

UCSF

UC San Francisco Electronic Theses and Dissertations

Title

Spectrome-AI: a Neural Network Framework for Inferring MEG Spectra

Permalink

<https://escholarship.org/uc/item/59g5h36z>

Author

Zhou, Jiamin

Publication Date

2019

Peer reviewed|Thesis/dissertation

Spectrome-AI: a Neural Network Framework for Inferring MEG Spectra

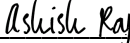
by
Jiamin Zhou

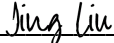
THESIS
Submitted in partial satisfaction of the requirements for degree of
MASTER OF SCIENCE

in
Biomedical Imaging

in the
GRADUATE DIVISION
of the
UNIVERSITY OF CALIFORNIA, SAN FRANCISCO


Approved:

DocuSigned by:

13BDA73A066E4E0... Ashish Raj
Chair

DocuSigned by:

DocuSigned by:
947C... Jing Liu

DocuSigned by:

DocuSigned by:
41E... Peder Larson

DocuSigned by:

F1E4E52A4E3D4D8... Janine Lupo

Committee Members

Acknowledgments

I would like to thank Dr. Ashish Raj for his expertise and developing the spectral graph model, Dr. Pablo Damasceno for simulating the data necessary for this work and his guidance throughout this entire project, and Xihe Xie for his work developing the graph model further. I would also like to thank my committee members Dr. Jing Liu, Dr. Peder Larson, and Dr. Janine Lupo for their support and advice. Special thanks to Dr. Jing Liu for being my main advisor initially when I was first starting to experiment with applying machine learning to biomedical imaging on a separate project.

Lastly, I'd like to thank my all of my classmates, who've been amazing people to learn from in both academic and social settings, and all the MSBI administrators and instructors for teaching and supporting all of us!

This paper was typeset in L^AT_EX.

Spectrome-AI: a Neural Network Framework for Inferring MEG Spectra

Jiamin Zhou

Abstract

Computational modeling is a tool that allows for biological systems involving large networks to be studied, such as in studying the correlations between structural connectivity and functional connectivity in the human brain. Raj et al.¹ proposed the spectral graph model in 2019 as a linear, low-dimensional alternative to conventional neural field and mass models that are more computationally expensive, especially when optimizing parameters, which is necessary in order to obtain quantitative and qualitative information about functional neural activity. The initial method used for inferring the spectral graph model parameters was Markov chain Monte Carlo (MCMC) sampling, which provided a robust way to estimate what the target parameter distributions were most likely to be. However, MCMC methods are still slow and computationally expensive. In this study, we trained a fully connected neural network on MCMC-simulated magnetoencephalography (MEG) data to perform parameter estimation for the spectral graph model in an accelerated manner. We found that the neural network was able to predict most parameters of interest without much loss in precision while generating the parameters in less than a second. This approach puts us closer to obtaining real time neurophysiological information from functional neuroimaging data for applications in diagnosis, prognosis, and characterization of various neurological diseases.

Keywords: *Deep learning, Computational modeling, Parameter estimation, Magnetoencephalography*

Contents

	Page
1 Introduction	1
1.1 <i>Modeling Brain Structure and Function</i>	1
1.2 <i>Global Parameter Inference</i>	2
1.3 <i>Neural Networks</i>	3
2 Methods	4
2.1 <i>Experimental Dataset Acquisition and Processing</i>	4
2.2 <i>Spectral Graph Model</i>	5
2.3 <i>Simulated Dataset Acquisition with MCMC</i>	5
2.4 <i>Fully Connected Neural Network</i>	6
2.5 <i>Performance Analysis</i>	8
3 Results	9
3.1 <i>MCMC Parameter Distributions</i>	9
3.2 <i>Deep Learning Model Performance</i>	10
3.3 <i>Applying the Generative Model</i>	11
4 Discussion	13
4.1 <i>Limitations</i>	13
4.2 <i>Future Steps</i>	14
References	15
Appendix	19

List of Figures

	Page
1.1 Spatial distributions of alpha band activity	2
2.1 Proposed neural network architecture	8
3.1 MCMC parameter distributions	10
3.2 Predicted vs true parameter values	11
3.3 Spectra generated from the forward model using default parameters and predicted parameters from the neural network	12

List of Tables

	Page
3.1 Regression statistics for the neural network predicted parameters	11
3.2 Default and predicted model parameter values	12

1 Introduction

1.1 *Modeling Brain Structure and Function*

Relating the brain's structural connectivity (SC) to its functional connectivity (FC) is a fundamental goal in neuroscience because it is capable of aiding our understanding of how the relatively fixed SC architecture underlies human cognition and diverse behaviors. The human brain consists of around 100 billion neurons,² which all dynamically interact in large-scale networks, such as networks of vision, motion, memory, and attention, and is known as the brain connectome.³ Recent studies have provided direct evidence through multiple methodologies that the patterns of SC and FC in the brain are correlated.⁴⁻⁷ Additionally, SC-FC coupling is not constant but rather exhibits significant changes during normal development⁸ and brain abnormalities,^{4,9} suggesting that understanding the underlying disruptions of SC or FC could shed light on the emergence and progression of cognitive dysfunction in neurological and neuropsychological diseases like autism, epilepsy, schizophrenia, and dementia.

Conventional computational models of functional neural activity involve numerical simulations of neural field¹⁰ or neural mass^{11,12} models that are coupled by anatomical connectivity, resulting in a large connected systems of ODEs describing functional networks due to local dynamics. The results of solving ODEs can help us make predictions about what will happen in the real system that is being modeled in response to perturbations in the system. However, these non-linear, stochastic simulations are very computationally demanding, and thousands of simulations are needed for an accurate description of functional activity patterns. Additionally, these models tend to have high dimensionality with hundreds of local parameters needed to describe regional neuronal dynamics. The spectral graph model proposed by Raj et al.¹ provides a linear mathematical model that is analytical and thus solvable given a set of parameters, not requiring stochastic simulations. The model is also low-dimensional, requiring only seven parameters that apply to brain dynamics globally and are also biophysiologicaly meaningful. The spectral graph model shows that certain stereotyped brain oscillations emerge from spectral graph properties of the structural connectome and influence resulting functional activity patterns. Furthermore, the model closely matches both the spectral and spatial patterns of alpha and beta rhythms seen in resting-state functional magnetoencephalography (MEG) recordings acquired from healthy subjects, as seen in **Figure 1.1**, adapted from Raj et al.¹

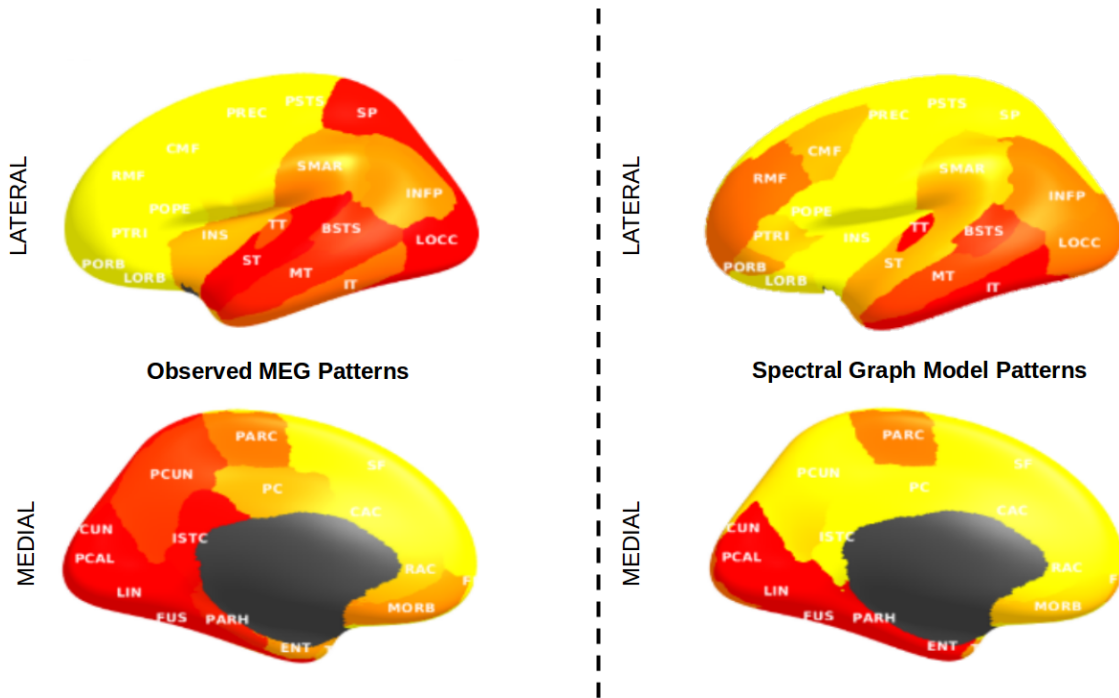


Figure 1.1: Patterns of spatial localization of alpha wave activity from real observed MEG data (left), displaying lots of activation in the posterior brain regions, and the corresponding spatial patterns that were generated using the spectral graph model (right), which replicates the expected posterior activation. *Adapted from Raj et al. (2019)¹*

Given an MEG spectrum, we'd like to find a set of parameters that describe the spectrum, with future applications in studying the relationship between the spectrum-parameter transformation and neurodegenerative diseases. The goal of our study was to accelerate the parameter inference process for the generative spectral graph model, which can capture group differences in functional brain activity, as well as reproduce all the characteristic wave signatures of the human brain in a simple and efficient manner, and thereby paving the way towards real time inference of neurophysiological information in neurocognitive disorders for diagnosis and prognosis.

1.2 Global Parameter Inference

Computational models require optimal parameters in order to obtain quantitative and qualitative predictions for the system being modeled. Parameter estimation is usually done by searching for the parameter values that, when used in a model simulation with differential equality and algebraic constraints, best fits experimental data with the least error.¹³ Various methods have been

proposed for parameter optimization in biological applications, including linear and non-linear least-squares fitting,¹⁴ simulated annealing,¹⁵ and evolutionary computation.¹⁶ For the aforementioned conventional neural field and mass models, their nonlinearity and high dimensionality result in non-convex optimization that significantly increase computational complexity.¹⁷ These problems are less prevalent in Raj et al.'s¹ spectral graph model, which is linear and low-dimensional.

We initially used Markov chain Monte Carlo (MCMC) sampling methods to generate a large enough sample from the posterior distribution so that possible spectral graph model parameter values can be accurately estimated without requiring much prior information about their actual distributions.¹⁸

The basic principle underlying MCMC methods is that a Markov chain can be constructed with a stationary distribution that is the joint posterior probability distribution of the parameters of the model. The parameters are assigned arbitrary initial values, and the chain is simulated until it converges to the stationary distribution. Observations from the chain are used to estimate the joint posterior probabilities of the parameters.¹⁹

Because MCMC parameter space is searched via local jumps leading to a relative lack of awareness of the global context of the distribution.²⁰ Chains are usually run for longer times with parallel processes initialized with significantly different initial states. In order to get more accurate estimates, the time required increases with the dimensionality of the parameters, which, for the Raj model, was in the order of hours.

1.3 *Neural Networks*

A data-driven approach using a deep neural network could be used to speed up the process of parameter inference for computational models. We hypothesize that the initial time to train the model will likely be the bottleneck in the process, but given a sufficiently large dataset,²¹ it would be possible for the algorithm to learn the relationship between observed spectra and model parameters. Studies using supervised machine learning methods for the purposes of parameter inference have mainly been applied in the fields of physics.^{22,23} Other studies looking into improving upon unsupervised stochastic sampling methods like MCMC, usually seek ways to improve the MCMC algorithm through exploiting the geometry of the target distribution(s), introducing scalability, parallel processing, or improving the proposal distribution(s) (see Robert et al.²⁰ for a review of these methods). The process of MCMC can result in tens of thousands of simulated data that

closely match the desired target problem, providing a large dataset ideal to be used as a training set for a neural network. In a neural network, input data (here, MEG spectra) are multiplied with adjustable parameters, known as weights, associated with each neuron in each layer, and then an activation function is applied before transferring it to further layers. Multiple layers of neurons give rise to a deep neural network. Each layer effects a transformation of the feature space of the target of interest, allowing a deeper network to be able to represent more complex transformations. The model is trained by minimizing a cost function associated with the difference between predicted and desired output via update of weights and biases through backpropagation²⁴ and gradient descent. By employing a trained neural network, we hope to then be able to transform an input MEG power spectrum into the required global parameters for the spectral graph model within seconds.

2 Methods

2.1 *Experimental Dataset Acquisition and Processing*

All model simulations were compared with ground truth resting-state MEG data acquired for 36 healthy adult subjects (23 males, 13 females; 26 left-handed, 10 right-handed; mean age 21.75 years (range: 7-51 years)).¹ All study procedures were approved by the institutional review board at the University of California, San Francisco (UCSF) and are in accordance with the ethics standards of the Helsinki Declaration of 1975 as revised in 2008. MEG recordings were acquired with a 275-channel CTF Omega 2000 whole-head MEG system from VSM MedTech (Coquitlam, BC, Canada). All subjects were instructed to keep their eyes closed for five minutes while their MEGs were recorded at a sampling frequency of 1200 Hz. MEG recordings were then down-sampled from 1200 Hz to 600 Hz, then digitally filtered to remove DC offset and any other noisy artifact outside of the 1 to 160 Hz bandpass range. Source localization to infer the neuronal activity that generated the observed signal was applied using the adaptive spatial filtering algorithms from the NUTMEG software tool written in house²⁵ in MATLAB (The MathWorks, Inc., Natick, Massachusetts, United States). All sources were labeled based on the Desikan-Killiany atlas available in the FreeSurfer software,²⁶ parcellated into 68 cortical regions and 18 subcortical regions.

2.2 Spectral Graph Model

The spectral graph model proposed by Raj et al.¹ is a hierarchical, linear graph model of brain activity at mesoscopic and macroscopic scales that yields a closed-form solution for relating the brain’s structural wiring to both spatial and spectral functional patterns of brain oscillations (MEG recordings described in **Section 2.1**). The steady state spectral response $\mathbf{X}(\omega)$ induced by the brain’s structural connections at angular frequency ω is described by **Equation 1**:

$$\mathbf{X}(\omega) = \sum_i \frac{\mathbf{u}_i(\omega)\mathbf{u}_i^H(\omega)}{i_G\lambda_i(\omega)F_e(\omega)}H_{local}(\omega)\mathbf{P}(\omega) \quad (1)$$

The spectral graph model of brain activity involves only seven global parameters: τ_e , τ_i , α , speed, τ_c , g_{ee} , and g_{ii} . The speed parameter describes the corticocortical fiber conduction speed, which is assumed to be a global constant independent of the pathway under question. The time constants τ_e and τ_i describe the delays in neural responses of excitatory and inhibitory neurons, while τ_c is thought to capture the dynamics of long-range afferents as well as a global or ”graph” time constant. The parameter α represents a global coupling constant that controls the relative weight given to long-range afferents compared to local signals. Finally, g_{ei} , and g_{ii} describe the gain constants for excitatory and inhibitory cells in the neuronal signal (g_{ee} is fixed at a value of 1). For the purposes of this study, we reduced the parameters of interest to the first five. The gain parameters g_{ei} and g_{ii} only produce a global shift in the power spectrum.

2.3 Simulated Dataset Acquisition with MCMC

MCMC was used for parameter inference for the spectral graph model, as well as the basis for the simulated training dataset for the fully connected neural network described in **Section 2.4**. Affine-invariant ensemble MCMC sampling²⁷ was used for faster convergence by updating an ensemble of model parameters each ”generation” as opposed to one set of parameters at a time, making it easier to parallelize over each set of correlated parameters within a generation. This method involves simultaneously evolving an ensemble of K walkers where the proposal distribution for one walker k is based on the current positions of the K_1 walkers in the complementary ensemble. To update the position of a walker at position X_k , a walker X_j is drawn randomly from the remaining walkers and a new position is proposed. This process is repeated until convergence.

The `emcee`²⁸ ensemble sampler was built in Python 3.7 and run on ten Intel Xeon E5-2680v3 CPUs (Intel Corporation, Santa Clara, CA, USA) with 5 GB memory per CPU in order to generate proposal distributions for the five global parameters of interest: τ_e , τ_i , α , speed, and τ_c . 16 walkers in the ensemble searched for 5000 iterations (steps) in nine parallel processes with initial parameter values of 0.3 s, 0.3 s, 0.8, 14.4 m/s, and 0.2 s, respectively. During the process, proposal parameters are run through the spectral graph model and compared with empirical MEG power spectra from the experimentally acquired MEG dataset. For comparing model-generated spectra with real source-localized MEG data, Pearson’s correlation coefficient (also known as Pearson’s r) was used as a cost function for each of the 86 regions of the brain. If error, defined as 1-Pearson’s r , is reduced, these parameters are added to a chain of parameter values with a certain probability determined by how much better it is. Extra positional arguments were used to accelerate the MCMC process, including *a priori* proposal distributions for each parameter (outlined in **Appendix Tables A1** and **A2**), the expected MEG signal range of 2 to 45 Hz, and an example of a downsampled MEG power spectrum from the experimentally acquired MEG dataset.

2.4 Fully Connected Neural Network

We implemented a fully connected neural network with three hidden layers, with the last layer consisting of five nodes for each of the five global parameters: τ_e , τ_i , α , speed, and τ_c . **Figure 2.1** is a diagrammatic representation of the neural network. The initial architecture consisted of five hidden layers, but after visualizing the distribution of node weights in each layer (**Appendix Figure A1**) we reduced the network by two layers as most of the initial weights were zero and thus not contributing to the transformation algorithm. Each hidden layer is followed by batch normalization, which normalizes the output of a previous activation layer by subtracting the batch mean and dividing by the batch standard deviation and thereby improving the stability of the network, and ReLU activation, which transforms data non-linearly for faster training.²⁹ We used mean-squared error (MSE) as the cost function, as defined by **Equation 2**, where \hat{y}_i is the predicted value of the i -th sample and y_i is the corresponding true value for the total of n samples. The

network was built in Python 3.7 using the Keras deep learning library with TensorFlow backend.³⁰

$$\epsilon_{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2)$$

When tuning neural network hyperparameters, multiple GPUs are usually required to speed up the search.³¹ This work used the Extreme Science and Engineering Discovery Environment (XSEDE) cluster Comet, hosted by the San Diego Supercomputer Center at UC San Diego, through allocation TG-IBN180015.^{32,33} Training was run on two NVIDIA Tesla P100 GPUs (NVIDIA Corporation, Santa Clara, CA, USA) with 128 GB memory.

The dataset used to train the neural network was composed of 230,400 simulated MEGs. Each MEG was generated a step taken in the MCMC process, after ignoring the initial 2,000 unstable "burn-in" steps during which most of the sampled parameters are unlikely to be physically relevant. Each MEG set consists of spectra for the 86 parcellated regions of the brain, divided into 40 frequency bands, and their corresponding five global parameters. This simulated data was then split into a training set of 184,400 MEGs and an unseen validation set of 46,000 MEGs, used to evaluate the neural network performance. Fully connected networks tend to memorize whatever is put before them, i.e. the training data, in a process known as overfitting. As a result, its often useful in practice to track the performance of the network on a hold-out validation set.

Some additional preprocessing was also performed on the data to help the neural network converge faster and learn relevant information. The outputs, i.e. the five model parameters, were all scaled to be between 0 and 1, as the original scale and distribution are different for each parameter, which may increase the difficulty of the problem being modeled. A target variable with a large spread of values may result in large error gradient values, causing weight values to change dramatically, making the learning process unstable. The frequency inputs were also rescaled to a dB scale, which is more semantically relevant as the different frequency bands on MEG have a logarithmic relationship.

We used a training batch size of 800 MEGs, which was the maximum limit for the allocated GPU resources. He initialization³⁴ was used for initializing weights, involving random initialization of weights close to zero to break symmetry and prevent every neuron from performing the same computation, as well as initialization of different ranges of weights depending on the size of the previous layer of neurons. This helps attain a global minimum of the cost function, mean-squared

error (MSE), faster and more efficiently. The Adam algorithm for stochastic optimization³⁵ was used with an initial learning rate of 0.001, β_1 of 0.9, β_2 of 0.999, ϵ of 0, and scheduled decay of 0.004. Optimization involves finding a set of weights that minimizes the cost function, MSE. Adam uses the RMSprop optimization method with the addition of momentum, which helps accelerate stochastic gradient descent in the relevant direction when around local optima and dampens oscillations. Adam also uses bias correction, which computes bias-corrected first and second moment estimates to update parameters. The learning rate determines the size of the steps taken to reach a local minimum, and was changed to 0.0001 after 150 epochs. The figures in this paper were produced after 200 epochs of training.

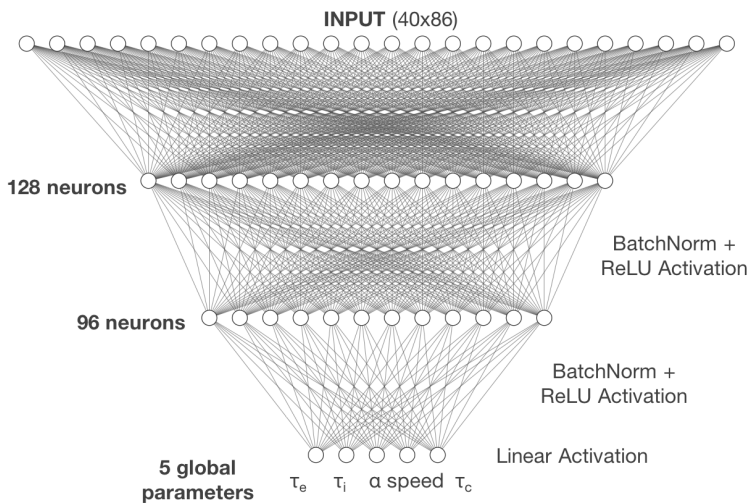


Figure 2.1: Proposed Spectrome-AI model architecture with an input size of 3,440 neurons per whole brain MEG spectrum, two hidden layers with batch normalization and ReLU activation, and a final layer with linear activation outputting the five global parameters for the spectral graph model. Number of neurons shown in each layer are not to scale.

2.5 Performance Analysis

The coefficient of determination (R^2) score was used to evaluate the network’s predictions on a random set of 4200 MEGs in both the training and cross-validation datasets, and was implemented using the `scikit-learn` machine learning library in Python.³⁶ R^2 represents the proportion of variance of y that has been explained by the independent variables in the model, and is defined in **Equation 3**, where $\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$, and $\sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n \epsilon_i^2$. It provides an indication of

goodness of fit and therefore a measure of how well unseen samples are likely to be predicted by the model, through the proportion of explained variance. As a result, the best possible R^2 score is 1.0, and an R^2 score of 0.0 represents a constant model that always predicts the expected value of y , disregarding the input features. Negative R^2 scores indicate model performance that is arbitrarily worse than constant.

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (3)$$

Ordinary least squares (OLS) linear regression was also run on the scatterplots using the `scikit-learn` library in Python. A line $\hat{y} = \beta_1 y + \beta_2$ is drawn through the scatterplot such that the sum of all squared deviations from the line (i.e. sum of squared residuals, described in **Equation 4**, where b is a candidate value for the parameter vector β) is minimized.

$$S(b) = \sum_{i=1}^n (\hat{y}_i - y_i^T b)^2 \quad (4)$$

The gold standard regression in this case would be an identity line, where the slope $\beta_1 = 1$, intercept $\beta_2 = 0$, and correlation coefficient $r^2 = 1.0$, corresponding to predicted parameter values that are the exact same as the ground truth parameter values.

3 Results

3.1 MCMC Parameter Distributions

The posterior parameter distributions that were generated after 5000 MCMC steps are shown in **Figure 3.1**. Each row corresponds to a patient. Most time constant parameters tended to have values close to 0 ms, with τ_e showing more variation and bimodality than τ_i or τ_c . Most α values were around 1, with some spread to higher values, and speed parameter values tended to localize around 15 m/s. The MCMC process was run for a total of 15 min (900 s) to achieve these posterior distributions.

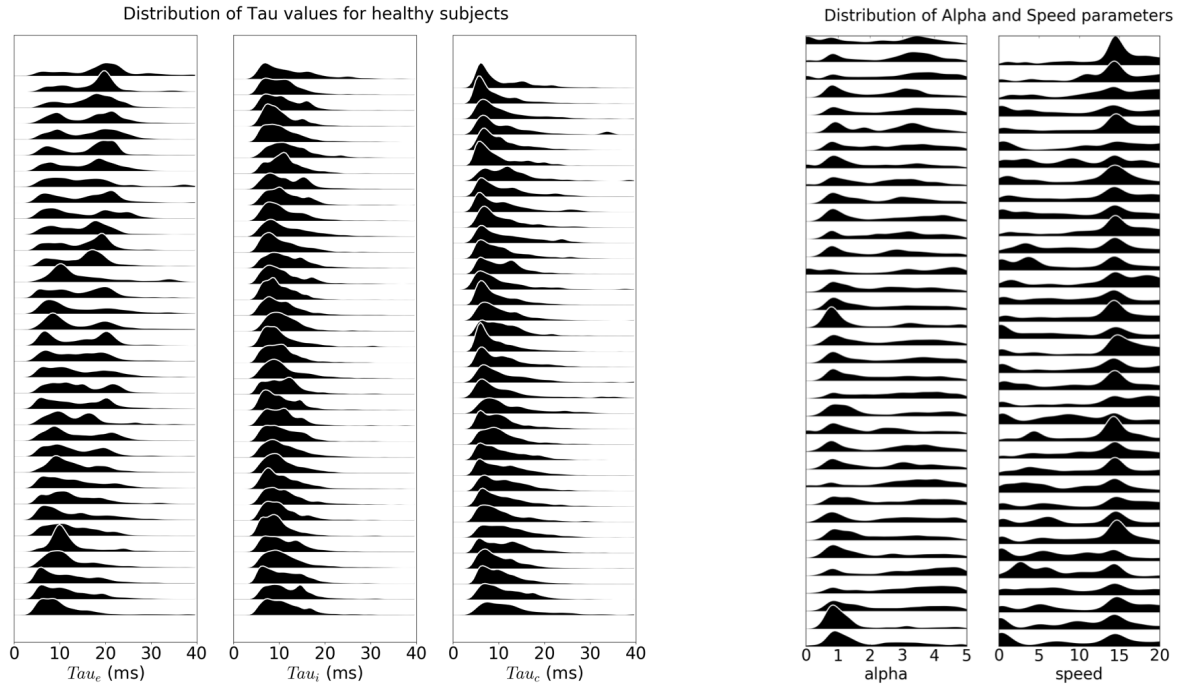


Figure 3.1: The posterior probability distributions after running the MCMC process. *Courtesy of Pablo F. Damasceno, PhD.*

3.2 Deep Learning Model Performance

Figure 3.2 shows the resulting neural network performance. **Table 3.1** compares the different regression statistics. In general, the neural network predicted the speed parameter well, with R^2 scores of close to 1.0 when predicting on both training and unseen cross-validation MEG data. The global coupling constant α and excitatory time constant τ_e are also relatively precise for both the training and cross-validation sets. However, the neural network performed poorly when estimating the τ_e and τ_i parameters, with some overfitting occurring with slightly better performance on the training data than the cross-validation data. Additionally, non-physiological values of less than zero are also being inferred by the neural network. Notably, performing parameter inference on an input MEG spectrum using this trained neural network takes only 45 ms, which is 20,000 times faster than the MCMC method.

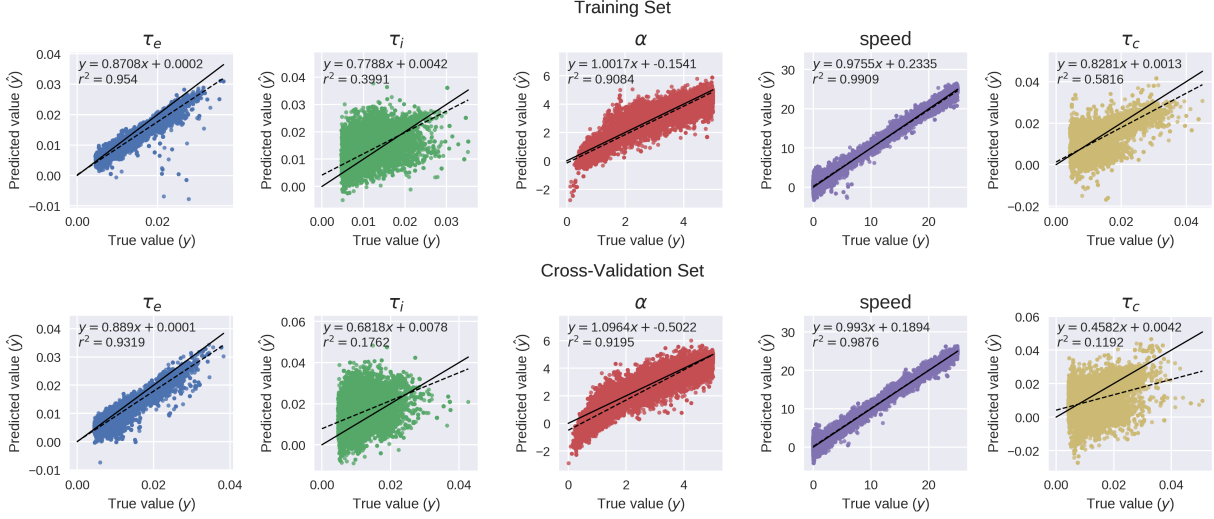


Figure 3.2: Predicted vs true parameter values on 4200 randomly selected training (top) and cross-validation (bottom) MEG datasets. R^2 scores for each are displayed in the top left of each plot, while the solid black line indicates the desired identity line where the predicted value is equal to the true value ($\hat{y} = y$.)

Table 3.1: Coefficient of determination (R^2) score for each parameter prediction, and resulting slope β_1 , intercept β_2 , and correlation coefficients after running linear regression for $\hat{y} = \beta_1 y + \beta_2$ on the scatterplots. The gold standard regression would be an identity line, where $\beta_1 = 1$, $\beta_2 = 0$, and $r^2 = 1.0$.

	Training					Cross-Validation				
	τ_e	τ_i	α	speed	τ_c	τ_e	τ_i	α	speed	τ_c
R^2	0.8567	-0.2035	0.8858	0.9907	0.4698	0.8825	-2.6823	0.8442	0.9872	-0.9618
β_1	0.8708	0.7788	1.0017	0.9755	0.8281	0.889	0.6818	1.0964	0.993	0.4582
β_2	0.0002	0.0042	-0.1541	0.2335	0.0013	0.0001	0.0078	-0.5022	0.1894	0.0042
r^2	0.954	0.3991	0.9084	0.9909	0.5816	0.9319	0.1762	0.9195	0.9876	0.1192

3.3 Applying the Generative Model

We used a default set of model parameters to develop and test the spectral graph model, as described in **Table 3.2**. The model-generated spectra from these default parameters were used as a toy example to visualize neural network performance in a neurophysiological context. The weights from the trained neural network were loaded and used to predict the τ_e , τ_i , α , speed, and τ_c parameters

from the input model-generated spectrum, producing the estimated parameters described in **Table 3.2**. These predicted parameters were then fed through the forward spectral graph model to produce power spectra, as shown in comparison with the default parameter-generated spectra in **Figure 3.3**. Since our neural network only predicted the τ_e , τ_i , α , speed, and τ_c parameters, the default parameter values of $g_{ei} = 4.0$ and $g_{ii} = 1.0$ were used.

Table 3.2: Default and predicted model parameter values

	τ_e	τ_i	α	speed	g_{ei}	g_{ii}	τ_c
Default Parameters	0.012	0.003	1.0	5.0	4.0	1.0	0.006
NN Predicted Parameters	0.0078	0.0095	3.0	5.9	-	-	0.019

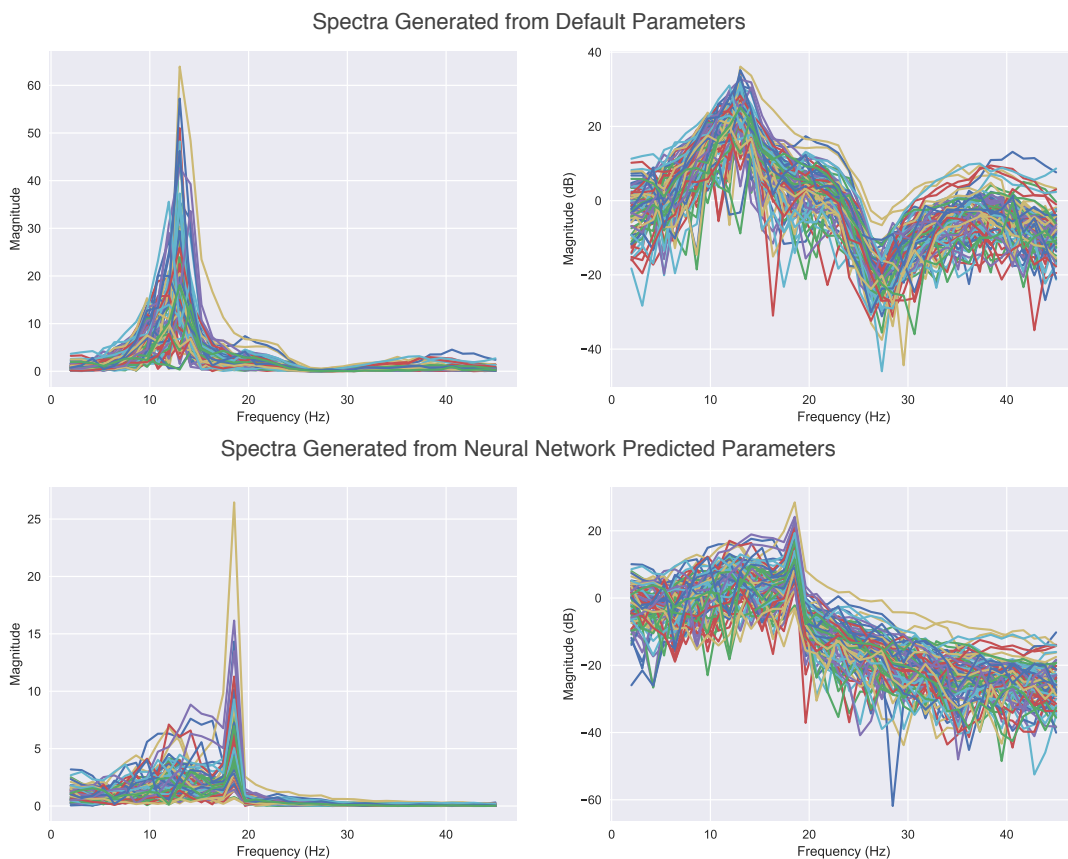


Figure 3.3: Power spectra generated from the spectral graph model using default parameters (top) and the predicted parameters (bottom) that were inferred using the trained neural network. Both linear data (left) and data converted to dB (right) are shown for each spectrum.

4 Discussion

The aim of this study was to provide a tool in the form of a deep neural network that can be used to infer global parameters for the spectral graph model provided by Raj et al.¹ faster than stochastic sampling methods without forfeiting too much in precision. A fully connected neural network with three hidden layers was utilized, where training the network took approximately 60 hrs but predicting parameters with the trained network took only 45 ms, which is 20,000 times faster than the previously used MCMC process. The overall prediction performance was assessed for each parameter, with some parameters correlating better with their true values than others. The power spectra generated with these predicted parameters can be visually compared with that generated by known true parameters, as seen in **Figure 3.3**. There are slight differences in frequency peaks, possibly due to the differences in the α parameter.

4.1 *Limitations*

The benefit of computational models is the potential for generating large amounts of data, which was taken to our advantage in generating a potential dataset for training a deep neural network. However, the data is still simulated, and differences still exist between the experimentally acquired MEG power spectra and those generated from the spectral graph model. The spectral graph model itself is robust at characterizing alpha and beta rhythms, but could be improved further by incorporating more local dynamics in order to describe higher frequency waves such as gamma rhythms.

An additional layer of potential errata exists from using the MCMC methods as ground truth, as they are stochastic simulations that approximate well the posterior parameter distributions for the spectral graph model, but still leave room for error. Potential improvements include incorporating real MEG data during training as well to fine-tune the neural network.

Typically, training is run until the loss in the hold-out validation set does not decrease further, usually after about 200 epochs without improvement. However, the network in this study was only trained for 200 epochs given the time and resources at hand, suggesting that further training may be required for convergence to occur.

4.2 *Future Steps*

The current trained neural network has been evaluated on an unseen validation dataset of simulated MEG spectra, but future work would include applying the trained neural network to infer parameters from the experimental source localized MEG spectra themselves and assessing the performance of those parameters in the spectral graph model compared with the original.

Other potential deep learning models may be explored in both predictive power and processing time, such as a convolutional neural network or a generative adversarial network, which may be able to find more complex relationships that better describe the transformation from an input power spectrum to the spectral graph model parameters. Nevertheless, as seen in **Appendix Figure A1**, even additional hidden layers may not be necessary in this problem. Hyperparameters for the network could also be explored further for faster convergence while also employing regularization to allow for better generalization to unseen examples and not overfitting.

Additionally, all data in this study were acquired or derived from healthy subjects at resting state. We can extend the applications of this real time parameter inference to study group differences in functional neural activity, both resting-state and task-induced, between healthy controls and cohorts with various neurocognitive disorders such as schizophrenia or Alzheimer's. We can potentially further train another deep learning model to classify these cohorts based solely on raw input MEG data.

References

- [1] A. Raj, C. Cai, X. Xie, E. Palacios, J. Owen, P. Mukherjee, and S. Nagarajan, “Spectral graph theory of brain oscillations,” *bioRxiv*, 2019.
- [2] S. Herculano-Houzel, “The human brain in numbers: a linearly scaled-up primate brain,” *Frontiers in human neuroscience*, vol. 3, p. 31, 2009.
- [3] O. Sporns, G. Tononi, and R. Ktter, “The human connectome: A structural description of the human brain,” *PLOS Computational Biology*, vol. 1, 09 2005.
- [4] J. Alstott, M. Breakspear, P. Hagmann, L. Cammoun, and O. Sporns, “Modeling the impact of lesions in the human brain,” *PLOS Computational Biology*, vol. 5, pp. 1–12, 06 2009.
- [5] F. Abdelnour, H. U. Voss, and A. Raj, “Network diffusion accurately models the relationship between structural and functional brain connectivity networks,” *NeuroImage*, vol. 90, pp. 335 – 347, 2014.
- [6] E. M. Izhikevich and G. M. Edelman, “Large-scale model of mammalian thalamocortical systems,” *Proceedings of the national academy of sciences*, vol. 105, no. 9, pp. 3593–3598, 2008.
- [7] P. Skudlarski, K. Jagannathan, V. D. Calhoun, M. Hampson, B. A. Skudlarska, and G. Pearlson, “Measuring brain connectivity: diffusion tensor imaging validates resting state temporal correlations,” *Neuroimage*, vol. 43, no. 3, pp. 554–561, 2008.
- [8] P. Hagmann, O. Sporns, N. Madan, L. Cammoun, R. Pienaar, V. J. Wedeen, R. Meuli, J.-P. Thiran, and P. E. Grant, “White matter maturation reshapes structural connectivity in the late developing human brain,” *Proceedings of the National Academy of Sciences*, vol. 107, no. 44, pp. 19067–19072, 2010.
- [9] P. Skudlarski, K. Jagannathan, K. Anderson, M. C. Stevens, V. D. Calhoun, B. A. Skudlarska, and G. Pearlson, “Brain connectivity is not only lower but different in schizophrenia: a combined anatomical and functional approach,” *Biological psychiatry*, vol. 68, no. 1, pp. 61–69, 2010.
- [10] A. Destexhe and T. J. Sejnowski, “The wilson–cowan model, 36 years later,” *Biological cybernetics*, vol. 101, no. 1, pp. 1–2, 2009.

- [11] G. Deco, M. Senden, and V. Jirsa, “How anatomy shapes dynamics: a semi-analytical study of the brain at rest by a simple spin model,” *Frontiers in computational neuroscience*, vol. 6, p. 68, 2012.
- [12] O. David and K. J. Friston, “A neural mass model for meg/eeg:: coupling and neuronal dynamics,” *NeuroImage*, vol. 20, no. 3, pp. 1743–1755, 2003.
- [13] G. Lillacci and M. Khammash, “Parameter estimation and model selection in computational biology,” *PLOS Computational Biology*, vol. 6, pp. 1–17, 03 2010.
- [14] P. Mendes and D. Kell, “Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation.,” *Bioinformatics (Oxford, England)*, vol. 14, no. 10, pp. 869–883, 1998.
- [15] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, “Equation of state calculations by fast computing machines,” *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.
- [16] C. G. Moles, P. Mendes, and J. R. Banga, “Parameter estimation in biochemical pathways: a comparison of global optimization methods,” *Genome research*, vol. 13, no. 11, pp. 2467–2474, 2003.
- [17] D. E. Knuth, “Postscript about np-hard problems,” *SIGACT News*, vol. 6, pp. 15–16, Apr. 1974.
- [18] G. Roberts and A. Smith, “Simple conditions for the convergence of the gibbs sampler and metropolis-hastings algorithms,” *Stochastic Processes and their Applications*, vol. 49, no. 2, pp. 207 – 216, 1994.
- [19] B. Rannala, “Identifiability of Parameters in MCMC Bayesian Inference of Phylogeny,” *Systematic Biology*, vol. 51, pp. 754–760, 09 2002.
- [20] C. P. Robert, V. Elvira, N. Tawn, and C. Wu, “Accelerating mcmc algorithms,” *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 10, no. 5, p. e1435, 2018.

- [21] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, pp. 436–444, May 2015.
- [22] X. Fan, J. Li, X. Li, Y. Zhong, and J. Cao, “Applying deep neural networks to the detection and space parameter estimation of compact binary coalescence with a network of gravitational wave detectors,” *SCIENCE CHINA Physics, Mechanics & Astronomy*, vol. 62, no. 6, p. 969512, 2019.
- [23] C. Dreissigacker, R. Sharma, C. Messenger, R. Zhao, and R. Prix, “Deep-learning continuous gravitational waves,” *arXiv preprint arXiv:1904.13291*, 2019.
- [24] D. E. Rumelhart, G. E. Hinton, R. J. Williams, *et al.*, “Learning representations by back-propagating errors,” *Cognitive modeling*, vol. 5, no. 3, p. 1, 1988.
- [25] S. S. Dalal, J. Zumer, V. Agrawal, K. Hild, K. Sekihara, and S. Nagarajan, “Nutmeg: a neuromagnetic source reconstruction toolbox,” *Neurology & clinical neurophysiology: NCN*, vol. 2004, p. 52, 2004.
- [26] B. Fischl, D. H. Salat, E. Busa, M. Albert, M. Dieterich, C. Haselgrove, A. Van Der Kouwe, R. Killiany, D. Kennedy, S. Klaveness, *et al.*, “Whole brain segmentation: automated labeling of neuroanatomical structures in the human brain,” *Neuron*, vol. 33, no. 3, pp. 341–355, 2002.
- [27] J. Goodman and J. Weare, “Ensemble samplers with affine invariance,” *Commun. Appl. Math. Comput. Sci.*, vol. 5, no. 1, pp. 65–80, 2010.
- [28] D. Foreman-Mackey, D. W. Hogg, D. Lang, and J. Goodman, “emcee: The MCMC Hammer,” *Publications of the Astronomical Society of the Pacific*, vol. 125, p. 306, Mar. 2013.
- [29] X. Glorot, A. Bordes, and Y. Bengio, “Deep sparse rectifier neural networks,” in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 315–323, 2011.
- [30] F. Chollet *et al.*, “Keras.” <https://keras.io>, 2015.

- [31] G. Luo, “A review of automatic selection methods for machine learning algorithms and hyperparameter values,” *Network Modeling Analysis in Health Informatics and Bioinformatics*, vol. 5, pp. 1–16, 2016.
- [32] J. Towns, T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. D. Peterson, R. Roskies, J. R. Scott, and N. Wilkins-Diehr, “Xsede: Accelerating scientific discovery,” *Computing in Science & Engineering*, vol. 16, pp. 62–74, Sept.-Oct. 2014.
- [33] N. Wilkins-Diehr, S. Sanielevici, J. Alameda, J. Cazes, L. Crosby, M. Pierce, and R. Roskies, “An overview of the xsede extended collaborative support program,” in *High Performance Computer Applications - 6th International Conference, ISUM 2015, Revised Selected Papers*, vol. 595 of *Communications in Computer and Information Science*, (Germany), pp. 3–13, Springer Verlag, 1 2016.
- [34] K. He, X. Zhang, S. Ren, and J. Sun, “Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification,” *arXiv:1502.01852 [cs]*, Feb. 2015. arXiv: 1502.01852.
- [35] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *arXiv:1412.6980 [cs]*, Dec. 2014. arXiv: 1412.6980.
- [36] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

Appendix

Table A1: *A priori* proposal Gamma distributions for parameters τ_e , τ_i , and τ_c as extra positional arguments for the ensemble MCMC sampler.

$$f(x, a, b) = \frac{1}{\Gamma(a)b} \left(\frac{x}{b}\right)^{a-1} e^{-\frac{x}{b}}$$

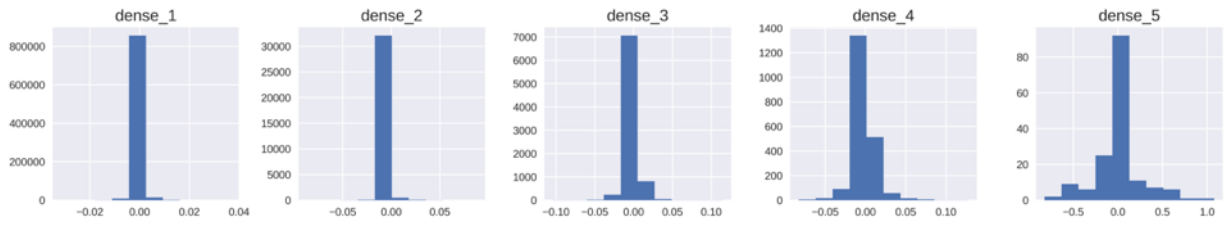
	τ_e	τ_i	τ_c
alpha (a)	2	2.006014687419703	2
scale (b)	0.0032375996670863947	0.0025497764353055712	0.0029499360144432463
loc (x)	0.004492887509221829	0.004662441067103153	0.004211819821836749

Table A2: *A priori* proposal uniform distributions for parameters α and speed as extra positional arguments for the ensemble MCMC sampler.

$$f(x, a, b) = \begin{cases} \frac{1}{b-a} & a \leq x \leq b \\ 0 & x < a, b < x \end{cases}$$

	α	speed
lower (a)	0	0
upper (b)	5	25

A



B

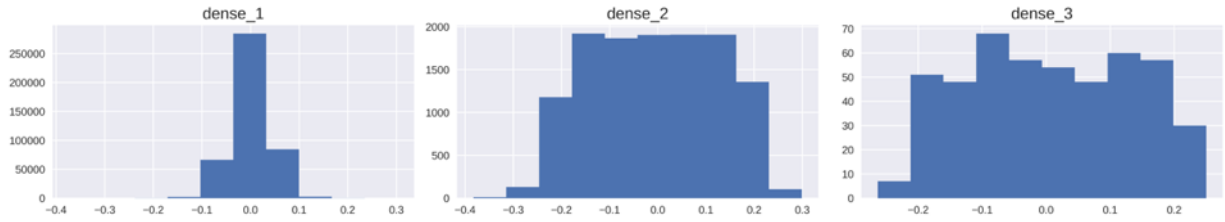



Figure A1: Histogram of weights in each fully connected layer in a **A)** five-layer network and a **B)** three-layer network.

Publishing Agreement

It is the policy of the University to encourage the distribution of all theses, dissertations, and manuscripts. Copies of all UCSF theses, dissertations, and manuscripts will be routed to the library via the Graduate Division. The library will make all theses, dissertations, and manuscripts accessible to the public and will preserve these to the best of their abilities, in perpetuity.

Please sign the following statement:

I hereby grant permission to the Graduate Division of the University of California, San Francisco to release copies of my thesis, dissertation, or manuscript to the Campus Library to provide access and preservation, in whole or in part, in perpetuity.



Author Signature

9/6/19

Date