CONNECTOME PRIORS IN DEEP NEURAL NETWORKS TO PREDICT AUTISM

Colin J. Brown, Jeremy Kawahara*, and Ghassan Hamarneh*

Medical Image Analysis Lab, School of Computing Science, Simon Fraser University, Canada

ABSTRACT

We propose a novel element-wise layer for deep neural networks that incorporates general priors designed for connectomes. In contrast to regular images, connectomes, expressed as adjacency matrices, are composed of elements that capture a relationship between two brain regions. As each element in the connectome has an anatomical meaning that is consistent across samples, prior knowledge about the structure of the connectome can be encoded and used to regularize learning based approaches. Thus in this work, we introduce a novel trainable element-wise layer for deep neural networks, with data-dependent anatomically-informed prior regularization terms designed for connectomes. We validate our approach on 1013 functional connectomes from the autism brain imaging data exchange (ABIDE) dataset, and show that our proposed layer and regularization terms improves the accuracy of predicting patients with autism spectrum disorder from controls within a deep learning framework.

1. INTRODUCTION

Stacked learning models like artificial neural networks (ANN) have been successful on a wide variety of prediction tasks for which large training sets are available [1]. However, for domains like human connectome data, database sizes currently range from only dozens to hundreds of samples and additional data is expensive and time consuming to acquire [2]. ANN models often contain thousands to millions of parameters making overfitting to the training set a major challenge when training these models on connectome data. Nevertheless, the relationships between connectome features and clinical presentations of brain disorders (e.g., autism spectrum disorder, ASD) are often complex, and may be better modelled by stacked, non-linear models. One way to address the problem of limited data is through incorporating priors and regularizing the models.

There are many general regularization techniques for ANNs. For instance, dropout involves random deactivation of hidden units during training [3]. Salimans et al. normalized weights in each layer by their magnitudes to improve conditioning during optimization [4]. Both weight normalization and dropout do not require modifying the model's architecture or substantial changes to the loss function, and we leverage both methods in this work.

Other methods have explored adding additional loss terms to regularize the model weights. Collins et al. imposed sparsity-encouraging $L1$ and $L0$ norm regularization terms across all weights in a CNN [5]. They found that using these priors enabled training a sparse model with reduced memory requirements, while maintaining similar predictive performance as a non-sparse model. We instead initially design a model with reduced parameters, and impose anatomical priors on specific layers. Kulkarni et al. also imposed $L1$ regularization, but only on diagonal matrices placed between pairs of fully connected layers, designed to enabled sparse weighting of individual responses [6]. In contrast, our proposed element-wise layers are implemented as separate (non-diagonal) filters with non-linear activation functions, regularized by priors designed for brain network data that Brown et al. previously showed to successfully regularize linear prediction models for connectome data [7].

More generally, a variety of works have tackled the problem of prediction using connectome data [2]. For instance, Abraham et al. studied the problem of predicting ASD from controls on the ABIDE dataset [8]. They explored a variety of different processing pipelines and prediction models.

In terms of deep neural networks for connectomes, Kawahara et al. proposed a CNN with convolutional filters that consider the topology of connectome data [9]. Specifically, they proposed trainable edge-to-edge, edge-to-node, and node-to-graph filters that summarize information across edges, nodes, and the entire graph, respectively.

In this paper, we introduce a novel element-wise layer, regularized with data-driven structural priors designed for brain networks. We combine this layer with convolutional filters designed for connectome data [9], and validate our approach on the task of classifying ASD subjects from controls on a dataset of 1013 functional connectomes from the autism brain imaging data exchange (ABIDE). We show that the addition of the proposed regularized element-wise layer to a deep learning framework improves prediction accuracy.

2. METHOD AND MATERIALS

A patient's connectome is represented by a symmetric adjacency matrix, X, (Fig. 1 *left*) which describes a graph

*Joint first authors

composed of *nodes* representing brain regions, and *edges* (elements in the adjacency matrix) that capture the relationships between brain regions (e.g., connectivity/functional correlation). Given X, we desire a model that predicts a corresponding label, y (e.g., ASD, clinical score). Here we describe our proposed model and dataset used for evaluation.

Element-wise Layer The proposed element-wise layer is a trainable layer that multiplies a set of learned weights to its inputs via the Hadamard product (i.e., element-wise multiplication). When applied directly to the connectome input data, this scales the magnitudes of edge strengths between regions in the brain, with the goal of learning weights specific to each edge in the brain network.

Let $A^{\ell,m} \in \mathbb{R}^{I^\ell \times J^\ell}$ represent the responses in layer ℓ of the network ($A^0 = X$), where $I^\ell \times J^\ell$ is the size of the spatial dimensions, and m indicates the feature maps/channels. We define the element-wise layer with an activation function as:

$$A^{\ell+1,n} = \tanh\left(\sum_{m=1}^{M^\ell}(W^{\ell,m,n} \odot A^{\ell,m})\right) \quad (1)$$

where $W^{\ell,m,n} \in \mathbb{R}^{I^\ell \times J^\ell}$ are the trainable weights of the element-wise layer connecting the mth feature map in layer ℓ to the nth feature map in layer $\ell+1$; $\odot$ represents the element-wise (Hadamard) product; M^ℓ are the number of feature maps/channels in layer ℓ; and $\tanh(\cdot)$ indicates an *element-wise* hyperbolic tangent function used as the activation function. We define Eq. 1 as a sum over the M^ℓ channels in layer ℓ, but depending on application, this sum could be omitted to maintain the channel dimensions.

We note that our element-wise layer differs from a 1×1 convolutional filter [10] as the element-wise layer learns a unique weight for each element in its input, rather than convolving a single weight over the entire feature map. A unique weight for each element is desirable given that the connectome stores its elements in a fixed ordering (i.e., an element is defined as an edge connected to specific brain regions) as opposed to images where pixel orderings are not inherently meaningful. Note that the element-wise layer does not sum over the spatial dimensions, resulting in the spatial dimensions of $A^{\ell+1}$ matching those of A^ℓ. Maintaining the spatial dimensions of the early layers of a deep neural network is desirable as it allows for stacking subsequent layers that are designed to leverage the full topology of the connectome data (e.g., edge-to-node and node-to-graph layers [9]). Also note that we omit a bias term, as it would result in non-zero responses to elements where the input is zero; an element with zero in a connectome indicates no connection between two brain regions. The hyperbolic tangent function was chosen as it produces a zero response when the connectome elements are zero, and scales the responses between -1 and 1 for subsequent network layers.

Model Loss Function Here we describe our modified loss function with the additional regularized element-wise layers. While $L2$ (Euclidean norm) regularization is commonly imposed on weights throughout deep networks to help prevent individual weights from becoming very large (e.g., [9, 11]), we apply different regularization to the element-wise layers, $L_E \subseteq L$, where $L = \{0, \cdots, |L|-1\}$ is the set of all layers in a model. Given a dataset D=$\{\mathbf{X}, \mathbf{y}\}$, let $\mathbf{X} \in \mathbb{R}^{H \times I^0 \times J^0}$ represent a set of H input samples (e.g., connectomes or images), $\mathbf{y}$ represent the associated H labels of those samples, and W represents the model's trainable parameters. Our proposed training loss function to minimize takes the form:

$$\mathcal{L}(h(\mathbf{X};\mathbf{W}),\mathbf{y}) = \lambda_{L2} \sum_{\ell \in L \setminus L_E} ||W^\ell||_2 + \sum_{\ell \in L_E} \mathcal{R}^\ell(D, W^\ell), \quad (2)$$

where $\mathcal{L}(\cdot)$ represents the categorical cross-entropy loss between the output of the neural network $h(\mathbf{X};\mathbf{W})$ being trained and the true labels $\mathbf{y}$; and $\mathcal{R}^\ell(\cdot)$ is a special regularization function for an element-wise layer at layer ℓ. $\mathcal{R}^\ell(\cdot)$ is indexed by layer since we may wish to regularize different element-wise layers in different ways, depending on prior knowledge about the features at that layer. Note that for all non element-wise layers, a standard $L2$ norm is used.

In this paper, we assume the element-wise layer is only applied to the input of the ANN (i.e., $L_E = \{0\}$), but this layer type can be inserted after any layer, especially after layers for which the responses can be interpreted semantically and for which an informed prior can be imposed.

Element-wise Data-dependent Regularization If there is a semantic interpretation of the learned features/weights, domain specific prior knowledge can be used to regularize the weights of the first, and possibly subsequent layers. Here, the application is brain network data, for which a variety of anatomically informed structural priors have been proposed. In particular, Brown et al. suggested that a subnetwork of predictive edges should be reasonably sparse, well connected, and anatomically plausible. Thus, they imposed $L1$, and novel connectivity and backbone network priors on their learned weights [7]. Specifically, they defined a network backbone prior matrix B_D as a binary diagonal matrix that penalizes assigning a weight to those elements that were found to have a low signal to noise ratio in the training data. They defined a connectivity prior matrix C as composed of 0 and -1's, where a -1 is assigned to edges that share a common node, and thus incentivizes patterns in the weights across edges that share common nodes. For our element-wise layer acting directly on the connectome data input, these priors are incorporated into our regularization function $\mathcal{R}(D, W)$:

$$\lambda_{L1}||W||_1 + \lambda_B||B_D \odot W||_2 + \lambda_C\left(\phi(W)^T C \phi(W)\right) \quad (3)$$

where each term is weighted by hyper-parameters, λ_{L1}, λ_B, and λ_C; and $||W||_1$ is an $L1$ norm sparsity regularization term. To create the backbone B_D and connectivity C matrices, we follow the criteria defined in [7], but modify this approach originally designed to work for vectors to work with matrices. Specifically, we form B_D as a binary matrix the same size as W, using the training data D to determine elements with low signal to noise ratio (Fig. 1 *middle*). We note how the commonly used $L2$ regularization is a special case of when B_D is composed of entirely ones (i.e., uniform prior).

Also differing from [7], we apply the hyperbolic tangent function, element-wise, to each of the squared vectorized weights, $\phi(W) = \tanh(W^2(:))$. We square elements of W to account for negative weights, vectorize W as $W(:)$ to allow for matrix multiplication with C, and apply $\tanh(\cdot)$ to bound the contribution to the loss function of any individual weight to between 0 and 1, inclusive, reducing large incentives for weights with large magnitudes.

Architecture The architecture of our deep neural network is based on BrainNetCNN [9], with our proposed element-wise layer as the first layer of this network. To further reduce the number of trainable parameters, we use 6 feature maps (as opposed to 32) in each of the edge-to-edge (E2E) and edge-to-node (E2N) layers, and use only a single fully-connected node-to-graph (N2G) layer in the final output layer. All layers use leaky-rectified linear units (leaky value of 0.33), except for the element-wise layer which uses the hyperbolic tangent function, and the final output layer which uses a softmax activation function. Dropout with a rate of 0.5 was applied to E2E and E2N, and a rate of 0.6 to the element-wise layer.

ABIDE Dataset We applied our proposed method to the task of predicting ASD using functional connectome data. The dataset comprises 1013 resting-state functional connectomes derived from the ABIDE dataset and accessed through the Preprocessed Connectomes Project website [12]. Functional MR images were preprocessed via the connectome computation system [13] with band-pass filtering and global signal regression and then registered to the automated anatomical labeling (AAL) atlas [14]. Time series were averaged within 90 atlas regions and functional connectivity between each pair of regions was computed using Pearson's correlation to produce an $I^0 \times J^0 = 90 \times 90$ weighted connectome adjacency matrix for each scan (Fig.1 *left*). 99 of the original 1112 fMRIs were removed due to erroneous time series data, leaving a total of 1013 functional connectomes. The subject in each scan was labelled either as control (n=539) or ASD (n=474) (i.e., $\mathbf{y} \in \{0,1\}^{N\times 1}$).

3. RESULTS

A variety of different model architectures for prediction of ASD were validated on the ABIDE data (Table 1). Each architecture was assessed using 5-fold cross validation. Within each fold, 20% of the training data was used as a validation set to decide the optimal number of training iterations. Three models were trained on the remaining 80% of the training data and the predicted class probabilities on the test data were averaged to form the final prediction. For all architectures and experiments, we weighted the terms ($\lambda_{L1} = 0.0002, \lambda_{L2} = 0.0001, \lambda_B = 0.0005, \lambda_C = 5E$-06) the same, except where otherwise noted. To optimize Eq. 2, we use the Adam optimizer with weight normalization [4]. All weights used data-dependent initializations [4] except for the element-wise layer, which were sampled from a uniform random distribution between ± 0.5.

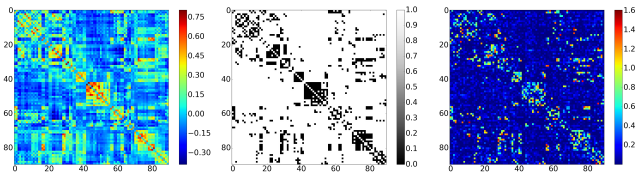

Fig. 1. (*Left*) The ABIDE functional connectomes averaged over all scans. (*Middle*) Our data-specific regularization matrix, B_D. (*Right*) The *absolute value* of the weights of a trained regularized Elmwise layer. As desired, the learned weights are visibly influenced by the B_D matrix.

We start with baseline and competing approaches. For our first experiment (*row a*), we train a linear model on the vectorized upper-triangular values of the symmetric connectome with $L2$ regularization. Using this same architecture, in *row b* we add the regularization from Eq. 3, set λ_{L1}=0.001 and increase the other default λ values by a factor of 10 as [7] reported improvements with sparse models. We then change model architectures (*row c*) to use the edge-to-edge (E2E) and edge-to-node (E2N) layers [9] with $L2$ regularization, taking as input the full symmetric matrix. This yields similar accuracy to the regularized linear model (*row b*). In *row d* we add an additional E2E layer using the same activation function and dropout rate as the element-wise layer, which does not improve results and indicates that simply adding layers and parameters does not always improve performance.

In order to test if our element-wise layer (Elmwise) aids the model, we add the Elmwise layer (Eq. 1) to the start of the model from *row c*. This improves overall accuracy (*row e*) when compared to not using the Elmwise layer (*row c*), and when using a model with the same number of layers and a similar number of parameters (*row d*).

We add in the connectivity term C from Eq. 3 which results in a small improvement (*row f*) to accuracy when combined with $L2$ regularization. We then only use the B_D regularization, which results in the highest accuracy combined with the proposed Elmwise layer (*row g*), outperforming the standard $L2$ regularization (for a fairer comparison between *row e* and row g in *row e* we set the Elmwise $\lambda_{L2} = \lambda_B$).

	Method	# Pars	Elmwise	E2E	E2N	Dense	L2	L1	B_D	C	ACC.	SE.	SP.
(a)		8.0K				1	✓				0.650	0.617	0.681
(b)	[7]	8.0K				1		✓	✓	✓	0.661	0.619	0.702
(c)	[9]	8.6K		1	1	1	✓				0.659	0.625	0.692
(d)	[9]	15.1K		2	1	1	✓				0.657	0.609	**0.704**
(e)	*ours*	16.8K	1	1	1	1	✓				0.675	0.680	0.671
(f)	*ours*	16.8K	1	1	1	1	✓			✓	0.681	0.657	**0.704**
(g)	*ours*	16.8K	1	1	1	1			✓		**0.687**	**0.692**	0.683
(h)	*ours*	16.8K	1	1	1	1		✓	✓	✓	0.681	0.677	0.685

Table 1: Comparison of model architectures to predict ASD. *# Pars* indicates the number of model parameters. B_D and C indicate the backbone and connectivity priors. *ACC, SE, SP* indicate accuracy, sensitivity, and specificity, respectively.

We note that this model that uses the Elmwise layer with the data-specific prior exhibits less overfitting (training accuracy = 82% at final epoch) compared to *row d* (training accuracy = 99%), partly due to dropout over fewer feature maps.

Combining the $L1$, C, and B_D terms did not yield improvements in our tests (*row h*). While we found that a relatively low weight for the B_D term λ_B=0.0005 resulted in improved accuracy, in order to better visualize the learned regularized weights of the Elmwise layer, we repeat the experiment of *row h*, except with increased B_D regularization λ_B=0.003. While this increased regularization results in a modest decrease to accuracy (0.679), it enables the influence of the priors to be visualized more clearly (Fig. 1 *right*).

Our results are comparable to the work of [8], which reported a classification accuracy of 0.678 over the same dataset using their top performing support vector machine trained on functional connectomes derived from dictionary learning based parcellations of the scans. Recent work by Subbaraju et al. report an accuracy of 0.773 across the entire ABIDE dataset using a support vector machine trained on features extracted from time-series data, that were projected to separate classes using a spatial filter [15]. We note that the differences in connectome preprocessing pipelines use as well as the differences in experimental setup (e.g., because of a different fold size, our model trains on substantially less training data than models in [8] and [15]) makes direct comparisons to these results unclear, and that our main contribution is equipping a deep-learning framework with the proposed element-wise layer and data-dependent regularization, which showed *relative* performance improvements.

4. CONCLUSIONS

We proposed a novel element-wise layer for neural networks that incorporated connectome and data-specific priors to learn an edge-specific weighting. Integrating this layer within a deep network framework designed for brain network data improved ASD prediction accuracy when tested over 1013 functional connectomes from the ABIDE dataset.

5. REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] C. J. Brown and G. Hamarneh, "Machine Learning on Human Connectome Data from MRI: A Survey," *arXiv*, 2016.

[3] G. E. Hinton *et al.*, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[4] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," in *NIPS*, 2016.

[5] M. D. Collins and P. Kohli, "Memory bounded deep convolutional networks," *arXiv preprint arXiv:1412.1442*, 2014.

[6] P. Kulkarni *et al.*, "Learning the Structure of Deep Architectures Using L1 Regularization," in *BMVC*. BMVA Press, 2015.

[7] C. J. Brown *et al.*, "Predictive Subnetwork Extraction with Structural Priors for Infant Connectomes," in *MICCAI*, S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal, and W. Wells, Eds. Springer International Publishing, 2016, pp. 175–183.

[8] A. Abraham *et al.*, "Deriving reproducible biomarkers from multi-site resting-state data : An Autism-based example," *NeuroImage*, vol. 147, pp. 736–745, 2016.

[9] J. Kawahara *et al.*, "BrainNetCNN: Convolutional neural networks for brain networks; towards predicting neurodevelopment," *NeuroImage*, vol. 146, pp. 1038–1049, 2017.

[10] M. Lin, Q. Chen, and S. Yan, "Network in network," in *ICLR*, 2013.

[11] H. Li *et al.*, "Identification of faulty DTI-based sub-networks in autism using network regularized SVM," *ISBI*, vol. 6, pp. 550–553, 2012.

[12] C. Craddock *et al.*, "The neuro bureau preprocessing initiative: open sharing of preprocessed neuroimaging data and derivatives," *Frontiers in Neuroinformatics*, no. 41, 2013, http://preprocessed-connectomes-project.org/.

[13] T. Xu, Z. Yang, L. Jiang, X.-X. Xing, and X.-N. Zuo, "A connectome computation system for discovery science of brain," *Science Bulletin*, vol. 60, no. 1, pp. 86–95, 2015.

[14] Tzourio-Mazoyer *et al.*, "Automated anatomical labeling of activations in SPM using a macroscopic anatomical parcellation of the MNI MRI single-subject brain," *Neuroimage*, vol. 15, no. 1, pp. 273–289, 2002.

[15] V. Subbaraju *et al.*, "Identifying differences in brain activities and an accurate detection of autism spectrum disorder using resting state functional-magnetic resonance imaging : A spatial filtering approach," *Med Image Anal*, vol. 35, pp. 375–389, 2017.