

Deep Learning Architecture with Dynamically Programmed Layers for Brain Connectome Prediction

Vivek Veeriah
University of Central Florida
4000 Central Florida Blvd
Orlando, FL 32816
vivekveeriah@knights.ucf.edu

Rohit Durvasula
University of Central Florida
4000 Central Florida Blvd
Orlando, FL 32816
drohit@knights.ucf.edu

Guo-Jun Qi
University of Central Florida
4000 Central Florida Blvd
Orlando, FL 32816
guojun.qi@ucf.edu

ABSTRACT

This paper explores the idea of using deep neural network architecture with dynamically programmed layers for brain connectome prediction problem. Understanding the brain connectome structure is a very interesting and a challenging problem. It is critical in the research for epilepsy and other neuropathological diseases. We introduce a new deep learning architecture that exploits the spatial and temporal nature of the neuronal activation data. The architecture consists of a combination of Convolutional layer and a Recurrent layer for predicting the connectome of neurons based on their time-series of activation data. The key contribution of this paper is the dynamically programmed layer that are critical in determining the alignment between the neuronal activations of pair-wise combinations of neurons.

1. INTRODUCTION

Accurately modeling the brain of mammals is the most challenging and interesting problem in the field of neuroscience. The structure of brain has always been the driving factor for developing computational intelligence systems, which in-turn help in understanding the structure of the brain. The main reason for it to be challenging is because of the presence of a large number of neurons - the human brain is estimated to have 100 billion neurons each being connected to thousands of neurons. Moreover, understanding the structure of brain can solve a number of problems in different fields. Understanding the brain structure is important for the treatment of epilepsy and other neuropathological diseases. At the neuronal level, obtaining the exact connectivity information from the brain, where each neuron has on average 7000 synaptic connections between neurons is a daunting task. There are some traditional methods for determining the connection between the neurons [8]. But they are inaccurate, inefficient or require invasive approaches.

Connectomics is study of connectomes, which are maps of connection within an organism's nervous system. Inferring the network topology from patterns of neural activity

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

KDD '15 Sydney, Australia

Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...\$15.00.

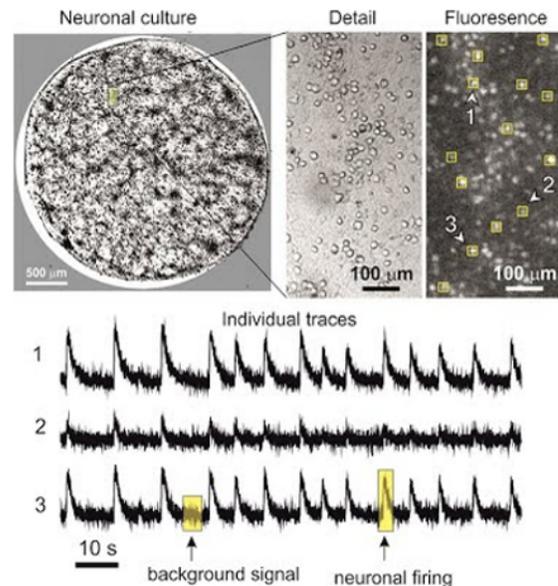


Figure 1: Plot of time series of activations of brain neurons.[6]

are not new. The aim of connectomics is to derive the detailed structure of the entire neural system. Some early approaches that were seminal in the recent developments of connectomics were those done by White et. al.[20] - a 300 cell nervous system of a nematode worm was accomplished. There have also been several active research efforts in the recent years to produce and analyze connectome datasets at meso and macro scales. These involve non-invasive imaging techniques of brain activity such as functional magnetic resonance imaging (fMRI), including the Human Connectome Project which is being led by the WU-Minn Consortium [18]. At a smaller scale, the effort of reconstructing networks from neural activity can be traced back to work done by Eytan-Maron in the year 2006 [1]. In their work, the in-degree of neurons were estimated based on the simple logic of inferring connections based on the higher firing frequency of activations. Another major study which brought in a breakthrough was the one done by the Paninski group at Columbia which was aimed at reconstructing the connectivity from calcium fluorescence imaging data [19] [10]. The idea was to infer spike times as a Bayesian inverse problem and then estimate the Generalized Linear Model (GLM) [9]

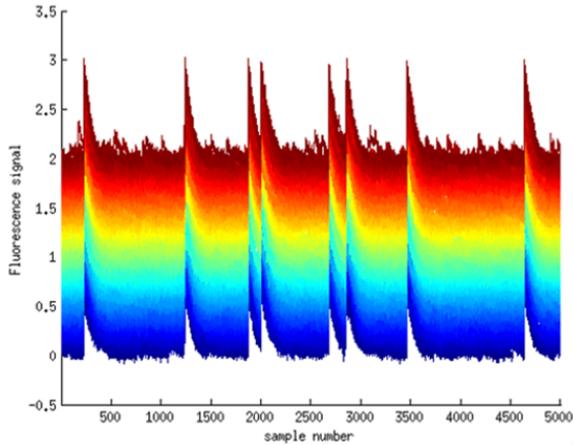


Figure 2: Time series of fluorescence activities of all neurons with a small vertical offset

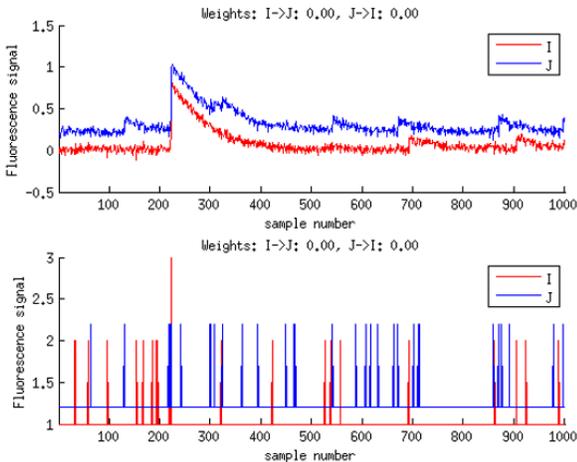


Figure 3: Time series and discretized activities of two neurons with a small vertical offset

kernels which were a measure of connectome weights of neurons. Their work builds rigorously on the study of GLM models demonstrating the reconstruction of spike data [17]. These methods have their own drawbacks but are far more scalable than axonal tracing.

To address this problem in a different way rather than adopting the conventional method, a new computation technique is employed. A recent breakthrough for observing the neural activities of several neurons has been designed. This new methodology takes advantage of optical imaging techniques. The imaging of calcium influx into neurons provides an indirect and accurate measure of action-potential generation within individual neurons. Certain fluorescent molecules are used which respond to the binding of calcium ions by changing their fluorescence properties [4].

These optical data are captured and were made public for reverse-engineering the structure of brain. An open competition was also based on this released dataset. The dataset consists of time-series data of neuronal activations which are generated by realistic simulations of real networks of neurons observed via calcium fluorescence recordings. The data

closely resembles the real recordings of cultured neurons and provides unequivocal ground truth of neuronal connections. The data was generated using a simulator that was extensively studied and validated [15]. The dynamic behavior of neurons was adapted to be reproduced by the simulator as observed in the cultures. The model also simulates the limitations and defects of the calcium fluorescence imaging technology - limited time resolution where the optical imaging technique does not allow to separate between individual spikes of neurons and light scattering effects where a particular neuron gets stimulated by the activity of nearby neurons. The data simulator used for synthesizing the data conforms with realistic scenarios at three levels [15]. The challenge is to infer the directed connections of neural networks from patterns of neural activity of individual neurons. The problem can therefore be thought as a causal structure reconstruction problem given the time series activation data of neurons. Similar instances of this problem can be found in genomics, climatology, epidemiology and econometrics.

In this paper, we present our unique approach which utilize the time-series nature of the neuronal activation data. The proposed model exploits both the spatial and temporal structure present in the activation time-series data through convolutional and max-pooling layers that output the features invariant to the background noises and local translation and warping of time-series data. The temporal structure of these data are exploited by the recurrent layers, a structure modeling the evolution of hidden states underlying the time-series data. This makes our model better generalized in temporal domain. One of the key contributions of this paper is the dynamically programmed layer whose structure is customized to reveal how two brain neurons are causally activated by aligning the compressed and generalized representation of activation sequences obtained from the recurrent layers.

The remainder of this paper is organized as follows. In Section 2, we review the background and related work on using time series of the activations of neurons to discover the brain connectome. The problem is formally defined in Section 3, followed by the proposed deep learning architecture in Section 4. In Section 5 we present the training of the proposed architecture through back propagation algorithm. The experiment results on the real competition datasets are demonstrated in Section 6. We finally conclude the paper in Section 7.

2. RELATED WORK

Understanding the structure of brain through its connectomes is the key for understanding the changes made to these structures through diseases like epilepsy and Alzheimer’s disease, thereby providing effective treatment for these. In order to achieve this goal of understanding the brain, the first ChaLearn Neural Connectomics Challenge was held. The dataset released under this challenge consisted of the time series activation data of 1000 neurons and their unequivocal ground truths that had been synthesized from a data simulator as explained in previously. The method that won the challenge was the work done by Suter et. al. [16]. Their method takes advantage of simple first order filters for smoothing and selectively amplifying the time series activations. Moreover, they also used a hard threshold filter for differentiating between the local and global activations of neurons. These thresholded signals were then subjected

to a selective amplification filter which amplifies only the local activations. The local activations are responsible for throwing the light on brain connectome structure. These local activation data are then used for calculating the partial correlation coefficients for pairs of brain neuron. These coefficients are calculated for every pair of neurons and compared with the ground truth data. It is clearly observed that their method is quite sensitive to the choice of the filters and thresholds used. Moreover, their method does not utilize the ground truth for tuning their parameters and also fails to utilize the neuronal position data provided with the dataset. Also, it does not perform well when the direction of causality of signals is considered.

Another interesting method that performed better when the direction of causality was considered was the work done by Lukasz Romaszko [12]. This method used a deep learning approach for processing the time series activation data. The brain neural activations were thresholded into four possible values - a drop, no change, a small increase and a spike based on some thresholds. Then every pair of these thresholded activation data was given as input to a Convolutional Neural Network. The architecture consisted of two Convolutional layers followed by a max pooling layer and another Convolutional layer. The max pooled output was then subjected to a fully connected layer. The softmaxed output was then used for calculating the error based on the ground truth, which was propagated back for correcting the weights in the neural network. Since every pair of neurons were provided as input, this method performed better when the direction of causality was considered. However, this method did not utilize the entire time series of the neuronal activations. They had empirically chosen the activation data upto 330 time steps claiming that these had the necessary statistics for determining the connectivity labels.

Work done by Czarnecki et. al. [12] was also one of the methods that produced good connectivity scores. In their approach, the neuron connectivity problem was modeled as a simple binary classification problem. Each pair of neuron was represented as a constant size vector where each dimension consisted of feature values produced by certain feature extractors like cross-correlation, cross-correlation with one-time frame lag, Generalized Transfer Entropy and Information Gain. Some topological features like Normalized Difference, Geometrical Closure, Markov Closure, Network De-Convolution etc. were also used for improving the classification process. These feature extractors were determined through validation over the validation set. In their paper, it was observed that only the Network De-Convolution feature that was extracted from the time-series activation data had consisted of the necessary information for predicting the connectivity between the neurons. All the other feature do not provide much information for classification. These extracted features for pairwise neurons was augmented into a complete vector which was then classified based on a random forest classifier. The feature extractors used in this approach were arbitrarily selected and the Network De-Convolution feature was the only pre-dominant feature that seemed to contribute to the classification process.

A popular method for inferring the brain connectomes is the one done by Stetter et. al. [15]. They explore the idea of modeling the brain into a Generalized Linear Model (GLM). Their objective was to improve the Transfer Entropy and is called the Generalized Transfer Entropy (GTE). The main

drawback of this method is that their assumption of modeling the brain into a GLM is not valid. This constrains the connectome problem to a GLM kernel estimation problem which has been observed to be incorrect [13].

It could be observed that all the recent works on connectomics are simple linear models that do not exploit the temporal structure present in the dataset. In contrast to these methods, our proposed method utilizes both the spatial and temporal information present in the entire time-series activations of the neurons. In contrast to the conventional deep neural networks, the structure is dynamically determined to reflect the alignment between the activity patterns of connected brain neurons.

3. PROBLEM FORMULATION

We begin with a formal definition of the problem in this section. Given two activation time-series data $\mathbf{x} = [x_1, \dots, x_T]$ and $\mathbf{x}' = [x_1, \dots, x_{T'}]$ of length T and T' from two brain neurons, we wish to predict if the former causes the activations of the latter one or its converse, and thus we can predict if the two neurons are causally connected along with its direction of causality. In other words, we wish to predict the directed connection between neurons.

No prior knowledge about the brain neural topology is given, making the competition very challenging. The only information available to decide the brain connectome are time-series activations, and a training set $\mathcal{D} = \{(\mathbf{x}, \mathbf{x}'), y\}$ on which the connection label $y \in \{0, 1\}$ is given to denote if a pair of neuron is causally connected or not. We wish to build a prediction model with competitive generalization ability on the test set without the given neural connections.

4. PROPOSED ARCHITECTURE

In this paper, we propose a novel architecture for predicting the connectomes based on the time-series neuronal activation data. The proposed architecture is as shown in Figure 4. The model consists of two separate convolutional layers which are subsequently followed by a recurrent layer.

The convolutional layer is responsible for learning of low-level filters that process the time-series data, and the recurrent layer is responsible for learning a compressed and generalized time-series representation of these filtered time-series activation data. This generalized representation obtained from the recurrent layers are then used in the dynamically programmed layer which is responsible for revealing the dynamic structure present between the two brain neurons. This dynamic structure reflects the alignment between the activity patterns of connected neurons.

During training the neurons are taken in a pair-wise manner along with their ground truth labels. Each neuron activation is passed through this model to obtain the time-series compressed representation from a recurrent network. The activations of each neuron in a pair are being dynamically aligned in order to reveal how two neurons in the brain are activated. In other words, the alignment throws light on the inherent structure that would be present between the pair of neurons considered. Based on these dynamically obtained alignments, the final output prediction layer then classifies these alignments as either connected or not connected. Then, the error obtained through the predicted and the ground truth label is then differentiated with respect to the parameters in each layer present in the model. The gra-

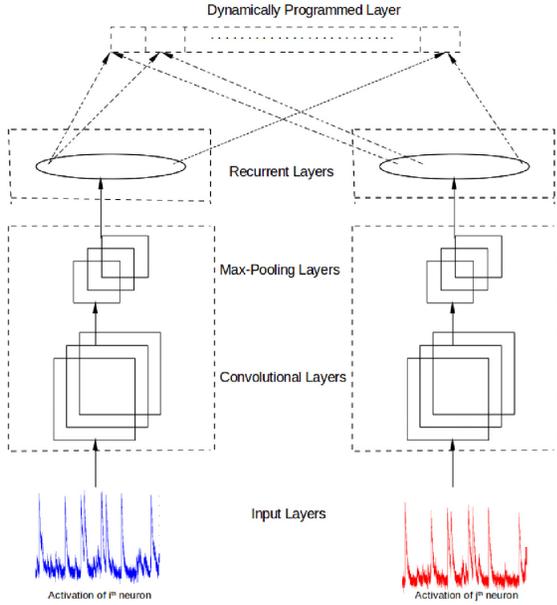


Figure 4: Proposed deep learning architecture. The input time series of activations of two brain neurons are processed through two separate networks, consisting of convolutional layers, max-pooling layers and recurrent layers from the bottom up. Then a dynamically programmed layer is built to align the output sequences of salient temporal patterns from the two recurrent layers, and predict the connectome between the brain neurons.

dients thus obtained are used for back propagation of errors through the layer. The training of our proposed model is based on back propagation.

Before we start presenting the details about each type of layers, we briefly introduce their structures and roles below.

Convolutional layers: They consist of one or more pairs of convolution and max-pooling layers. The convolution layer applies a set of filters that produces responses to small regions of the input signals [7]. These filters are replicated over the entire input signals to extract the strong temporal correlation from the time-series activations of brain neurons. A convolutional layer is followed by a max-pooling layer. This sub-sampling layer produces downsampled version of the filter responses by taking the maximum filter activations from different positions within a specified region of the filter responses. This sub-sampling layer produces invariance to translation and makes the responses tolerant to minor differences of positions of neural activation patterns over the temporal space.

Recurrent layers: The fundamental idea behind the Recurrent Neural Network [14] is that it has atleast one feedback connection, making the activations flow around the loop. This enables these neural networks to process temporal sequences and provide a good generalization over these temporal data. These have been actively used in speech recognition and handwriting recognition applications [3]. The simplest form of these recurrent neural networks are the multi-layer perceptrons with the previous set of hidden activations feeding back into the network along with

the current input sequence. According to the Universal Approximation theorem, any non-linear dynamical system can be approximated to any accuracy by a recurrent neural network.

Dynamically programmed layer: The intuition of having this dynamically programmed layer is due to the fact that it would be easier to find the alignment between the time-series of activations of any pairs of neurons. This intuition is based on the fact that two causally connected neurons tend to have a similar time-series of activations. These can be observed from the Figure 1. Specifically, the role of this layer is used to generate the optimal alignment between the output sequences from the recurrent layers. While the recurrent layers extract the salient temporal patterns from the activations, this dynamically programmed layer aligns the two time-series activations by maximizing the accumulative correlations between the extracted temporal patterns.

4.1 Convolutional Layers

The dataset consists of time-series activation data of about $N = 1000$ neurons. Each neuron n has an activation time series $\mathbf{x} = [x_1, \dots, x_T]$ of approximately $T = 180,000$ steps. To process such enormous amount of data and to exploit its temporal structure, a convolutional layer is first built to process the time-series data. This convolutional layer consists of K convolutional kernels of different sizes, each of which is represented by a vector of filter coefficients $\mathbf{W}^k \in \mathbb{R}^{n^k}$.

Each kernel can be seen as a filter that extracts the salient patterns of temporal structure from the neuron activations, such as activation peaks, periodic changes and zero-crossing frequencies. In contrast to the conventional signal filters, these kernels are integrated into a deep architecture, whose filtering coefficients are decided by maximizing the alignment between activation time-series of connected neurons. We will present the details of learning algorithm later.

Formally, the output feature maps produced by each filter kernel \mathbf{W}^k in the convolutional layer is given by the following equation:

$$h_t^k = \sigma((\mathbf{W}^k * \mathbf{x})_t + b^k)$$

where $*$ is the convolution operator defined as

$$(\mathbf{W}^k * \mathbf{x})_t = \sum_{i=0}^{n^k-1} W_i^k x_{t-i}, \quad (1)$$

n^k is the size of k th filter kernel, $\mathbf{h}^k = [h_1^k, \dots, h_L^k]$ is the feature map output of the k^{th} filter and b^k is the bias.

4.2 Max-Pooling Layer

Following the convolutional layer, a max-pooling layer is used which down-samples the output feature maps by selecting the maximum feature value every P time steps. Formally, the max-pooling is performed by

$$h_t^k \leftarrow \max\{h_{t-P}^k, h_{t-P-1}^k, \dots, h_{(t-1)-P+1}^k\}$$

for $k = 1, \dots, K$ and $t = 1, \dots, \lfloor \frac{T}{P} \rfloor$.

The max-pooling layer has twofold advantages. On one hand, it can generate a more robust sequence, which is invariant to small local translation and warping over time axis by only retaining the most salient local responses. On the other hand, by down-sampling the sequence obtained from the convolutional layer, the max-pooling can produce a more

compact representation of the time-series of activation data. This can remarkably reduce the cost of processing the time-series data in the consecutive layers without loss of too much information.

4.3 Recurrent Layers

The recurrent layer uses a multi-layer perceptron structure with a feedback loop. These recurrent loops exploit the powerful capability of memorizing the temporal context of time-series activations for this layer.

The input into the recurrent layer is the K feature map outputs from the max-pooling layer $\mathbf{h}^k, k = 1, \dots, K$. In other words, at each time step, we have a K dimensional input feature vector $\mathbf{h}_t = [h_t^1, \dots, h_t^K] \in \mathbb{R}^K$. The recurrent layer outputs are represented as $\mathbf{o}_t \in \mathbb{R}^L$ for each time t , and the behavior of a classical recurrent layer can be described by the following equations:

$$\begin{aligned} \mathbf{s}_{t+1} &= f_s(\mathbf{W}_{sx}\mathbf{x}_t + \mathbf{W}_{ss}\mathbf{s}_t) \\ \mathbf{o}_t &= f_o(\mathbf{W}_{os}\mathbf{s}_t) \end{aligned} \quad (2)$$

where \mathbf{s}_t is the hidden state vector of the recurrent layer, and f_s are f_o are nonlinear hidden and output activation functions (e.g., hyperbolic function and sigmoid function), along with three connection matrices \mathbf{W}_{sx} between hidden state and input vector, \mathbf{W}_{ss} between two consecutive hidden state vectors and \mathbf{W}_{os} between the output vector and the hidden state vector.

The hidden states of this dynamical system are the set of values that summarizes all the information about the past behavior of the system that is necessary to provide a unique description of its future behavior. This recurrent layer is learned through back-propagation of error through time, a natural extension of the standard back-propagation algorithm that performs gradient descent on a complete unfolded network.

4.4 Dynamically Programmed Layer

The output sequences from the two recurrent layers are used for dynamical alignment of the activation sequences. This is performed to reveal the causal activation between two brain neurons, and hence their connectivity.

The algorithm developed for this dynamical alignment is based on the following intuition. The brain neurons when activated by another neuron would have a similar response over time as that of the stimulating neuron. So these neurons would have a maximum dot product at that particular time instance. Therefore, based on this intuition, the dynamically programmed layer aligns the two representations of the neurons that would give the maximum dot product accumulated over time instances.

Figure 5 illustrates the structure of this Dynamic Programmed Layer (DPL). Formally, consider two vector sequences \mathbf{o}_t and \mathbf{o}'_t obtained from two lower recurrent layers. We define a sequence of alignment nodes $\{1, \dots, M\}$ in DPL. Then, each DPL node m is connected to a pair of recurrent layer outputs \mathbf{o}_{π_m} and $\mathbf{o}'_{\pi'_m}$ respectively, representing the time- π_m activation of the first neuron causes the time- π'_m activation of the second neuron. Because the activation between neurons is directed, we set causality constraint $\pi_m \leq \pi'_m$ so the first neuron's activation is always no later than the corresponding activation of the second neuron assumed to be activated by the first one.

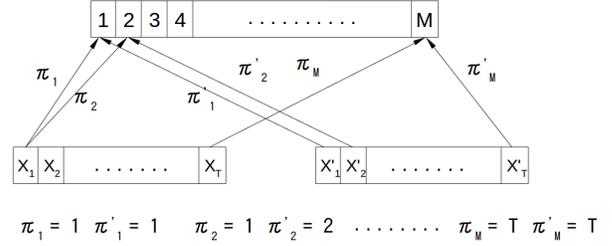


Figure 5: Dynamically Programmed Layer. The hidden state activations of two recurrent neural networks are dynamically aligned, to reveal the inherent structure present in the activations of the two brain neurons.

We use the dot product to measure the correlation between \mathbf{o}_{π_m} and $\mathbf{o}'_{\pi'_m}$. Then the best alignment π and π' can be found by maximizing the following cumulative dot product over time

$$\begin{aligned} \pi^*, \pi'^* &= \arg \max_{\pi, \pi'} \sum_{m=1}^M \langle \mathbf{o}_{\pi_m}, \mathbf{o}'_{\pi'_m} \rangle \\ \text{subject to } &\pi_m \leq \pi'_m, m = 1, \dots, M \\ &\pi_m \leq \pi_{m+1}, \pi'_m \leq \pi'_{m+1}, m = 1, \dots, M-1 \end{aligned} \quad (3)$$

where the first inequality is the causal constraint, and the constraints in the last line are to enforce the alignment for each sequence preserves the temporal order in DPL.

Solving the above maximization problem yields an optimal alignment of activation sequences between two neurons that are causally connected. This DPL model is dynamically decided, where the optimal assignment decided by π and π' between a pair of neurons differs from that between another pair. For this reason, we call this layer dynamically programmed. Moreover, once the output sequences from the lower recurrent layers change, the best assignment will change accordingly. This results in a joint optimization problem of finding the best parameters for the recurrent layer along with the best alignment between the activation sequences.

Inspired by the dynamic programming technique used in dynamic time warping [5], the optimal solution to the above maximization problem can be solved by the following recursive equation:

$$G(t, t') = \langle \mathbf{o}_t, \mathbf{o}'_{t'} \rangle + \max\{G(t, t'), G(t-1, t'), G(t, t'-1)\} \quad (4)$$

where $G(t, t')$ represents the maximum cumulative inner product between $\mathbf{o}_{1:t}, \mathbf{o}'_{1:t'}$ up to t and t' respectively, with initial condition $G(1, 1) = \langle \mathbf{o}_1, \mathbf{o}'_1 \rangle$. Then the maximum value of Eq. (3) is given by $G(M, M)$.

Moreover, to enforce the causality constraint, we set

$$G(t, t') = -\infty, \quad t > t', \quad (5)$$

which avoids reverse-causal connection between neurons, the scenario where the first neuron causes the activation of the second neuron backward in time.

Once the best π and π' are found, the output of each DPL node is given by

$$d_m := \langle \mathbf{o}_{\pi_m}, \mathbf{o}'_{\pi'_m} \rangle.$$

The DPL output is sensitive to the direction of causality between the two brain neurons under consideration. At a given time, it is assumed that only one brain neuron activates the other. At any given time, it is not possible for both the neurons to activate each other. The maximum accumulative dot product is calculated in a way that the direction of causality is preserved while the computation proceeds. The alignment is performed based on the assumption that the first input neuron is causing the activation of the second input neuron. Therefore, this alignment layer would give the best possible value if the above assumption happens to be true. Otherwise, the alignment would have a very poor value.

4.5 Output Prediction Layer

These alignment produced by the DPL is used for making a prediction of connections between the considered pairs of brain neurons.

Formally, we can use an average pooling over all the outputs $\{d_m|m = 1, \dots, M\}$ from the DPL layer

$$\bar{d} = \frac{1}{M} \sum_{m=1}^M d_m \quad (6)$$

to predict whether two brain neurons are causally connected. The larger the mean \bar{d} , the more likely that the first neuron causes the activation of the second neuron, and thus they are causally connected. Specifically, a sigmoid function of \bar{d} is applied to model the probability of the two brain neurons being connected $y = 1$ or not $y = 0$:

$$\Pr(y = 1|\bar{d}) = \text{sigm}(\bar{d}) := \frac{1}{1 + \exp(-\bar{d})} \quad (7)$$

4.6 Feed-Forward Prediction

Stacking the aforementioned layers atop one another, a pair of input activation sequences \mathbf{x} and \mathbf{x}' from two brain neurons will go up through these layers to generate the output label y that predicts two brain neurons are causally connected with a probability $\Pr(y = 1|\bar{d})$. Algorithm 1 summarizes this feed-forward prediction process. In the next section, we will discuss the training algorithm which decides the parameters for such a multi-layered deep network.

5. TRAINING THROUGH BACK PROPAGATION

The output layer gives a sigmoid score based on the pooling over the DPL output that measures the alignment produced between the pair of neurons considered. If the two brain neurons considered are causally connected, then this dynamically programmed layer would output a much large value as the score. On the contrary, if the two neurons that are considered are not causally related, then this dynamically programmed layer would achieve a very poor alignment, thus yielding a poor score of alignment. This alignment score is then used for classifying a presence or absence of a connectome in the brain.

The cost function for this proposed model is a simple cross-entropy loss which is derived based on the principle of maximum likelihood estimation. Suppose we use $y \in \{0, 1\}$ to represent if the two neurons are causally connected or not, then the cross-entropy error made by $\Pr(y|\bar{d})$ is defined

Algorithm 1 Feed-Forward Prediction

Input: A pair of activation time-series \mathbf{x} and \mathbf{x}' from two input brain neurons

Convolutional Layer: $\mathbf{h}^k = \sigma(\mathbf{W}^k * \mathbf{x} + b^k)$ and $\mathbf{h}'^k = \sigma(\mathbf{W}^k * \mathbf{x}' + b^k)$ for $k = 1, \dots, K$;

Recurrent Layer:

$$\mathbf{s}_{t+1} = f_s(\mathbf{W}_{sx}\mathbf{x}_t + \mathbf{W}_{ss}\mathbf{s}_t), \quad \mathbf{o}_t = f_o(\mathbf{W}_{os}\mathbf{s}_t)$$

and

$$\mathbf{s}'_{t+1} = f_s(\mathbf{W}_{sx}\mathbf{x}'_t + \mathbf{W}_{ss}\mathbf{s}'_t), \quad \mathbf{o}'_t = f_o(\mathbf{W}_{os}\mathbf{s}'_t)$$

Dynamically Programmed Layer: Find the optimal connections π and π' by solving Eq. (3);

Output Prediction Layer: $\bar{d} = \frac{1}{M} \sum_{m=1}^M \langle \mathbf{o}_{\pi_m}, \mathbf{o}'_{\pi'_m} \rangle$;

Output: $\Pr(y = 1|\bar{d}) = \frac{1}{1 + \exp(-\bar{d})}$.

as

$$J(\Theta) = -y \log \Pr(y = 1|\bar{d}) - (1 - y) \log \Pr(y = 0|\bar{d}) \quad (8)$$

where the set Θ contains the model parameters, including the connection matrices $\{\mathbf{W}_{sx}, \mathbf{W}_{ss}, \mathbf{W}_{os}\}$ in recurrent layer, and filter coefficients $\{\mathbf{W}^k|k = 1, \dots, K\}$ in convolutional layer.

The training algorithm proceeds by alternating between the feed-forward prediction and back-propagation. In the feed-forward prediction as depicted in Algorithm 1, the optimal alignments π and π' in DPL is decided. Then they are fixed in back propagation algorithm which computes the derivatives to each parameter $\theta \in \Theta$ by chain rule to update the deep network parameters. Therefore, this training algorithm jointly optimizes the alignments between activation sequences in DPL, along with the optimal network parameters.

The back-propagated errors through the network include the following derivatives to each parameter $\theta \in \Theta$

$$\frac{\partial J}{\partial \theta} = \frac{\partial J}{\partial \bar{d}} \frac{\partial \bar{d}}{\partial \theta} \quad (9)$$

where

$$\frac{\partial J}{\partial \bar{d}} = -y(1 - \text{sigm}(\bar{d})) + (1 - y)\text{sigm}(\bar{d}) \quad (10)$$

and

$$\begin{aligned} \frac{\partial \bar{d}}{\partial \theta} &= \frac{1}{M} \sum_{m=1}^M \frac{\partial d_m}{\partial \theta} = \frac{1}{M} \sum_{m=1}^M \frac{\partial}{\partial \theta} \langle \mathbf{o}_{\pi_m}, \mathbf{o}'_{\pi'_m} \rangle \\ &= \frac{1}{M} \sum_{m=1}^M \left(\left\langle \frac{\partial}{\partial \theta} \mathbf{o}_{\pi_m}, \mathbf{o}'_{\pi'_m} \right\rangle + \langle \mathbf{o}_{\pi_m}, \frac{\partial}{\partial \theta} \mathbf{o}'_{\pi'_m} \rangle \right) \end{aligned} \quad (11)$$

where $\frac{\partial}{\partial \theta} \mathbf{o}_{\pi_m}$ and $\frac{\partial}{\partial \theta} \mathbf{o}'_{\pi'_m}$ are the errors back-propagated into the recurrent layers, which can be computed by using Back Propagation Through Time (BPTT) [11] as in classical recurrent neural networks.

It is worth noting that the above equation suggested that the back-propagated errors can be additively combined through the errors back-propagated through each \mathbf{o}_{π_m} and $\mathbf{o}'_{\pi'_m}$. Thus

Algorithm 2 Back Propagation

Input: A training set $\mathcal{D} = \{(\mathbf{x}, \mathbf{x}', y)\}$ of pairs of activation time-series, and step size $\epsilon > 0$;

repeat

Randomly pick up a pair of \mathbf{x} and \mathbf{x}' from \mathcal{D} ;
Feed-forward prediction with \mathbf{x} and \mathbf{x}' ;

for each $m \leftarrow 1 : M$ **do**

Run BPTT to compute $\frac{\partial}{\partial \theta} \mathbf{o}_{\pi_m}$ and $\frac{\partial}{\partial \theta} \mathbf{o}'_{\pi'_m}$;

Update sequentially or in parallel

$$\theta \leftarrow \theta - \epsilon \frac{1}{M} \frac{\partial J}{\partial d} \left\langle \frac{\partial}{\partial \theta} \mathbf{o}_{\pi_m}, \mathbf{o}'_{\pi'_m} \right\rangle$$

and

$$\theta \leftarrow \theta - \epsilon \frac{1}{M} \frac{\partial J}{\partial d} \left\langle \mathbf{o}_{\pi_m}, \frac{\partial}{\partial \theta} \mathbf{o}'_{\pi'_m} \right\rangle;$$

end for

until Stop condition is satisfied;

Output: Model parameters Θ .

we can update the model parameters individually with these back-propagated errors:

$$\theta \leftarrow \theta - \epsilon \frac{1}{M} \frac{\partial J}{\partial d} \left\langle \frac{\partial}{\partial \theta} \mathbf{o}_{\pi_m}, \mathbf{o}'_{\pi'_m} \right\rangle$$

and

$$\theta \leftarrow \theta - \epsilon \frac{1}{M} \frac{\partial J}{\partial d} \left\langle \mathbf{o}_{\pi_m}, \frac{\partial}{\partial \theta} \mathbf{o}'_{\pi'_m} \right\rangle$$

with a positive step size ϵ .

The above update rules can be applied sequentially or in parallel to the model parameters without affecting the final result. We have exploited the parallel implementation of updating the model parameters with back-propagated errors, and the results showed that significant orders of speedup can be achieved with many cores of Graphical Processing Units (GPUs).

Algorithm 2 summarizes the learning algorithm estimating the model parameters. The algorithm consists of both feed-forward prediction and backward error propagation processes. In the feed-forward prediction, the input activation time-series go up through the deep networks, and the best alignment between them are made upon the output sequences from recurrent layers. In the backward error propagation process, the model parameters are adjusted in the gradient direction. This process usually changes the output of recurrent layers, and thus the best alignment obtained by the feed-forward prediction process should also be adjusted. Therefore, the model parameter and the best alignment are alternately updated, and a convergence can be eventually reached since the accumulative inner product is bounded¹ and monotonically nondecreasing in each round over the connected brain neurons.

6. EXPERIMENTS AND RESULTS

In this section, we evaluate the proposed algorithm on the dataset published in the open competition that aims to advance the state of the art in prediction of brain connectomes.

¹This boundedness condition can be satisfied by setting the output activation f_o to hyperbolic function \tanh that bounds the each entry of recurrent layer output vector \mathbf{o}_t on $[-1, 1]$.

6.1 Dataset and Background

The competition was organized for finding a way to understand the structure of the brain. Finding the structure of brain would yield benefits in many fields of research.

The dataset that was released as a part of this competition consisted of six large datasets, each consisting of 1,000 neurons and their corresponding time-series activities. Each of these six datasets has its own degree of connectivity but different levels of clustering coefficients. These degrees of connectivity and clustering coefficients pertaining to each of the dataset was not released as this information can be used to over-fit the model. This implies that the neurons in each of this dataset have different connectivity patterns across the other neurons present in the same dataset. Therefore, a good generalization of the designed algorithm can be obtained when a model is trained over different datasets and tested over a small subset of the examples that have not been used for training.

It is also ensured that the dataset from which the testing set is obtained is not used for training. By evaluating the performance over such a scenario would shed light on the level of generalization achieved by designed algorithm because it is being trained over different sets of training data and being tested over a completely different set of data - since, the clustering coefficients would be different for different datasets.

6.2 Experiment Setting

For the sake of fair comparison, we follow the similar competition protocol performed by the existing methods [12]. These existing methods are trained over the same training dataset and tested over the same testing set used for evaluating our model. Their corresponding performances are compared in this section.

The overall performance comparison can be observed from the Area under the Receiver Operating Characteristic (AUROC) curve as shown in Figures 6(a)-6(f). It corresponds to the area under the curve obtained by plotting the true positive ratio against the false positive ratio by varying the prediction values to determine the classification results. The AUROC is calculated using the trapezoid method. In experiments, neuron pairs are ranked by the likelihood that they are causally connected with each other. The more likely a pair of causally connected neuron is ranked higher than a disconnected or non-causally connected pair, the larger the AUROC is.

6.3 Model Architecture

The designed deep architecture consists of three filters in the convolutional layer. Each of these filters created a feature map through repeatedly convolution of the one dimensional input signal with the linear filter, adding a bias term and then applying a non-linear sigmoid activation function. Each filter used in the convolutional layer had a dimension of 5×1 . Each feature map is then max-pooled over 47×1 contiguous regions. Table 1 summarizes the details of convolutional layer and max-pooling layer.

The resulting signals are then processed by the recurrent layer. As shown in Table 2, the number of hidden units in the recurrent neural network is set to five, where each hidden unit outputs a sequence of hidden states by applying a hyperbolic activation function. The output sequences obtained from the recurrent layer are the inputs to the dynamically

Layer	Input	Filter Size	Output
Activation sequence	-	-	179497 x 1
Convolution layer	3	5 x 1	179493 x 3
Max-Pooling layer	3	47 x 1	3819 x 3

Table 1: Details of convolutional layer and max-pooling layer

	Units	Dimensions
Input	3	3819 x 3
Output(Hidden)	5	3819 x 5

Table 2: Details of recurrent layer

programmed layer. The number of hidden nodes in DPL varies between different pairs of sequences to be aligned by this layer. Finally, an averaging-pooling layer was used to generate the output prediction on the brain neuron connectivity.

6.4 Compared Algorithms

We compare the proposed architecture with the following algorithms which have achieved the state-of-the-art performance in the competition.

- Cross Correlation [15]: Computes the connectivity scores based on simple cross-correlation over the time-series neuron activation data.
- Granger Causality [2]: This is a statistical hypothesis test which computes a score based on whether one time-series is useful in predicting the other time-series data.
- Information Gain - Entropy and Gini Index [15]: These method treats all the points of the time-series activation data as independent and identical distributions and computes a connectivity score based on the information gain over these distributions.
- Trained Classifier [15]: This method computes partial correlation coefficient based on some basic features computed over the time-series data of neurons.
- Random Score [15]: Generates a random score of connectivity. This method is a simple baseline method used for comparison.
- Generalized Transfer Entropy (GTE) [13]: This is a non-parametric statistic which measures the amount of directed transfer of information between two random processes. This is also one of the state of the art method used for predicting the connectomes of the brain neurons.
- Partial Correlation Statistics [16]: This method consists of several lower order high pass and low pass filters applied over the time-series data. These filtered time-series signals are used for calculating a connectivity score based on partial correlation coefficient. This is the state of the art method in the open competition.

6.5 Results

Method	AUROC
Random Score	0.4034
Generalized Transfer Entropy (GTE)	0.5472
Information Gain - Gini Index	0.5856
Information Gain - Entropy	0.6057
Granger Causality	0.4736
Trained Predictor	0.5985
Cross Correlation	0.5572
Partial Correlation Statistics	0.7698
Our Method	0.8309

Table 3: Comparison of average AUROCs over six test sets achieved by different algorithms on predicting causal connections. The best performance is highlighted in bold.

Figures 6(a)-6(f) compare the AUROCs obtained by different algorithms over six test sets. From the Figures 6(a)-6(f), it can be clearly observed that our proposed method surpasses all the existing methods in the evaluation scenario as described above. Some of the existing methods are unsupervised learning methods - they do not make use of the ground truth provided with the dataset during the training process. Hence, these methods are directly evaluated over the test set.

The proposed model identifies the direction of causality present in the brain neurons. The basic idea is to check whether the the time-series variables are symptomatic to any underlying process. The direction of causality of brain neurons can be better predicted based on the past values of either of these neurons. For example, if neuron A is the cause for activation of neuron B then the activations of neuron B at any point in time can be better predicted based on the past activations of neuron A. If these two neurons are not causal, then it would be impossible to make the above prediction. Therefore, in this case, the dynamically programmed layer would have a poor value due to the mis-alignment of these two neurons A and B, i.e. unable to predict the future values of one neuron based on the past values of the other neuron.

It is a well known fact that causal relationships can be confounded. The fact that two neurons A and B are correlated does not imply that there is a physical connection between these two neurons. There might be a third neuron that is responsible for these causes. This problem is difficult to be solved and hence, this competition used the AUROC as the evaluation metric. It stresses over the importance of revealing the presence of a connection between the two brain neurons rather than the absence of connection between them.

Our method makes an assumption that the first neuron would be the cause for activation of the second neuron. Sometimes, the second neuron might be the cause for activation of the first neuron. This can be detected by swapping the order of brain neurons. In order to accommodate this scenario, both the training and testing datasets have been augmented by forming a set of ordered pairs of neurons. Therefore, our proposed algorithm would be able to predict the causal directions with a better accuracy between any pair of neurons as long as they have been observed in the dataset.

This improvement of accuracy of prediction of brain con-

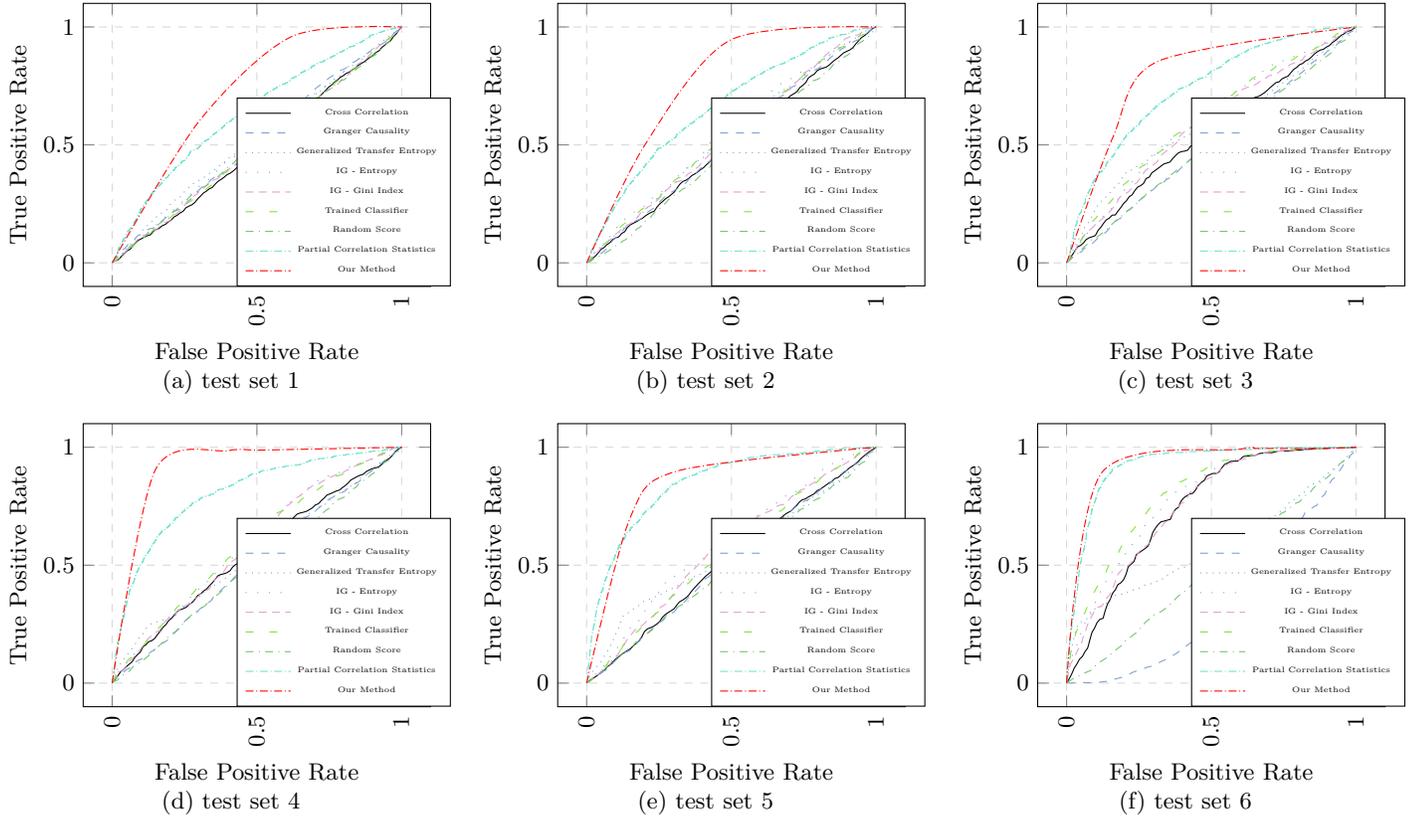


Figure 6: AUROC Comparison on six test Sets. The figure is best viewed in color.

nectomes, with the direction of causality under consideration, is achieved mainly because of these non-linear representations of the time-series of activations of these brain neurons which are obtained through the combination of convolutional and recurrent layers. This compressed and generalized representations of these time-series activations of the brain neurons are then used for obtaining an alignment score based on the dynamically programmed layer.

This exploits the fact that two causally connected neurons would have similar time-series of activations, which are subject to background noises, local warping and translation. Through the back-propagation training process, the proposed deep architecture could extract invariant patterns to these changes, resulting in the robust alignment between different time series of brain neuron activations.

6.6 Learned Convolutional Filters

Finally, we look into some details about the obtained deep architecture. After training the model over the training sets, three convolutional filters were obtained as illustrated in Figure 8. Based on this figure, it could be clearly observed that these convolutional filters that have been trained through back-propagation form a band pass filter (filter - 1), band reject filter (filter - 2) and a selective amplification filter (filter - 3).

These three filters extract different patterns of local activation activity from a sliding window of size 5 superimposed over the input activation time series, yielding a variety of complementary output features for further processing

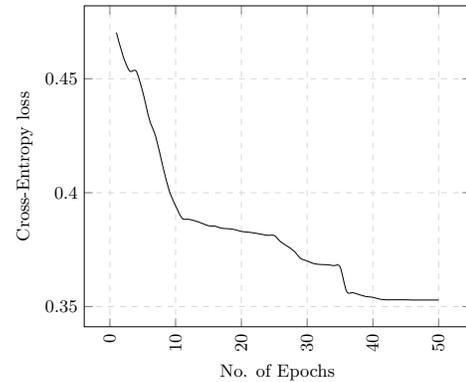


Figure 7: The learning curve illustrates the training errors versus the number of echoes in the training process.

through the deep networks.

7. CONCLUSION

We proposed a novel deep learning algorithm for predicting the brain connectomes based on the time-series activations of brain neurons. The proposed architecture is both scalable and easy to train in terms of resources and time. The improvement obtained in terms of prediction accuracy

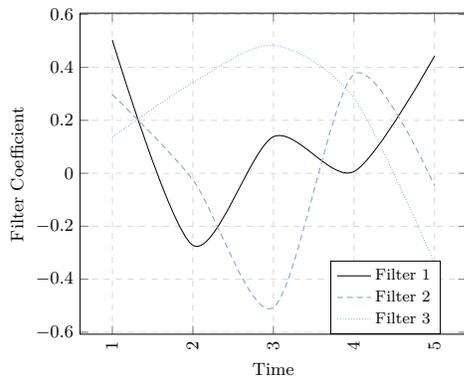


Figure 8: The three convolutional filters obtained through the training of the proposed deep architecture.

is critically due to the exploitation of the deep architecture, which jointly extracts sequences of salient patterns of activation and aligns them to predict the causal connectivity between brain neurons. Experiment result on an open competition dataset shows that the proposed method outperforms the state-of-the-art algorithms.

8. ACKNOWLEDGEMENTS

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

9. REFERENCES

- [1] D. Eytan and S. Marom. Dynamics and effective topology underlying synchronization in networks of cortical neurons. *The Journal of neuroscience*, 26(33):8465–8476, 2006.
- [2] C. W. Granger. Some recent development in a concept of causality. *Journal of econometrics*, 39(1):199–211, 1988.
- [3] A. Graves, A.-R. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6645–6649. IEEE, 2013.
- [4] C. Grienberger and A. Konnerth. Imaging calcium in neurons. *Neuron*, 73(5):862–885, 2012.
- [5] E. Keogh and C. A. Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and Information Systems*, 7(3):358–386, 2005.
- [6] C. Learn. For Imaging to Connectivity. <http://connectomics.chalearn.org/help/setting>, 2014. [Online; accessed 02-18-2015].
- [7] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [8] J. Livet, T. A. Weissman, H. Kang, R. W. Draft, J. Lu, R. A. Bennis, J. R. Sanes, and J. W. Lichtman. Transgenic strategies for combinatorial expression of fluorescent proteins in the nervous system. *Nature*, 450(7166):56–62, 2007.
- [9] H. Madsen and P. Thyregod. *Introduction to general and generalized linear models*. CRC Press, 2011.
- [10] Y. Mishchenko, J. T. Vogelstein, L. Paninski, et al. A bayesian approach for inferring neuronal connectivity from calcium fluorescent imaging data. *The Annals of Applied Statistics*, 5(2B):1229–1261, 2011.
- [11] M. C. Mozer. A focused back-propagation algorithm for temporal pattern recognition. *Complex systems*, 3(4):349–381, 1989.
- [12] J. G. Orlandi, B. Ray, D. Battaglia, I. Guyon, V. Lemaire, M. Saeed, J. Soriano, A. Statnikov, and O. Stetter. First connectomics challenge: From imaging to connectivity.
- [13] J. G. Orlandi, O. Stetter, J. Soriano, T. Geisel, and D. Battaglia. Transfer entropy reconstruction and labeling of neuronal connections from simulated calcium imaging. *PLoS one*, 9(6):e98842, 2014.
- [14] M. Schuster and K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.
- [15] O. Stetter, D. Battaglia, J. Soriano, and T. Geisel. Model-free reconstruction of excitatory neuronal connectivity from calcium imaging signals. *PLoS computational biology*, 8(8):e1002653, 2012.
- [16] A. Sutura, A. Joly, V. François-Lavet, Z. A. Qiu, G. Louppe, D. Ernst, and P. Geurts. Simple connectome inference from partial correlation statistics in calcium imaging. *arXiv preprint arXiv:1406.7865*, 2014.
- [17] W. Truccolo, U. T. Eden, M. R. Fellows, J. P. Donoghue, and E. N. Brown. A point process framework for relating neural spiking activity to spiking history, neural ensemble, and extrinsic covariate effects. *Journal of neurophysiology*, 93(2):1074–1089, 2005.
- [18] D. C. Van Essen, S. M. Smith, D. M. Barch, T. E. Behrens, E. Yacoub, K. Ugurbil, W.-M. H. Consortium, et al. The wu-minn human connectome project: an overview. *Neuroimage*, 80:62–79, 2013.
- [19] J. T. Vogelstein, B. O. Watson, A. M. Packer, R. Yuste, B. Jedynak, and L. Paninski. Spike inference from calcium imaging using sequential monte carlo methods. *Biophysical journal*, 97(2):636–655, 2009.
- [20] J. White, E. Southgate, J. Thomson, and S. Brenner. The structure of the nervous system of the nematode *caenorhabditis elegans*: the mind of a worm. *Phil. Trans. R. Soc. Lond.*, 314:1–340, 1986.