Hz and input into a validated EMGD musculoskeletal model of the neck (Barrett et al., 2021) (Figure 32). The

outputs of the model included intervertebral moments calculated using inverse dynamics and compression and shear joint contact forces in consideration of both passive and active musculoskeletal force contributions. This model included rigid bodies representing the seven cervical vertebrae, T1 and the skull as well as 218 muscle elements and 511 ligament elements spanning from C7-T1 to the base of the skull. Active muscle force contributions were calculated using an EMG-assisted optimization routine (Gagnon et al., 2011; Cholewicki & McGill, 1994). The helmet system was represented within the model as its own rigid body with inertial parameters (mass and moment of inertia) obtained from CAD models of each helmet configuration. They were then attached to the skull-segment with a weld joint, with the center of mass offset relative to the head's center of mass specific to the helmet configuration. The anatomical dimensions of the model were scaled to match a 50th percentile male.

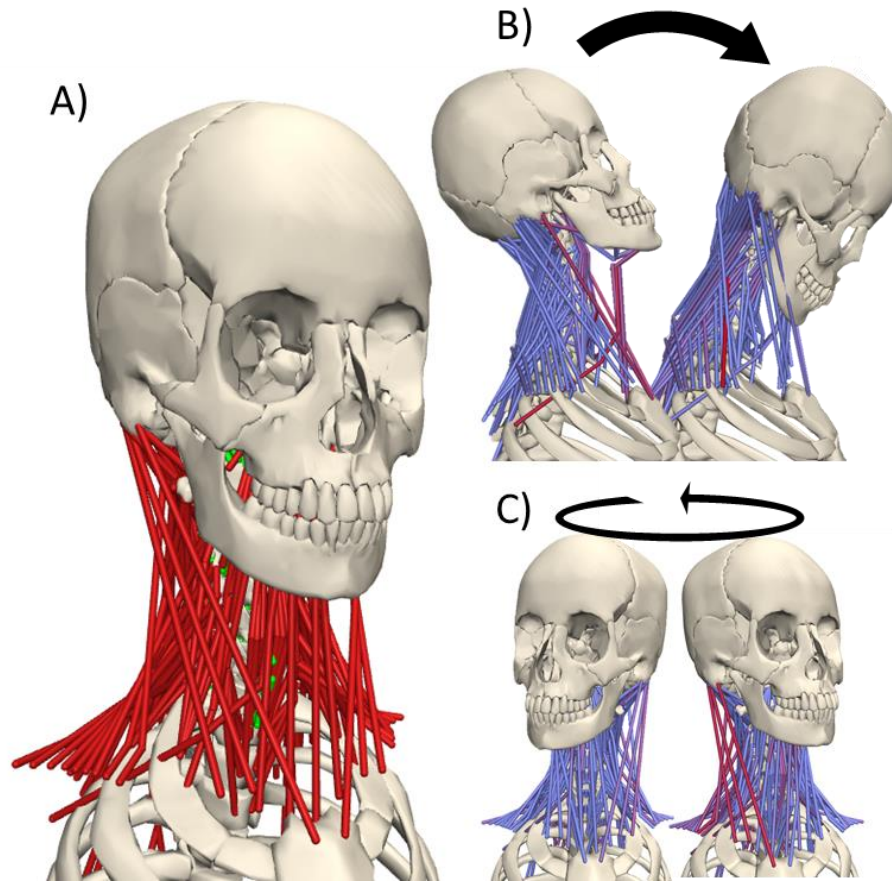


Figure 32: EMG-driven model of the neck adapted from the OpenSim model of Barrett et al. (2021). (A) the model in a neutral posture with muscle elements represented in red, and ligaments in green. (B) demonstration of the pitch rotation kinematics, which involved reciprocal flexion-extension (flexion shown here). (C) the axial-twisting yaw rotation. The left image in (C) is the static portion of the trial, whereas the image on the right is taken during the accelerating phase of the movement, hence the notable activation in the sternocleidomastoid.

Model validation, as described by Barrett et al., 2021, was performed in two stages: validation of the osteoligamentous elements of the model and validation of the passive muscle tension contributions. The overarching model was composed of individual functional spinal

unit (FSU) models where ligament force-elongation curves were based on the work of Mattuci et al. (2013) and the contributions of the intervertebral disc and zygapophyseal joints were represented by non-linear bushing elements (Christophy et al., 2013). The resultant passive-moment angle curve of each FSU was then calibrated with experimental data in flexion-extension (Camacho et al., 1997; Panjabi et al., 1988) lateral-bending (Yoganandan et al., 2007) and axial-rotation (Yoganandan et al., 2008). The kinematic response of the osteoligamentous cervical spine model to applied forces was validated against experimental data (Ivancic, 2013; Panjabi et al., 2001). Next, the 218 muscle elements were added to the model and the whole spine's passive moment angle curve was validated against equivalent *in vivo* passive curves, where participants were moved through their range-of-motion by an externally applied force (McClure et al., 1998; McGill et al., 1994).

Alternative EMGD neck models, namely those of Vasavada et al. (1998) and Huber (2013) have been developed to serve the same purpose as that of Barrett, (2016). However, issues related to nonlinearities between EMG drive and subsequent muscle force generation have been identified in the Vasavada model (Netto et al., 2008). Meanwhile, the Huber (2013) model includes just 3 neck muscle elements whose paths were not modelled with via points for muscle wrapping. In light of these limitations, and the extensive validation of the Barrett (2016) model, it is likely the best available model for use in this study.

3.4 Artificial Neural Network Development

A dataset consisting of head-trunk angular kinematics (Chapter 3.2), and corresponding EMGD model-derived C6-C7 triaxial moments, compression and shear (Chapter 3.3) was created and then split such that the data collected from 16/26 participants were exclusively used for ANN training, 5/26 were exclusively used for ANN cross validation and 5/26 were exclusively used for ANN testing. Each dataset contained data from a roughly equal number of male and female participants.

Custom Python (v3.7.7, Python Software Foundation., DE, USA) scripts were written to develop ANNs using TensorFlow (Abadi et al., 2016) and Keras (Cholet et al., 2015) software libraries. Python is an open-source language and resultant models (stored in a Python-specific file format) are thus easily accessible without major financial barriers. The use of Keras and TensorFlow libraries simplified the creation of ANN development pipelines by eliminating the need to write custom code for certain aspects of the development pipeline.

It is not feasible to identify the exact combination of ANN type, input and hyperparameters that yield the absolute best possible results. This is because there is a virtually infinite number of possible ANN parameter combinations that can be tested. Therefore, an extensive two-stage hyperparameter tuning process was completed to select an ANN from a range of reasonable hyperparameter values. This two-stage approach was performed to reduce overall computational demands by first identifying a promising network type (e.g. LSTM vs. FF) and architecture before tuning other hyperparameters. In the first

stage of this process, I aimed to identify the most promising combination of a variety of ANN types, architectures and input features. To do so, FF ANNs, LSTM ANNs and ANNs consisting of a combination of LSTM and FF layers were developed with a variety of input features and network architectures. The performance of these ANNs was used to inform selection of the most promising combination of ANN type, inputs and architecture. In stage two of the hyperparameter tuning process I aimed to identify a combination of learning rate, dropout rate and mini-batch size values that further improved ANN performance.

3.5 Hyperparameter Tuning Stage One: Selecting ANN Inputs, Type of ANN and ANN Architecture

3.5.1 Hyperparameter Tuning Stage One Methods

Three groups of ANNs were developed. “Dual-rotation” ANNs were trained and evaluated on both pitch and yaw head movement trials. “Labelled-trial” ANNs were also trained and evaluated on both pitch and yaw trials. However, these ANNs received binary trial labels as additional inputs. “Rotation-specific” ANNs were trained and evaluated exclusively on data from pitch or yaw trials. Within each of these ANN groups, five different sub-groups of ANNs (Table 3) were developed. These sub-groups are defined by the type of layers in the ANNs as well as their input features. The sub-groups contained ANNs consisting of only densely connected feedforward layers, only stacked LSTM cells, and ANNs consisting of an LSTM cell followed by FF layers (Figure 33). ANN inputs were either head-trunk angular displacement, velocity and acceleration as well as tri-axial C6-C7 moments or head-trunk displacement and C6-C7 moments. In labelled-trial ANNs, trial labels were additional one-

hot encoded inputs where values of zero or one were concatenated to each frame of data within pitch and yaw trials respectively. One-hot encoding is a method of converting categorical data to numerical data such that ANNs are able to receive this information as an input feature. The inputs and ANN layer types within each sub-group are presented in Table 3.

The decision to investigate the use of head-trunk angular displacement and C6-C7 moments as sole inputs was made to indirectly examine whether LSTM ANNs would account for velocity and acceleration information by recalling displacement inputs across timesteps. This would be advantageous as using fewer input features would reduce the likelihood of overfitting (Reed & Marks 1999), reduce the associated computational demands and simplify end-user (e.g. helicopter helmet design team) usage. The outputs of all ANNs developed in this thesis are time-series C6-C7 compression and AP shear.

Table 3: Description of each of the five ANN sub-groups that were developed. “ALL” subscripts denote ANNs who received head-trunk angular displacement, velocity, acceleration and moment inputs. “DM” subscripts denote ANNs that received head-trunk angular displacement and moment inputs.

Sub-Group	Type of layers	Inputs
FF _{ALL}	Only feed-forward	Tri-axial head-trunk angular displacement, velocity, acceleration and C6-C7 moments
LSTM _{ALL}	Only LSTM	Tri-axial head-trunk angular displacement, velocity, acceleration and C6-C7 moments
LSTM-FF _{ALL}	LSTM and feed-forward	Tri-axial head-trunk angular displacement, velocity, acceleration and C6-C7 moments
LSTM _{DM}	LSTM only	Tri-axial head-trunk angular displacement and C6-C7 moments
LSTM-FF _{DM}	LSTM and feed-forward	Tri-axial head-trunk angular displacement and C6-C7 moments

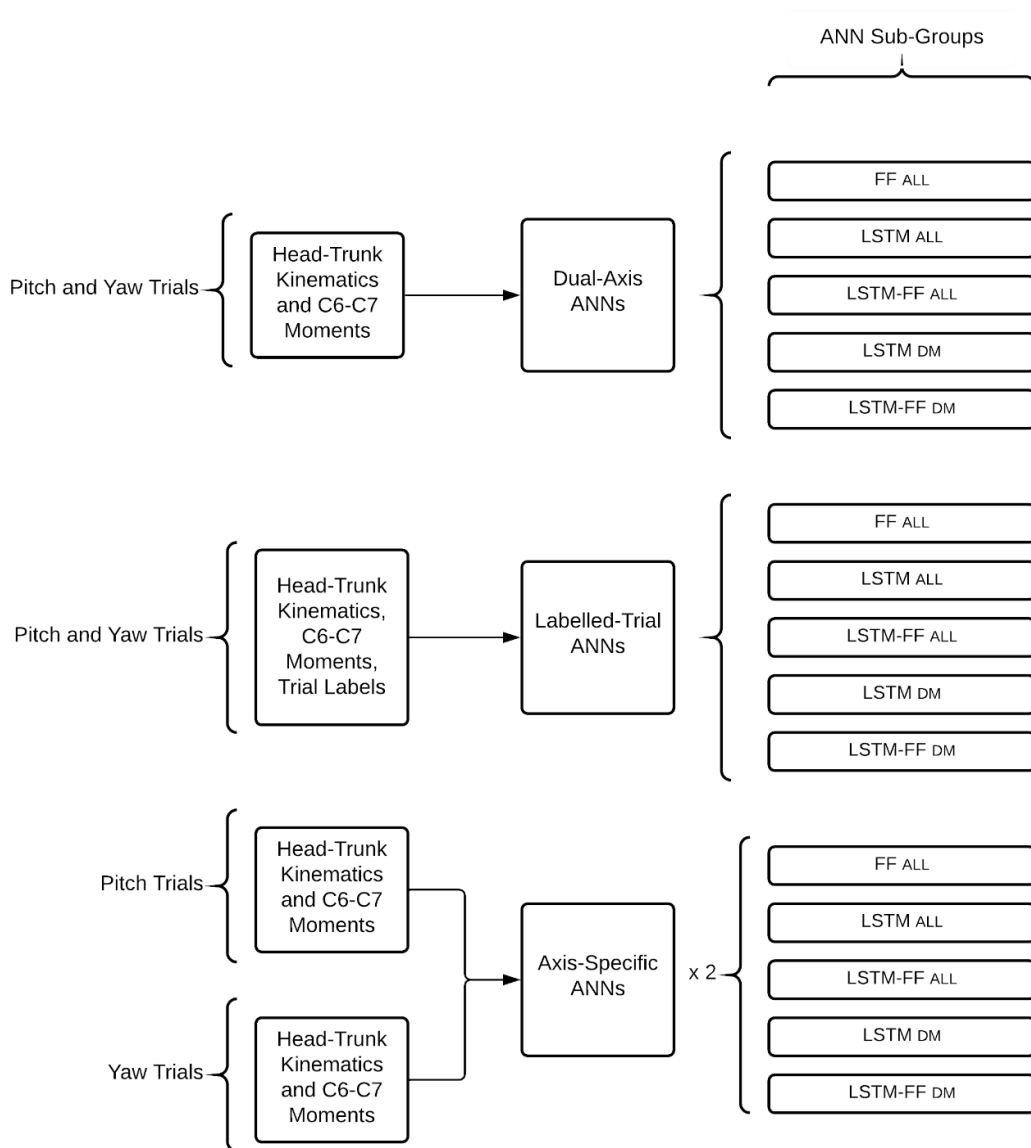


Figure 33: Overview of the groups and sub-groups of ANNs developed in the first stage of hyperparameter tuning.

Within each ANN sub-group, 75 ANNs were developed with a combination of different numbers of hidden layers and numbers of nodes per layer. The number of hidden layers was iterated from one to five for FF and LSTM-FF ANNs. In the case of LSTM ANNs, ANNs consisting of between one and five LSTM layers followed by a two-node FF output layer were developed. This range of hidden layers was selected as it spanned the range observed in relevant literature (Table 1). The number of nodes per hidden layer was set to either 16, 32, 64, 128 or 256 in consideration of Ng et al. (n.d.). Implementing a constant number of nodes across hidden layers is consistent with expert recommendations (Bengio, 2012). At this stage, each ANN was assigned the same default hyperparameters (Table 4) based on general recommendations.

Table 4: Default hyperparameters applied to all ANNs during first hyperparameter tuning stage.

Hyperparameter	Value
Dropout rate	0.5, as recommended by Srivastava (2014)
Early stopping patience	6, as recommended by Prechelt (2016)
Learning rate	0.01, as recommended by Bengio (2012)
Mini-batch size	10

All hidden and input layers were assigned hyperbolic tangent activation functions. This allowed for positive and negative node activation values which aligned with the fact that ANN inputs included both positive and negative values. Output layers consisted of two nodes which were assigned linear activation functions as was done by Lu et al. (2013) and Parkinson & Callaghan (2009). The use of linear activation functions in the output layer is likely beneficial in regression contexts as it allows for an unbounded range of possible output values. In contrast, other activation functions constrain values to within a given range (e.g. tanh outputs range from -1 to 1). Input layers consisted of 12 or 13 nodes as the ANNs had 12 or 13 inputs. With the exception of trial label input features, each input and output feature was individually scaled to the range of -1 to 1 where the maximum value in the training set was set to 1 and the minimum value was set to -1 (Halilaj et al., 2018). Ensuring that all ANN input features are within the range of -1 to 1 ensures that each feature has equal opportunity to contribute to the loss function which in turn improves training speed (Bishop, 1995). Weights were initialized through Glorot initialization (Glorot & Bengio., 2010) while biases were initialized as zero (Bengio, 2012). Briefly, Glorot initialization sets weights such that they are normally distributed with a mean of zero. This initialization method was selected as it has been demonstrated to improve both training speed and network performance relative to traditional uniform initialization (Glorot & Bengio, 2010). All ANNs were trained using the Adam optimization algorithm (Kingma & Ba, 2015) where minimization of MSE was the objective function. The Adam optimization algorithm is the recommended default learning algorithm in deep learning applications (Karpathy et al. 2016).

As in Parkinson & Callaghan (2009), three ANNs with each combination of architectural parameters were trained to ensure stable estimates of performance. This is required because glorot initialization yields pseudo-random initial weights and biases. The initial weighting of a given node influences how future training iterations will adjust the weight/bias of the node. Therefore, two ANNs with the same architecture and training parameters, but different initial node weightings, will likely have different weight and bias configurations upon completion of training. This results in varying levels of predictive performance between networks with the same hyperparameters.

Consistent with best practices (Halilaj et al., 2018), the performance and generalizability of each ANN was evaluated by calculating the MSE of predicted time series compression and shear relative to corresponding EMGD model outputs on the validation set. A composite error score for each network was calculated by dividing the MSE of predicted compression and shear values by the mean of the corresponding EMGD model-derived values and then summing the quotients. This allowed for error within both compression and shear outputs to be considered within a single error metric

3.5.2 Hyperparameter Tuning Stage One Results

3.5.2.1 Comparing Dual-rotation, Labelled-Trial and Rotation-specific ANNs

Comparison of the dual-rotation and labelled-trial ANNs suggests that including trial labels improved predictive performance. In fact, both the mean and minimum composite error scores of labelled-trial ANNs were lower than those of dual-rotation ANNs across all sub-

groups (Table 5). This was to be expected as the identification of trials as pitch or yaw may assist ANNs in accounting for differences in muscle activation profiles that would in turn influence joint contact forces.

Table 5: Summary statistics of composite error scores within each ANN sub-group. The best performing ANNs within each group are highlighted in bold text.

	Sub-Group	Mean Composite Error Score (SD)	Minimum Composite Error Score
Dual-Rotation	FF _{ALL}	28.2 (3.5)	24.6
	LSTM _{ALL}	29.6 (3.5)	25.2
	LSTM-FF_{ALL}	25.0 (1.2)	23.6
	LSTM _{DM}	30.3 (3.4)	26.2
	LSTM-FF _{DM}	26.5 (1.3)	24.0
Labelled-Trial	FF _{ALL}	25.8 (3.3)	22.0
	LSTM _{ALL}	27.6 (4.1)	21.2
	LSTM-FF_{ALL}	23.0 (1.6)	20.3
	LSTM _{DM}	29.2 (3.7)	24.4
	LSTM-FF _{DM}	25.1 (1.6)	22.7
Pitch-Specific	FF _{ALL}	30.0 (5.7)	24.6
	LSTM _{ALL}	35.4 (6.8)	25.5
	LSTM-FF_{ALL}	26.6 (2.5)	23.5
	LSTM _{DM}	37.0 (6.1)	30.1
	LSTM-FF _{DM}	30.2 (2.3)	26.8
Yaw-Specific	FF _{ALL}	19.3 (1.8)	17.0
	LSTM _{ALL}	24.9 (3.1)	18.7
	LSTM-FF_{ALL}	17.7 (1.1)	16.2
	LSTM _{DM}	24.7 (2.2)	20.3
	LSTM-FF _{DM}	19.1 (1.2)	17.6
Mean of Yaw and Pitch-Specific	FF _{ALL}	24.6 (6.8)	20.8
	LSTM _{ALL}	30.1 (7.4)	22.1
	LSTM-FF_{ALL}	22.1 (4.8)	19.9
	LSTM _{DM}	30.9 (7.7)	25.2
	LSTM-FF _{DM}	24.6 (5.9)	22.2

While the use of a single model for application to both pitch and yaw head movements would simplify the use of ANNs in the helmet design process, rotation-specific ANNs outperformed the dual axis and labelled-trial ANNs. Yaw-specific ANNs achieved lower mean and minimum composite error scores than both the dual-rotation and labelled-trial ANNs. The pitch-specific ANNs had greater composite error scores than the labelled-trial ANNs. However, this is due to the magnitude of compression and shear values being greater during pitch movements. The mean of the composite error scores of both pitch and yaw-specific ANNs indicate that the combined use of pitch and yaw-specific ANNs yields better overall performance than the labelled-trial ANNs (Table 5).

3.5.2.2 Comparing ANN Sub-Groups (Types of Layers and Inputs)

Within the dual-rotation, labelled-trial and rotation-specific groups, ANNs within the LSTM-FF_{ALL} (LSTM cell followed by hidden FF layers with displacement, velocity, acceleration and moment inputs) sub-group had the lowest mean and minimum composite error scores (Table 5) as well as the lowest standard deviation, indicating consistently strong performance. In the dual-rotation, labelled-trial, and yaw-specific groups, LSTM-FF_{DM} ANNs had the next best mean composite error score. In contrast, FF ANNs had the second best performance in the pitch-specific ANN group. The results discussed in this chapter motivated a focus on further assessment of LSTM-FF_{ALL} ANNs in subsequent analysis of hyperparameter tuning results.

3.5.2.3 Comparing Architectural Parameters

ANN architectural parameters were evaluated by calculating the composite error scores of each LSTM-FF_{ALL} ANN architecture configuration within each of the dual-rotation, labelled-trial and rotation-specific groups. With the exception of the dual-rotation ANN group, the best-performing ANNs had 256 nodes per hidden layer. In the dual-rotation ANN group, ANNs with 16 nodes per hidden layer performed best. The number of hidden FF layers that provided the strongest performance differed across ANN groups. In the dual-rotation group, ANNs with one hidden layer had the strongest performance. In pitch and yaw-specific groups, ANNs with 3 hidden layers performed best. In the labelled-trial ANN group, ANNs with 4 hidden layers had the strongest performance. The performance levels of LSTM-FF_{ALL} ANNs are presented in Figure 34. Detailed depictions of ANN performance, inclusive of the other 4 ANN sub-groups, are presented in Appendix A.

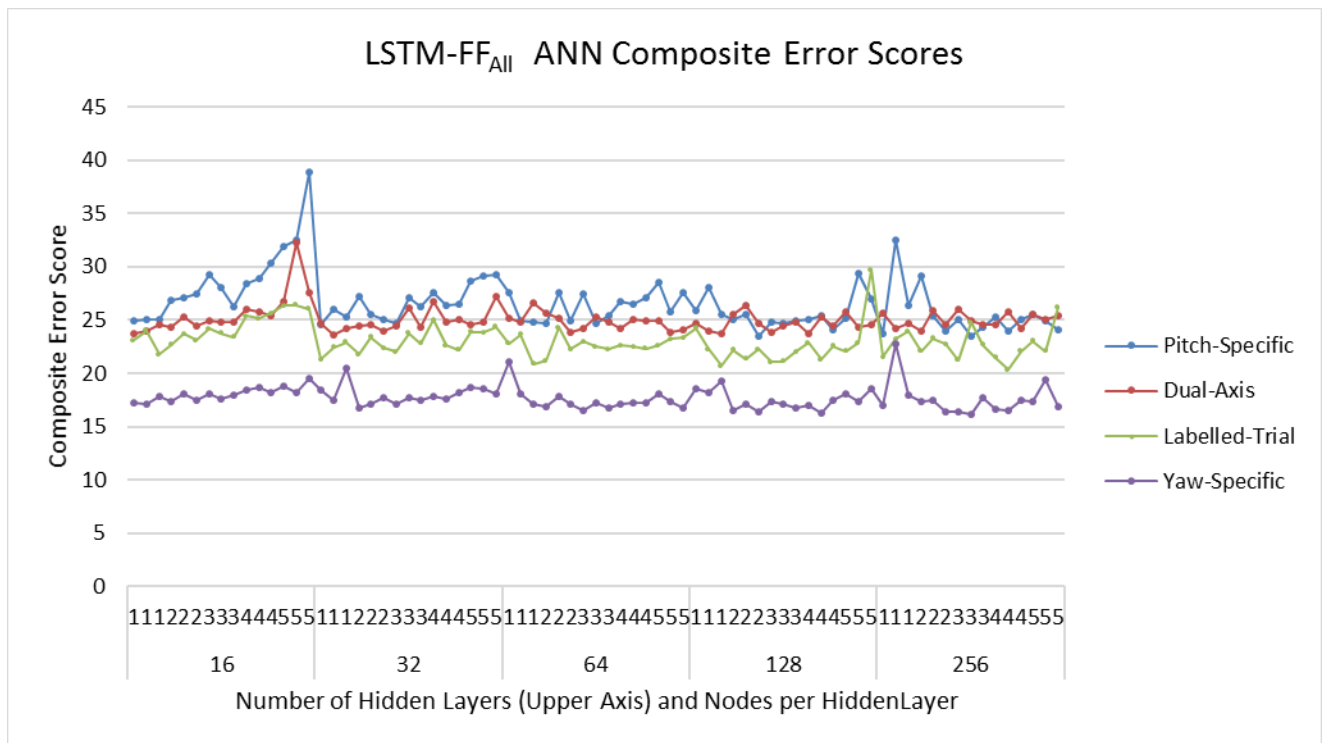


Figure 34: Performance of LSTM-FF_{ALL} ANNs with a variety of number of hidden layers and nodes per hidden layer.

3.6 Hyperparameter Tuning Stage Two: Selecting Initial Learning Rate, Dropout rate and Mini-Batch Size

3.6.1 Hyperparameter Tuning Stage Two Methods

Within each ANN group, 60 LSTM-FF_{ALL} ANNs, with the appropriate number of hidden feed-forward layers and nodes per layer (Chapter 3.5), were developed with a variety of learning rate, dropout rate and mini-batch size values. Initial learning rates were 0.1, 0.01, 0.001 or 0.0001 as recommended by Bengio (2012). Dropout rate was set to 0.1, 0.25 or 0.5 to mimic the range of values recommended by Srivastava et al., (2014). Mini-batch sizes were set to 2, 4, 8, 16 or 32 in consideration of the findings of Masters & Luschi (2018).

Each ANN was evaluated by calculating composite error scores in the same manner as in the previous chapter.

3.6.2 Hyperparameter Tuning Stage Two Results

Composite error scores (Table 6) across all ANN groups appeared to be predominantly influenced by learning rate. More detailed depictions of ANN hyperparameter tuning results from this stage are presented in Appendix B. Of note, the 0.1 learning rate caused ANNs to output extremely large or “not a number” (NaN) values. NaN is a data type used in Python to represent undefinable values. The occurrence of these NaN outputs is a known symptom of the exploding gradient problem (Zhang et al., 2019). This issue is common in recurrent neural networks and is further exacerbated in deeper networks (Pascanu, Mikolov & Bengio 2012).

Table 6: Mean composite error scores (N) of ANNs grouped by learning rate. Except in the dual-rotation group, ANNs with a learning rate of 0.1 often yielded values that were undefinable/not a number. Models that yielded NaN values were excluded from calculation of mean values.

Initial Learning Rate	Dual-rotation	Labelled Trial	Pitch-Specific	Yaw-Specific	Mean of Pitch and Yaw-Specific
0.1	25.56	236.91	151.27	278.40	214.84
0.01	22.76	43.35	37.82	21.94	29.88
0.001	22.55	23.18	26.16	18.08	22.12
0.0001	23.71	24.65	27.97	19.00	23.49

Within each ANN group, the ANN with the lowest composite score was identified (Table 7). Surprisingly, the best-performing dual-rotation ANN achieved stronger performance than the corresponding labelled trial ANN. Because a simpler model is preferable for practical reasons, and because the more complex labelled-trial ANN did not provide stronger performance than the dual-rotation ANN, the labelled-trial ANN was not

considered for further analysis. Hypothesis testing was only performed on the best-performing dual-rotation and rotation-specific ANNs.

Table 7: Overview of the hyperparameters that yielded the lowest composite error score within each ANN group.

ANN Group	ANN Type	# of Hidden Layers	# of Nodes per Layer	Initial Learning Rate	Dropout Rate	Mini-Batch Size	Composite Error Score
Dual-rotation	LSTM- FF _{ALL}	1	16	0.01	0.1	32	20.94
Labelled-Trial	LSTM- FF _{ALL}	4	256	0.001	0.5	8	21.33
Pitch-Specific	LSTM- FF _{ALL}	3	256	0.001	0.25	8	23.99
Yaw-Specific	LSTM- FF _{ALL}	3	256	0.001	0.1	8	16.15

3.7 Hypothesis Testing

While MSE is useful as a model evaluation metric, peak and cumulative compression and shear values are more clinically relevant. Therefore, hypothesis testing was performed by comparing the error in peak and cumulative compression and shear ANN outputs relative to the EMGD model to differences in EMGD model-derived loading under two relevant helmet configuration conditions.

3.7.1 Generating Variables of Interest

The dual-rotation, pitch and yaw-specific LSTM-FF_{ALL} ANNs with the lowest composite error score were applied to the test dataset. The time series compression and shear outputs of the dual axis, pitch and yaw-specific ANNs and EMGD model on the test set were processed further to generate peak and cumulative compression and shear values over individual head movement cycles. First, each 30-second trial in the test set was split into head movement cycles that were identified given head-trunk angular velocity waveforms (Figure 35). A head movement cycle was defined as looking towards a new target before returning to the original target (e.g. during yaw trials: gaze began as fixed on the right target, then the head moved to acquire a target on the left, and then returned to acquire the initial target on the right). Within each head movement cycle, peak and cumulative compression and shear values were calculated from both the ANN and EMG model compression and shear time series outputs. Cumulative values were calculated in MATLAB R2019b (Mathworks Inc., USA) through trapezoidal integration of the corresponding time series compression and shear outputs over

the course of individual head movement cycles. Calculation of cumulative shear values was performed based on the absolute value of the time-series shear outputs. This was required as shear outputs had both positive negative values representing anterior and posterior shear respectively. The durations over which the ANN and EMG model outputs were integrated were identical as the kinematics used to define the movement cycles were identical. Absolute error (AE) values between ANN and EMG model outputs were calculated for the peak and cumulative compression and shear values within each head movement cycle. Next, absolute difference (AD) values were calculated between EMG-model derived peak and cumulative compression and shear values between hNVG and hNVGCW conditions within the test set. Differences between these particular helmet conditions were chosen for this analysis as they had previously been found to not exhibit statistically significant differences in cervical spine loading outcomes (Barrett, 2019).

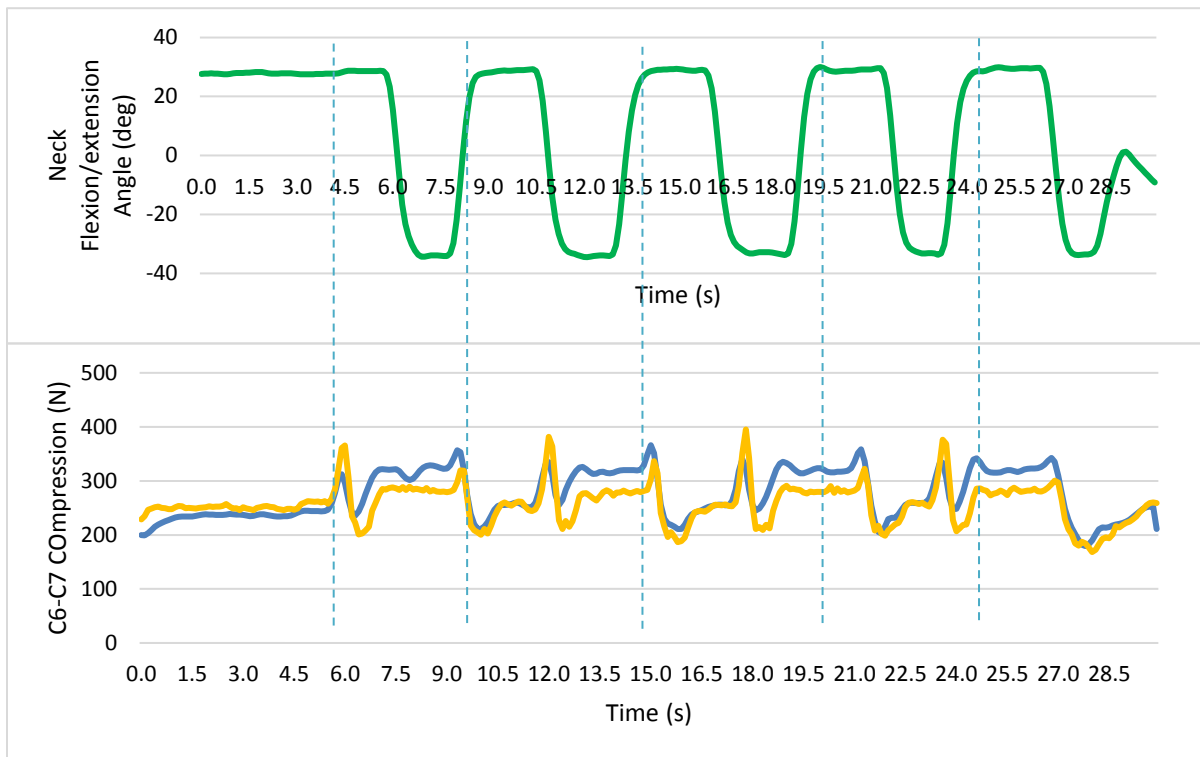


Figure 35: Head-trunk flexion angle (upper figure) and C6-C7 time-series compression (lower figure) during a pitch trial. The blue curve represents ANN-predicted C6-C7 compression while the yellow curve represents EMGD model C6-C7 compression outputs. Vertical dotted lines approximate where head movement cycles would be defined as beginning and ending.

Hypothesis Testing

Hypothesis testing was performed in SPSS Version 26 (IBM, Armonk, NY) to compare the AE values between ANN and EMGD model outputs to corresponding AD values within the test set. For each of the dual-rotation, pitch-specific and yaw-specific ANNs, four one-tailed Welch's *t*-tests were performed to determine if the AE values between ANN and EMG model outputs were significantly ($p < 0.05$) different from the relevant AD values between hNVG

and hNVGCW conditions. Specifically, Welch's t -tests were performed to compare the distribution of AE values to the corresponding hNVG-hNVGCW AD values for each of the eight variables of interest (peak and cumulative compression and shear in pitch and yaw trials). The Welch's t -test was used as it is more robust to Type one error relative to Student's t -tests when comparing samples of unequal sample size and/or variance (Delacre, Lakens & Leys, 2017)

3.7.2 Secondary Analysis: Mean Time-Series Error

This research focused on the ability of ANNs to accurately predict peak and cumulative compression and shear metrics. These measures are meaningful predictors of pain incidence within the lumbar spine (Norman et al., 1998) and these findings may extend to the cervical spine. Additionally, these measures have previously been associated with helicopter flight-related neck injury (Forde et al., 2011). As a result, these measures would be of primary interest to a helmet designer. However, for some practitioners, the ability to observe trends and events within time series compression and shear data as an individual moves their head may also be important. To evaluate the LSTM-FF_{ALL} ANNs in this context, I calculated the mean error of predicted compression and shear relative to EMG model-derived values within the test set. Mean error was calculated across all frames of data within the test set as the mean difference between ANN and EMGD model outputs.

3.7.3 Secondary Analysis: Temporal Error in ANN-Predicted Peak Compression and Shear

Given the targeted end use of the ANNs for *in-silico* design, it is important to assess the agreement between the ANNs and the EMGD model in terms of the timing of peak compression/shear occurrence. The ability of the ANNs to accurately predict when peak loading occurs may be useful within the helmet design process by helping designers to understand loading-posture relationships. To assess the ANNs in this context, the indices of the frames in which ANN-predicted and EMGD-derived peak compression and shear occurred within each head movement cycle were extracted. The absolute temporal differences between these indices were then expressed as a percentage of the duration of the head movement cycle.

Chapter 4: Results

4.1.1 Statistical Testing

The ANN for pitch rotations predicted cumulative compression (mean absolute error (MAE) \pm SD = 116.87 ± 62.87 Ns), peak compression (MAE = 28.81 ± 21.20 N), cumulative shear (MAE = 7.26 ± 6.21 Ns) and peak shear (MAE = 2.73 ± 2.51 N) outputs whose absolute error values relative to the criterion EMG-model based outputs were all significantly lower ($p < 0.05$) than corresponding AD values (Table 8) between hNVG and hNVGCW conditions.

Table 8: Mean absolute differences in compression and shear measures during hNVG vs. hNVGCW conditions.

	Compression		Shear	
	Peak (N) (SD)	Cumulative (Ns) (SD)	Peak (N) (SD)	Cumulative (Ns) (SD)
Pitch	39.94 (30.34)	226.04 (186.82)	4.92 (5.14)	19.38 (12.05)
Yaw	33.71 (51.52)	203.81 (154.62)	7.01 (8.61)	23.99 (18.46)

The ANN for yaw rotations generated cumulative compression (MAE = 74.10 ± 44.51 Ns) and cumulative shear (MAE = 17.68 ± 13.32 Ns) values whose absolute error values were significantly lower ($p < 0.05$) than AD values (Table 8) between hNVG and

hNVGCW conditions. Absolute error values were not significantly lower than corresponding AD values in peak shear (MAE = 6.46 ± 6.45 N) ($p = 0.16$) and peak compression (MAE = 19.89 ± 21.11 N) ($p = 0.09$). Overall, the use of rotation-specific ANNs yielded error levels that were below corresponding between-helmet condition differences in six of eight variables

When applying the dual-rotation ANNs to pitch trials, error levels in cumulative compression (MAE = 656.68 ± 214.87), peak compression (MAE = 131.49 ± 46.16), cumulative shear (MAE = 62.13 ± 27.35) and peak shear (MAE = 15.74 ± 10.30) were all significantly greater ($p < 0.05$) than corresponding between-helmet condition differences. Similarly, applying the dual-rotation ANNs to yaw trials yielded outputs whose error levels in cumulative compression (MAE = 438.92 ± 311.03), peak compression (MAE = 148.72 ± 88.55), cumulative shear (MAE = 77.98 ± 24.18) and peak shear (MAE = 14.64 ± 8.62) were significantly greater than corresponding between-helmet condition differences. The dual-rotation ANN exhibited error levels that were greater than between-helmet condition differences in all variables during both pitch and yaw trials.

4.1.2 Mean Time-Series Error

Mean error data (Table 9) suggest that both pitch and yaw specific LSTM-FF_{ALL} ANNs overpredicted compression magnitudes by an average of approximately 22 and 15 N during pitch and yaw trials respectively. ANN-predicted shear values underestimated EMGD values by an average of less than 2 N. This can be observed in the exemplar data presented in Figures 36 and 37.

Table 9: Mean error of rotation-specific LSTM-FF_{ALL} ANN-predicted compression and shear relative to EMG model-derived compression and shear.

	Mean Compression Error (N)	Mean Shear Error (N)
Pitch	21.94	0.22
Yaw	14.81	1.22

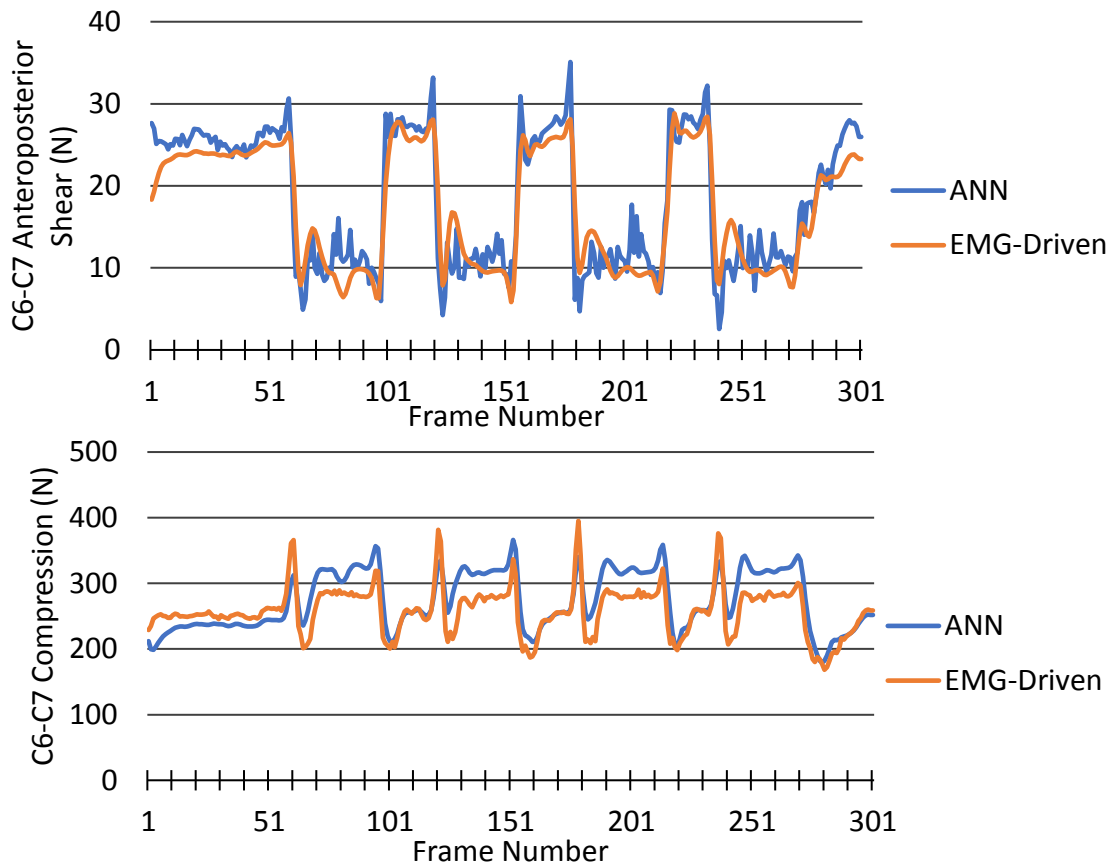


Figure 36: Exemplar compression (bottom) and shear (top) LSTM-FF_{ALL} ANN outputs for a single pitch trial selected from the test set. These specific waveforms were selected as they are representative of mean error values calculated across all pitch trials within the test set.

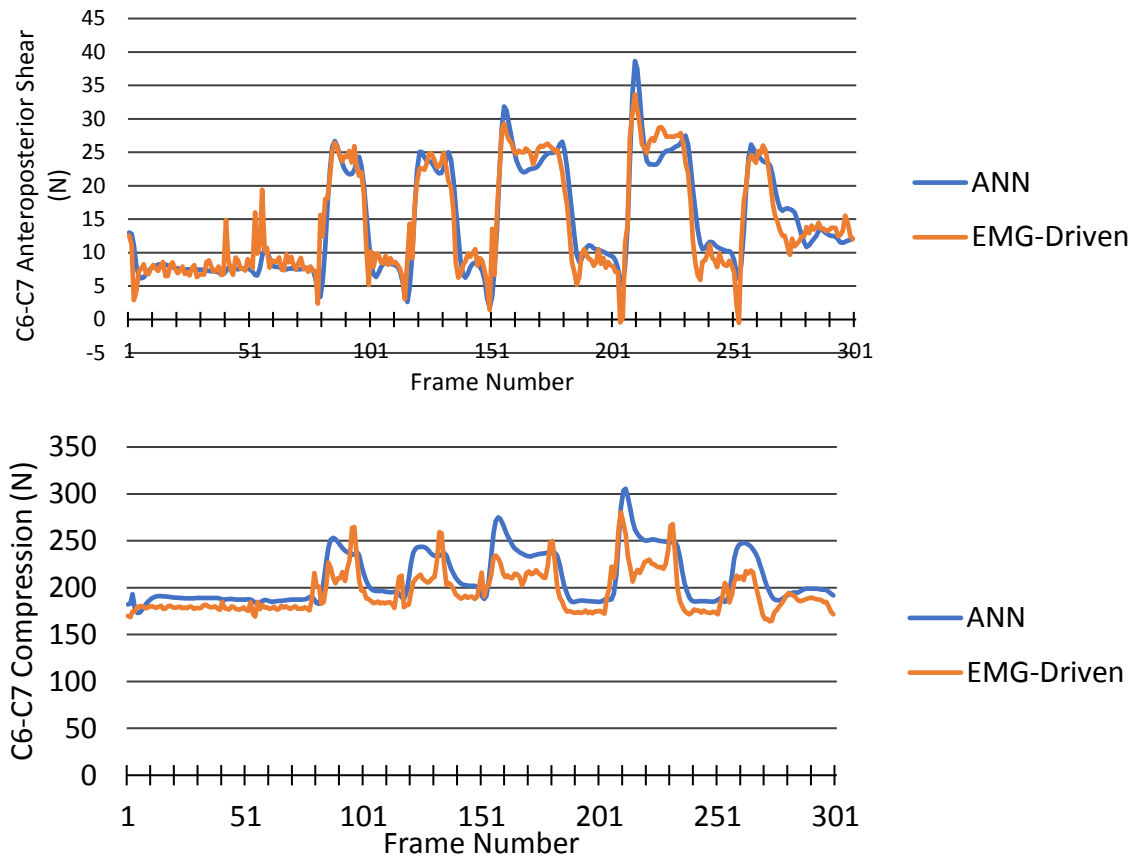


Figure 37: Exemplar compression (bottom) and shear (top) LSTM-FF_{ALL} ANN outputs for a single yaw trial selected from the test set. These specific waveforms were selected as they are representative of mean error values calculated across all yaw trials within the test set.

4.1.3 Temporal Error in Timing of ANN-Predicted Peak Compression and Shear

The pitch- and yaw-specific LSTM-FF_{ALL} ANNs appear to exhibit similar magnitudes of mean absolute error in the timing of peak loading occurrence (Table 10). The standard deviations of these error values are also relatively high. Visual examination of C6-C7 loading waveforms (Figures 36 and 37) suggests that there are typically two peaks in loading

magnitude within each head movement cycle. This is likely related to the fact that each head movement cycle contained two acceleration-braking cycles. Consideration of the frequency of different ranges of error (Figures 38-41) suggests that a substantial portion of temporal error in peak loading timing may be due to ANNs predicting peak loading to occur during the incorrect acceleration/braking phase within a head movement cycle. This would likely lead to a temporal offset in peak loading occurrence of approximately 50% of cycle duration. The error in timing of peak shear and compression occurrences was between zero and ten or 40 and 60 percent in 72% and 89% of pitch trials respectively (Figures 38 and 39). Yaw trials exhibited a similar trend but with weaker performance in the prediction of peak compression timing. Specifically, 53% and 82% of peak compression and shear occurrences were predicted with zero to ten or 40 to 60 percent temporal error (Figures 40 and 41).

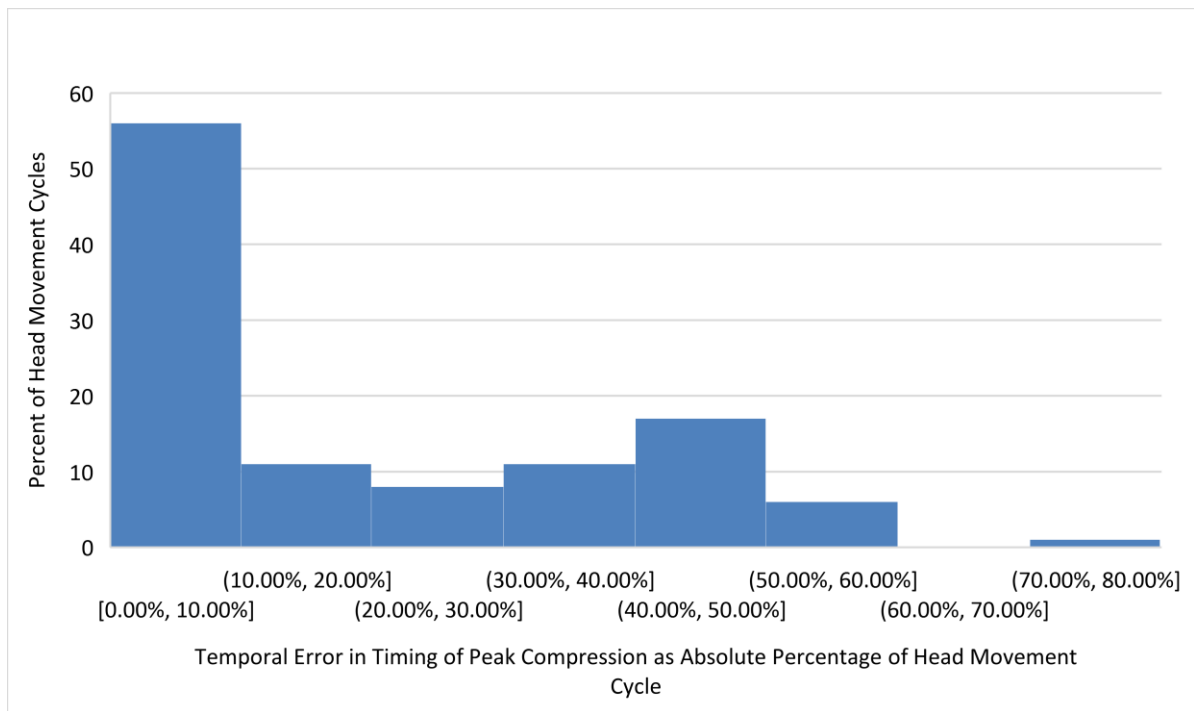


Figure 38: Histogram indicating proportion of head movement cycles with various magnitudes of absolute error in timing of predicted peak compression loading during pitch trials within the test set.

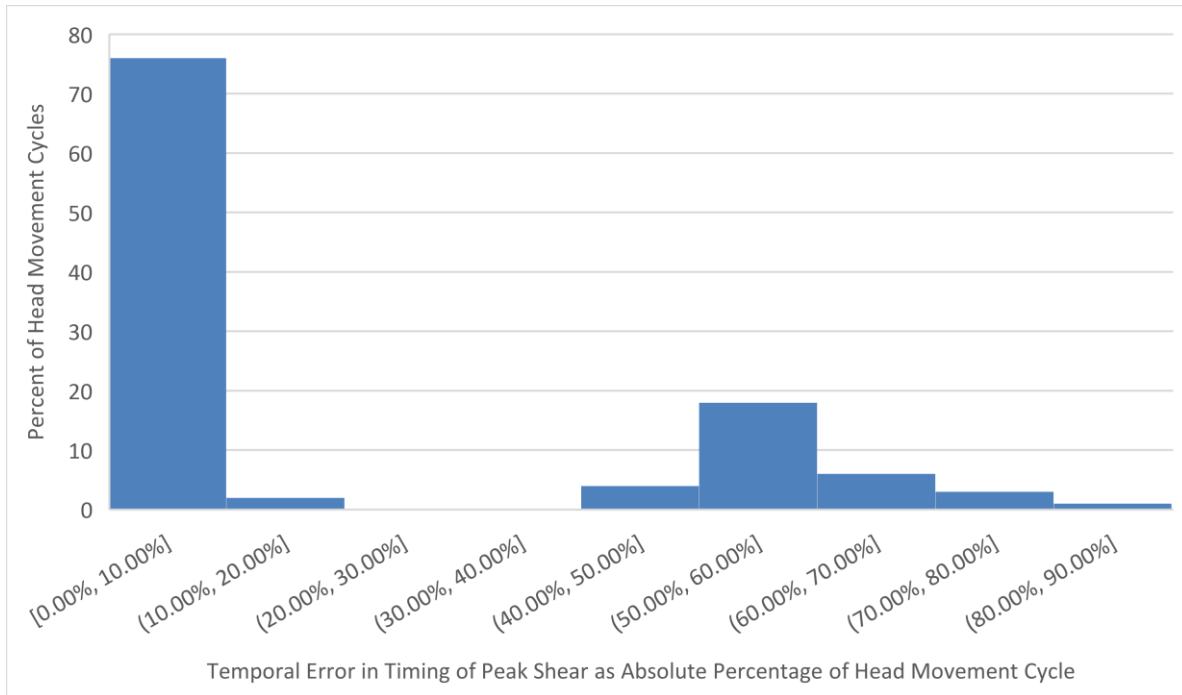


Figure 39: Histogram indicating proportion of head movement cycles with various magnitudes of absolute error in timing of predicted peak shear loading during pitch trials within the test set.

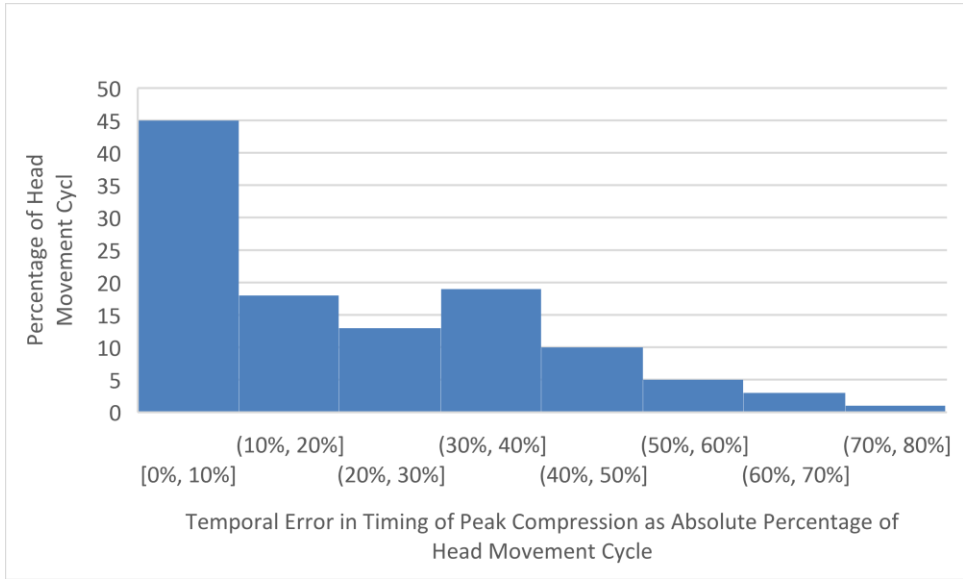


Figure 40: Histogram indicating proportion of head movement cycles with various levels of error in timing of predicted peak compression loading during yaw trials within the test set.

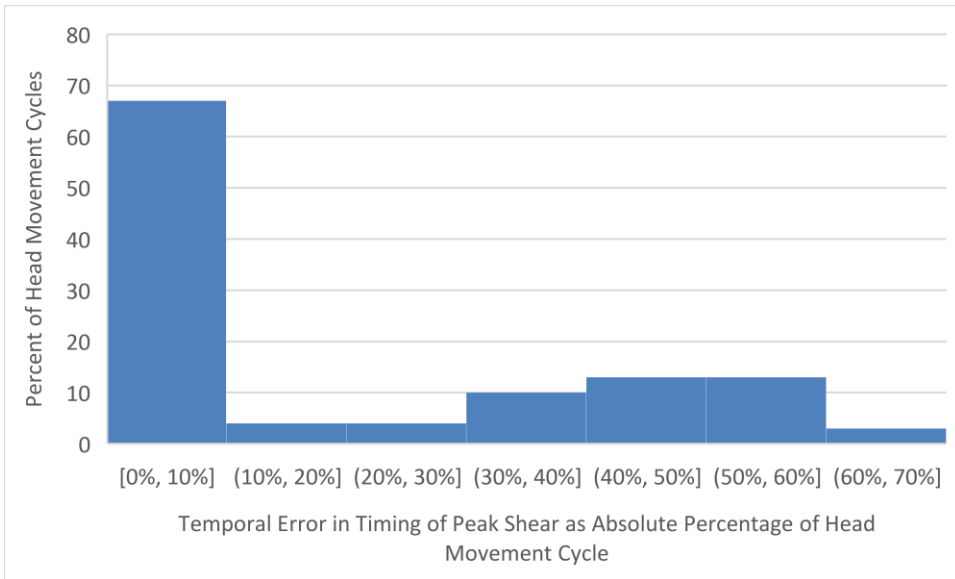


Figure 41: Histogram indicating proportion of head movement cycles with various levels of error in timing of predicted peak shear loading during yaw trials within the test set.

Table 10: Mean absolute difference in timing of peak loading within pitch- and yaw-specific LSTM-FF_{ALL} ANNs relative to EMGD model outputs as a percentage of head movement cycle duration.

	Mean Absolute Difference in Timing of Peak Loading (SD)	
	Pitch	Yaw
Compression	17.9% (20.2%)	22.0% (18.5%)
Shear	19.1% (26.5%)	19.3% (21.8%)

Chapter 5: Discussion

5.1 ANN Performance

This study successfully developed rotation-specific ANNs to predict C6-C7 joint contact forces to permit rapid, iterative virtual helmet design evaluation. The study hypothesis was that an ANN could predict C6-C7 compression and AP shear where the absolute error, relative to a criterion standard EMGD model, is less than AD values between EMGD model-computed C6-C7 compression and AP shear during flight-relevant head movement under hNVG vs. hNVGCW helmet conditions. Data support this hypothesis as absolute error values were significantly lower than between-helmet condition differences in all variables of interest with the exception of peak shear and compression during yaw trials. While model generalizability can be a concern when applying ANNs, the ANNs developed in this thesis were tested on data from participants who were not included in the training set to measure the ability to generalize.

Rotation-specific ANNs were successful in achieving error less than associated between-helmet condition differences but dual-rotation ANNs were not. In fact, dual-rotation ANNs failed to achieve acceptable performance across all four variables of interest in both pitch and yaw trials. This was likely due to biomechanical differences between pitch and yaw movements such as differing muscle activation profiles. Muscle force contributions strongly influence intervertebral contact forces (Granata & Marras, 1993), and are likely to be consequentially different between pitch and yaw head movements. It appears that the dual-rotation ANNs in this thesis were unable to account for these differences as they did not

receive EMG input. Overall, rotation-specific ANNs can be used as a tool in a helmet CAD process to differentiate between relevant helmet designs in consideration of biomechanical exposures. While the etiology of neck pain is multifactorial, reduction of joint contact forces in helicopter pilots through improved helmet design would likely be beneficial in reducing the incidence of neck pain and injury.

5.2 Comparison to Previous ANN Models

Because no previous studies have applied ANNs to predict cervical spine loading, parallels are drawn between the ANNs developed in this study to previous applications of ANNs to predict lumbar spine loading. Compression and shear magnitudes are generally an order of magnitude greater in the lumbar spine (Callaghan & McGill, 2001) than the cervical spine (Barrett, McKinnon & Callaghan, 2020) during unloaded conditions. To account for differences in the scale of target value magnitudes, the ANNs in this thesis are compared to the models of Parkinson & Callaghan (2009) and Arjmand et al. (2013) as they reported coefficients of correlation and determination respectively.

The ANN of Arjmand et al. (2013) yielded an R^2 value of 0.99 between ANN-predicted compression and corresponding kinematics-driven FE model outputs. Interestingly, their study also developed quadratic regression equations which also achieved a R^2 of 0.99, albeit with greater RMSE. Their ANN and regression models outperformed the rotation-specific ANNs in this thesis which generated compression predictions with R^2 values of 0.50 and 0.72 in pitch and yaw trial conditions respectively (Appendix C). The near-perfect R^2

values of the models of Arjmand et al. may be related to the method by which target compression and shear values were generated. As a reminder, target values were generated by a kinematics-driven FE model (Arjmand & Shirazi-Adl, 2006) of the thoracolumbar spine. The inputs to the ANN model were the mass of the handled load, the anterior and right lateral distance to the L5-S1 joint centre, whether the load was lifted with one or two hands as well as trunk flexion angle. Data used to develop the ANN and regression models were generated by simulating static lifting postures where the anterior and lateral distances of the load were iterated from 0 cm to 65 cm and 0 cm to 85 cm respectively. A driver of the very high observed R^2 value of the ANN in the study of Arjmand et al. may be the deterministic nature of the FE model where sets of input features have consistent relationships with each other. Specifically, changes in muscle activation profiles, which would greatly influence lumbar contact forces, were not accounted for in the FE model. In contrast, the ANNs in this thesis were trained and tested on data derived from an EMGD model whose inputs did not have constant relationships (i.e. a given set of kinematic inputs was not always accompanied by the same EMG inputs). This, combined with the fact that ANNs in this thesis did not receive the same inputs as the EMGD model that they tried to mimic is a likely reason for their weaker performance relative to the ANNs of Arjmand et al. (2013).

The ANN of Parkinson & Callaghan (2009) achieved an average r value of 0.6 between ANN-predicted time series L4-L5 compression and corresponding EMG-assisted model values when applied to novel data. Correlation coefficients for shear outputs were not reported. The pitch-specific ANN in this thesis achieved an r value of 0.71 between time-

series C6-C7 compression outputs and corresponding EMGD outputs. The yaw-specific ANN had an r value of 0.85 with respect to compression outputs. Scatterplots depicting ANN predictions vs. corresponding EMGD model outputs are presented in Appendix C.

Parkinson & Callaghan (2009) speculated about potential avenues by which ANN performance could be improved. These considerations were factored in to the design of the ANN development process in this thesis and may partially explain the observed improvements in r values. As mentioned in Chapter 2.13.3, the ANNs of Parkinson & Callaghan used a RMSD loss function. They speculated that an alternate loss function had the potential to improve performance outcomes. In this thesis, a MSE loss function was applied. MSE loss functions are the default choice when applying ANNs to regression problems (Moshagen, Adde & Rajgopal, 2021; Reed, 1999). MSE loss functions penalize the model for larger errors exponentially more than for smaller errors, e.g. an error of 10 units is considered to be more than twice as bad as an error of 5 units. In other words, MSE loss is highly sensitive to outliers, thus “encouraging” the ANN to better map peaks and valleys within a wave-form (Moshagen, Adde & Rajgopal, 2021).

The ANN of Parkinson & Callaghan was concluded to be promising for use in large scale studies of cumulative loading across a group of subjects. However, they noted that their ANN was not a reasonable approach to obtain estimates of exposure on an individual level. In this thesis, the standard deviations of the absolute errors in peak and cumulative compression and shear are relatively high compared to their respective means. A high degree of variability in absolute error across head movement cycles suggests that, similar to the

ANN of Parkinson & Callaghan (2009), predictive performance may not be consistent between individual head movement cycles. Therefore, practitioners who use the ANNs from this thesis should aim to draw conclusions from mean results generated over several visual scanning trials. These ANNs should not be used as a tool for injury risk assessment at an individual level. Overall, these results suggest that the ANNs developed in this study are useful in understanding how different helmet conditions may influence biomechanical exposures of the cervical spine across a group of head movements.

Parkinson & Callaghan (2009) speculated that implementation of recurrence in their ANN could improve their results by better accounting for inter-dependencies between frames of data. Therefore, this thesis compared the performance of FF and LSTM-based ANNs. Results demonstrate a consistent pattern across the dual-rotation, labelled-trial and rotation-specific ANN groups. LSTM-FF ANNs generally performed best followed by FF ANNs then LSTM ANNs. The improved performance of LSTM-FF vs. FF ANNs suggests that there were important interdependencies between timepoints and that the LSTM-FF ANNS were able to leverage this information for improved performance.

5.3 ANN Input Features

Conceptually, the ANNs in this thesis were tasked with learning the relationship between inverse kinematics data and the compression and shear outputs of an EMGD model. However, to be useful in applied practice, the ANNs were required to do so without EMG inputs. With that in mind, ANN inputs were selected based on whether they were likely to

influence muscle activation profiles. Head-trunk kinematics were investigated as they are likely to influence muscle activation profiles through force-velocity (Hill, 1938) and force-length (Gordon, Huxley & Julian, 1966) relationships. C6-C7 joint moment inputs were also included as they are likely related to the magnitude of muscle forces required to generate observed head movements. However, the addition of other input features may have improved ANN performance. For example, C6-C7 reaction forces, which are also available through inverse dynamics and MBD modelling, contribute to compression and shear contact forces. Similarly, a helmet mass input feature may have improved ANN predictions by providing explicit information by which the ANN could differentiate between helmet configurations. While the inclusion of reaction force and helmet mass input features may improve ANN performance, this would also increase the risk of overfitting (Ying, 2018). Additionally, the use of a reaction force input feature would increase data requirements when applying these ANNs in practice. Cognizant of these issues and that the rotation-specific ANNs developed in this thesis achieved error levels below corresponding between-helmet condition differences in six of eight variables, additional ANN input features were not formally evaluated in this thesis. Nonetheless, the evaluation of additional input features may be useful in cases where acceptable error thresholds are lower.

The ability of recurrent ANNs to model interdependencies between timepoints, was indirectly examined by assessing whether the inclusion of time-derivative measures (angular velocity and acceleration) influenced predictive performance. Across all recurrent models in this thesis (LSTM and LSTM-FF), predictive performance was improved when kinematic

input features included angular velocity and acceleration in addition to angular displacement. The angular velocity input feature may have improved predictive performance by allowing the ANN to account for the muscle force-velocity relationship (Hill, 1938). Similar to the C6-C7 moment inputs, the angular acceleration input feature may have provided information related to muscle activation patterns required to produce the observed head movements. Given that LSTM-based ANNs can remember information over multiple timesteps, it is possible that they could have derived velocity and acceleration data from angular displacement inputs. However, results suggest that this was not the case in the ANNs developed in this thesis.

5.4 Future Directions

In regression tasks, traditional supervised ANN models have typically had single outputs (Xu et al., 2019). However, the ANNs developed in this thesis are multi-output models. Multi-output ANNs have the advantage of simplifying the application of these ANNs for the end-user and to provide improved computational efficiency (Borchani et al., 2015). Additionally, the machine learning literature includes several recent examples of multi-output ANNs providing similar or better performance than their single-output counterparts in a variety of applied contexts. These include prediction of stock prices (Sharma et al., 2017), waste gasification (Pandey et al., 2016) and meteorological variables (Raza & Jothiprakash, 2014). Nonetheless, it is possible that a single-output ANN would have provided stronger performance than multi-output ANNs (Claveria, Monte & Torra, 2015). For example, an ANN whose sole output was C6-C7 compression may achieve faster and more effective

convergence on the set of weights and biases that provide optimal prediction of compression values. This is because the loss function would only consider compression rather than compromising between consideration of both compression and shear outputs (Xu et al., 2019). A future direction that could yield improved performance, would be to develop single-output ANNs to predict cervical spine compression and shear separately.

This thesis focused on the use case of ANNs to facilitate *in-silico* assessment of military helicopter helmet designs. However, these ANNs may be useful in biomechanical assessment of other head-supported masses in which individuals perform similar head movements. For example, virtual reality headsets, which can increase flexion/extension moments about the cervical spine (Penumudi et al., 2020), may require biomechanical assessment for ergonomic design of headsets. While the application of these ANNs to other use cases may be possible, this should be undertaken with caution as well as with the understanding that ANN error would likely be greater.

5.5 Limitations

In this study, ANNs were developed based on data collected from a university student population. It is possible that trained military helicopter pilots exhibit distinct muscle recruitment strategies that could influence compression and shear outcomes. Additionally, real-world helicopter flight exposes pilots to whole-body vibration which may also influence neck muscle recruitment strategies (Law et al., 2017). Helicopter flight may also induce accelerations of the head due to rapid helicopter maneuvers. These factors may influence

cervical spine contact forces in a manner that the ANNs would not be able to account for. However, these concessions in external validity were required to enable the collection of reliable motion capture and EMG data from a reasonable number of participants. A future direction may be to replicate the data collection while participants are seated on a vibrating platform to mimic helicopter flight.

As with any ANN, performance may be negatively affected when exposed to input data that are outside the range of input values in this study. While the training set consisted of 256 30-second trials collected from 16 individuals, these data only represent yaw and pitch motions performed under head supported mass conditions consistent with helicopters. The model may not generalize as well to conditions that require alternative rotation axes or head supported mass parameters beyond those tested here. Another consideration regarding the data used to develop these ANNs is that target values were derived from EMGD model estimates which may have differed from true compression and shear values. However, I am confident in the validity of the EMGD model as it exhibits good agreement with previous modelling of spinal load during isometric contractions (Choi & Vanderby, 2000) and has undergone an extensive sensitivity analysis (Barrett et al, 2021). Unfortunately, it is not feasible to directly verify the outputs of the EMGD model with *in vivo* measurements as this would likely require surgically implanted force transducers within the spine.

The data that were used to train and evaluate ANNs were derived from the EMGD model whose anatomical dimensions were that of a 50th percentile adult male. OpenSim does have the capability of scaling the model to individual participant characteristics; however, it

is unclear if the properties of the passive-tissues, most notably the lumped passive contributions from intervertebral disc and facet joints, scale to a similar degree with size. Considerable efforts were made to validate the model's passive elements for a 50th percentile male and altering structural dimensions would have added extra uncertainty into the EMG model-derived compression and shear forces. While anatomical differences between participants were not considered, subject-specific active motor patterns were incorporated using individual EMG measurements.

Assessment of the agreement between ANN and EMGD model outputs for the timing of peak loading occurrence indicate that ANNs exhibited errors in timing of approximately 20% of head movement cycles. As discussed in Chapter 4.1.3, this may be related to the tendency of head movement cycles to include two peaks and ANNs having difficulty in correctly identifying the peak that exhibits the maximum loading magnitude. These findings suggest that the ANNs developed in this thesis should not be used to investigate the timing of peak loading within the helmet CAD process.

5.6 Appropriate Application of ANNs Within a Military Helicopter Helmet Design Process

The application of the ANNs developed in this thesis to a helicopter design process should be performed with consideration of the following constraints. As discussed in Chapter 5.2, the ANNs in this thesis should be used for assessment of cervical spine loading at a population level rather than the individual level. These assessments may include consideration of peak and cumulative loading during pitch head movements. However, assessment of loading

during yaw movements should be constrained to cumulative loading as error levels in predicted peak loading indicated insufficient sensitivity to differentiate between helmet designs. Additionally, the pitch- and yaw-specific ANNs should only be applied to visual scanning trials where participants perform head rotations solely about the relevant axis. Kinematic data collected during off-axis movements may lie outside the range of data used to develop the ANNs in this thesis and would likely lead to greater ANN error. Helmet design teams who wish to understand posture-loading relationships should do so with the understanding that the ANNs may predict the timing of peak loading occurrence to occur during the wrong acceleration-braking phase within a given head movement cycle. Finally, as discussed in Appendix D, design teams should apply the ANNs to sequences of input data that include several head movement cycles as ANN performance appears to improve in the latter movement cycles within a trial. Specifically, ANN performance is improved during the second vs. first movement cycle during pitch trials. During yaw trials, ANN performance was improved during the third and fourth vs. first and second movement cycles.

5.7 Novel Contributions

The ANNs that were developed in this thesis only require triaxial C6-C7 moments and head-trunk kinematics as inputs to generate compression and shear outputs. Such data are readily available from digital human models or MBD models of the neck and head system. The addition of these ANNs allows helmet designers to quickly evaluate the cervical spine contact force consequences of novel designs, avoiding the resource intensive process of laboratory-based human participant research. Data from Lu et al. (2013) also support the

feasibility of an MBD + ANN method, where they developed an ANN to estimate cartilage deformation within the knee by developing an ANN to replace a more computationally expensive finite element modeling approach. The findings of this thesis continue to support that robustly designed ANNs can provide a reasonable, time efficient alternative to more resource intensive approaches in order to support a rapid, iterative, helmet CAD process.

In addition to developing a tool to improve helmet CAD processes, this thesis identified important considerations regarding the use of ANNs as a surrogate for musculoskeletal models. I attribute the ability of the ANNs in this thesis to generate accurate joint contact force predictions without important information regarding muscle activation to the use of LSTM networks. LSTM networks are distinct amongst recurrent neural networks in that they can learn interdependencies between multiple timesteps in the past and a current timestep. This property makes LSTM networks especially powerful and versatile (Van Houdt, Mosquera & Napoles, 2020). As machine learning increases in popularity within the field of ergonomics and human factors (Lau et al., 2018), LSTM networks may be an especially efficacious and versatile tool in human simulation contexts.

This thesis explored how ANN performance was affected by different sets of kinematic input features (displacement only vs. displacement, velocity and acceleration). A novel finding is that although LSTM networks can recall information over multiple timesteps, performance was improved when angular velocity and acceleration inputs were included in addition to angular displacement inputs. Therefore, future applications of ANNs

for musculoskeletal simulation/modelling in the absence of EMG inputs should consider including these higher order kinematic inputs.

The importance of considering direction of motion is another key finding of this work. Although the dual-rotation ANNs in this thesis achieved relatively low MSE across time series data, they were not successful in generating clinically relevant peak and cumulative values with error less than between-helmet condition differences. In contrast, rotation-specific ANNs generated clinically relevant values with error less than between-helmet condition differences. Future applications of ANNs to modeling of joint contact forces should consider movement direction by developing ANNs for specific movement types.

Chapter 6: Conclusion

This thesis developed ANNs as a tool to inform helicopter helmet design through *in-silico* design. Overall, rotation-specific LSTM-based ANNs are useful as a helmet design tool as they exhibit adequate sensitivity to differentiate between relevant helmet designs in terms of associated cervical spine compression and shear. While dual-rotation ANNs were unsuccessful in generating clinically relevant outputs with adequate sensitivity to differentiate between helmets, pitch and yaw-specific ANNs were. Pitch-specific LSTM-FF ANNs were successful in predicting four of four variables of interest with adequate sensitivity to differentiate between helmets while yaw-specific ANNs were successful in two of four variables.

This thesis included an extensive hyperparameter tuning process in which a total of 1,740 ANNs were developed and assessed. This process involved the development of FF, LSTM and LSTM-FF ANNs with a variety of network sizes, input features, learning rates, mini-batch sizes and dropout ratios. From these ANNs, the best-performing dual-rotation, pitch-specific and yaw-specific ANNs were identified. Across these groups, the LSTM-FF ANNs performed best which highlights the benefit of LSTM-based ANNs and their ability to account for interdependencies between multiple frames within a sequence. Additionally, the ANNs performed best when angular velocity and acceleration were included in addition to angular displacement input features. Overall, the rotation-specific ANNs represent a useful

tool in assisting the design of next generation helicopter helmets that reduce cervical spine joint contact forces with the goal of preventing neck pain in military helicopter aircrew.

References

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). Tensorflow: A system for large-scale machine learning. In *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)* (pp. 265-283).
- Adam, J. (2004). *Results of NVG-induced neck strain questionnaire study in CH-146 Griffon aircrew*. Defence Research and Development Toronto (Canada).
- Agatonovic-Kustrin, S., & Beresford, R. (2000). Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. *Journal of pharmaceutical and biomedical analysis*, 22(5), 717-727.
- Aghazadeh, F., Arjmand, N., & Nasrabadi, A. M. (2020). Coupled artificial neural networks to estimate 3D whole-body posture, lumbosacral moments, and spinal loads during load-handling activities. *Journal of biomechanics*, 102, 109332.
- Almosnino, S., Pelland, L., Pedlow, S. V., & Stevenson, J. M. (2009). Between-day reliability of electromechanical delay of selected neck muscles during performance of maximal isometric efforts. *BMC Sports Science, Medicine and Rehabilitation*, 1(1), 1-8.
- Äng, B., & Harms-Ringdahl, K. (2006). Neck pain and related disability in helicopter pilots: a survey of prevalence and risk factors. *Aviation, space, and environmental medicine*, 77(7), 713-719.
- Arjmand, N., & Shirazi-Adl, A. (2006). Model and in vivo studies on human trunk load partitioning and stability in isometric forward flexions. *Journal of biomechanics*, 39(3), 510-521.
- Arjmand, N., Ekrami, O., Shirazi-Adl, A., Plamondon, A., & Parnianpour, M. (2013). Relative performances of artificial neural network and regression mapping tools in evaluation of spinal loads and muscle forces during static lifting. *Journal of biomechanics*, 46(8), 1454-1462.

- Barrett, J. (2016). *An EMG-Driven cervical spine model for the investigation of joint kinetics: With application to a helicopter pilot population* (Master's thesis, University of Waterloo).
- Barrett, J. M., McKinnon, C., & Callaghan, J. P. (2020). Cervical spine joint loading with neck flexion. *Ergonomics*, 63(1), 101-108.
- Barrett, J. M., McKinnon, C. D., Dickerson, C.R. & Callaghan, J. P. (2021). An EMG-Driven Cervical Spine Model in OpenSim. Manuscript submitted to *Journal of Applied Biomechanics*. [Submission ID: JAB.2020-0384]
- Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade* (pp. 437-478). Springer, Berlin, Heidelberg.
- Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2), 157-166.
- Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford university press.
- Borchani, H., Varando, G., Bielza, C., & Larranaga, P. (2015). A survey on multi-output regression. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(5), 216-233.
- Callaghan, J. P., & McGill, S. M. (2001). Low back joint loading and kinematics during standing and unsupported sitting. *Ergonomics*, 44(3), 280-294.
- Camacho, D. L., Nightingale, R. W., Robinette, J. J., Vanguri, S. K., Coates, D. J., & Myers, B. S. (1997). Experimental flexibility measurements for the development of a computational head-neck model validated for near-vertex head impact. *SAE transactions*, 3989-4002.
- Chafé, G. S., & Farrell, P. S. (2016). Royal Canadian Air Force CH-146 Griffon aircrew 2014 spinal musculoskeletal trouble survey. *Defence R&D Canada. Toronto: DRDC Toronto scientific report*, (179).
- Chiu, C. C., Sainath, T. N., Wu, Y., Prabhavalkar, R., Nguyen, P., Chen, Z., ... & Bacchiani, M. (2018, April). State-of-the-art speech recognition with sequence-to-sequence models. In *2018*

- IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 4774-4778). IEEE.
- Choi, H. & Vanderby Jr, R. (2000). Muscle forces and spinal loads at C4/5 level during isometric voluntary efforts. *Medicine and science in sports and exercise*, 32(4), 830-838.
- Cholewicki, J., & McGill, S. M. (1994). EMG assisted optimization: a hybrid approach for estimating muscle forces in an indeterminate biomechanical model. *Journal of biomechanics*, 27(10), 1287-1289.
- Chollet, F., & others. (2015). Keras. GitHub. Retrieved from <https://github.com/fchollet/keras>
- Christophy, M., Curtin, M., Senan, N. A. F., Lotz, J. C., & O'Reilly, O. M. (2013). On the modeling of the intervertebral joint in multibody models for the spine. *Multibody System Dynamics*, 30(4), 413-432.
- Claveria, O., Monte, E., & Torra, S. (2015, January). Multiple-input multiple-output vs. single-input single-output neural network forecasting. In *Research Institute of Applied Economics* (pp. 2015-02). Barcelona University.
- Côté, P., Cassidy, J. D., Carroll, L. J., & Kristman, V. (2004). The annual incidence and course of neck pain in the general population: a population-based cohort study. *Pain*, 112(3), 267-273.
- Cortes, C., Gonzalvo, X., Kuznetsov, V., Mohri, M., & Yang, S. (2017, July). Adanet: Adaptive structural learning of artificial neural networks. In *International conference on machine learning* (pp. 874-883). PMLR.
- Croft, P. R., Lewis, M., Papageorgiou, A. C., Thomas, E., Jayson, M. I., Macfarlane, G. J., & Silman, A. J. (2001). Risk factors for neck pain: a longitudinal study in the general population. *Pain*, 93(3), 317-325.
- Csáji, B. C. (2001). Approximation with artificial neural networks. *Faculty of Sciences, Eötvös Loránd University, Hungary*, 24(48), 7.

- Delacre, M., Lakens, D., & Leys, C. (2017). Why psychologists should by default use Welch's t-test instead of Student's t-test. *International Review of Social Psychology*, 30(1).
- Delp, S. L., Anderson, F. C., Arnold, A. S., Loan, P., Habib, A., John, C. T., ... & Thelen, D. G. (2007). OpenSim: open-source software to create and analyze dynamic simulations of movement. *IEEE transactions on biomedical engineering*, 54(11), 1940-1950.
- Derouin, A. J., & Fischer, S. L. (2019). Validation of a three-dimensional visual target acquisition system for evaluating the performance effects of head supported mass. *Applied ergonomics*, 76, 48-56.
- Dibblee, J., Worthy, P., Farrell, P., Hetzler, M., Reid, S., Stevenson, J., & Fischer, S. (2015). Evaluating a prototype device designed to alleviate night vision goggle induced neck strain among military personnel. *Ergonomics*, 58(12), 2067-2077.
- Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).
- Elman, J. L. (1990). Finding structure in time. *Cognitive science*, 14(2), 179-211.
- Farrell PSE, Shender BS, Goff CP, Baudou J, Crowley J, Davies M, Day SE, Di Muzio V, Dodson WD, Duvigneaud N, Feberg S, Fleischer H, Keillor J, Lopes M, van den Oord MHAH, Shivers BL, Sovelius R, Slungaard E, Smith A, Smith E, Weme T, Wong T, Wright Beatty H. Aircrew Neck Pain Prevention and Management. North Atlantic Treaty Organization (NATO) Task Research Group Human Factors and Medicine 252 (TR-HFM-252) Final Report. 2020.
- Fischer, A. (1992). A special Newton-type optimization method. *Optimization*, 24(3-4), 269-284.
- Forde, K. A., Albert, W. J., Harrison, M. F., Neary, J. P., Croll, J., & Callaghan, J. P. (2011). Neck loads and posture exposure of helicopter pilots during simulated day and night flights. *International Journal of Industrial Ergonomics*, 41(2), 128-135.

- Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* (Vol. 1, No. 10). New York: Springer series in statistics.
- Gagnon, D., Arjmand, N., Plamondon, A., Shirazi-Adl, A., & Larivière, C. (2011). An improved multi-joint EMG-assisted optimization approach to estimate joint and muscle forces in a musculoskeletal model of the lumbar spine. *Journal of biomechanics*, *44*(8), 1521-1529.
- Geman, S., Bienenstock, E., & Doursat, R. (1992). Neural networks and the bias/variance dilemma. *Neural computation*, *4*(1), 1-58.
- Gholipour, A., & Arjmand, N. (2016). Artificial neural networks to predict 3D spinal posture in reaching and lifting activities; Applications in biomechanical models. *Journal of biomechanics*, *49*(13), 2946-2952.
- Gilbert, J. C., & Nocedal, J. (1992). Global convergence properties of conjugate gradient methods for optimization. *SIAM Journal on optimization*, *2*(1), 21-42.
- Glorot, X., & Bengio, Y. (2010, March). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249-256). JMLR Workshop and Conference Proceedings.
- Goh, G. B., Hodas, N. O., & Vishnu, A. (2017). Deep learning for computational chemistry. *Journal of computational chemistry*, *38*(16), 1291-1307.
- Goodfellow, I., Bengio, Y., Courville, A.,. (2016). *Deep learning* (Vol. 1, No. 2). Cambridge: MIT press.
- Granata, K. P., & Marras, W. S. (1993). An EMG-assisted model of loads on the lumbar spine during asymmetric trunk extensions. *Journal of biomechanics*, *26*(12), 1429-1438.
- Granata, K. P., Lee, P. E., & Franklin, T. C. (2005). Co-contraction recruitment and spinal load during isometric trunk flexion and extension. *Clinical Biomechanics*, *20*(10), 1029-1037.
- Gurney, K. (1997). *An introduction to neural networks*. CRC press.

- Halilaj, E., Rajagopal, A., Fiterau, M., Hicks, J. L., Hastie, T. J., & Delp, S. L. (2018). Machine learning in human movement biomechanics: Best practices, common pitfalls, and new opportunities. *Journal of biomechanics*, *81*, 1-11.
- Harrison, M. F., Neary, J. P., Albert, W. J., Veillette, D. W., McKenzie, N. P., & Croll, J. C. (2007). Physiological effects of night vision goggle counterweights on neck musculature of military helicopter pilots. *Military medicine*, *172*(8), 864-870.
- Hastie, T., Tibshirani, R., & Friedman, J. (2009). Unsupervised learning. In *The elements of statistical learning* (pp. 485-585). Springer, New York, NY.
- Haykin, S. (2004). A comprehensive foundation. *Neural networks*, *2*(2004), 41.
- Healey LA, Derouin AJ, Callaghan JP, Cronin DS, Fischer SL. 2021. Night vision goggle and counterweight use affect neck muscle activity during reciprocal scanning. *Aerospace Medicine and Human Performance*. In Press. Manuscript ID: AMHP5673
- Healey, L. (2019). *The effects of operationally relevant head supported mass on neck muscle activity during a rapid scanning task* (Master's thesis, University of Waterloo).
- Hill, A. V. (1938). The heat of shortening and the dynamic constants of muscle. *Proceedings of the Royal Society of London. Series B-Biological Sciences*, *126*(843), 136-195.
- Hill, A. V. (1938). The heat of shortening and the dynamic constants of muscle. *Proceedings of the Royal Society of London. Series B-Biological Sciences*, *126*(843), 136-195.
- Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.
- Hinton, G., Srivastava, N., & Swersky, K. (2012). Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, *14*(8).
- Hochreiter, S. (1991). Untersuchungen zu dynamischen neuronalen Netzen. *Diploma, Technische Universität München*, *91*(1).

- Hochreiter, S., & Schmidhuber, J. (1997a). Long short-term memory. *Neural computation*, 9(8), 1735-1780.
- Hochreiter, S., & Schmidhuber, J. (1997b). LSTM can solve hard long time lag problems. *Advances in neural information processing systems*, 473-479.
- Hochreiter, S., Bengio, Y., Frasconi, P., & Schmidhuber, J. (2001). Gradient flow in recurrent nets: the difficulty of learning long-term dependencies.
- Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5), 359-366.
- Hou, Y., Zurada, J. M., Karwowski, W., Marras, W. S., & Davis, K. (2007). Estimation of the dynamic spinal forces using a recurrent fuzzy neural network. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(1), 100-109.
- Huber, Z. E. (2013). *Creation and Validation of a Dynamic, EMG-Driven Cervical Spine Model* (Doctoral dissertation, The Ohio State University).
- Ivancic, P.C., 2013. Effects of orthoses on three-dimensional load-displacement properties of the cervical spine. *Eur. Spine J.* 22, 169–177. <https://doi.org/10.1007/s00586-012-2552-0>
- Karpathy, A. (2016). Cs231n convolutional neural networks for visual recognition. *Neural networks*, 1(1).
- Karpathy, A., 2016. *CS231n Convolutional Neural Networks for Visual Recognition*.
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Kingma, I., Baten, C. T., Dolan, P., Toussaint, H. M., van Dieën, J. H., de Looze, M. P., & Adams, M. A. (2001). Lumbar loading during lifting: a comparative study of three measurement techniques. *Journal of Electromyography and Kinesiology*, 11(5), 337-345.
- Law, A. H., Beatty, H. E. W., Keillor, J., & Wickramasinghe, V. (2017, May). Pilot head and neck response to helicopter whole body vibration and head-supported mass. In *Proceedings of the AHS 73rd Annual Forum, Fort Worth, TX, USA* (pp. 9-11).

- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436-444.
- Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*.
- Lu, Y., Pulasani, P. R., Derakhshani, R., & Guess, T. M. (2013). Application of neural networks for the prediction of cartilage stress in a musculoskeletal system. *Biomedical signal processing and control*, 8(6), 475-482.
- Manoogian, S. J., Kennedy, E. A., Wilson, K. A., Duma, S. M., & Alem, N. M. (2006). Predicting neck injuries due to head-supported mass. *Aviation, space, and environmental medicine*, 77(5), 509-514.
- Masters, D., & Luschi, C. (2018). Revisiting small batch training for deep neural networks. *arXiv preprint arXiv:1804.07612*.
- Mattucci, S. F., Moulton, J. A., Chandrashekar, N., & Cronin, D. S. (2013). Strain rate dependent properties of human craniovertebral ligaments. *Journal of the mechanical behavior of biomedical materials*, 23, 71-79.
- McClure, P., Siegler, S., Nobilini, R., 1998. Three-dimensional flexibility characteristics of the human cervical spine in vivo. *Spine (Phila. Pa. 1976)*. 23, 216–223.
<https://doi.org/10.1097/00007632-199801150-00013>
- McGill, S.M., Jones, K., Bennett, G., Bishop, P.J., 1994. Passive stiffness of the human neck in flexion, extension, and lateral bending. *Clin. Biomech. (Bristol, Avon)* 9, 193–8.
[https://doi.org/10.1016/0268-0033\(94\)90021-3](https://doi.org/10.1016/0268-0033(94)90021-3)
- McKinnon, C. D., Dickerson, C. R., Laing, A. C., & Callaghan, J. P. (2016). Neck muscle activity during simulated in-flight static neck postures and helmet mounted equipment. *Occupational Ergonomics*, 13(3-4), 119-130.
- Moré, J. J. (1978). The Levenberg-Marquardt algorithm: implementation and theory. In *Numerical analysis* (pp. 105-116). Springer, Berlin, Heidelberg.

- Moshagen, T., Adde, N. A., & Rajgopal, A. N. (2021). Finding hidden-feature depending laws inside a data set and classifying it using Neural Network. *arXiv preprint arXiv:2101.10427*.
- Mousavi, S. H., Sayyaadi, H., & Arjmand, N. (2020). Prediction of the thorax/pelvis orientations and L5–S1 disc loads during various static activities using neuro-fuzzy. *Journal of Mechanical Science and Technology*, *34*(8), 3481-3485.
- Netto, K. J., Burnett, A. F., Green, J. P., & Rodrigues, J. P. (2008). Validation of an EMG-driven, graphically based isometric musculoskeletal model of the cervical spine. *Journal of biomechanical engineering*, *130*(3).
- Ng, A., Katanforoosh, K., Mourri, B., Lecture 1.10 – Long Short Term Memory, COURSERA: Sequence Modelling. Online Course, Accessed October 1, 2020.
- Nielsen, M. A. (2015). *Neural networks and deep learning* (Vol. 25). San Francisco, CA: Determination press.
- Norman, R., Wells, R., Neumann, P., Frank, J., Shannon, H., Kerr, M., & Study, T. O. U. B. P. (1998). A comparison of peak vs cumulative physical work exposure risk factors for the reporting of low back pain in the automotive industry. *Clinical biomechanics*, *13*(8), 561-573.
- Pandey, D. S., Das, S., Pan, I., Leahy, J. J., & Kwapinski, W. (2016). Artificial neural network based modelling approach for municipal solid waste gasification in a fluidized bed reactor. *Waste management*, *58*, 202-213.
- Panjabi, M., Dvorak, J., Duranceau, J., Yamamoto, I., Gerber, M., Rauschnig, W., & Bueff, H. (1988). Three-dimensional movements of the upper cervical spine. *Spine*, *13*(7), 726-730.
- Panjabi, M.M., Crisco, J.J., Vasavada, A., Oda, T., Cholewicki, J., Nibu, K., Shin, E., 2001. Mechanical Properties of the Human Cervical Spine as Shown by Three-Dimensional Load–Displacement Curves. *Spine (Phila. Pa. 1976)*. *26*, 2692–2700.
<https://doi.org/10.1097/00007632-200112150-00012>

- Parkinson, R. J., & Callaghan, J. P. (2009). The role of dynamic flexion in spine injury is altered by increasing dynamic load magnitude. *Clinical biomechanics*, 24(2), 148-154.
- Pascanu, R., Mikolov, T., & Bengio, Y. (2013, May). On the difficulty of training recurrent neural networks. In *International conference on machine learning* (pp. 1310-1318). PMLR.
- Penumudi, S. A., Kuppam, V. A., Kim, J. H., & Hwang, J. (2020). The effects of target location on musculoskeletal load, task performance, and subjective discomfort during virtual reality interactions. *Applied ergonomics*, 84, 103010.
- Philipp, G., Song, D., & Carbonell, J. G. (2018). Gradients explode-deep networks are shallow-resnet explained.
- Prechelt, L. (1998). Early stopping-but when?. In *Neural Networks: Tricks of the trade* (pp. 55-69). Springer, Berlin, Heidelberg.
- Reed, R., & Marks II, R. J. (1999). *Neural smithing: supervised learning in feedforward artificial neural networks*. Mit Press.
- Reeves, N. P., & Cholewicki, J. (2003). Modeling the human lumbar spine for assessing spinal loads, stability, and risk of injury. *Critical Reviews™ in Biomedical Engineering*, 31(1&2).
- Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1985). *Learning internal representations by error propagation*. California Univ San Diego La Jolla Inst for Cognitive Science.
- Shannon SG, Mason KT. U.S. Army Aviation Life Support Equipment Retrieval Program: Head and Neck Injury Among Night Vision Goggle Users in Rotary-Wing Mishaps. USAARL Rep Number 98-02. 1997;(98).
- Sharma, A., Bhuriya, D., & Singh, U. (2017, April). Survey of stock market prediction using machine learning approach. In *2017 International conference of electronics, communication and aerospace technology (ICECA)* (Vol. 2, pp. 506-509). IEEE.

- Sheela, K. G., & Deepa, S. N. (2013). Review on methods to fix number of hidden neurons in neural networks. *Mathematical Problems in Engineering*, 2013.
- Shrestha, A., & Mahmood, A. (2019). Review of deep learning algorithms and architectures. *IEEE Access*, 7, 53040-53065.
- Sidorov, A., & Sidorov, A. (2018, June 26). Transitioning entirely to neural machine translation. Retrieved April 9, 2021, from <https://engineering.fb.com/2017/08/03/ml-applications/transitioning-entirely-to-neural-machine-translation/>
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1), 1929-1958.
- Stokes, I. A. (2005). Relationships of EMG to effort in the trunk under isometric conditions: force-increasing and decreasing effects and temporal delays. *Clinical Biomechanics*, 20(1), 9-15.
- Sutskever, I., Martens, J., Dahl, G., & Hinton, G. (2013, May). On the importance of initialization and momentum in deep learning. In *International conference on machine learning* (pp. 1139-1147). PMLR.
- Tetko, I. V., Livingstone, D. J., & Luik, A. I. (1995). Neural network studies. 1. Comparison of overfitting and overtraining. *Journal of chemical information and computer sciences*, 35(5), 826-833.
- van den Oord, M. H., De Loose, V., Meeuwssen, T., Sluiter, J. K., & Frings-Dresen, M. H. (2010). Neck pain in military helicopter pilots: prevalence and associated factors. *Military medicine*, 175(1), 55-60.
- Van Houdt, G., Mosquera, C., & Napoles, G. (2020). A review on the long short-term memory model. *Artificial Intelligence Review*, 53, 5929-5955.

- Vasavada, A. N., Li, S., & Delp, S. L. (1998). Influence of muscle morphometry and moment arms on the moment-generating capacity of human neck muscles. *Spine*, 23(4), 412-422.
- Wu, G., Siegler, S., Allard, P., Kirtley, C., Leardini, A., Rosenbaum, D., ... & Stokes, I. (2002). ISB recommendation on definitions of joint coordinate system of various joints for the reporting of human joint motion—part I: ankle, hip, and spine. *Journal of biomechanics*, 35(4), 543-548.
- Xu, D., Shi, Y., Tsang, I. W., Ong, Y. S., Gong, C., & Shen, X. (2019). Survey on multi-output learning. *IEEE transactions on neural networks and learning systems*, 31(7), 2409-2429.
- Ying, X. (2019, February). An overview of overfitting and its solutions. In *Journal of Physics: Conference Series* (Vol. 1168, No. 2, p. 022022). IOP Publishing.
- Yoganandan, N., Pintar, F.A., Stemper, B.D., Wolfla, C.E., Shender, B.S., Paskoff, G., 2007. Level-dependent coronal and axial moment-rotation corridors of degeneration-free cervical spines in lateral flexion. *JBJS* 89, 1066–1074.
- Yoganandan, N., Stemper, B.D., Pintar, F.A., Baisden, J.L., Shender, B.S., Paskoff, G., 2008. Normative segment-specific axial and coronal angulation corridors of subaxial cervical column in axial rotation. *Spine* (Phila. Pa. 1976). 33, 490–496.
- Zadeh, L. A. (1996). Fuzzy sets. In *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by Lotfi A Zadeh* (pp. 394-432).
- Zhang, T., Gao, C., Ma, L., Lyu, M., & Kim, M. (2019, October). An empirical study of common challenges in developing deep learning applications. In *2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE)* (pp. 104-115). IEEE.

Appendix A: Results of Hyperparameter Tuning Stage Two

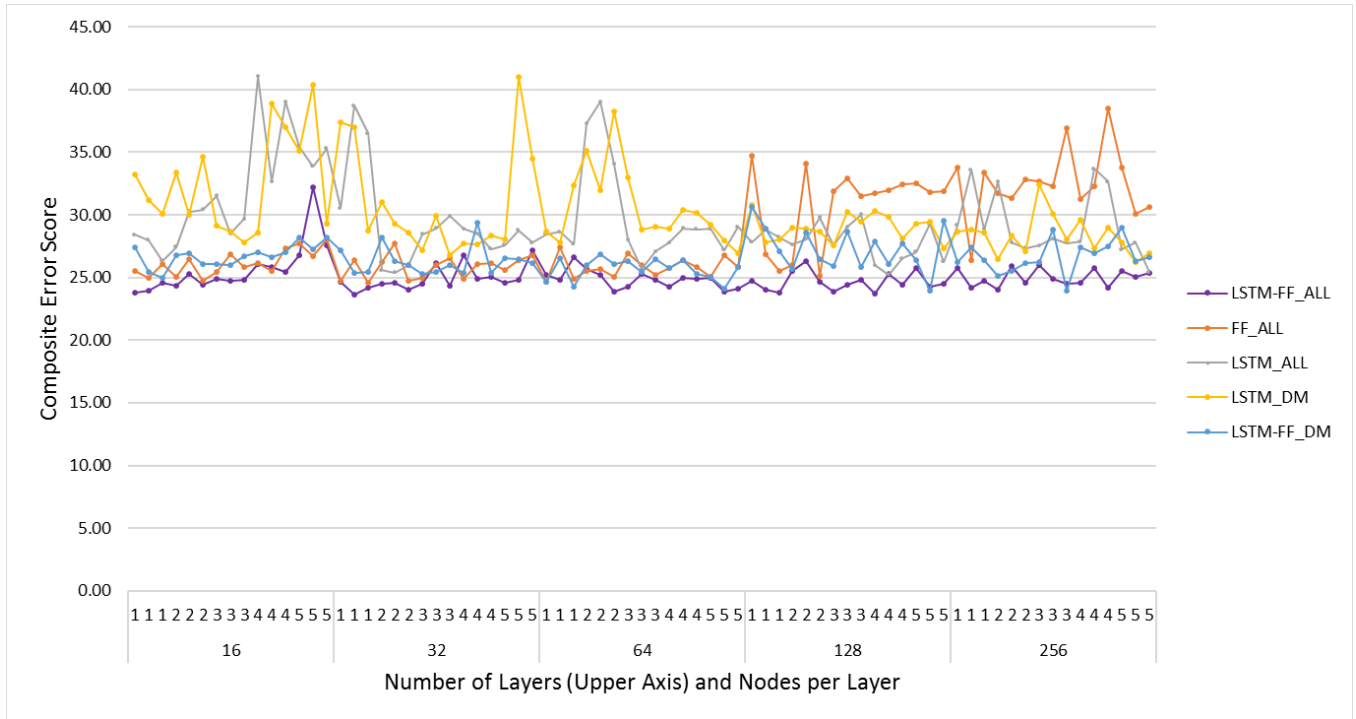


Figure 42: Composite error scores of dual-rotation ANNs of various types with a range of number of layers and nodes per layer.

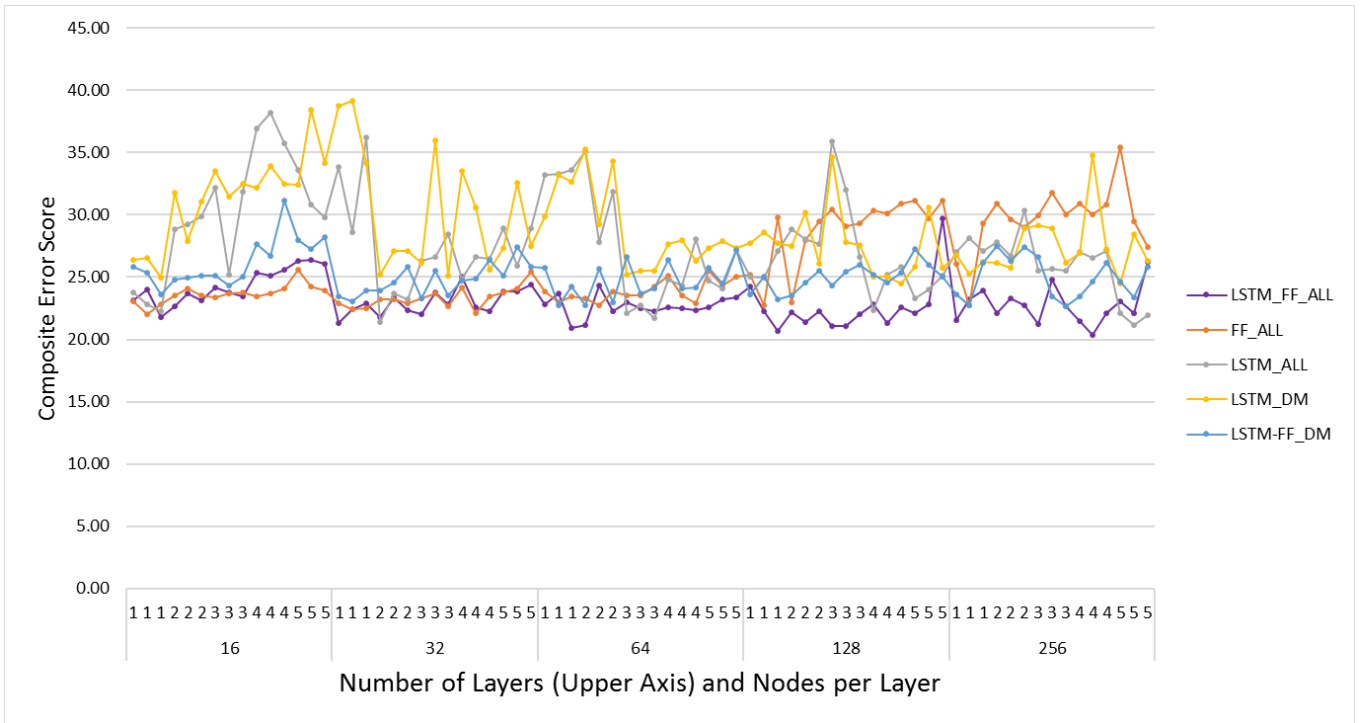


Figure 43: Composite error scores of labelled-trial ANNs of various types with a range of number of layers and nodes per layer.

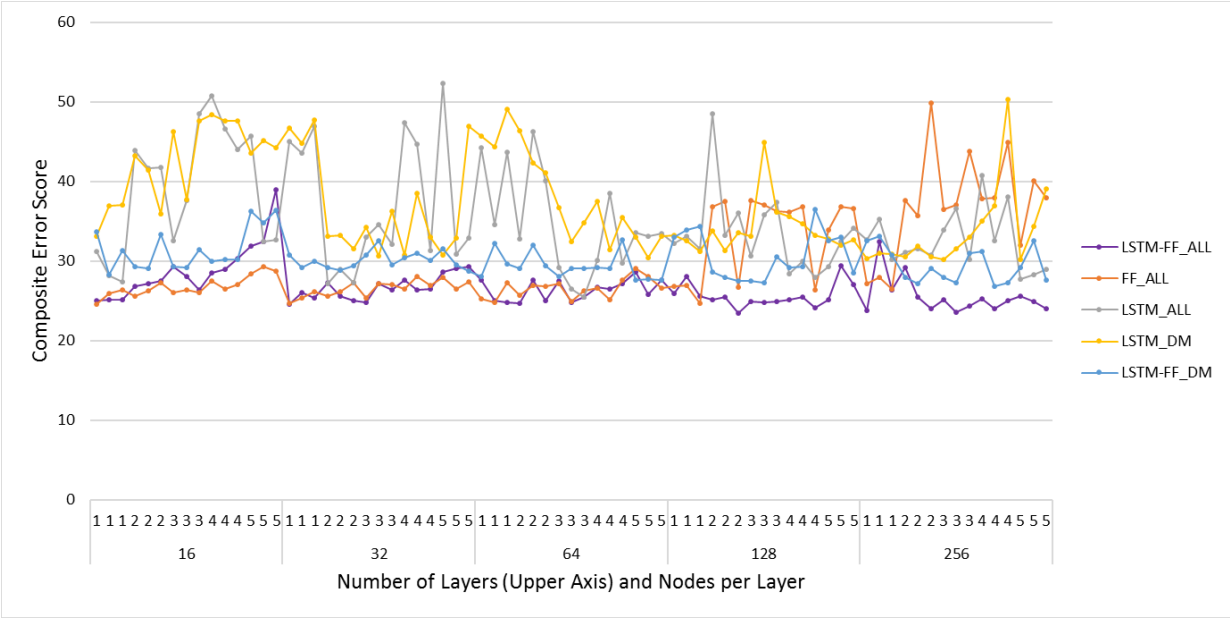


Figure 44 Composite error scores of pitch-specific ANNs of various types with a range of number of layers and nodes per layer.

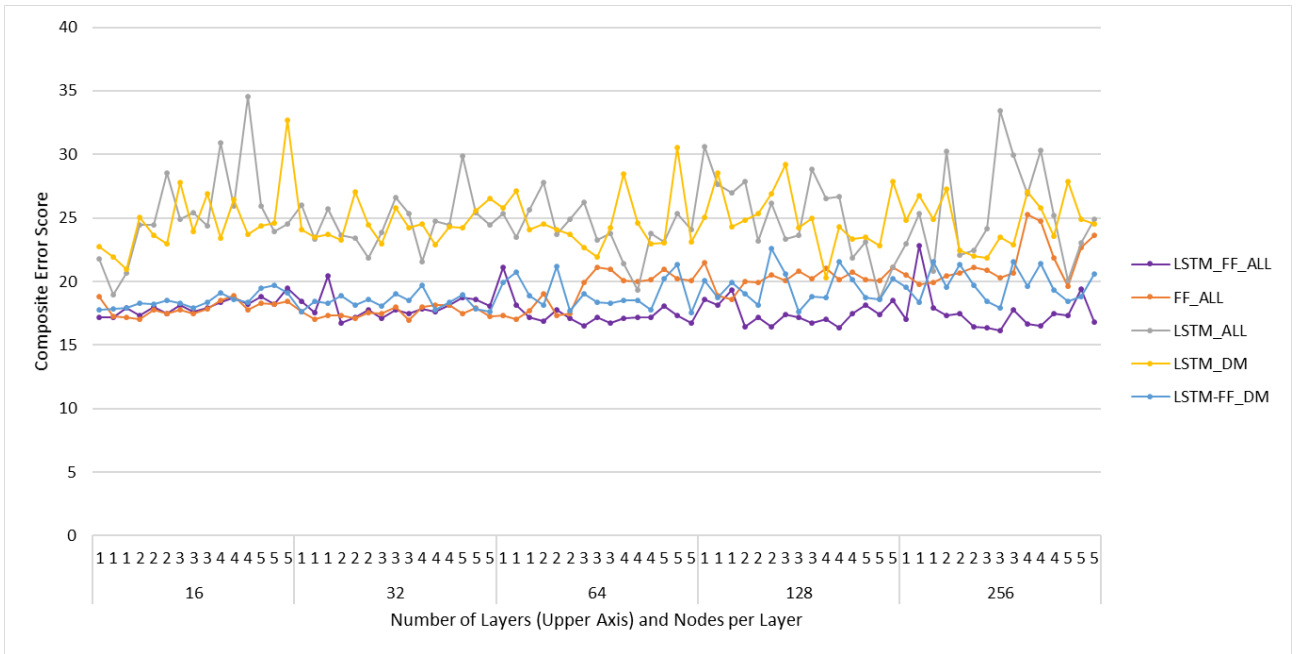


Figure 45: Composite error scores of yaw-specific ANNs of various types with a range of number of layers and nodes per layer.

Appendix B: Results of Hyperparameter Tuning Stage Two

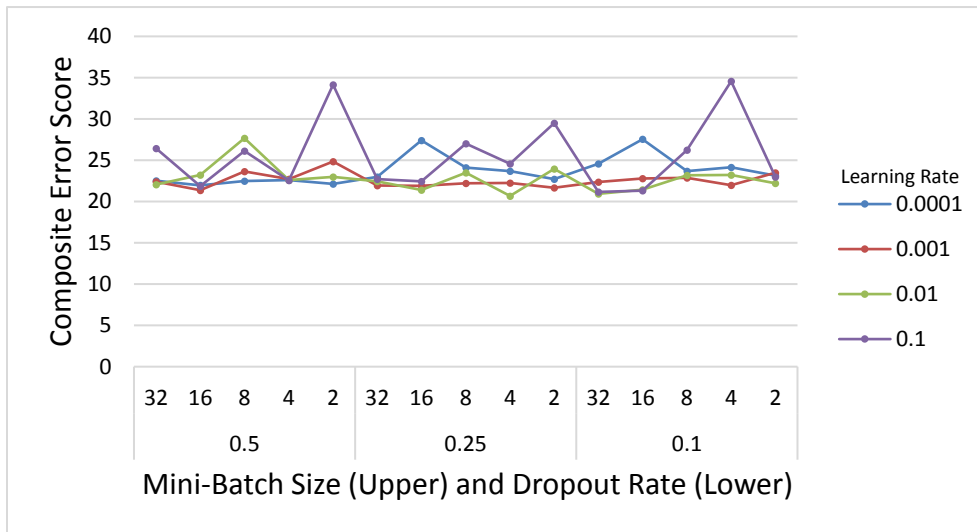


Figure 46: Composite error score of dual-rotation LSTM-FF ANNs with 3 hidden layers and a variety of learning rates, mini-batch sizes and dropout rates when applied to the validation set.

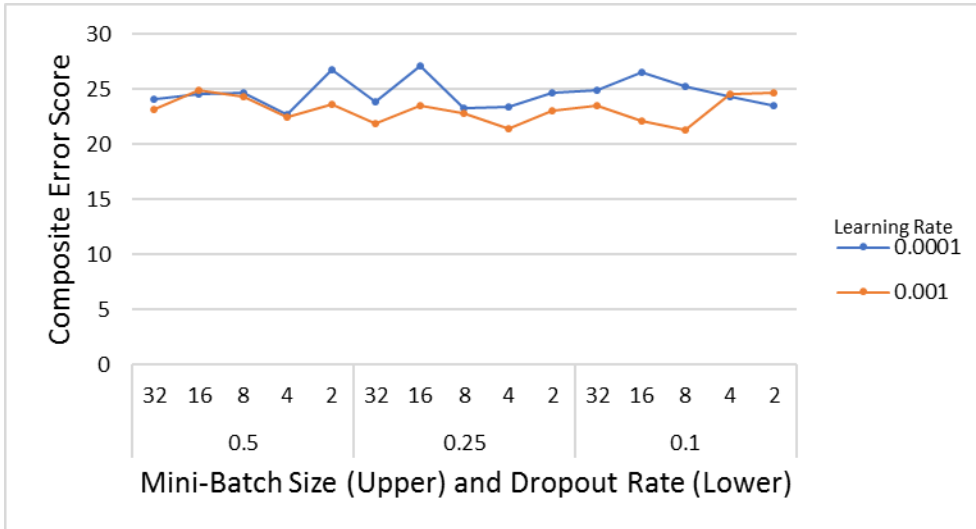


Figure 47: Composite error score of labelled-trial LSTM-FF ANNs with 3 hidden layers and a variety of learning rates, mini-batch sizes and dropout rates when applied to the validation set. Although ANNs with a learning rate of 0.1 and 0.01 were developed, they yielded composite error scores that were consistently 2 to 5 times higher than ANNs with other learning rates and were therefore excluded from the figure.

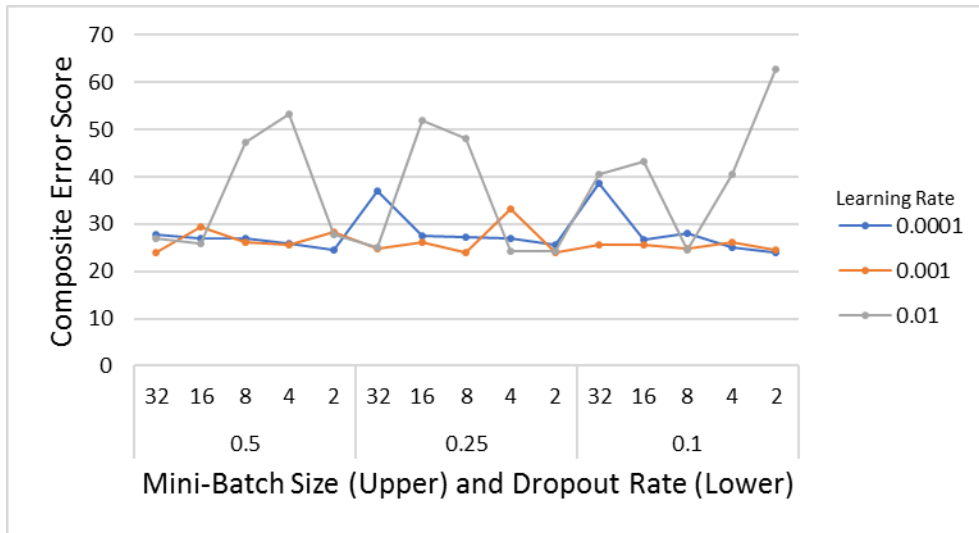


Figure 48: Composite error score of pitch-specific LSTM-FF ANNs with 3 hidden layers and a variety of learning rates, mini-batch sizes and dropout rates when applied to the validation set. Although ANNs with a learning rate of 0.1 were developed, they yielded composite error scores that were consistently 2 to 5 times higher than ANNs with other learning rates and were therefore excluded from the figure.

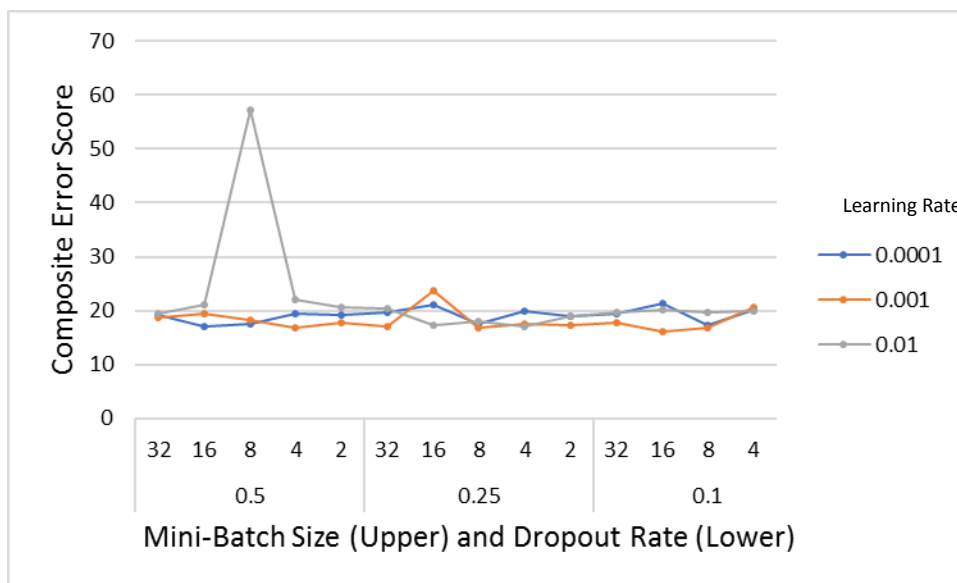


Figure 49: Composite error score of yaw-specific LSTM-FF ANNs with 3 hidden layers and a variety of learning rates, mini-batch sizes and dropout rates when applied to the validation set. Although ANNs with a learning rate of 0.1 were developed, they yielded composite error scores that were consistently 2 to 5 times higher than ANNs with other learning rates and were subsequently excluded from the figure.

Appendix C: Scatterplots of ANN-Predicted vs. EMG-Driven Model C6-C7 Compression and Shear Outputs

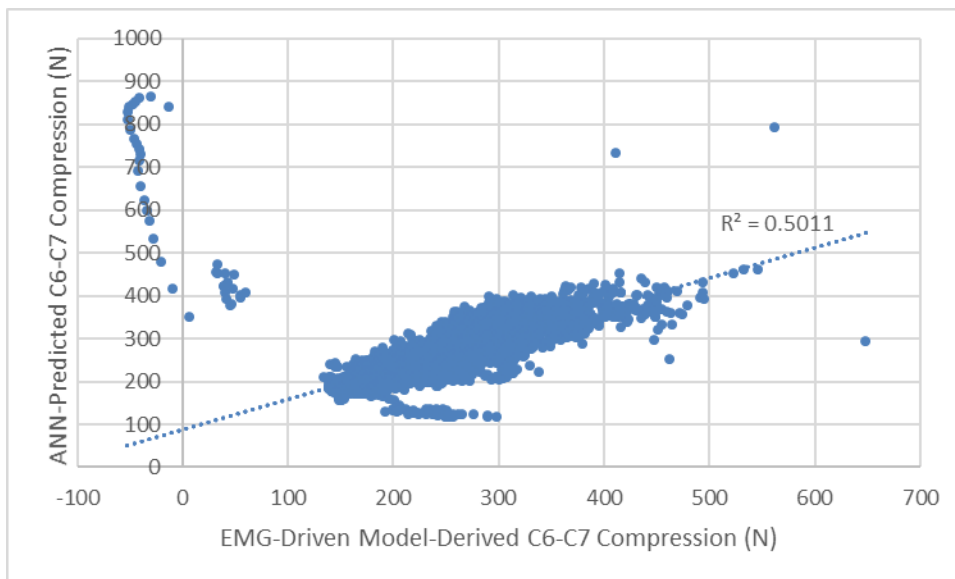


Figure 50: Scatterplot comparing C6-C7 compression outputs of the best performing LSTM-FF_{ALL} ANN and corresponding EMGD model-derived C6-C7 compression during pitch rotation trials. Each point represents data from one frame from the test dataset.

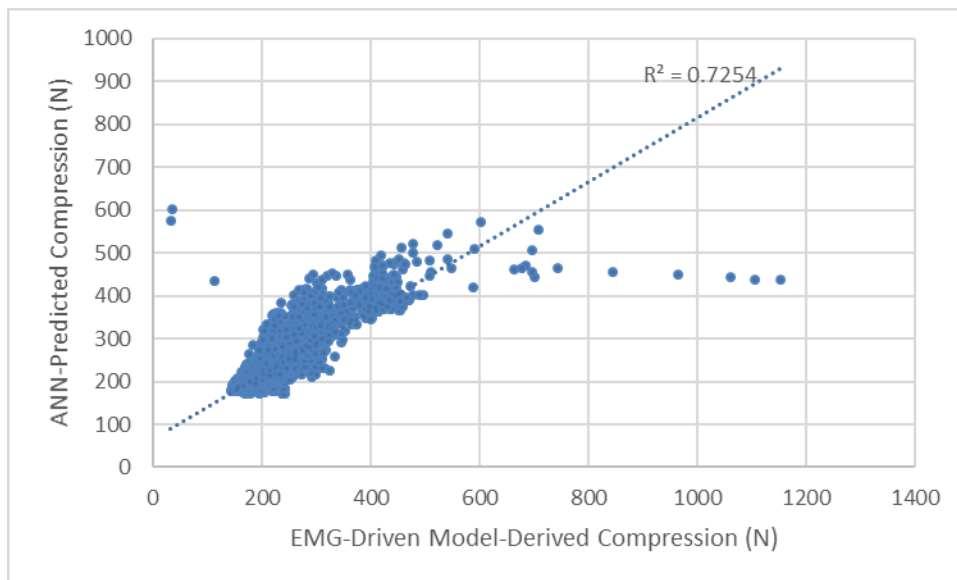


Figure 51: Scatterplot comparing C6-C7 compression outputs of the best performing LSTM-FF_{ALL} ANN and corresponding EMGD model-derived C6-C7 compression during yaw rotation trials. Each point represents data from one frame from the test dataset.

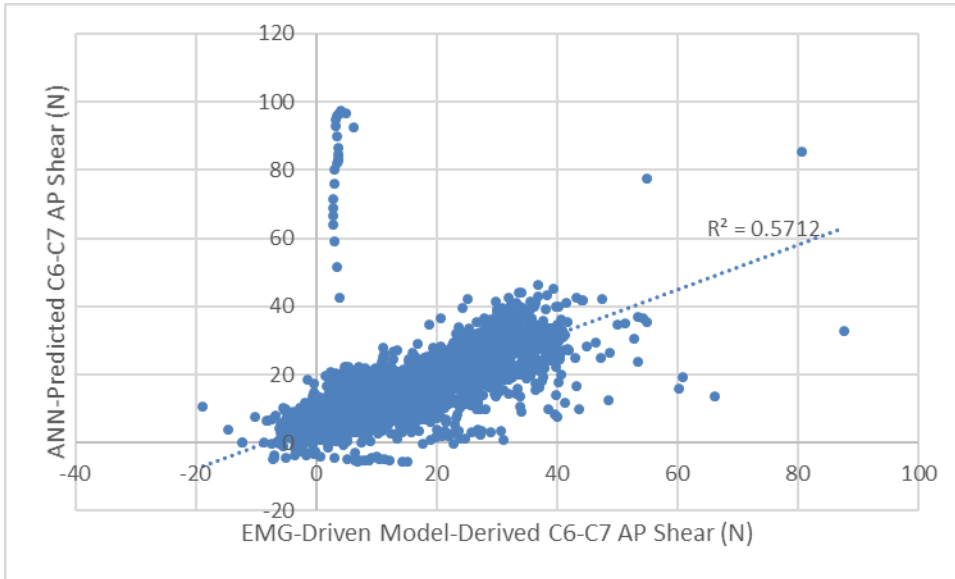


Figure 52: Scatterplot comparing C6-C7 shear outputs of the best performing LSTM- FF_{ALL} ANN and corresponding EMGD model-derived C6-C7 shear during pitch rotation trials. Each point represents data from one frame from the test dataset.

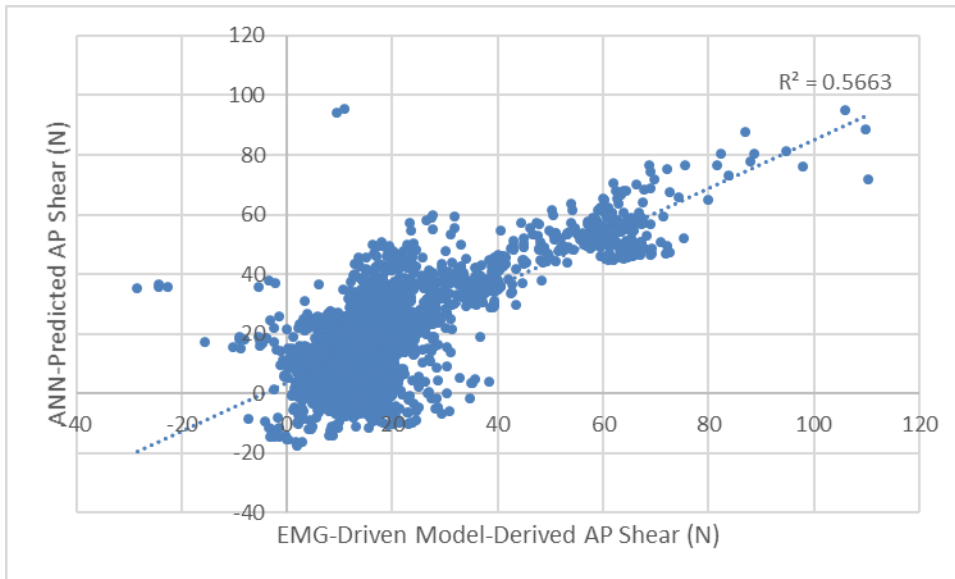


Figure 53: Scatterplot comparing C6-C7 shear outputs of the best performing LSTM- FF_{ALL} ANN and corresponding EMGD model-derived C6-C7 shear during yaw rotation trials. Each point represents data from one frame from the test dataset.

Appendix D: ANN Performance Across Multiple Cycles Within a Trial

The recurrent nature of the best-performing ANNs in this study, specifically the ability of LSTM-based ANNs to recall information over long durations, raises the question of whether ANN performance progressively improves over the course of a sequence of inputs. To investigate this, ANN error magnitudes in peak and cumulative compression and shear (as calculated in Chapter 3.7.1) were compared across head movement cycles within each trial of the test set. The minimum number of completed head movement cycles within the pitch and yaw trials was two and four cycles respectively. Therefore, comparison of ANN performance was performed between the first and second head movement cycle during pitch trials and between the first, second and third and fourth head movement cycles during yaw trials.

Repeated measures ANOVAs (RMANOVAs) were performed to detect differences in absolute error magnitudes of ANN-predicted vs. EMGD model-derived peak and cumulative compression and shear values within the first, second, third and fourth head movement cycles of yaw trials from the test set. Absolute error of ANN-predicted peak and cumulative loading values were the dependent variable while cycle number was the independent variable. Greenhouse-Geisser corrections were applied where assumptions of sphericity were not met as determined by Mauchly's test. Post hoc Tukey's pairwise comparisons were performed to identify significant differences between individual head movement cycle numbers where appropriate. A RMANOVA failed to detect significant differences in cumulative compression error magnitudes across head movement cycles (p adj. = 0.59). However,

significant differences were detected for error magnitudes of cumulative shear (p adj. > 0.01), peak compression (p adj. = 0.12) and peak shear (p adj. > 0.01). The results of pairwise comparisons are presented in Figures 54-57.

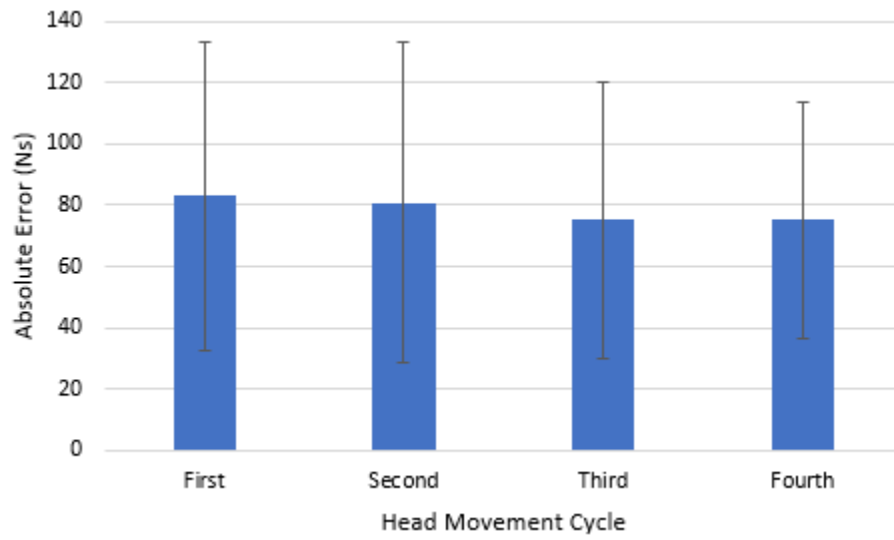


Figure 54: Absolute error of ANN-predicted C6-C7 cumulative compression vs. EMGD model-derived values across the first four head movement cycles of yaw trials within the test set.

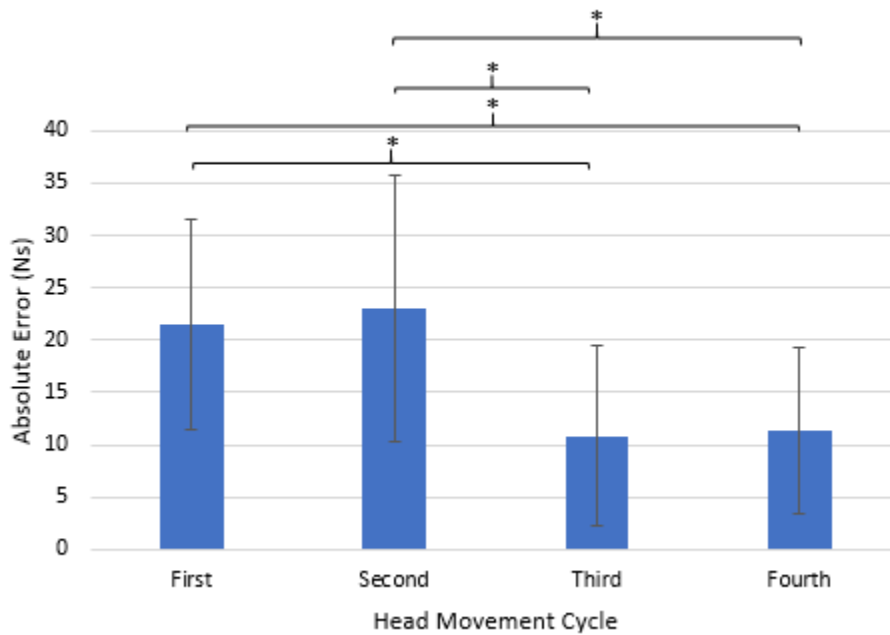


Figure 55: Absolute error of ANN-predicted C6-C7 cumulative shear vs. EMGD model-derived values across the first four head movement cycles of yaw trials within the test set. Asterisks denote significant pairwise differences ($p < 0.05$).

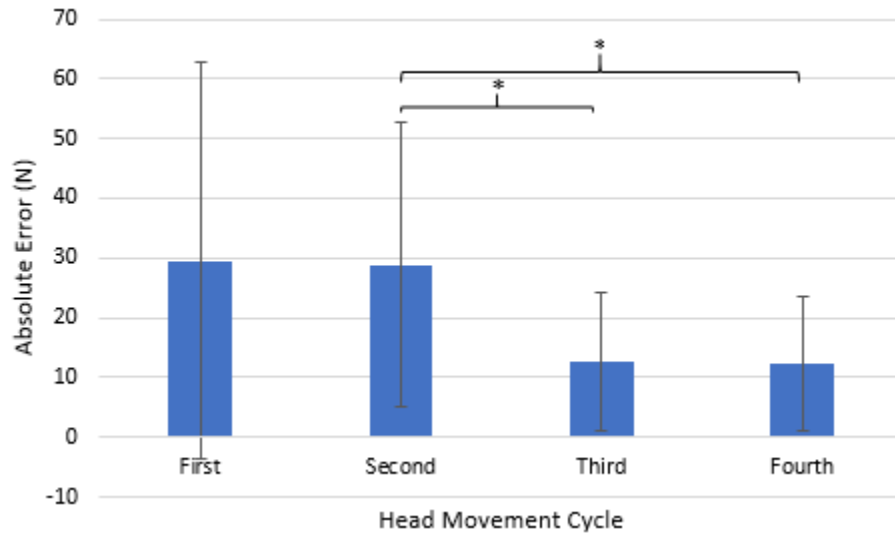


Figure 56: Absolute error of ANN-predicted C6-C7 peak compression vs. EMGD model-derived values across the first four head movement cycles of yaw trials within the test set. Asterisks denote significant pairwise differences ($p < 0.05$).

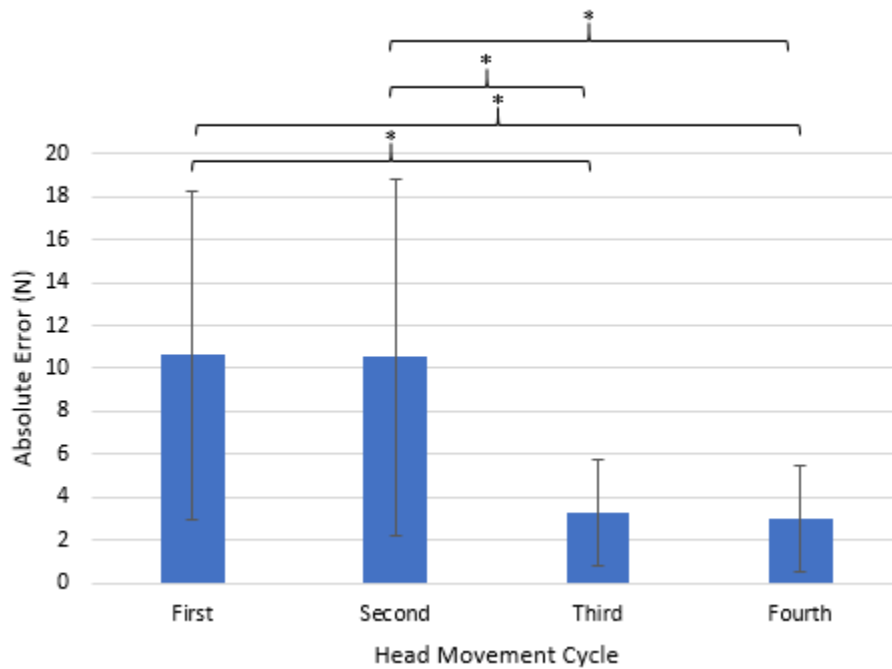


Figure 57: Absolute error of ANN-predicted C6-C7 peak compression vs. EMGD model-derived values across the first four head movement cycles of yaw trials within the test set. Asterisks denote significant pairwise differences ($p < 0.05$).

Two-tailed paired sample t-tests were performed to compare the magnitudes of absolute error in ANN-predicted peak and cumulative compression and shear between the first two head movement cycles of pitch trials within the test set (Table 11). Similar to in yaw trials, there were no significant differences in cumulative compression ($p = 0.87$) between cycle numbers. In contrast, absolute error magnitudes were significantly lower during the second vs. first head movement cycle in cumulative shear ($p = 0.03$), peak compression ($p = 0.049$) and peak shear ($p = 0.04$).

Table 11: Absolute error of ANN-predicted vs. EMGD model-derived values across the first two head movement cycles of pitch trials within the test set.

	First Cycle	Second Cycle
Cumulative Compression (Ns) (SD)	121.52 (71.47)	118.84 (53.41)
Cumulative Shear (Ns) (SD)	9.69 (6.79)	6.09 (3.63)
Peak Compression (N) (SD)	37.95 (28.92)	24.52 (18.19)
Peak Shear (N) (SD)	4.29 (3.09)	2.44 (1.91)

Results indicate that the LSTM-based ANNs in this thesis perform more effectively on later head movement cycles within a given trial. During yaw trials, cumulative shear, peak compression and peak shear error magnitudes were lower in the third and fourth movement cycles vs. the first and second cycles. During pitch trials, error in these variables of interest was lower in the second vs. first head movement cycle. Cumulative compression error magnitudes did not exhibit statistically significant differences across head movement cycles during both pitch and yaw trials. Overall, decreases in error magnitudes in later movement cycles are likely related to the ability of LSTM-based ANNs to recall information over long durations within a sequence.

Although this analysis did not assess whether trends in improved performance continue past two and four movement cycles in pitch and yaw trials respectively, it appears that ANN performance generally improved over the course of a trial. When applying the ANNs from this thesis within a helicopter helmet CAD process, designers should apply the ANNs to input sequences that include several head movement cycle repetitions. Additionally, they should draw conclusions based on ANN predictions from later cycles within a trial where available.