UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

A CONSTRAINT DRIVEN APPROACH TO NEURAL AND MUSCLE RECRUITMENT IN WRIST MOTOR TASKS

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

By

MANUEL LOPEZ - SANTILLANA
Norman, Oklahoma
2020

A CONSTRAINT DRIVEN APPROACH TO
NEURAL AND MUSCLE RECRUITMENT IN WRIST MOTOR TASKS

A THESIS APPROVED FOR THE
GALLOGLY COLLEGE OF ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. Andrew H. Fagg, Chair

Dr. Amy McGovern

Dr. Christan Grant

Acknowledgements

I would like to thank Dr.Andrew H. Fagg for his continued support and guidance throughout my academic journey. This thesis would not have been possible without his help. I would also like to thank my family and friends who have encouraged me to follow my dreams. I would like to thank the thesis committee members (Dr. Andrew H. Fagg, Dr. Amy McGovern, and Dr.Christan Grant) who provided valuable insights and the reader for taking the time to make my efforts worthwhile.

Contents

Abstract

The transformation from visual stimulus to muscle recruitment is a non-linear one: it must take into account the configuration of the body and the lines of action of the muscles. The primary cortex plays a role in the muscle recruitment process. Any limb movement can be represented in multiple coordinate frames; a musculoskeletal representation, where forces are enacted through muscles, which will be referred to as an intrinsic representation, or by tracking the resulting visual movement, taking the perspective of a onlooker, providing an extrinsic representation of the movement. A coordinate system can also be drawn in respect to the joint orientation. How limb movements are represented in the primary motor cortex has been an intense focus of research.

Georgopoulos et al. (1982) established that the activity of individual neurons in the primary cortex can be fit to a cosine curve, where the peak represents an extrinsic direction for which they are maximally activated when performing limb movements. Due to the association of the neuron's activity with an extrinsic direction, Georgopoulos et al. (1982) referred to the peak activation as the preferred direction (PD), favoring an extrinsic representation of neural activity. Later on, Kakei et al. (1999) focused on wrist movement and

the corresponding neural activity in a center-out wrist task. Kakei et al. (1999) observed that the preferred direction of neurons in certain wrist postures: the supinated, the pronated, and the midrange. Through this observation Kakei et al. (1999) found that a subgroup of the neurons seem to shift the location of the preferred direction depending on the wrist posture. In this experiment, the neurons in the primary motor cortex (MI) region of the brain had their preferred direction shift less than the actual rotation of the wrist. Oby et al. (2012) expanded upon this, finding that the corresponding wrist force task also had PD shifts, with both authors concluding that the resulting PD shifts, on average, were less than the wrist rotation, with the average MI PD shift being less than the average wrist force PD.

To model what sort of transformations are possible in the primary motor cortex, Craig (2013) devised a neural network model as an abstract representation of this transformation process. Craig's model is an recurrent neural network (RNN) that computes the transformation from visual input to forces, having no assumptions of the transformations that should occur. The model of Craig seeks to minimize the MSE of the forces produced and regularization, the activity of the RNN is based upon the minimization of the loss.

In this thesis, I provide an extension of the original neural network model of force generation in a center-out wrist task that was developed by Craig (2013). The Craig's model inputs are the extrinsic representation of the target, wrist configuration, and task cues. The model is tasked with generating the corresponding extrinsic wrist force trajectories given those inputs. My model takes advantage of the Python library of Keras (François, 2015), allowing

for an ease of access to deep learning tools. The model is trained from the force trajectory of Todorov and Jordan (1998) allowing for a more modern approach by employing dropout and having a more accurate force depiction. My model also makes the distinction between target direction and the onset of forces. The disassociation between these two inputs allows for a better understanding of the mechanics and transformations that are associated with each cue. Incorporation of the supinated position into my model permits the model's activity to be evaluated in regard to this posture.

In my model, a sub population of neurons consistently exhibit cosine tuning through various neuron choices and regularization parameters. The sub population was present for all postures, with the supinated posture consistently containing less stable neurons than the pronated and midrange postures Other neurons of the model exhibited activity shifts that mainly were time dependent, with little change in activity with respect to target location. The MI neurons did not exhibit a PD shift across changes in wrist posture. However, the muscles exhibited a significant difference in PD shift.

Chapter 1

Introduction

When performing a movement, the primary motor cortex (MI) region of the brain is involved in generating signals that travel down the spinal cord, to produce muscle recruitment patterns, that result in the external generation of forces.

For this encoding to take place, primary motor cortex neurons must be able to modulate in order to perform the desired outcome. Georgopoulos et al. (1982) concluded that neurons in the primary motor cortex modulate based on the form and timing of the movement direction. Georgopoulos et al. (1982) observed the primary motor cortex neurons' activity levels while a monkey performed a center-out reaching task, in which the monkey first held its hand at a center location, then, when seeing a visual target appear along the circle centered around the monkey's hand, the monkey moved the hand to the visually indicated target. During this time, the hand trajectory and the neural activity were recorded. Georgopoulos et al. (1982) modeled the neural

activation patterns as a cosine of the difference between the direction of movement and a neuron's "preferred direction." A *Preferred Direction* (PD) is the direction in which a cell is maximally activated. Georgopoulos et al. (1986) theorized that neurons implement a voting scheme in which each neuron's level of activity implies the strength of vote. These votes are then summed to give a net movement direction of the arm.

Georgopolus worked within an extrinsic Cartesian coordinate frame. His work was in relation to that coordinate frame. However, it has been shown that neurons also modulate in accordance to limb posture (Scott and Kalaska, 1997), with movement direction (Ajemian et al., 2001), the inertial load's direction (Mussa-Ivaldi, 1988), and movement force direction (Kalaska et al., 1989).

Kakei et al. (1999) devised an experiment to clarify the debate on how movement is encoded in MI through the relationship between the wrist and the intrinsic and extrinsic coordinate frames. Kakei et al. (1999) created trials where there was a disassociation between wrist posture from corresponding generated extrinsic movement. Kakei et al. (1999) examined the neural modulation in a wrist-based center-out task where the forearm was in one of three postures: pronated, midrange, and supinated. In these trials, the monkey was given the task of moving a manipulandum to a target location. This was done by moving its wrist along the flexion/extension and radial/ulnar deviation. The monkey was tasked with moving from a center position to one of the eight targets. The targets were equally spaced around a unit circle which was centered on the hand in its resting position. By observing the PDs of each

forearm configuration, Kakei et al. (1999) quantified each neuron's PD change, referred to as the *PD shift* from the pronated posture to the supinated one. From these trials, Kakei et al. (1999) described a bimodal distribution, with one population of neurons exhibiting almost no PD shift, while the other had shifts that corresponded with changes in muscle activation PDs. The former population was labeled as extrinsic, where the latter, having a correspondence to PDs, was deemed *muscle-like*.

Kakei et al. (1999) proposed that the processing that occurred during these trials was of a serial nature, where the extrinsic visual information underwent a transformation to an intermediate state or a muscle-like representation before reaching spinal motor neurons. Since both extrinsic and muscle-like populations were found in the primary motor cortex, Kakei et al. (1999) came to the conclusion that the primary motor cortex partially handled this transformation process. Kakei et al. (1999) came to this conclusion after viewing two different populations of neurons simultaneously present at different stages of the process.

Oby et al. (2012) provided an extension of this research by creating an isometric wrist force experiment, in which a monkey applied forces on a manipulandum. The movement of the monkey's cursor location corresponded to the generated force trajectory on the manipulandum. The monkey was tasked with applying the forces to reach an indicated target location, where the target locations were evenly spaced across a unit circle that was centered at the monkey's initial resting position. Oby et al. (2012) showed that the resulting PD shift distribution from pronated to midrange was unimodal. The neuron's

PD shift mean was found to be closer to that of a muscle-like distribution than that of an extrinsic coordinate system.

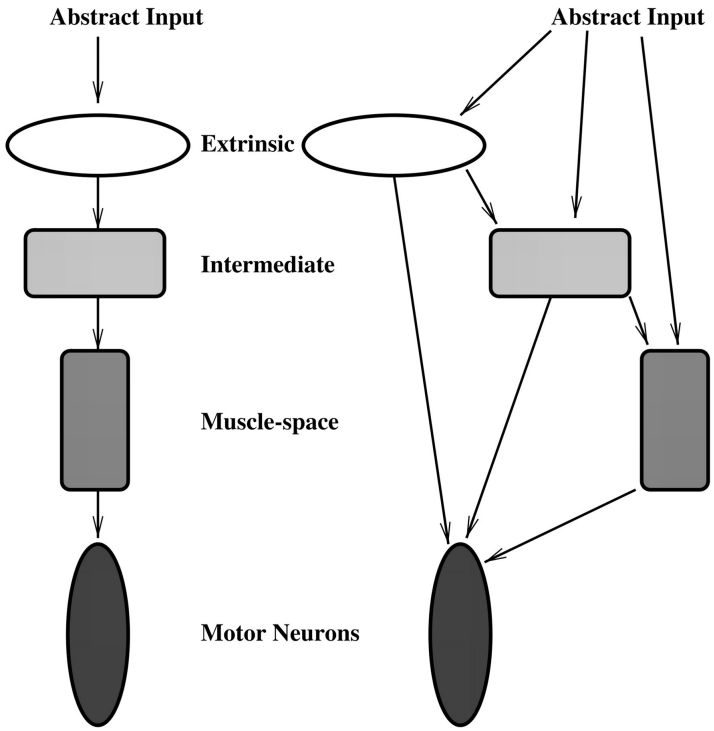

Figure 1.1: The parallel processing scheme as proposed by Shah et al. (2004), used with permission.

The idea that MI neurons were based around a muscle-like representation was challenged by Shah et al. (2004), demonstrating that a linear transformation exists from extrinsic neurons that have a posture-dependent depth of modulation or DoM to the right muscle activation patterns. It was from this result that a parallel processing scheme was proposed in which an abstract input can take on different forms to produce the same motor function.

In Shah's scheme, shown in Figure 1.1, the input can be transformed from an extrinsic state to an intermediate state, and from an intermediate repre-

sentation to a muscle space representation. Each of these MI encodings can directly drive motor neurons and have a direct encoding from the abstract input. However, the process in which these movements and transformations are encoded remains obscure, with neurons representing a wide range of transformations and not conforming to a single coordinate frame. This is especially prevalent with hyper-redundancies of neural hardware, where a movement execution is encoded many times in different ways to ensure a proper execution of that action (Moorman et al., 2017).

The recurrent neural network framework proposed by Craig (2013) is structured to make the transformation from extrinsic stimulus to wrist-muscle activation and force trajectory in a manner that is not predetermined. The model is constrained to minimize muscle and neural activation levels. The neurons in the model show a modulation of movement direction and distribution of PD shifts across postures. These modulations and PD shifts are similar to those seen in the monkey experiments. The neuron behavior of the model is selected by a learning algorithm so as to produce the desired forces while minimizing the weights and activation of the model.

In this thesis, I propose an extension of the model of Craig (2013). Recent advancements in the development of deep learning tools can now be applied to further the research that was done by Craig (2013), including the Python libraries of Keras and Tensorflow (François, 2015).

In this thesis, I merge the basis of Craig's model with the tools of Keras to provide a deeper and more succinct evaluation of the center-out wrist task by providing regularization and dropout constraints to the model, including

a supinated position into the model. I seek to remedy the problem of the model developed by Craig (2013), where only a few neurons experienced cosine tuning. By enforcing dropout in my model, I temporarily removed neurons from the recurrent neural network, along with all of the connections associated with those neurons. This forces other neurons to compensate for the dropped neurons, and build redundancies into the neural network. These redundant behaviors should provide more similar activation patterns between neurons across the network.

I demonstrate that the architecture is suitable to reproducing the force results seen in the monkey trials, with similar neural PD shifts to those found in the primary motor cortex of the Rhesus Macaque monkey in the trials of Oby et al. (2012). I also show that the neurons exhibit depth of modulation (DoM) changes, which were demonstrated by Shah et al. (2004) to be another way that neurons can encode complex movement.

Chapter 2

Prior Work

The central nervous system must perform a transformation on some abstract input in order to create the movement response associated with some goal. How this transformation is done is largely unknown. However, this task can be represented as a series of coordinate transformations between different representations: an extrinsic coordinate frame where movement is directed by location that some animal wants to inhabit, an intrinsic coordinate frame where movement is directed by the muscular anatomy of the animal, or an intermediate representation that is comprised of both.

Georgopoulos et al. (1982) discovered that a subset of the neurons in the primary motor cortex shows activation that varies as a function of the cosine of the angle between the current movement direction and a neuron's "preferred direction" (PD). The PD is the neuron's direction for which it is maximally active. The neuron activities were observed when a monkey performed a planar center-out reaching task to eight different target locations which were spaced

evenly around a central point. The activity of a neuron followed the form:

$$y = b_0 + c_1 \cos(\theta - \theta_0) \tag{2.1}$$

where the activity of the neuron y is represented by the PD θ_0, the depth of modulation (DoM) c_1, the baseline activation b_0, and the direction of movement θ. Georgopoulos et al. (1986) theorized that the extrinsic movements were encoded through this sub population of neurons. Each neuron makes a contribution to recruiting certain muscle groups, where the pulling directions strength and magnitude is calculated in relation to the neuron's baseline activity. A neuron's magnitude and direction were thought to be directly proportional to the contribution that neuron makes to the net extrinsic movement in a Cartesian space.

The belief that encodings of neurons are based purely on the extrinsic movement direction has since been challenged, as primary motor cortex (MI) neurons also modulate with limb posture (Scott and Kalaska, 1997), showing that a model based on the extrinsic movement direction is not sufficient for capturing the information encoded in MI neurons. Scott and Kalaska (1997) suggested that there is an encoding of intrinsic information taking place.

Kakei et al. (1999) expanded on these studies by having monkeys perform a center-out wrist task where a monkey would have to move its wrist to one of eight targets evenly spaced in a circle around the center starting position.The monkey's wrist movements directed a screen's cursor, and the monkey was required to move the cursor to a target location for three different forearm

postures. By performing this task in different postures, a clear distinction was made between the extrinsic (visual) movement direction, the movement of the wrist (flexion/extension, radial/ulnar deviation), and the intrinsic muscle activity. It was observed that a large proportion of neurons followed the cosine tuning that Georgopoulos et al. (1986) had modeled. However, it was also shown that the preferred direction of many neurons was dependent on the arm posture. Figure 2.1 demonstrates the change in orientation of preferred directions in each posture (C), with the muscle groups denoted by the bold lettering adjacent to the PD vectors. As the wrist rotates so do the PD vectors. The wrist postures are shown in (A), with the data in relation to that wrist posture represented as the column in which that wrist posture occurs.The average generated movement to the target location in (B), where the anatomical movement is represented by the lettering adjacent to the x and y axis. The anatomical movement is in reference to the wrist. As the wrist rotates, so does the anatomical movement. Kakei et al. (1999) examined the effect of an upwards extrinsic movement across postures. The wrist configurations in (D) show the rotation of the wrist in regards to the pronated position. Neurons exhibit a PD shift in accordance to wrist posture, however the shift in PD is about half of the rotation of the wrist.

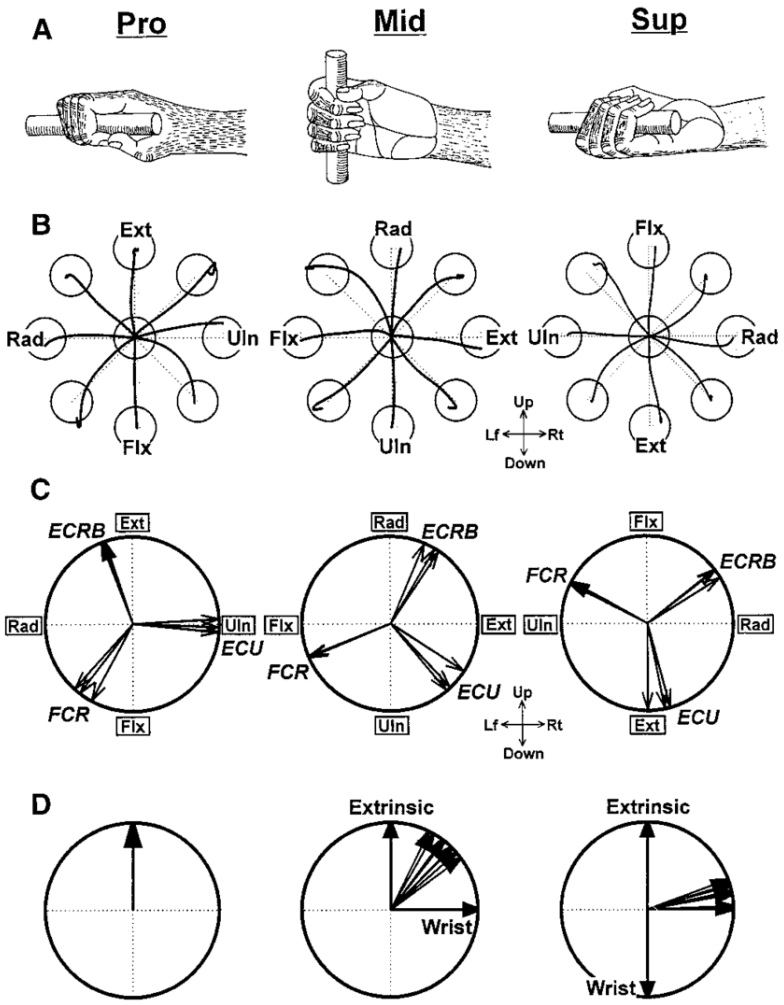

Figure 2.1: A) Schematic of monkey right hand gripping the handle of the manipulandum in three wrist postures. (Left) pronated; (center) midway between the two. (right) supinated; (B) Average movement trajectories to eight peripheral targets in three wrist postures with the associated anatomical movement:flexion (Flx), extension(Ext), radial (Rad) deviation, and ulnar (Uln) deviation. (C) Distribution of preferred directions (PDs) of three wrist muscles when the limb was in the three wrist postures, along with the corresponding anatomical movement labels of B. (D) Normalized shifts of PDs with wrist rotations from Pronated to Midrange and from Pronated to Supinated From Kakei et al. (1999), used with permission.

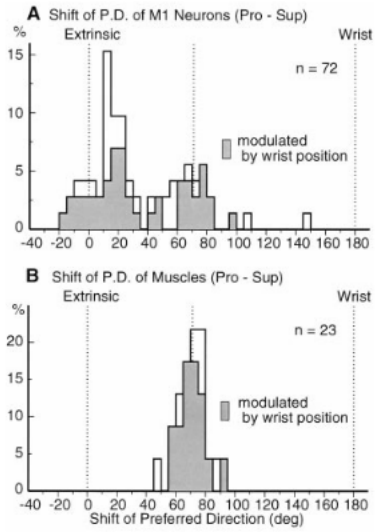

Figure 2.2: Distribution of PD shifts of motor neurons going from the Pronated to Supinated posture (A), Distribution of PD shifts of muscle going from Pronated to Supinated postures (B). From Kakei et al. (1999), used with permission.

Figure 2.2 shows the distribution of PD shifts from the pronated position to the supinated position. Kakei et al. (1999), observing the apparent bimodal distribution, split the neurons into two categories "extrinsic" neurons with no shift in PD, and neurons that behaved similarly to muscles. The muscle-like neurons' PDs showed a 60-80 degree shift in relation to the rotation of the wrist. The boundary between the muscle-like neurons population and the extrinsic neurons population was determined qualitatively. Kakei et al. (1999) proposed that the bimodal structure seen in the PD shift population was happening due to the need for the MI to transform the visual representation to a muscular one, where the two populations observed represented different stages of the process.

Another explanation for the distribution to be bimodal is presented by the

parallel processing scheme in Figure 1.1, in which both extrinsic and intrinsic populations exert control of the muscles directly (Shah et al., 2004; Dum and Strick, 2002). In order to show that such a system could work, Shah et al. (2004) developed a neural network model composed solely of extrinsic neurons. Due to all neurons being extrinsic, there was no PD shift between postures. However, depending on the posture, certain neurons experienced a larger or smaller fluctuation in activity over different movement directions.

The resulting model generated muscle activation patterns that were similar to those observed by Hoffman and Strick (1999), which used the same center out wrist task used by Kakei et al. (1999). The success of the model in producing muscle activations revealed that extrinsic neural encodings can directly drive muscles. The demonstration of purely extrinsic neurons driving muscles challenged the serial model developed by Kakei et al. (1999), as Kakei et al. (1999) described the transformation as being a bimodal distribution. The idea that there are different representations of neurons, each that can directly drive muscles supported the parallel processing scheme of Shah et al. (2004), shown in Figure 1.1

Oby et al. (2012) expanded on this research with an isometric wrist task. In these trials, a Rhesus Macaque monkey performed a center-out wrist task by applying forces to a manipulandum in two postures: midrange and pronated. Oby et al. (2012) applied the same procedure as Kakei et al. (1999), who observed the neural activity after the monkey had been trained for two months. The observations made in the primary cortex were through the use of a 10x10 microelectrode array.

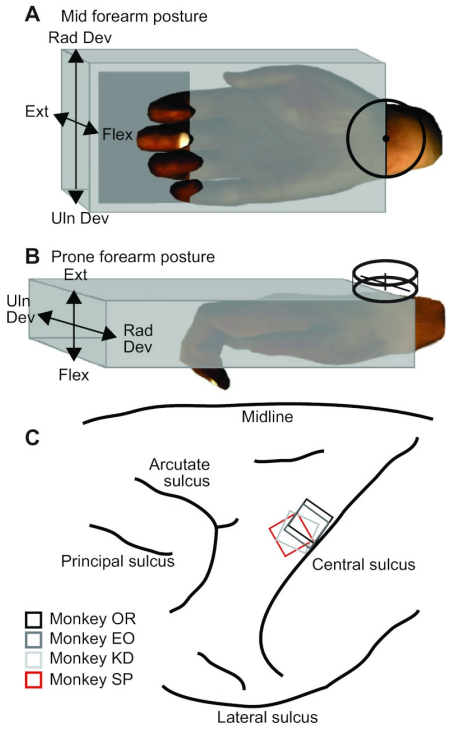

Figure 2.3: The set up of Oby et al. (2012), where a monkey has to provide wrist forces in the (B) pronated and (A) midrange postures. The microelectrode array placements within the hand area of the primary motor cortex (C) of the monkeys (OR, EO, KD and SP), used with permission.

To see the muscle activity throughout the trials, electrodes were also implanted into the wrist muscles. The resulting forces that were produced by the monkey were recorded with a load cell sensor. Cursor position was linearly related to the force applied to the load cell. The direction was visible to the monkey, who, when receiving a visual cue indicating target direction, was required to respond after waiting 0.5 seconds. The monkey was tasked with producing the associated force trajectory for moving the cursor to the given visual cue. It then had to hold the target position for 0.5 seconds and complete the task within five seconds for the trial to be deemed successful.

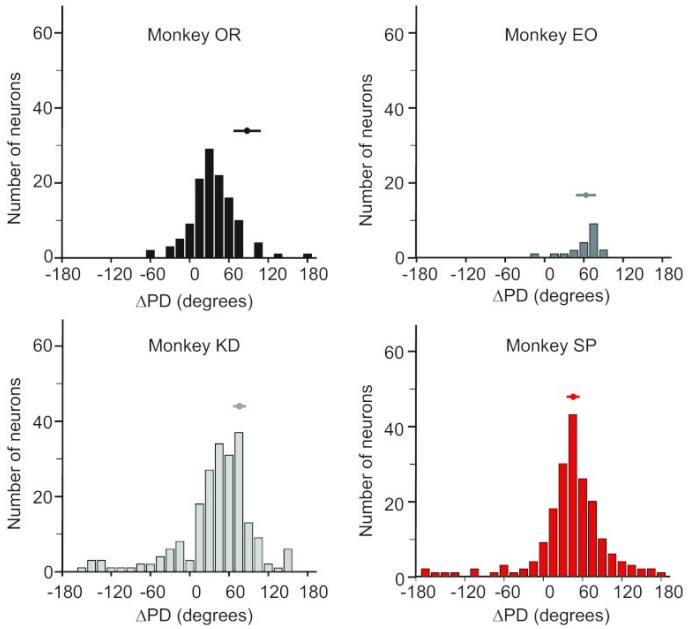

Figure 2.4: Resulting PD shifts of four different monkeys: OR, EO, KD, and SP. (Oby et al., 2012),used with permission

Figure 2.4 shows a unimodal distribution of approximately 50 degrees, which differs from the bimodal results recorded by Kakei et al. (1999). This mode is in between Kakei's subpopulations of extrinsic and muscle-like neurons. Oby et al. (2012) also made a linear model that could transform cortical MI neuron data, regardless of wrist posture, into intrinsic muscle activations. This transformation demonstrated that, given a posture-dependent model of muscle activations to hand forces, the model could generate the associated hand forces.

Due to the musculoskeletal changes with each wrist posture, no linear model exists from extrinsic, non- posture-dependent modulated neurons to muscle activation. MI neurons may have stronger associations with muscles than the corresponding extrinsic forces.

Craig (2013) expanded on the isometric wrist experiment by Oby et al. (2012), modeling the experiment with a recurrent network. The purpose of implementing a recurrent neural network is to investigate the types of neural activity that arise in the transformation between visual input and the corresponding wrist forces. Craig (2013) discovered that a neural network could be trained by a minimum jerk representation of force. The minimum jerk representation transitioned from the center resting position to the target location. The resulting neural activity of the network was derived from the optimization of the model's loss and not from a model constructed to conform to a biological process. The resulting neural network was successfully trained to perform the wrist task with the two postures. Many different model configurations can achieve similar performance metrics. After training multiple models, Craig (2013) evaluated the activity of the neurons found in the model. Craig (2013) found that the resulting neural activity of the RNN had similarities to those of the monkey. The optimization of the loss generated tuned neurons. The neurons of the model exhibited PD shifts and depth of modulation in accordance with wrist posture. The various tunings of neurons across the models reinforced the theory that the activity found in the primary motor cortex cannot be well defined on a neuron by neuron basis, but processes in the primary motor cortex can be evaluated by their aggregate behavior (Fetz, 1992).

Chapter 3

Approach

3.1 The Neural Network Model

Through the training of a recurrent neural network, the neural activity that arises from training is due to the optimization of force trajectories. By employing this approach, a further understanding of the question of how the nervous system implements the transformation from visual (extrinsic) input to muscle activity patterns is gained. This work is an extension of the work seen in Craig (2013), who showed similarities in the PD shifts to those observed by Oby et al. (2012). By employing techniques such as regularization and dropout, as well as extending the model to the supinated posture with more target positions and better training, I hope to help establish a more generalized model to implement the transformation from visual input to muscle activity patterns.

Since the procedure of an isometric wrist trial is of time varying nature,

the model is tasked with producing time varying behavior. To accomplish this, I provide the model with a visual indication of target location, a "go" signal, and an indication of which wrist posture is used. The output is comprised of no expected forces until the "go" signal occurs, at which point the expected forces are compared to the force transition of a maximal smoothness curve (Todorov and Jordan, 1998). The trajectory function of Todorov and Jordan (1998) is tasked with giving a force pathing from the center location to the target location within the time constraints of being between the start of the "go signal" and the start of where the positions are expected to be held. When the position is expected to be held, the force remains constant. The recurrent neural network is tasked with performing a suitable transformation. This transformation gets passed to a muscle activation layer that produces extrinsic forces. A representation of the data flow through the model is given by Figure 3.1.

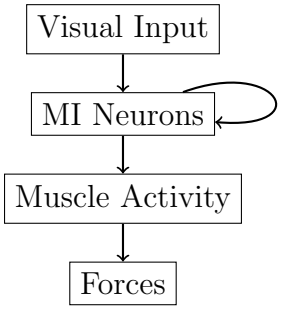

Figure 3.1: The neural network architecture of the model.Visual input is passed to a RNN, indicated by the recurrent loop of the MI neuron activity layer that is transferred to a muscle activity layer to help produce the resulting forces.

The transformation from visual data to extrinsic forces is highly posture dependent. A distinction of which posture is used is necessary for the input

layer. This is done through the use of a *one hot encoding*, or a discrete representation of the posture, where each entry in the row is assigned a binary value in accordance to whether or not that posture is active. The first entry is an indicator of the pronated position and the last entry of the vector is an indicator of the supinated position. The associations between posture and the one hot encoding are given in figure 3.1:

$$\rho = \begin{cases} [0,0,1] & \text{for supinated,} \\ [0,1,0] & \text{for midrange, and} \\ [1,0,0] & \text{for pronated,} \end{cases} \tag{3.1}$$

where ρ takes on one of these one hot encodings, and remains constant throughout the trial. By training one model over all three postures, generalized neural activity develops, having a subpopulation of neurons that are active for all three postures of the trial.

To perform the transformation of the neural activity over the time of a trial, the trial is decomposed into discrete time steps. For this model, a 5 second trial is broken into 100 time steps. The neural network is forced to compute a transformation from the input to the output at a sampling rate of 50 milliseconds. The target position is decomposed into a Cartesian form:

$$\omega_i = \begin{Bmatrix} x_i \\ y_i \end{Bmatrix}, \tag{3.2}$$

where the subscript i denotes the target location of the ith trial. The targets are on the unit circle, therefore, the decomposition of the target orientation q_i

to a Cartesian representation of (x_i, y_i) is as follows:

$$x_i = \cos(q_i), y_i = \sin(q_i). \tag{3.3}$$

To present visual data to the model, a conversion from a polar to a Cartesian coordinate system is also necessary for the preferred direction:

$$PD_j = [\cos(\phi_j), \sin(\phi_j)], \tag{3.4}$$

The variable ϕ_j represents the polar angle of the jth neuron. The resulting visual input is then represented by Gaussians of the form:

$$v_i = e^{-PD_j{}^T \omega_i}, \tag{3.5}$$

where θ_{PD_j} denotes the jth preferred direction of the visual cells and ω_i represents the ith target location. The result of a matrix multiplication in the exponent is a $\cos(\theta)$ where θ is the angle difference of the two vectors.

$$||\vec{a}||\ ||\vec{b}||\ \cos(\theta) = \vec{a}^T\ \vec{b} \tag{3.6}$$

The equation 3.6 is used to translate the function visual input equation 3.5 into the form $e^{\cos(\theta)}$: a visual input that is a Gaussian shape that is centered around the difference of the PD and the target orientation.

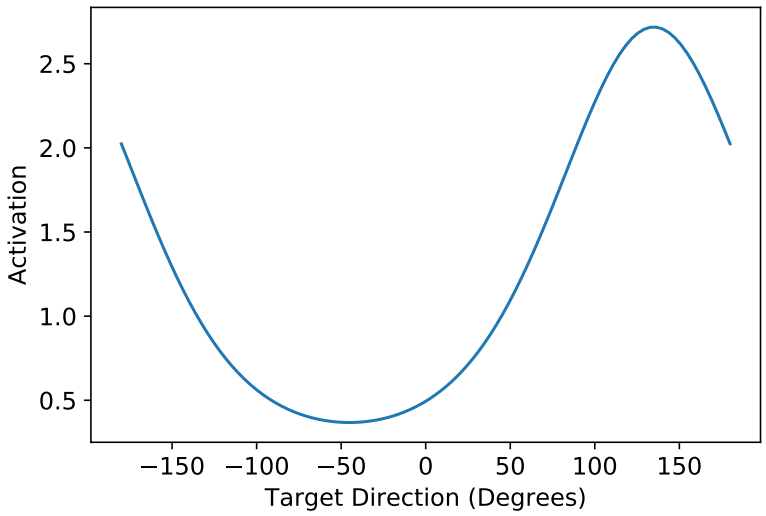

Figure 3.2: The model's visual input of target location

In order for this model to be robust, mimicking the results from Craig (2013), trial noise is introduced to the pure signal. Harris and Wolpert (1998) suggest that the noise in neural activation is proportional to the magnitude of the activation, I adopt this approach here:

$$v_i^{new} = v_i + \epsilon_i, \text{ where} \tag{3.7}$$

$$\epsilon_i \sim \mathcal{N}(0, \alpha v_i). \tag{3.8}$$

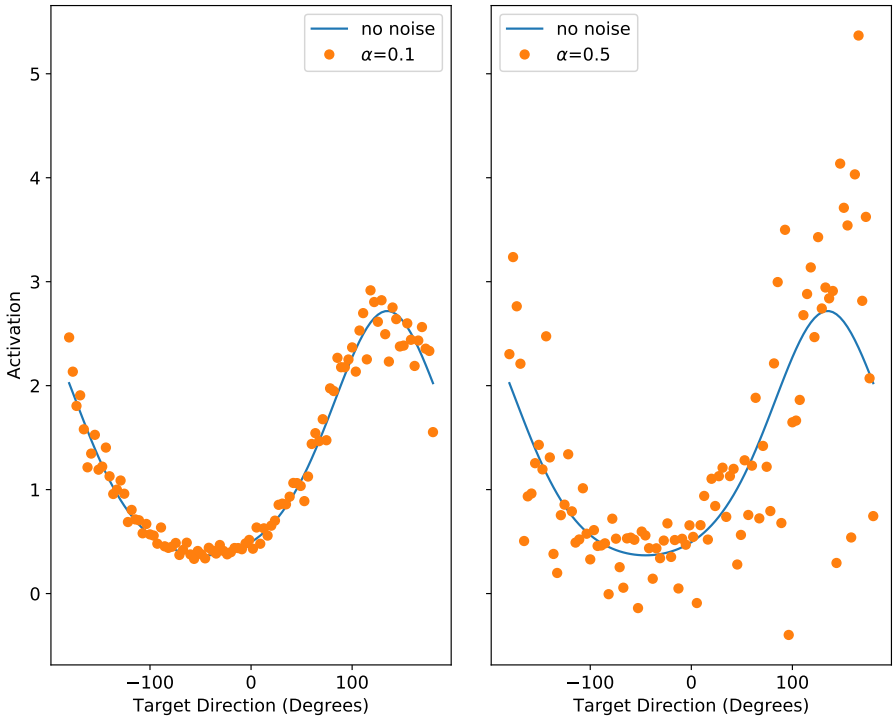

Figure 3.3: The amount of signal dependent noise produced by varying the α value.

The signal without noise is represented as the blue line in Figure 3.3. The orange data points represent the noise added to the signal. The target direction for training can be anywhere along the unit circle, so the domain of the function is from -180 degrees to 180 degrees. The indication of the onset of forces and muscle activity is known in literature as the *go signal*, and the timing of the associated activities will be denoted by a subscript *go* in reference to the signal. For the training of this model, the onset of the cue is between 1 to 2 seconds of the 5 second trial, leaving room for the planning phase. The cue is represented by a binary value at each timestep, indicating a truth value of whether forces

are supposed to be exerted.

$$g(t) = \begin{cases} 0 & t < T_{go}, \text{ and} \\ 1 & t \geq T_{go}. \end{cases} \tag{3.9}$$

The inputs of a force cue, a visual indication of the goal trajectory and an encoding of the associated wrist posture of the trial, are summarized in each timestep of the trial as an input into the model:

$$\mathbf{I}_t = \begin{bmatrix} v_1^{new} \\ v_2^{new} \\ \vdots \\ v_n^{new} \\ \rho \\ g(t) \end{bmatrix}, \tag{3.10}$$

where I_t represents the input layer at time t, v_n^{new} being the nth neuron's visual input of location with the associated noise.

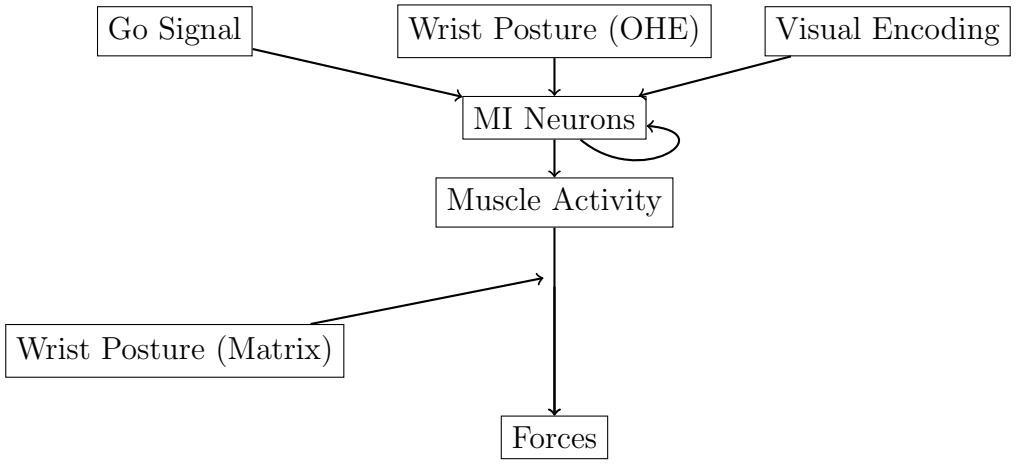

Figure 3.4: The data interactions that take place in the model.

The inputs in the model are passed to the RNN, which models the MI neurons in the brain. The RNN's output is the input to a layer representing the muscles of the model. The output of the muscle layer is a product of the muscle force orientations to provide the model's prediction of force.

For the model to be reliable, the numerical representation and stability of RNN's needs to be addressed. Due to the non-negative nature of neuron activations and a desire to be able to replicate activity behavior found in biological models, a constraint of having the activation function as non-negative is applied. A fully differentiable activation function is neccessary for gradient descent to work properly. A fully continuous activation function is also a necessity as the neurons in the model have to apply an activation function to the input values provided.

Having defined all of the operations needed in the model, the model dynamics of information are demonstrated with a forward pass of the model.

I denote the input at all timesteps as:

$$I_{T_{ALL}} = \begin{bmatrix} I_1 & I_2 & I_3 & \cdots & I_{T_{end}} \end{bmatrix}, \tag{3.11}$$

where T_{end} is the ending timestep of the trial. Each layer of the model has an associated weight for a value. The value is assigned based on the relative numerical importance to the neuron. The input to a neuron is multiplied by a neuron's weights, an association of the importance of the input in reference to that neuron. I denote the weight matrix of all of the neurons in the model as W, where the superscript 1 on the w and W represents the first layer of the weight element and matrix, respectively:

$$W^{(1)} = \begin{bmatrix} w^1_{11} & w^1_{12} & w^1_{13} & \cdots & w^1_{1H_{end}} \\ w^1_{21} & w^1_{22} & w^1_{23} & \cdots & w^1_{2H_{end}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ w^1_{N1} & w^1_{N2} & w^1_{N3} & \cdots & w^1_{NH_{end}} \end{bmatrix}, \tag{3.12}$$

This weight matrix is used an input to the Recurrent neural network. However, recurrent neural networks are known to have numerical stability issues in regards to their derivatives, with the gradients either vanishing or exploding (Pascanu et al., 2013), or stability issues in the forward pass.In order to remedy this, I created a custom activation function for the RNN layer.

3.2 Activation Function

My activation equation 3.13 is modeled after the **Exponential Linear Unit** or (**ELU**). However, due to the ELU's linear region, a neuron can activate itself in a unstable manner. To remedy this, a sublinear section is added to the right hand side of the activation function. In accordance with the naming convention of the ELU, the **Exponential Sub-Linear Unit** (**ESLU**) is defined as follows:

$$eslu(z) = \begin{cases} \frac{1}{\beta}(z^{\beta} - 1) + 2 & z > 1, \\ z + 1 & 1 \geq z \geq 0, \text{ and} \\ e^{z} & 0 < z, \end{cases} \tag{3.13}$$

where $0 < \beta < 1$. For numerical simplicity a value of 0.5 was used for β in the model.

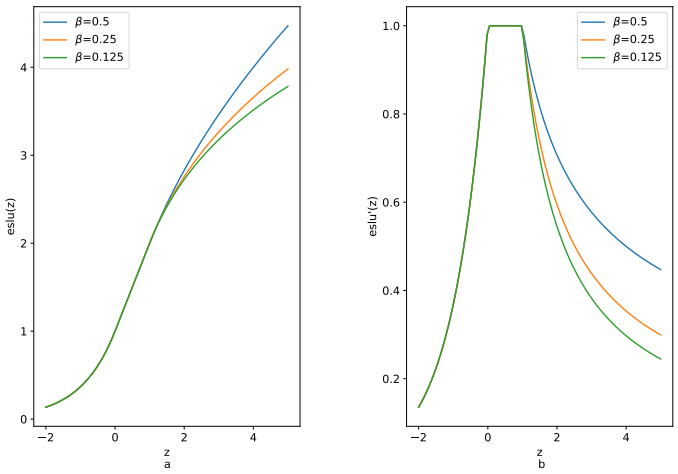

Figure 3.5: (a) The effect varying β values have on the ESLU's outputs.(b) Corresponding derivative of the activation function

This is a custom activation function and not included in the library provided by (François, 2015). This has to be built using the underlying Tensorflow architecture. N is the number of MI neurons in the RNN layer, and H_{end} is the last element of the input at a timestep. The resultant of this product is the input to the activation function, giving the neuron activity for the next layer:

$$A_t^{(2)} = eslu(W^{(1)} I_t + W^{(H)} A_{t-1}^{(2)}), \tag{3.14}$$

where $A_t^{(2)}$ and I_t are the activation and input at timestep t of the RNN layer, or second layer of the neural network. The neural network uses previous activation timesteps $A_{t-1}^{(2)}$ with a hidden weight matrix W^H to process the information changes across time. The output of the layer is saved to be used as input for the next timestep of the neural network, this is the recurrent

behavior of the RNN. The activation of the RNN is then multiplied by another weight layer to proceed down the neural network:

$$W^{(2)} = \begin{bmatrix} w^2_{11} & w^2_{12} & \cdots & w^2_{1H_{end}} \\ w^2_{21} & w^2_{22} & \cdots & w^2_{2H_{end}} \\ \vdots & \vdots & \vdots & \vdots \\ w^2_{61} & w^2_{62} & \cdots & w^2_{6H_{end}} \end{bmatrix}, \tag{3.15}$$

there are only 6 muscles in the model, so the model only has 6 neurons. For this layer, the activation function is an ELU rather than an ESLU. The muscle layer of the network is not a recurrent layer and does not have issues with the gradient not reaching an equilibrium.

$$A^{(3)}_t = elu(W^{(2)} A^{(2)}_t) \tag{3.16}$$

The recurrent neural network passes information to an ELU layer which is representative of the muscle activity. The angles at which the muscles in the wrist exert forces are provided by Hoffman and Strick (1999) and are accessed by the model's muscle activations to provide the resulting force:

$$\eta = \begin{cases} [89.4, 355.5, 310.7, 280.5, 263, 179.1] & \text{for pronated,} \\ [159.4, 17.9, 1.2, 341.4, 313.9, 197.8] & \text{for midrange, and} \\ [218.6, 69.6, 42.4, 8.1, 321.8, 254.3] & \text{for supinated.} \end{cases} \tag{3.17}$$

The degrees of each posture are translated into a Cartesian form using equation

3.3. The postures used in each trial are constant, so the values of η correspond with the one hot encoding of ρ.

$$\rho_{mat} = \begin{bmatrix} \cos(\eta) & \sin(\eta) \end{bmatrix} \tag{3.18}$$

The muscle orientations are written in a Cartesian coordinate system and are used as a product to transform the muscle activation $A^{(3)}$ into the predicted forces of the model, through the use of the posture ρ_{mat}.

$$[F_x, F_y]^T = \rho_{mat} A_t^{(3)} \tag{3.19}$$

At any time t the input I_t is provided to the recurrent neural network. For the first timestep, only I_t is used and the output of the RNN is saved. For each successive timestep, the preceding output is used as input to help determine the time based structure of the trial. The activation of the RNN is transferred via weights to the muscle activity layer. The output of the muscle activity layer, as a factor with wrist posture, provides the Cartesian force of the model at time t.

3.3 Force Trajectory

There must be an idealized force trajectory in order for error to be calculated in the model. The trajectory translation is done through the work of Todorov and Jordan (1998). The trajectory model of Todorov and Jordan (1998) is a hybrid of a minimum jerk trajectory model (Flash and Hogan, 1985) with a

smoothness constraint, where the optimal trajectory is defined as the endpoint velocity and acceleration having a value of 0. The purpose of using this model is to account for the bell shaped speed profiles in straight reaching movements while accounting for the inverse relationship between speed and curvature in extemporaneous drawing movements. The introduction of a smoothness maximization shows a uniformly better performance than the 2/3 power law that preceded it (Lacquanti et al., 1983; Todorov and Jordan, 1998). My model takes on the assumptions of Todorov and Jordan (1998), that an underlying motor strategy to produce smooth movements is an ideal movement representation. While the force trajectories drawn by these trials are two dimensional, an extension of the movement and training can be made.

The hybrid jerk trajectory is applied at the timing of the force cue to transition from the centered resting position to the desired target goal in a natural manner that adheres to a biological behavior in the time provided (Todorov and Jordan, 1998). The transition occurs with 0.5 seconds to spare of the 5 second trial in order to ensure that ending forces are held for half a second to replicate the confirmation of the monkey's success of the trial.

3.4 Loss

I hypothesize that penalizing the activation and weights of the model will force the model to provide activation patterns that are more realistic. The reasoning behind these penalizations is based on biological neurons which have a general metabolic energy constraint (Falkowska et al., 2015). In order to

place an energy cost on the model's active neurons, a norm penalty term is added. However, only regularizing the activity allows for a neural network to rely on large weights to provide the values needed for force, minimizing the activity while maximizing the weights. In order to avoid this complication, a penalty is also applied to the weights in the form of a kernel regularization. This calculation of loss is applied to both the RNN and the muscle activation layer. The model loss is a trade-off between force trajectory reconstruction error and these regularization terms. At each timestep, t, the loss is:

$$\begin{aligned} L_t = \frac{1}{2}\sum_{i=1}^{2}(h_{it} - p_{it})^2 + \lambda_1 \sum_{i=1}^{NH_{end}} ||w_i^{(1)}||_2^2 \\ +\lambda_2 \sum_{i=1}^{N} ||\hat{a}_t^{(2)}||_2^2 + \lambda_3 \sum_{i=1}^{6} ||w_i^{(2)}||_2^2 \\ +\lambda_4 \sum_{i=1}^{6N} ||\hat{a}_t^{(3)}||_2^2. \end{aligned} \tag{3.20}$$

The terms λ_1 and λ_2 are the regularization hyper-parameters used to control all of the MI's N neurons weights, denoted by $w^{(1)}$, and activation levels, denoted by $\hat{a}^{(2)}$. The terms λ_3 and λ_4, likewise, are the regularization hyper-parameters of the muscle layer. $\hat{a}^{(3)}$ and $w^{(2)}$ are the respective activations and weights in the muscle layer. The mean squared error term is the difference between the tth hybrid jerk value, which is represented as h_{it}, and the model's corresponding predicted value: p_{it} which is computed from equation 3.19. The Cartesian coordinate system requires two outputs of the model to be compared. All of the norms used in the loss function are L2 norms, as denoted by the subscript 2. Backpropogation is done at each timestep to fit the model.

3.5 Training Details

In brains, there is a redundancy hardwired into the biological network to ensure that movement will execute with certainty. Individual biological neurons are stochastic in their behavior, yet the biological performance is robust. In order to model this stochasticity, I introduce a drop rate to the neural network. The drop rate indicates the percentage of neurons that are excluded from the neural network. At each training epoch neurons are chosen randomly and the neurons dropped vary from one batch to the next. The dropout forces trained neurons to compensate for the lost activity of dropped neurons, creating redundancies in the encoding. The generalization produced by a drop enforces the neural network to have more active neurons. For this model, a drop rate of $p = 0.5$ is used. The reasoning for this parameter choice comes from 0.5 being a hyperparameter of a optimal training performance (Srivastava et al., 2014).

In this model, I implemented early termination to the training. This is to ensure that the model stops the training process once learning has ceased. To gauge when this should occur, I used the validation loss as a metric for whether or not more learning has occurred. I have set a patience value; if the loss hasn't reached a new minimum in a set amount of epochs I terminate the process. Every time a new minimum loss value is found, the model is saved, allowing for the checkpoint of the model's lowest loss to be used. For the purposes of this study, a patience value of 100 was used. In order to get a full representation of the neuron activity of the model, and for the neurons to be generalized, the model was trained off of target sampling drawn from a

uniform distribution in the range from -180 to 180 degrees.

The optimization of the model's loss during training was done with the Adam optimizer (Kingma and Ba, 2015). The Adam optimizer requires a step size, two exponential decay rates for moment estimates, a stochastic objective function, and an initial parameter vector. The step size, exponential decay rates, and initial parameter vectors are chosen by default in Keras. The stochastic objective function, or loss function, is shown in equation 3.20.

3.6 Solving for Preferred Direction and Depth of Modulation

The model's neuron activity levels are sampled across 16 equally spaced positions, along with the corresponding wrist forces. The filtering process is used to separate activation patterns that do not exhibit a cosine tuning. The model's neuron population is divided into cosine tuned neurons and neurons that do not exhibit cosine tuning.

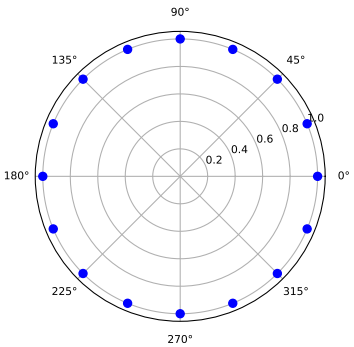

Figure 3.6: The evaluation points chosen for my model

To find a fit for each neurons cosine, the problem is reformulated as a regression problem:

$$U = \hat{\Psi}^T X + \epsilon, \tag{3.21}$$

where the neuron activity U is a function of the bias or average activity of the neuron ϵ, the Cartesian target location force X and the Cartesian PD of the neuron $\hat{\Psi}$. The cosine tuning function is then rewritten as:

$$U - \epsilon = \hat{\Psi}^T X, \tag{3.22}$$

$$\hat{\Psi} = (X^T X)^{-1} X^T (U - \epsilon) \tag{3.23}$$

$$\begin{aligned} \hat{\Psi} = & \left(\begin{bmatrix} F_{x_1} & F_{y_1} \\ F_{x_2} & F_{y_2} \\ \vdots & \vdots \\ F_{x_{16}} & F_{y_{16}} \end{bmatrix} \begin{bmatrix} F_{x_1} & F_{x_2} \cdots F_{x_{16}} \\ F_{y_1} & F_{y_2} \cdots F_{y_{16}} \end{bmatrix} \right)^{-1} \\ & \begin{bmatrix} F_{x_1} & F_{x_2} \cdots F_{x_{16}} \\ F_{y_1} & F_{y_2} \cdots F_{y_{16}} \end{bmatrix} \begin{bmatrix} a_1 & a_2 \cdots a_{16} \end{bmatrix} \end{aligned} \tag{3.24}$$

Georgopoulos et al. (1982) discovered that the activity of a sub-population of neurons followed the form of equation 2.1.The regression coefficients for fitting the cosine of a neuron be derived from the equations 3.25, 3.26 and 3.27.

$$b_0 = \frac{\sum_{i=1}^{R} a_i}{R}, \tag{3.25}$$

$$c_1 = ||\Psi||_2 \frac{\sum_{i=1}^{R} ||X_i||_2}{R}, \text{ and} \tag{3.26}$$

$$\theta_0 = \text{atan2}(\Psi_y, \Psi_x), \tag{3.27}$$

where R is a the number of subsamplings used, having the neural activity's sampling domain being based upon the degree of the target location. a is the activity of the neuron at that sample, X_i being the ith X-Force and Y-Force generated by the neural network, Ψ_x and Ψ_y are the respective x and y coordinates of the approximate PD and the first and second entries of Ψ, respectively. The L2 norm is denoted by the subscript 2 on the magnitude. The depth of modulation of the fit is denoted by c_1, with b_0 representing the average activity of the neuron and θ_0 representing the PD. The depth of modulation c_1 is a product of the magnitude of the PD with the mean magnitude of the force.

3.7 Model Evaluation

The purpose of this thesis is to show that with a simple architecture and appropriate regularization constraints, the model generates representations similar to those found in biology. The goal of this work is to measure the PD and DoM changes that happen throughout the model. Differentiating noise from the overall signals allows for a reliable representation of the changes that take place in the model. Comparing each neuron's activity to a cosine of best fit allows for a goodness of fit evaluation. The goodness of fit of a cosine clarifies whether or not the given neuron exhibits behavior similar to the activity

observed from Georgopoulos et al. (1986). Each neuron's goodness of fit is evaluated through the coefficient of determination. The coefficient of determination is used as a filter for deciding if a neuron is stable. In order to find the cosine of best fit from a data set, a methodology similar to (Stevenson et al., 2011) is used. The works of Stevenson et al. (2011) used a bootstrap process in which a subsampling of the data is used to generate a fit. In order for a neuron to be considered robust and tuned it must pass 3 filtering measures:

1. The coefficient of determination of a neuron's bootstrapped cosine fit must be greater than 0.2. For information on how this is computed, see section 3.6.

2. Cosine fitting preferred direction must not deviate by 25 degrees over the 150 bootstrapped samples of that neuron.

3. The cosine peak cannot be 1.5 times larger than the activity exhibited by the neuron.

4. The cosine peak has to be greater than 0.1

The cosine fitting has to be considered stable, so the deviation of the PD is restricted to 25 degrees to ensure that the representation of the cosine fit is accurate across the samplings. The peak being only 1.5 times greater comes from a limitation on the depth of modulation in relation to baseline activity. The cosine peak being greater than 0.1 comes from the basis that a cosine fit with little to no depth of modulation is a trivial case for a cosine fit.

Chapter 4

Results

The purpose of my work is to show that a basic RNN architecture with a regularization terms in the RNN and muscles is capable of producing stable cosine tuning behavior in neurons. An assessment of the loss of the model is necessary to determine whether or not my model has fit the basic constraints. The loss function is based on regularization and MSE, the model makes a compromise between the loss terms to prevent over fitting. The loss and validation loss are shown in Figure 4.1. Due to early stopping, the number of epochs trained varies between models.

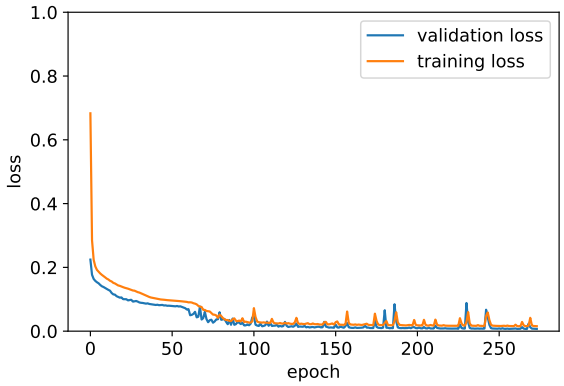

Figure 4.1: The mean training loss and validation loss of 20 runs

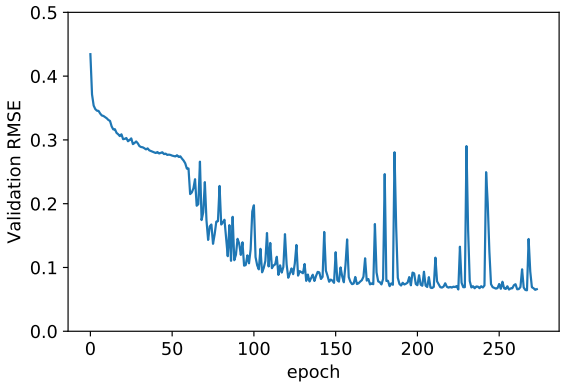

Figure 4.2: The mean validation RMSE of 20 runs.

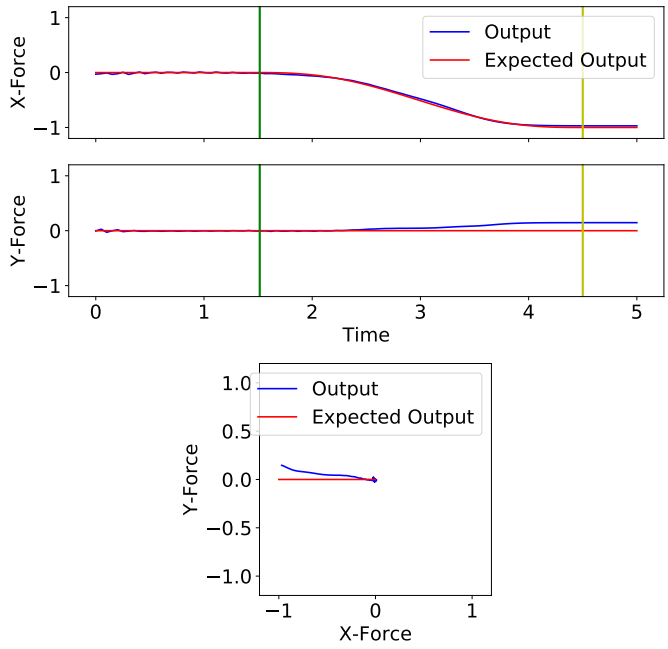

Figure 4.3: The model's attempts to match the given force path

The loss function contains regularization terms. A decrease in the loss can be attributed to minimizing regularization factors rather than the MSE error. While it can be seen that a fit is achieved from the model, the compromises made between regularization and the MSE loss metric is unknown. In order to remedy this, the average root mean squared error of the model is provided in Figure 4.2.

To visualize the translation from error to the corresponding wrist force task and how the error is distributed throughout the trial, an example of the wrist forces in a temporal manner is provided by Figure 4.3. The green line indicates the "go signal." The yellow line indicates the starting of the hold position in the trial. There is a slight oscillation of forces at the start of the trial. I hypothesize that this oscillation is due to the piece wise behavior of the

activation function of the RNN during backpropagation, where the derivative of the eslu is attempting to reach the linear portion of the activation function. Instead of directly reaching the linear portion, the activity is oscillating between values that have derivatives that are slightly below or above 1 and the activity is transformed into a small force.

A succinct metric is required to summarize the model's fit across all trials. Although the model doesn't make a distinction of prioritizing the MSE of the hold period, the evaluation of the success or failure of the wrist task is dependent on it. To provide a performance metric for the hold position portion of the trial, a RMSE is calculated for an intuitive description of how far the model strays from the provided force trajectory. The RMSE is calculated in reference to that portion of the trial:

$$RMSE = \sqrt{\frac{\left(\sum_{k=T_{hold}}^{T_{end}} \left(\vec{h}_k - \vec{p}\right)\right)^2}{T_{end} - T_{hold}}}, \tag{4.1}$$

where T_{hold} is the initial point in the trial, where the forces are expected to be held. T_{end} is the end of that duration, $\vec{h}_k$ is the force that the model is expected to match, and $\vec{p}$ is the model's constant predicted force. The equation 4.1 is with respect to a single trial.

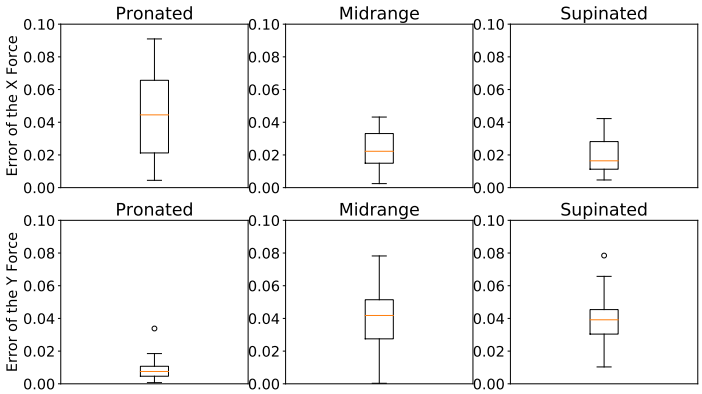

Figure 4.4: A comparison of the model's force errors across postures.

The distribution of error of the model in the hold position for 16 target locations is represented by the Figure 4.4. The target locations are uniformly distributed around the circumference of the circle. The target locations of the 16 trials are given by Figure 3.6.

4.1 Neuron Behavior

The neuron behavior observed in the model is in reference to the model's resulting force trajectory of a trial. Figures 4.5 and 4.7 show neuron activity as a function of target position and time. The Figures 4.9,4.7 and 4.9 below represent the MI neuron activity associated with a target location and time. There are only 16 evaluation points for target location, so the heatmap is broken into 16 rows. A constraint of the activation function is non-negativity. Thus, the heatmaps values are bounded to be non-negative. The strength of activity is represented by a shade of blue, where a stronger shade implies a

relative strength of activity in the neuron. The green and yellow lines are the respective start of the "go signal" and start of the hold period. For the model evaluation, the "go signal" is held at a constant time across trials, so it is a straight line. For the training and evaluation, the timing of hold position is held constant throughout trials, the forces held at the hold position are in reference to the target location of the trial. Figures 4.6, 4.8, and 4.10 show the corresponding cosine fits, a new cross sectional perspective is taken, where time is indicated by color, and MI activity is represented by data points. The cosine fits of MI activity are represented by the lines, and MI activity is only shown if there is an associated cosine fit. If the neuron's activity does not pass the stability requirements (see section 3.7) at a certain timestep, the MI activity and cosine fit at the timestep are not drawn. The angle at which the cosine fit peaks is the PD, represented by a red line on the heatmap. At any time, a neuron can experience an unstable fit. This is represented by a break in the red line and a missing portion of the color gradient in the cosine fits.

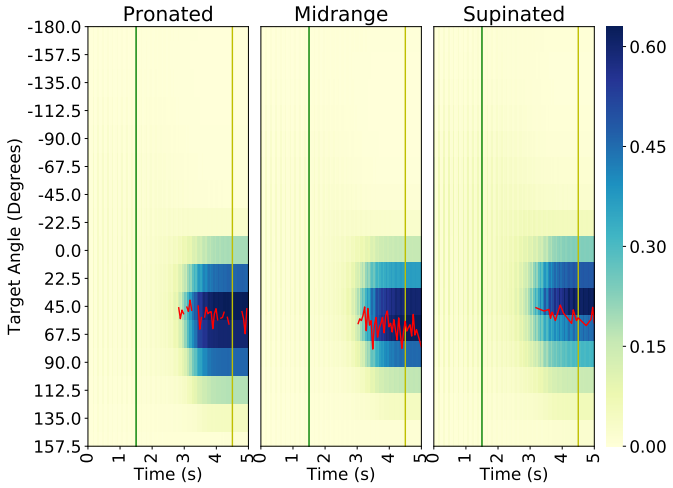

Figure 4.5: A neuron with a clear depth of modulation shift. The neuron activity is represented by the shades of blue, indicating the relative strength of the activation. The heatmap is divided among 16 trials that are evenly spaced along the unit circle. The red line marks the angle of the preferred direction of the cosine fits.The yellow line indicates the beginning of the hold period, and the green line indicates the onset of the go signal.

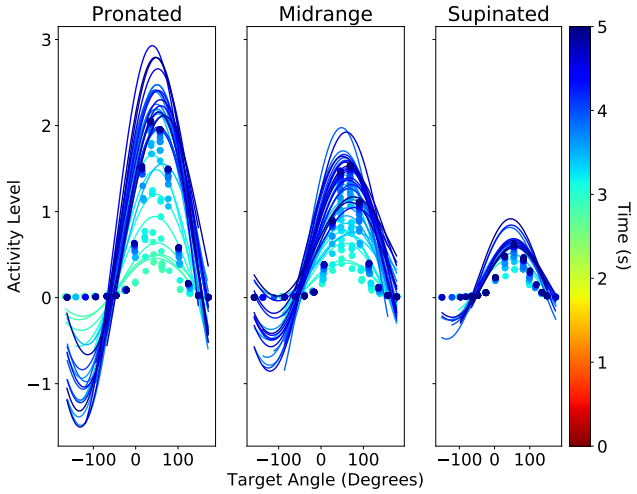

Figure 4.6: The associated cosine fits of the neuron activity.The neuron activity is represented by data points.The timing of the cosine fits is indicated by the colorbar. The model's neuron activity is generated from 16 trials with target locations equally spaced around the unit circle.

There are many different types of observed neurons from the population. The neuron shown in Figures 4.5 and 4.6 exhibits almost no activity during the start of the trial, as no blue shades are present in the heatmap. Then, at about 3 seconds, cosine fitting is present, represented by the red line and cosine fits being of a teal color. The cosine fit is be present throughout the remainder of the trial, with only a few timesteps showing a red line break, indicative of a missing cosine fit for that timestep. The depth of modulation shifts between postures, from a highly active pronated to a slightly active supinated position, shown by a the shades of blue in the heatmap getting progressively lighter as the depth of modulation of the cosine fit becomes smaller.

The neuron in Figures 4.7 and 4.8 exhibits a PD shift, in time as well as across postures. The PD is tracked by the red line. The PD of the pronated posture is slightly higher than that of the midrange and supinated. The amplitude of the cosine fit remains largely constant, while there is a noticeable phase shift from one posture to the next. The neurons of the model did not exhibit many PD shifts outside a 20 degree range.

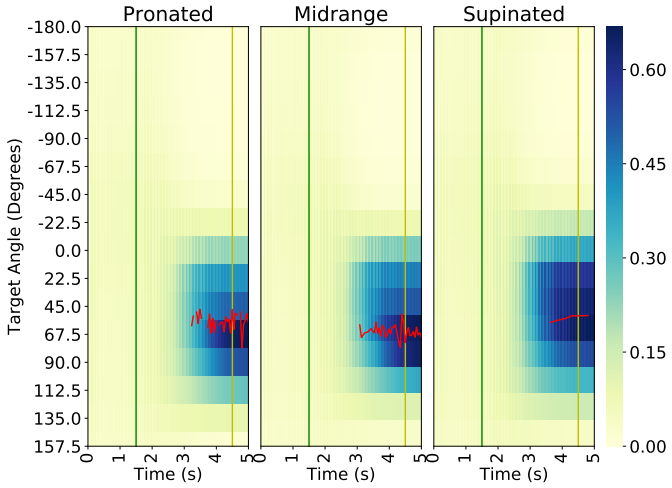

Figure 4.7: A neuron with a clear preferred direction shift.The neuron activity is represented by the shades of blue, indicating the relative strength of the activation. The heatmap is divided among 16 trials that are evenly spaced along the unit circle. The red line marks the angle of the preferred direction of the cosine fits. The yellow line indicates the beginning of the hold period, and the green line indicates the onset of the go signal.

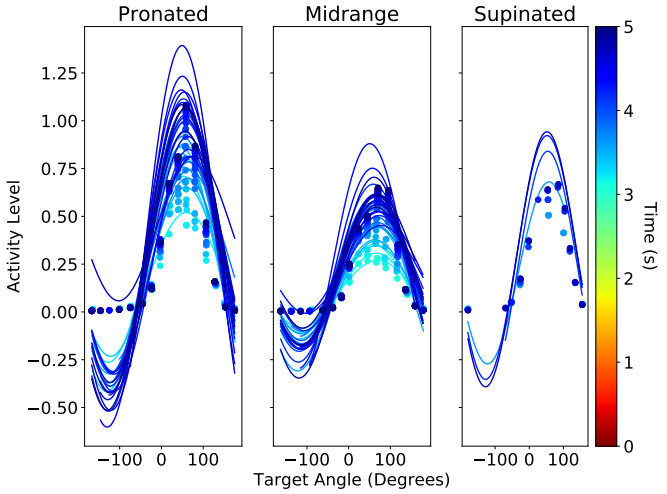

Figure 4.8: The associated cosine fit of the PD shift.The neuron activity is represented by data points.The timing of the cosine fits is indicated by the colorbar. The model's neuron activity is generated from 16 trials with target locations equally spaced around the unit circle.

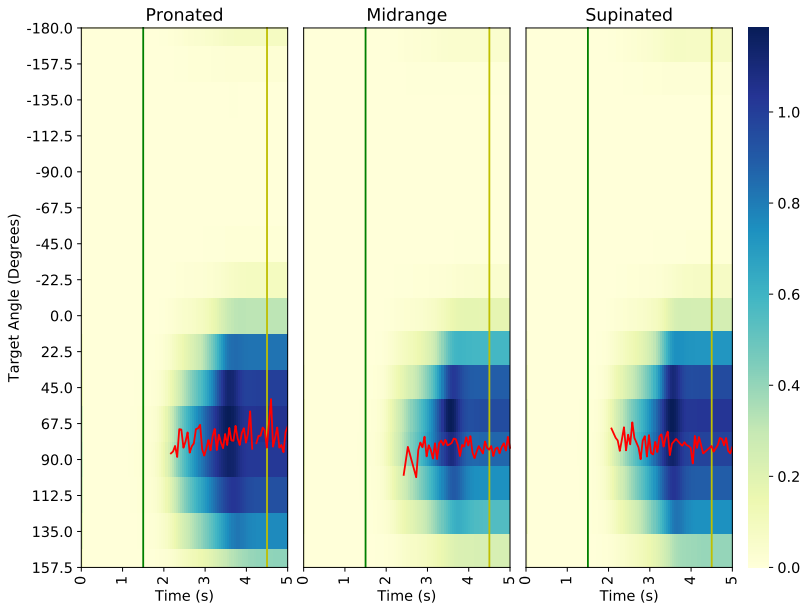

Figure 4.9: A neuron exhibiting depth of modulation and preferred direction changes.The neuron activity is represented by the shades of blue, indicating the relative strength of the activation. The heatmap is divided among 16 trials that are evenly spaced along the unit circle. The red line marks the angle of the preferred direction of the cosine fits.The yellow line indicates the beginning of the hold period, and the green line indicates the onset of the go signal.

The model's neurons tend to exhibit a depth of modulation (DoM) change more than a noticeable shift in preferred direction. However, the transformations that occur in this model are in a continuous domain, with models that perform PD shifts and DoM changes throughout the trial. The stable neurons of the models exhibit neither a fully extrinsic nor muscle-like behavior.

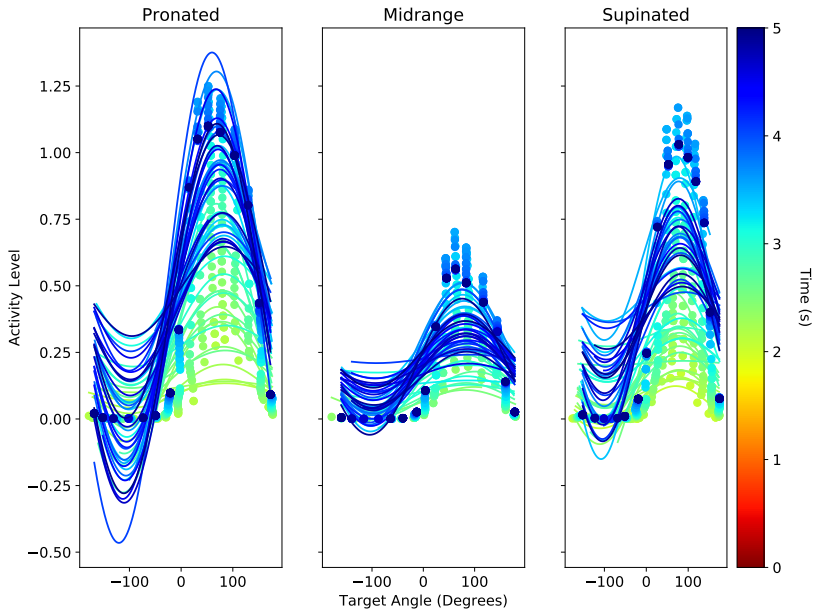

Figure 4.10: The resulting cosine fits of changes in depth of modulation and preferred direction.The neuron activity is represented by data points.The timing of the cosine fits is indicated by the colorbar. The model's neuron activity is generated from 16 trials with target locations equally spaced around the unit circle.

There are other neurons that exist that do not fall into any sort of cosine fitting, but are mainly time dependent.These neuron's are considered unstable, so cosine fits are not generated. However, they are prevalent throughout this model. While a higher regularization may enforce a more well defined shape, the cost of MSE was too great to consider justifying the expense.

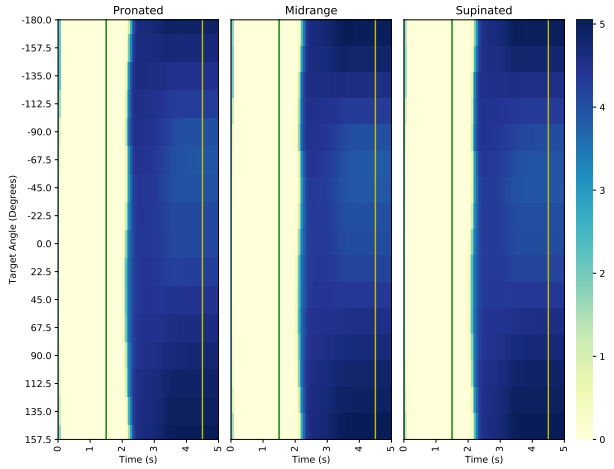

Figure 4.11: A neuron with largely a time based activation in the force generation period. The yellow line indicates the beginning of the hold period, and the green line indicates the onset of the go signal.

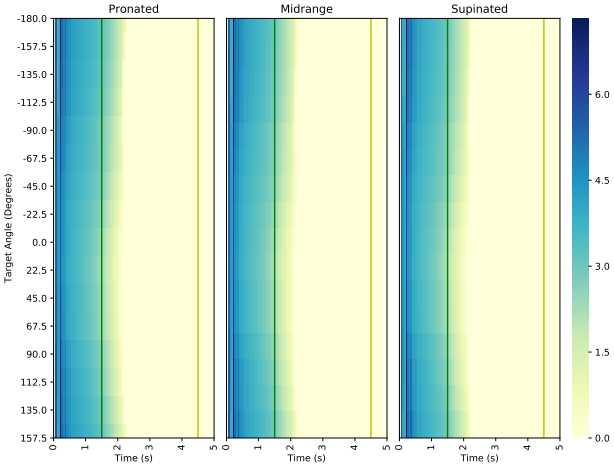

Figure 4.12: A neuron with largely a time based activation in the planning portion of the trial. The yellow line indicates the beginning of the hold period, and the green line indicates the onset of the go signal.

4.2 Total Neuron Behavior

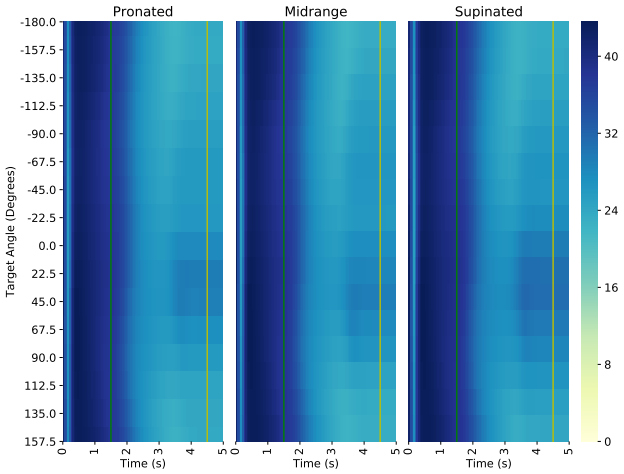

Figure 4.13: The net MI activity of the model in relation to each posture. The value of the net neuron activity is given by the shades of blue.The yellow line represents the start of the hold period and the green line represents the onset of the go signal.

All Neuron activity is summed in relation to each posture to generate the heatmap of the neural activity of the model. By examining the net RNN MI neural activity a general evaluation of information importance in regards to the MI neurons is presented by Figure 4.13. A non-posture dependent sum of activity is shown by Figure 4.14. The main activity of the neurons is in the planning phase, with the model showing a slight activity preference for a target in the 0-40 range in the hold period. A net DoM shift across postures is present for this PD.

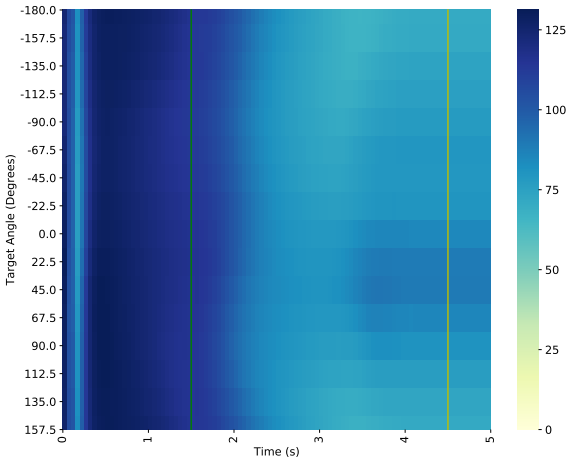

Figure 4.14: The total MI activity of the model across time.The value of the net neuron activity is given by the shades of blue.The yellow line represents the start of the hold period and the green line represents the onset of the go signal.

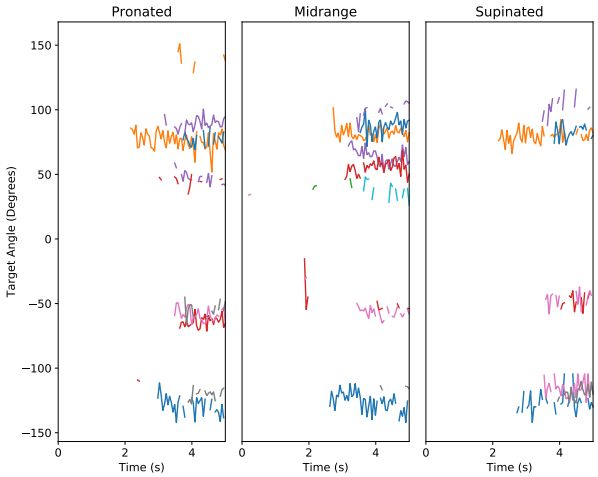

Figure 4.15: The PDs seen throughout all of the RNN neurons.

The PDs are mainly present after the "go signal," indicating that there is an association between preferred directions and muscle forces in the model.

4.3 Fraction of Variance Accounted For

In order to demonstrate that the results are not based on outliers, an evaluation of model stability is required. This takes the form of a Fraction of Variance Accounted For (FVAF) on a model level. The FVAF is evaluated with respect to the axis of Cartesian coordinate system, with a x-force and y-force FVAF.

$$FVAF = 1 - \frac{\sum_{i=1}^{16}(F_i - \hat{F}_i)^2}{\sum_{i=1}^{16}(F_i - \bar{F})^2}, \tag{4.2}$$

is comprised of F_i, the observed force of our model, $\hat{F}_i$, the predicted force, with $\bar{F}$ being the mean of the force values. This is equivalent to:

$$FVAF = 1 - \frac{SS_{err}}{SS_{tot}}, \tag{4.3}$$

meaning that we are comparing the sum of squares error to the sum of squares total. If there is a value close to 1, $SS_{err} \approx 0$. If there is a value less than one, $S_{err} > S_{tot}$, implying that there is more error than the mean prediction. A FVAF of 0 implies that the prediction of the mean is $SS_{err} = SS_{tot}$.

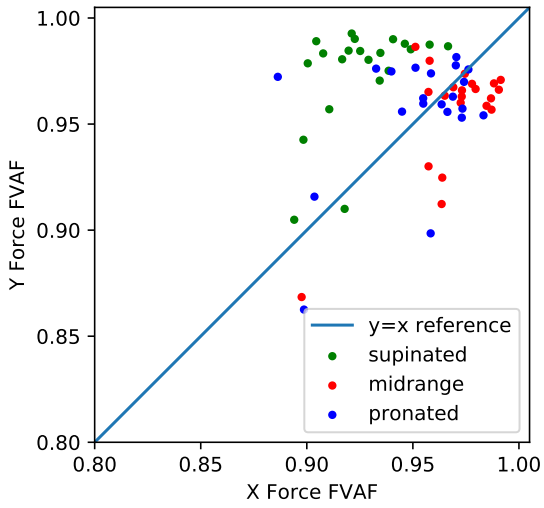

Figure 4.16: The Fraction of Variance Accounted For for each posture in multiple trials, with each color being representative of a different run's result. The mean PD shifts are calculated from an aggregate population of 20 runs. The line f(x)=x is shown for comparison.

The model's architecture predicts the distribution of the force fairly well, with trained models being in the 0.9 to 1 range, having the supinated postion's X force being the hardest to determine, with the mean being around 0.92 and the means of all other positions force being in the range from 0.95 to 0.98.

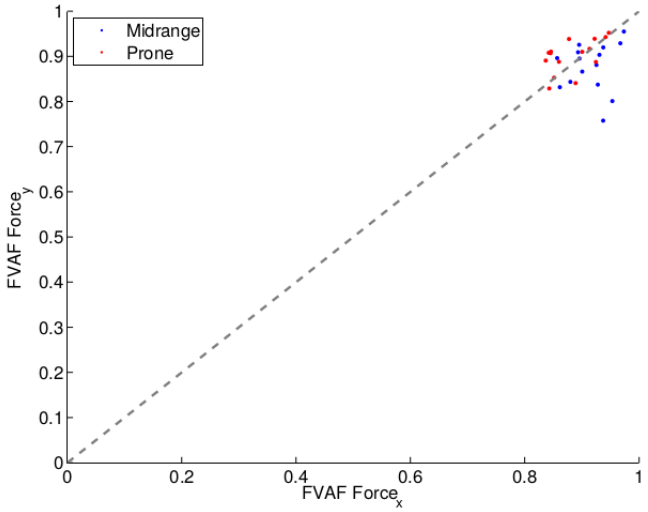

Figure 4.4: *Scatter plot of force FVAF for the X and Y dimensions. The indentity function $f(x) = x$ is shown for comparison.*

Figure 4.17: The FVAF of (Craig, 2013). The performance is consistent from model to model with all FVAF values greater than 0.8. The supinated wrist posture is not included in the model.

When incorporating the supinated position into the neural network, it appears that the FVAF of the pronated and midrange do not shift much to accommodate the supinated position, with the pronated being centered around the y=x axis and the midrange having it's data slightly skewed to the bottom right of the graph. This implies that the model treats optimizing the supinated position as an inferior problem, with the losses incurred on the midrange and pronated being too large to better optimize the supinated position.

To more fully evaluate the behavior seen in the model's stable neuron fits, an aggregate neuron behavior of 20 trials is built with cosine fit evaluations on the hold period of the trial. By aggregating for 20 trials, a stable neuron population is produced. The neural encoding of information can be described

by the depth of modulation and PD shift (Craig, 2013). In order to observe this, the stable neurons in Figure 4.18 are examined through these two metrics. It is from there that an analysis of individual hyperparameters takes place, showing the trade-offs that occur when choosing a particular model selection.

4.4 The Aggregate Preferred Direction and Depth of Modulation

In order to show figure legibility, the depth of modulation shifts are centered around the mean depth of modulation. The shifts are compared from the pronated to the other postures. P_{pro} is used to represent the pronated posture, with P_{pos} being the posture that is being compared. The 100 comes from the percent change from the mean.

$$\frac{100(P_{pos} - P_{pro})}{\frac{P_{pro}+P_{pos}}{2}} \tag{4.4}$$

The figures below are an aggregate of 20 runs in order show a robust representation of the type of behavior the population of stable neurons exhibit. The corresponding histograms are shown adjacent to the scatterplots with a bin size of 30 for each.

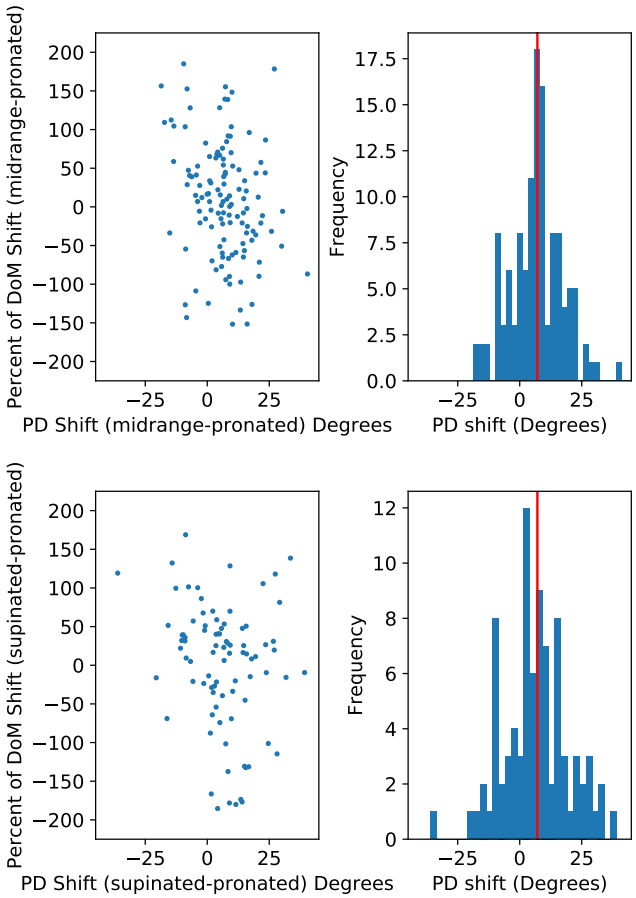

Figure 4.18: The depth of modulation and distribution of PD shifts across posture changes.The red line in the histogram indicates the mean PD shift of the populations

The model exhibits neurons with low PD shifts, as the unimodal distribution is centered around a 10 degree change for the pronated to supinated shift and the pronated to midrange shift. The low PD shifts that what were observed in the results section is a general trend of the data. According to a t-test there is no difference in PD shift between the midrange and supinated postures ($p < 0.6$) and there is no difference between depth of modulation shifts ($p < 0.4$). This choice differs from the observations of Oby et al. (2012)

and the modeling results of Craig (2013), who observed substantially more stable neurons in the hold position.

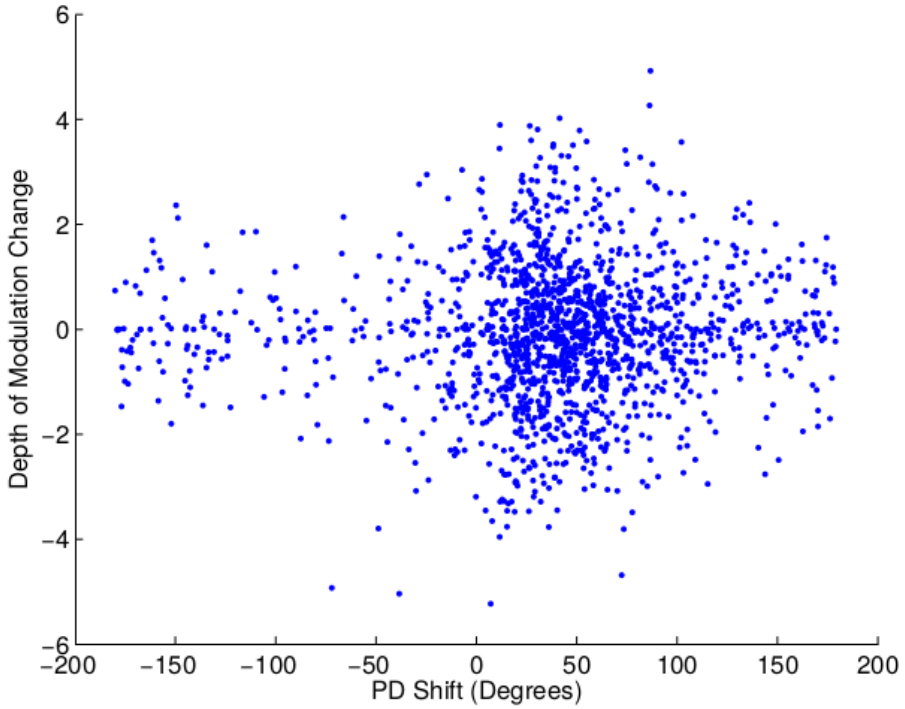

Figure 4.19: The stable neurons found in Craig's model

The model's neurons have low stability in comparison to those found in Craig's model. Craig (2013) discovered that regularization was a strong indicator of stable cosine fits. If that is true, then my model's lack of stable cosine fits is likely due to the activation values taking on more complex behaviors in order to perform the same tasks with less error. Kakei et al. (1999) observed a larger PD shift being present in the pronated to supinated than the pronated to midrange PD shift. This phenomenon is not seen in the stable neurons in my model, as the neurons have consistently been in the domain of +/- 20

degrees, (see Figure 4.18).

Similar to the model's MI neurons, the muscles in the model have a preferred direction and can be fit to cosine tuning curves. By analyzing a muscle neuron in the same framework of a MI neuron in the model, a clear comparison to the types of transformations they perform and representations of information are available. For Figure 4.20, the PD's of the neurons were computed during the hold position of the trial.

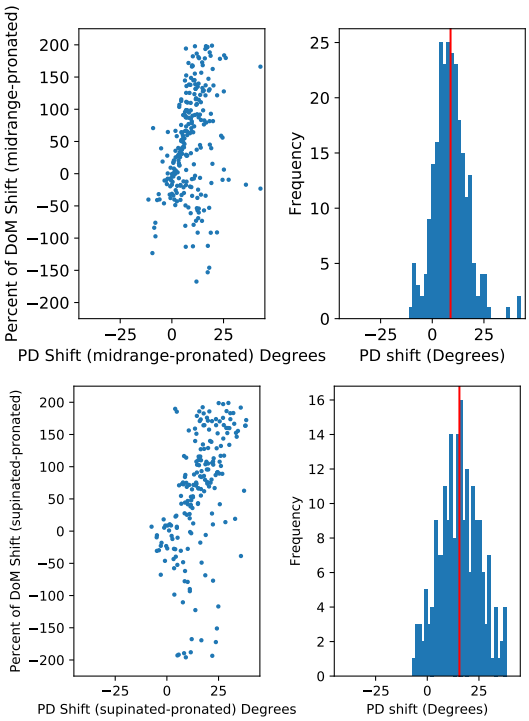

Figure 4.20: The PD shifts and DoM in the muscles. The red line in the histogram indicates the mean PD shift of the populations.

The model's muscles shift are a lot less pronounced than biological muscles. However, we have a clear unimodal distribution of the muscles that seems to shift slightly with wrist rotation. The shift of PDs being smaller than the

overall wrist rotation was observed by Oby et al. (2012). The bin size of 30 is used on both of the histograms in Figure 4.20. The model seems to have a linear association between the PD shift and DoM. According to a t-test there is a difference in the PD shift between the midrange and supinated postures ($p < 2 \times 10^{-14}$).

4.5 Neuron Analysis

A natural question to ask during modeling is whether the number of neurons in the RNN layer is sufficient for the problem. This question is largely empirical. However, by providing a metric of the performance of different models through the validation MSE, an expectation of how many neurons should be needed can emerge. The model initially underperforms by not having a sufficient number of neurons to fit the data trends, then has enough neurons and accurately predicts the force. Over fitting with too many neurons is not likely for the training set, since the training set is a discrete representation of target locations on the unit circle and not the unit circle itself. The model has preventative measures against over-fitting through the use of regularization. A visible case of over fitting is not present, with the model reaching an equilibrium.

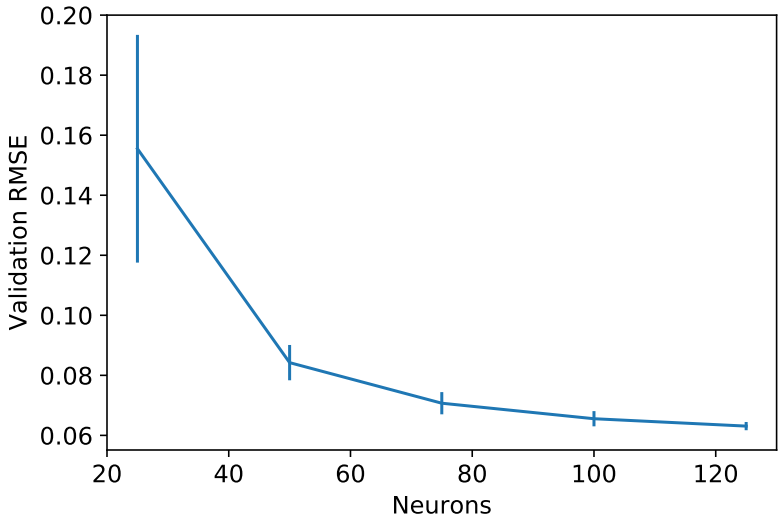

Figure 4.21: The validation RMSE of the RNN with varying neuron layer sizes.The vertical lines denote the standard deviation of the 20 trials with the mean given by the intersection of the line through this range.

To force more cosine tuned neurons, it is necessary to evaluate the activity regularization parameters in the model. This is due to the fact that penalizing neural activity has been shown by Craig (2013) to produce more cosine fitting. However, too strong of a regularization causes an under-fitting of the data. In order to show a more conclusive trend of the data and avoid outliers, the RMSE is averaged over 20 trained models. Through the use of a t-test I can conclude whether or not a statistically significant improvement of RMSE has occurred with an increase in the number of neurons. According to the t-test the distributions of validation RMSE between the 100 neuron model set and the 120 neuron model set are different ($p < 0.0006$). The difference between the 100 neuron model set and 120 neuron model set implies that the complexity of the information is not fully captured by an 100 neuron model. As the model

gains more neurons the statistical difference between two adjacent model sets narrows.

4.6 Activation Regularization Analysis

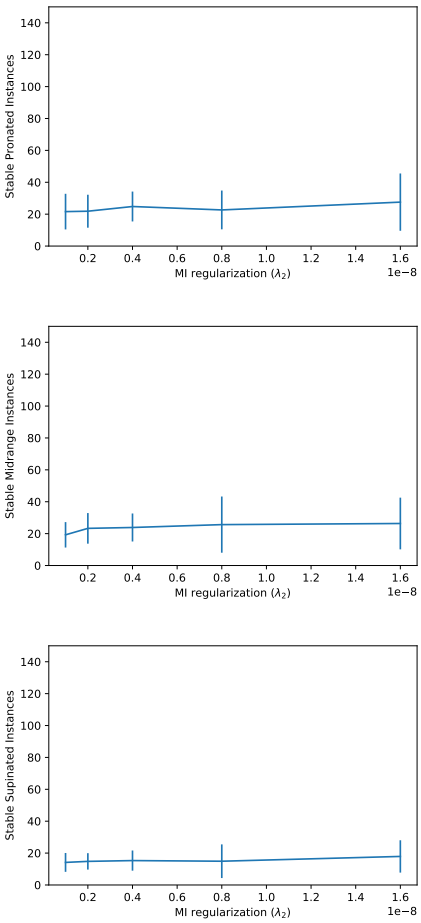

Figure 4.22: The mean number of stable neurons in the hold period with MI activity regularization. The standard deviation is given by the errorbars.

The supinated position consistently has fewer stable neurons than the midrange and pronated postures. According to a t-test there is no difference in the

number of stable neurons found in the pronated posture and supinated posture ($p < 0.83$). A t-test comparing the difference from the supinated posture other postures found that the pronated ($p < 0.048$) and midrange ($p < 0.06$) show small differences. I hypothesize that the reasoning behind this is that the neural associations made between the pronated and the supinated position are more similar to each other than the neural representation of the supinated position.

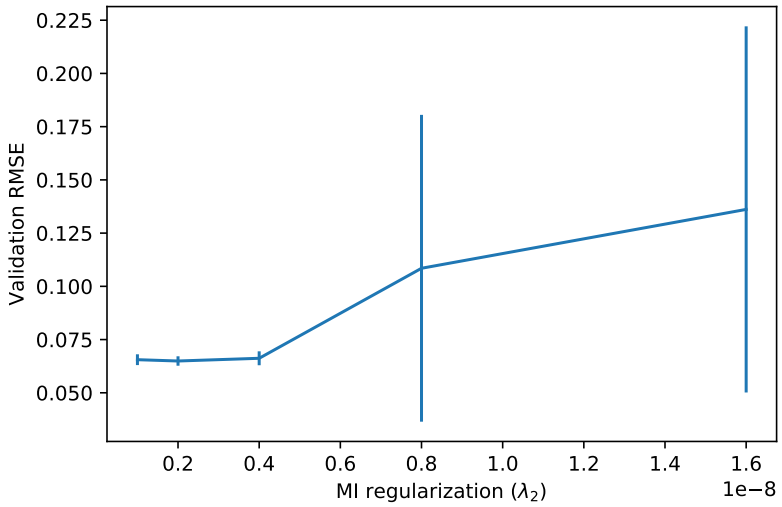

Figure 4.23: The mean RMSE of 20 runs with corresponding MI activity regularization.The standard deviation of RMSE is given by the errorbars.

As the MI regularization increases, the validation RMSE is also seen to increase. This implies that the regularization parameter is fairly sensitive change in this range.

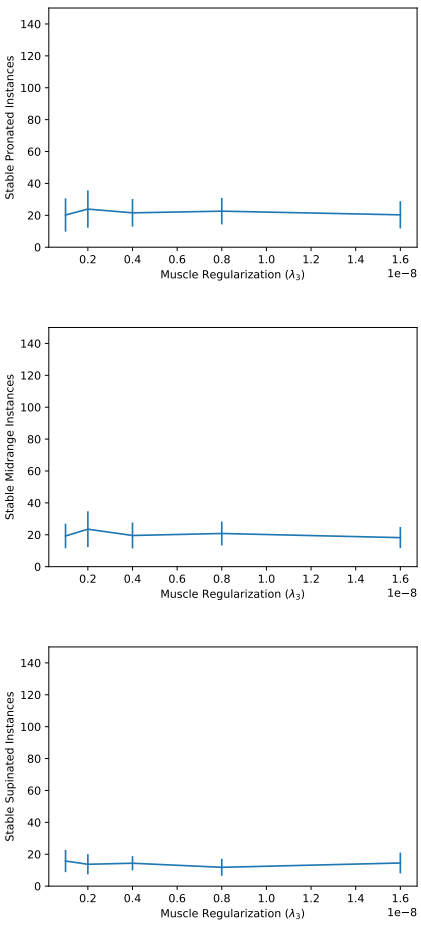

Figure 4.24: The mean number of stable neurons in the hold period with muscle activity regularization.The standard deviation is given by the errorbars.

According to a t-test there is no difference in the number of stable neurons found in the pronated posture and midrange posture ($p < 0.91$) across all λ_3 regularization values. A t-test for the differences between the supinated position to other postures show smaller p-values ($p < 0.15$), with the p-value fluctuating between each regularization value.

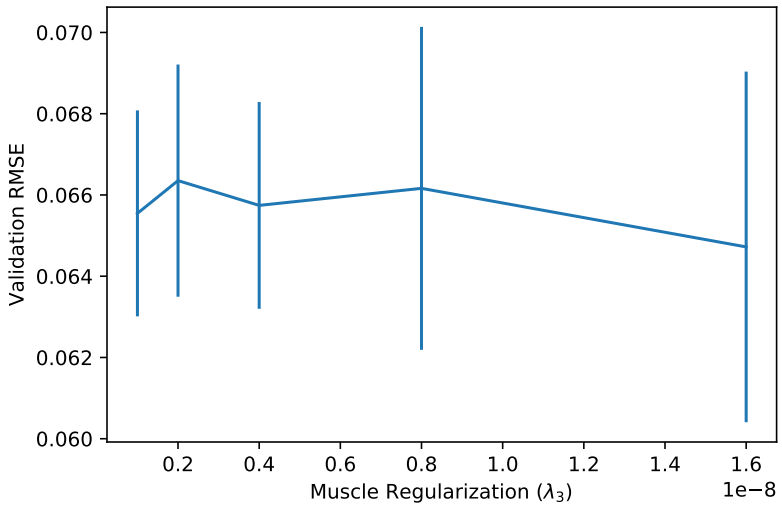

Figure 4.25: The mean RMSE of 20 runs with corresponding muscle activity regularization.The standard deviation of the RMSE is given by the errorbars.

The model has a lot more sensitivity in the MI layer than the muscle regularization layer. There is no statistically significant change that happens within these parameters of regularization. I hypothesize that the difference is mainly due to the complexity of the reduction of dimensionality occurs in the model.

In examining Figure 4.22, the model tends to have a stronger association with stable cosine fits in the midrange and pronated wrist postures

Since the model is based off of six muscles in the wrist and their orientations, the number of muscles used in the model cannot change without more information. This implies that the complexity of the model's transformation is currently limited to two layers, where one layer is fixed.

Without a decent kernel regularization in the muscle layer, the model produces neurons that have little to no modulation in time.

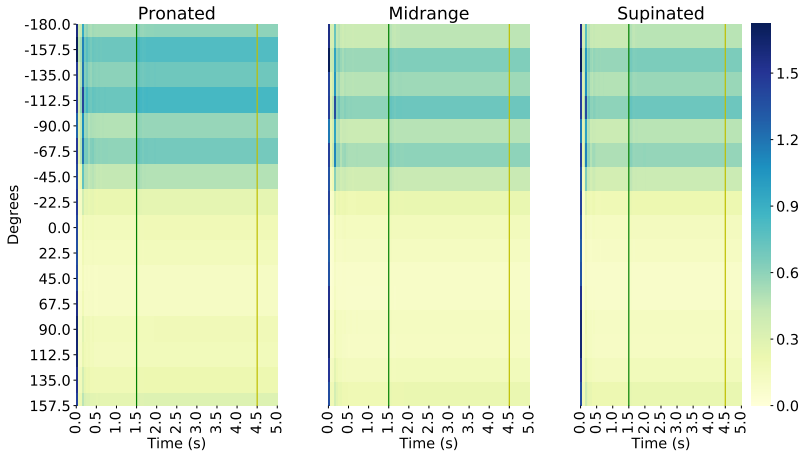

Figure 4.26: A neuron with only activation regularization.The activity of the neuron is indicated by the shades of blue. The yellow line represents the start of the hold period and the green line represents the onset of the go signal.

Chapter 5

Conclusion

In this thesis, a proposed new model using the Keras framework was analyzed, building off of the framework of Craig (2013). Inspired by the isometric hand force tasks of Oby et al. (2012), the model is able to perform a transformation from visual inputs to corresponding force output in a natural manner. The model performs this transformation with the use of enforced constraints, through regularization penalization and Adam optimization of the hybrid jerk force trajectory of Todorov and Jordan (1998). The generated neural activity was in response to the model fitting a transformation from visual input to force trajectory. The neural activity has tuning properties that are similar to those found in biology, with neurons exhibiting posture dependent PD shifts and depth of modulation changes.

Through analysis, I demonstrate that a subpopulation of neuron activity exhibits PD shifts. The regularization values are found to be crucial in changing the generated activity to be structured similarly to a neuron's cosine

tuning, with weight regularization being necessary for the model to respond in ways that are more predictable. This work shows that the model's transformation from visual input to the corresponding wrist forces does not need to be one that follows a well defined pattern on a neuron-by-neuron basis (Fetz, 1992). Instead, it is the aggregate effect of the neurons that generate the resulting force trajectory.

The PD's generated from the trials imply that there are certain areas in which the model's stable neurons reside. PD shifts from one posture to the next are substantially smaller than those seen in the MI of monkeys, with the majority being within a +/-20 degree range. Shah et al. (2004) demonstrated that neurons with no PD shift, but having posture dependent change in depth of modulation, are able to perform the transformation from neuron to muscle. The neuron activity in my model did not show a significant statistical difference when comparing the pronated to midrange PD shift and depth of modulation to the pronated to supinated PD shift and DoM. The model's MI activity was most present in the hold period of the model.

A key difference between the modeled task and that of Craig (2013) is that an instruction period is given. The target stimulus is presented and, after a delay, a movement is to be expected. The delay results in a sub-population of neurons that have largely uniform neuron activity with respect to target direction. The uniform activity is encountered in various time intervals of the trial, with other neurons being designated to produce muscle activity.

My model shows a few statistically significant distinctions. These distinctions and choices are yet to be explored. The observed cosine tuned neurons

of the muscle activity exhibited a linear trend, with an increase in PD shift of the neuron being related to an increase in DoM. A reasoning as to why the number of stable neurons in the supinated position is consistently significantly different than a midrange or pronated posture has yet to be explored.

The model is an abstract representation of the biological processes that take place in the CNS. Extending this model's framework to include other portions of the body could help demonstrate how other biological processes happen and explain complex interactions with fundamental tools.

My model relies on optimization of an RNN. Whether or not this neural network can be improved with a different neuron structure, such as incorporating a long short term memory configuration, has yet to be seen. Currently, the model's neurons operate with PD shifts that are rather minimal compared to the biological counterpart. A discovery, in either the biological processes of the primary motor cortex's transformation or in the machine learning community, can improve this model to being more applicable to medical needs.

The model takes on the assumptions of a correct trajectory being one of maximal smoothness (Todorov and Jordan, 1998), with prediction error increasing with duration of a trial. My model should see similar effects in the prediction error.

Chapter 6

Appendix

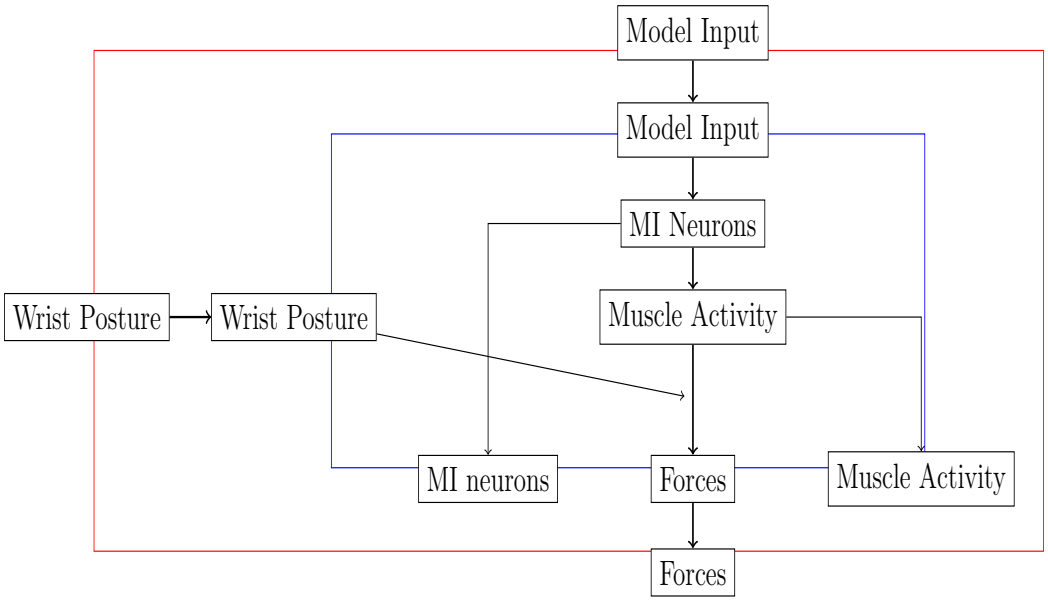

Figure 6.1: The Keras implementation of the model, with the blue rectangle representing the inner model and the red rectangle representing the outer model. The boxes represent the information that is contained inside each model. The variables on the perimeter of the red and blue boxes indicate the flow of information into and out of the model.

The outer model consists of the inputs of the wrist posture as a matrix (see equation 3.18) and the input (see equation 3.10). The computations are done in the inner model, where the MI neurons and muscle activity are located. The MI neurons are represented by an recurrent neural network. The nesting of models is done due to the fact that back-propagation happens over the output of the model. The constraint of MSE is in regards to force. Nesting a model inside another model allows for the back-propagation to train the inner layers of the model. An inner model allows for the viewing and analysis of the trained layers. Through the passing of inputs from the outer model to the inner model the model's functionality and architecture is preserved.

Bibliography

R. Ajemian, D. Bullock, and S. Grossberg. A model of movement coordinates in the motor cortex: Posture- dependent changes in the gain and direction of single cell tuning curves. *Cerebral Cortex*, 11:1124–1135, 2001.

M. A. Craig. Predicting wrist forces and muscle electromyograms from cortical data. Master's thesis, University of Oklahoma, 2013. School of Computer Science.

R. P. Dum and P. Strick. Motor areas in the frontal lobe of the primate. *Physiology & Behavior*, 77:677–682, 2002.

A. Falkowska, I. Gutowska, M. Goschorska, P. Nowacki, D. Chlubek, and I. Baranowska-Bosiacka. Energy metabolism of the brain, including the cooperation between astrocytes and neurons, especially in the context of glycogen metabolism. *Int J Mol Sci.*, 16(11):25959–25981, 2015.

E. E. Fetz. Are movement parameters recognizably coded in the activity of single neurons? *Behavioral and Brain Sciences*, 15(4):679–690, 1992.

T. Flash and N. Hogan. The coordination of arm movements: an experimen-

tally confirmed mathematical model. *Journal of Neuroscience*, 5:1688–1703, 1985.

C. François. Keras, github repository, 2015. URL `https://github.com/fchollet/keras`. Accessed March 18, 2019.

A. P. Georgopoulos, J. F. Kalaska, R. Caminiti, and J. T. Massey. the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *Journal of Neuroscience*, 2:1527–1537, 1982.

A. P. Georgopoulos, A. B. Schwartz, and R. E. Kettner. Neuronal population coding of movement direction. *Science*, 233:1416–1419, 1986.

C. Harris and D. Wolpert. Signal-dependent noise determines motor planning. *Nature*, 394:780–784, 1998.

D. Hoffman and P. Strick. Step-tracking movements of the wrist. iv. muscle activity associated with movements in different directions. *Journal of Neurophysiology*, 81(1):319–331, 1999.

S. Kakei, D. Hoffman, and P. L. Strick. Muscle and movement representations in the primary motor cortex. *Science*, 285(5436):2136–2139, 1999.

J. Kalaska, D.A.D, Cohen, Hyde, M.L., and M. Prud'Homme. A comparison of movement movement direction-related versus load direction-related activity in primate motor cortex, using a two-dimensional reaching task. *The Journal of Neuroscience*, 9:2080–2102, 1989.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

F. Lacquanti, C. Terzuolo, and P. Viviani. The law relating the kinematic and figural aspects of drawing movements. *Acta Psychologica*, 54:115–130, 1983.

H. G. Moorman, S. Gowda, and J. M. Carmena. Control of redundant kinematic degrees of freedom in a closed-loop brain-machine interface. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(6):750–760, 2017.

F. Mussa-Ivaldi. Do neurons in the motor cortex encode movement direction? an alternative hypothesis. *Neuroscience Letters*, 91(1):106–111, 1988.

E. Oby, C. Ethier, and L. Miller. Movement representation in primary motor cortex and its contribution to generalize emg predictions. *Journal of Neurophysiology*, 109:666–678, 2012.

R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *ICML*, 2013.

S. H. Scott and J. F. Kalaska. Reaching movements with similar hand paths but different arm orientations. *Journal of Physiology*, 77(2):826–852, 1997.

A. Shah, A. H. Fagg, and A. Barto. Cortical involvement in the recruitment of wrist muscles. *Journal of Neurophysiology*, 91(6):2445–2456, 2004.

N. Srivastava, G. Hinton, A. Krizhevsky, and I. Sutskever. Dropout: A sim-

ple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

I. Stevenson, A. Cherian, B. London, M. Sachs, N. A. Lindberg, E. Reimer, J. Slutzky, M. Hatsopoulos, N. Miller, and K. Kording. Statistical assessment of the stability of neural movement representations. *Journal of Neurophysiology*, 106:764–774, 2011.

E. Todorov and M. Jordan. Smoothness maximization along a predefined path accurately predicts the speed of profiles of complex arm movements. *Journal of Neurophysiology*, 80(2):696–714, 1998.