# Dissecting DNN with Co-Adaptive Modules

Jialin Lu

## ABSTRACT

We hypothesize that like human brain, Deep Neural Networks are also organized in distinct modules for different tasks.

In this paper we seek the dissection of a hidden layer in a pre-trained deep neural netwrok into disjoint groups of hidden neurons that are functionally related for predicting certain sets of similar data samples, i.e. samples with similar feature patterns. We treat such groups of neurons, which we call *co-adaptive module*s, as the basic functional unit in DNN and identify those modules using a biclustering approach based on the attribution score of individual hidden neurons in a data-driven manner.

We find that, unsurprisingly, the identified modules are highly sparse, i.e., only a small subset of neurons are important for predicting a set of similar samples and, while we do not use any label supervision, samples corresponding to the same module show coherent feature patterns. We also show in experimental results that found modules perform a critical role in discriminating data samples. We demonstrate that these modules learn suprisingly rich representations and are able to detecting feature patterns on several pre-trained models, including visual classification as well as a biological application case where different variations of DNA motifs are learned by these modules.

We suggest that more interpretable result for DNN could be obtained by taking a module-based perspective and demonstrate co-adaptive modules learn to detect very complex data patterns.

## CCS CONCEPTS

• **Computing methodologies** → **Computer graphics**;

## KEYWORDS

research

**ACM Reference format:**

Jialin Lu. 2017. Dissecting DNN with Co-Adaptive Modules. In *Proceedings of ACM Conference, Washington, DC, USA, July 2017 (Conference'17),* 8 pages.

https://doi.org/

*Conference'17, July 2017, Washington, DC, USA*
© 2017 Copyright held by the owner/author(s). Publication rights licensed to Association for Computing Machinery.
This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in *Proceedings of ACM Conference, July 2017.*

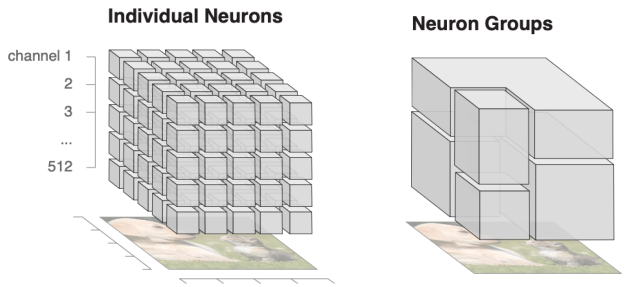
## 1 INTRODUCTION

We can often gain insight into the overall structure of a complex system by grouping elements into clusters based on their correlated properties and then conducting analysis on these clusters. Many types of biological networks have been modeled in terms of interactions between a relatively small set of "modules", such as protein–protein interaction, pollination networks,metabolic networks and food web and based on these modules as abstraction comes out many scientific breakthroughs. It is often that when we reduce the noisy complex basic units' into more abstract modules, we could describe and interprete the system in a more human-scale fashion. From this philosophy, we hypothesize that deep neural networks (DNN) could also be interpreted by the concept of modulation for different tasks.

While it is widely believed that the secret ingredient of DNN's success lies in the huge capacity of hidden neurons, most of research on this stream choose to dig into one-sample level and ignore the complex collective behavior of hidden neurons, thus not able to reveal what complex patterns are learned though these hidden layers on different samples. Previous study, either trying to visualize the detected feature or aiming at attributing an importance score to a input feature or hidden neuron in DNNs, will only make sense under the assumption that a single neuron, or a channel of neurons are responsible for detecting a certain feature, while this is not likely to be true since single neuron or channel-based visualization and attribution do not give satisfactory interpretations. A pracitcal drawback of those methods is that, interpretations would only be truly interpretable in the first convolutional layer, while in the deeper layers, neuron or channel based visualization typically give fancy but nonsense results.

We argue that more interpretable results could be obtained by identifying groups of hidden neurons that *co-adapts* together for discriminating certain feature patterns. The idea of co-adaptive neurons are first introduced by [Hinton et al. 2012], where the author claim that several hidden neurons *co-adapts* together as a module contribute to the making a prediction, where one neuron may not be able to work in the absence of other co-adaptive neurons. This phenomenon is in fact only observed though mind experiments and the concept of co-adaption is not defined formally, and ,despite some biological inspirations, how to identify such co-adaptive neurons largely remain unsolved. However this phenomenon itself inspires the invention of dropout [Hinton et al. 2012; Srivastava et al. 2014] which is arguably the robustest regularization technique for DNN.

In this paper, we hypothesize the existence of co-adaptive module, where multiple neurons work together with similar behaviors to detect certain feature patterns, and propose a biclustering approach based on attritbution score of hidden neurons for different samples. Basically, we first construct a neuron-sample matrix $M$ where each




(a) All neurons in a layer (b) grouped

**Figure 1: *Approach of finding co-adaption module***
**All neurons in a hidden layer are dissected into groups. Each group is a so-called co-adaption module where it detects certain feature patterns. Note that neurons in a co-adaptive module does not have to be spatially contiguous.**

entry $m_{ij}$ indicates how much a neuron $i$ in a given layer contributes to the prediction of a data sample $j$. $m_{ij}$ could be computed using several existing attribution methods such as gradient times input, guided propogation, DeepLIFT score or total conductance. Based on this matrix $M$, we use a spectral coclustering approach which simultaneously cluster rows and columns with higher values of data matrix $M$, to identify disinguishable groups of neurons that contributes unusually high value for predicting certain set of samples.

We find that these identified groups, which we call co-adaptive modules, dissect a hidden layer into functionally coherent parcellations and play a important role in discriminating data samples. We first observed that, unsurprisingly, the neuron-sample matrix is highly sparse, i.e., only a small subset of neurons are important for predicting a set of similar samples. Second, we found the neuron modules found by our biclustering method learns different variation of feature patterns. It is surprising that while we do not use any label supervision, samples corresponding to the same module show coherent feature patterns. Then, we also show in experimental results that found modules play a critical role in discriminating data samples. We perform ablation experiment to see the performance of DNN in the absence of co-adaptive modules and demonstrate that removing these modules would give the severest performance drop compared with other alternative methods for finding important neurons. We also demonstrate that these modules learn suprisingly rich representations and are able to detecting feature patterns tested on several pre-trained models, including visual classification as well as a biological application case where different variations of DNA motifs are learned by these modules. This biological application provids a pracitcal to find different variations of DNA motifs for transcription factor binding sites.

Our approach provides a powerful and perhaps the first way, as far as we know, to dissect a hidden layer into functionally related neuron modules in a unsupervised way. These found modules detect different feature patterns and play an important role in predicting samples. Our approach would be very helpful for intepreting DNN's behavior, and these modules could be directly used to reveal what what feature patterns are learned in the DNN.

## 2 RELATED WORK

We briefly introduce some representative approaches for interpreting DNNs and their drawbacks. Then we introduce several works on attributing DNN's prediction to its hidden neurons, which serve as the basis of our approach.

***Interpreting DNN.***. Some believe deep neural nets work as generic feature detectors and thus it is natural to interprete what features are detected by hidden neurons, using optimization-based method (maxmizing the activation of hidden neurons) or back propagation based methods (computing a attribution score for each hidden neuron).

We observe that such methods is too restrictive to gain any human-scale insights. For example, optimizing the input to maxmize certain neuron's activation do not give satisfactory and interpretable results, or even it does, it only works in the first layers. Maxmizing a channel in the first convolutional layers might by chance reveal certain feature patterns, but maxmizing neurons in deeper layers would typically give fancy but nonsense results.

We identify flaws of these approaches that they only work at single (hidden or output) neuron given single data sample :

- *single-neuron interpretation*: those methods restrict the interpretation on a single neuron or a channel of neurons, while in principle multiple hidden neurons, regardless of their spatial locations, could work together as a coherent group for feature extraction.

  Take a more concrete explanation, if we want to identify the important hidden neurons for discriminating "dog" images, these neurons do not neccessarily have to be a single one or within a channel with spatial proximity, but only have to be "functionaly" related.

- *single-sample interpretation*: such methods give interpretation only on a single data instance $x$, and is not guatanteed to generalize on other data instances. For example, attributing the prediction of "dog" on a single instance $x_1^{dog}$ to a hidden neuron does not mean that for another "dog" image $x_2^{dog}$ it will give consistent attribution. While in principle, a convincing attribution approach should be able to be consistent over different samples as long as certain conditions are met, e.g., several neurons are most of the time firing when given "dog" images, indicating that they are detecting some "dog" feature patterns.

These methods more or less ignore that, given the flexibility of capacity and notion of collective information encoding, it is more likely that a group of hidden neurons work together to perform feature extraction in a collective manner. Using one single neuron or a channel of neurons as the basic unit is probably not a good choice.

***Attribution on hidden neurons.*** There is a bunch of work on attributing a prediction to hidden neurons. Basically, the goal is to assign a score $a$ to a hidden neuron $n$ that represents how much important is this neuron for predicting a sample $x$ to a class $y$.

To name a few alternatives, [Shrikumar et al. 2017] propose DeepLIFT , assigning scores to the feature of input and generalize to assign a score of importance to a specfic hidden neuron. [Sundararajan et al. 2017] proposed intergrated gradients to assign scores into a input neurons and later generalize to total conductance [Dhamdhere et al. 2018; Shrikumar et al. 2018] which assign scores to hidden neurons. [Leino et al. 2018] proposed a influence-based attribution method.

Such methods has demonstrated to be able to identify the important input features as well as high-importance hidden neurons that is relevant with the DNN making a specific prediction on a class $y$ given a data instance $x$.

However, these methods do not explicitly consider that hidden neurons may be functionally related.

## 3 METHOD

We start with our notations and then introduce how we use an attribution methods mentioned above to construct a neuron-sample attribution matrix. Based on this matrix, we apply spectral coclustering to simultaneously group neurons and samples that have consistent high attribution values.

### 3.1 Notation

We denote a DNN as a function $f$ parameterized by $\theta$ such that $\hat{y} = f_\theta(x)$ is the prediction of sample $x$. We have a dataset $X = \{x\}$ of test samples which are not used in training.

We also have a attribution function $a$ which could be any of guidedprop, DeepLIFT score [Shrikumar et al. 2017], total conductance [Dhamdhere et al. 2018]. $a(n, x, y)$ measures how much a hidden neuron $n$ contributes to predict sample $x$ into class $y$.

### 3.2 Construct the neuron-sample attribution matrix

We first specify the neurons to be from a given layer $l$ and construct the neuron-sample matrix $M_{N_l * N_{data}}$ where $N = |l|$ is the amount of neurons in that layer and $N_{data} = |X|$ be the size of dataset $X = x_1, x_2, \ldots, x_N$ such that each entry in the matrix $e_{ij} = e(n_i, x_j)$. $e$ is a embedding function which measures the contribution of neuron $n_i$ for predicting data sample $x_j$.

Notice that the embedding function $e$ is not class-specfic and the attribution function $a$ needs an extra argument of which class to attribute to. We take $e_{ij}$ to be $a(n_i, x_j, \hat{y})$ where $\hat{y}$ is the predicted class of DNN $\theta$.

We then normalsize the socre by substracting the mean of all classes so that the final embedding function is

$$e_{ij} = a(n_i, x_j, \hat{y}) - \frac{1}{|C|} \sum_c a(n_i, x_j, c)$$

where $a$ could any of class-specfic attritbution function like [Dhamdhere et al. 2018; Leino et al. 2018; Shrikumar et al. 2017, 2018].

We choose the DeepLIFT [Shrikumar et al. 2017] score in our experiments, where in qualitive and quantitive evaluations these attribution methods generally do not differ so much. Each row in this neuron-sample matrix corresponds to a hidden neuron $n$ in layer $l$ and each column a data sample $x \in X$

### 3.3 Spectral Co-clustering

Given such a matrix, we use spectral coclustering to find biclusters with values higher than those in the corresponding other rows and columns. Each row (neuron) and each column (sample) belongs to exactly one bicluster. Note that this algorithm needs a predefined number of biclusters.

The spectral coclustering algorithm treats the input data matrix as a bipartite graph: the rows and columns of the matrix correspond to the two sets of vertices, and each entry corresponds to an edge between a row and a column. The algorithm approximates the normalized cut of this graph to find heavy subgraphs that have strong edges.

An approximate solution to the optimal normalized cut may be found via the generalized eigenvalue decomposition of the Laplacian of the graph. Usually this would mean working directly with the Laplacian matrix $M^{Laplacian}_{(N_l+N_{data})*(N_l+N_{data})}$, however it is possible in our case to work directly with $M$ , which is smaller and more efficient. The input neuron-sample matrix $M$ is preprocessed as follows:

$$M' = R^{-\frac{1}{2}} M C^{-\frac{1}{2}}$$

Where $R$ is the diagonal matrix with entry $i$ equal to $\sum_j M_{ij}$ and $C$ is the diagonal matrix with entry $j$ equal to $\sum_i M_{ij}$. The singular value decomposition, $M' = U \Sigma V^T$ , provides the partitions of the rows and columns of $M$. A subset of the left singular vectors gives the row partitions, and a subset of the right singular vectors gives the column partitions. The $\ell = \lceil \log_2 k \rceil$ singular vectors, starting from the second, provide the desired partitioning information. They are used to form the matrix $Z$ :

$$Z = \left[ R^{-1/2} U C^{-1/2} V \right]$$

where the columns of $U$ are $u_2, \ldots, u_{\ell+1}$ , and similarly for $V$. Then the rows of $Z$ are clustered using k-means. The first $n_rows$ labels provide the row partitioning, and the remaining $n_columns$ labels provide the column partitioning.

Each row (neuron) and each column (sample) belongs to exactly one bicluster, so rearranging the rows and columns to make partitions contiguous reveals these high values along the diagonal.

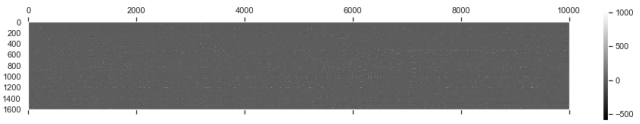
## 4 EXPERIMENTS

Most of our experiments are done on a pre-trained model on MNIST for visual digit classification. The architecture of this DNN is a 4 layer model where the first two layers are convolutional layer followed by two fully-connected layer. For attribution methods we choose DeepLIFT [Shrikumar et al. 2017] score.

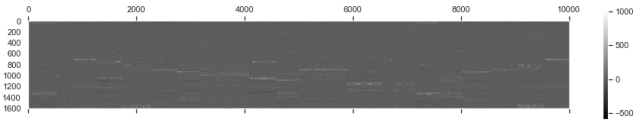
### 4.1 The neuron-sample matrix is highly sparse

We use the standard test set with 10000 data samples. We choose $l$ to be the second convolutional layer of shape $(5, 5, 64)$. The total number of hidden neurons in this layer is 1600. We useg the orginal code from authors to compute the DeepLIFT score [Shrikumar et al. 2017]. We set $k = 20$ to find $k$ biclusters in this neuron-sample matrix.

The raw neuron-sample matrix and the rearranged matrix is displayed in Figure 2.



(a) raw neuron-sample matrix



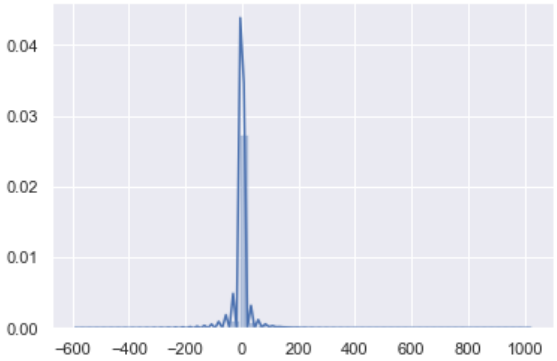
(b) neuron-sample matrix rearranged with bicluster labels

**Figure 2: *visualization of the neuron-sample matrix* Each row corresponds to one of the 1600 neurons in the second convolutional layer ($5 * 5 * 64$). Each column represents one of 10000 data samples.**

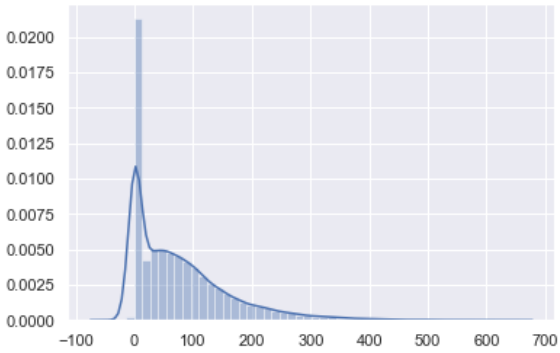
As we can see from Figure 2, this neuron-sample matrix is highly sparse, which indicates that only a small subset of neurons are important for predicting a set of similar samples. After applying the spectral coclustering algorithm, we see that the matrix does show a checkerboard-like pattern.

We also visualize the distribution of attribution score in the original matrix and in one found biclusters in Figure 3.

The overall distribution is centered around zero but the attribution scores in one bicluster are generally of a high postive score, indicating that the spectral coclustering algorithm does find good subgraphs that a subset of hidden neurons typically have higher attribution scores on a subset of samples.



(a) distribution of attribution score in one bicluster in the raw neuron-sample matrix



(b) distribution of attribution score in the id11-th bicluster

**Figure 3: *Distribution of attribution scores* Each row corresponds to one of the 1600 neurons in the second convolutional layer ($5 * 5 * 64$). Each column represents one of 10000 data samples.**



(a) samples from id@13-th bicluster

(b) samples from id@14-th bicluster

**Figure 4: *Data samples from one bicluster of total* 20 *biclusters***

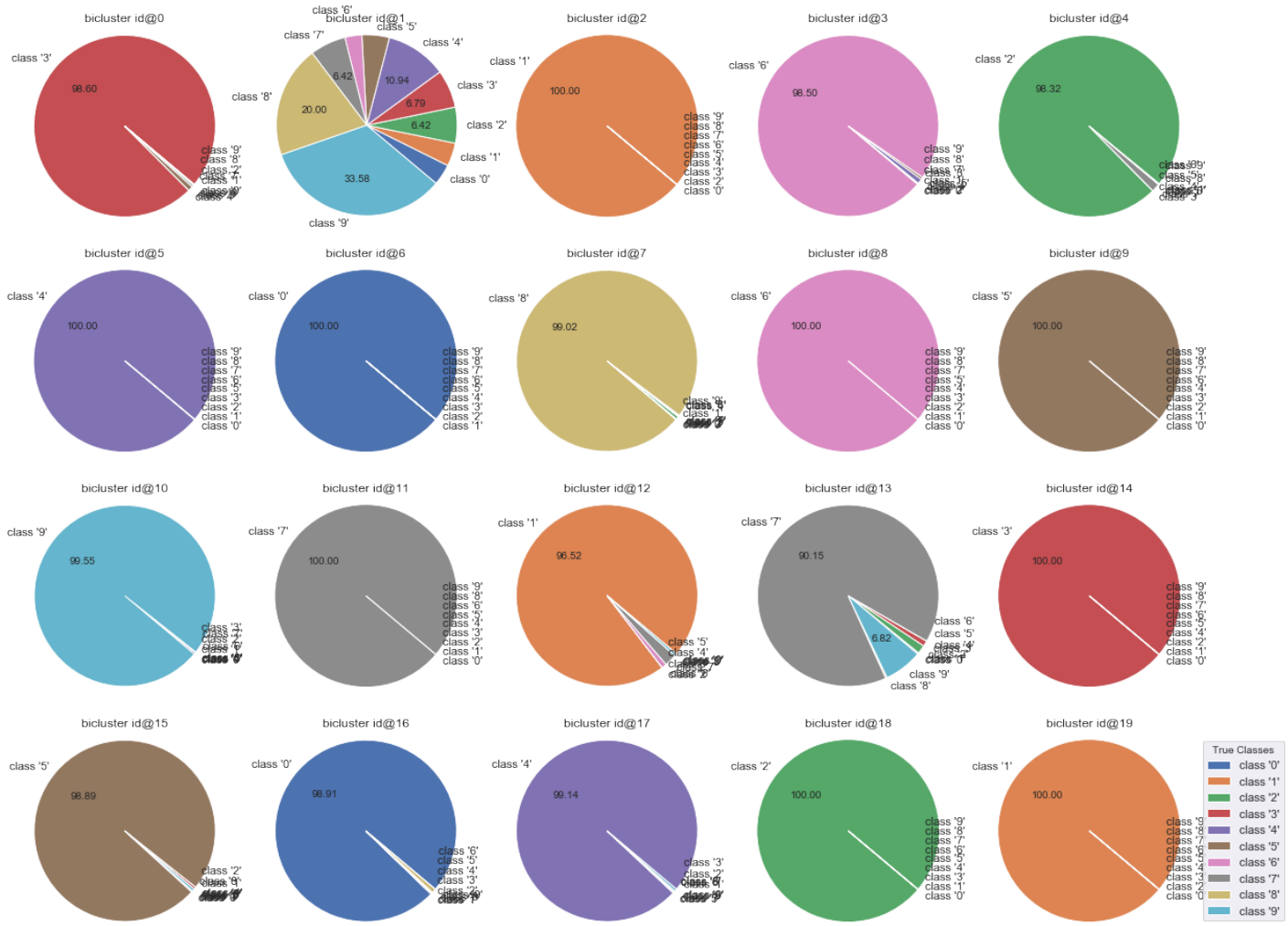



**Figure 5:** ***Distribution of class labels in each bicluster***

## 4.2 Samples corresponding to the same module show coherent feature patterns

In this section we wish to show that the samples in the one bicluster, i.e., samples corresponding to the same co-adaptive module show coherent and similar feature patterns.

We pick up one bicluster and randomly sample several samples from that bicluster.

Surprisingly, the samples in the id13-th bicluster shows very similar patterns that most of these digits are of class "7" and even for several samples of other classes, the digit images are very similar. Note that we do not use any label supervision in our approach, and we are achieving a very good unsupervised discriminating job.

Now we use the ground-truth labels of MNIST digit labels for $X$ to visualize the distribution of class labels in each biclusters.

As seen in the percentage pie chart of class label distribution in different biclusters (Figure 5), sample from each bicluster are almost of the same class label, which indicates that DNN is perhaps organized as disjoint modules for detecting different feature patterns.

## 4.3 Ablation Study: co-adaptive modules are critical for discriminating samples

We check the performance of DNN by removing those found co-adaptive neuron modules. In this experiment, we split the 10000 images $X$ further into two sets where the first 5000 samples as validation set are used to compute attribution score and perform spectral coclustering, the rest 5000 samples as test set are used for testing prediction metrics such as accuracy and recall. We set the number $K$ of biclusters to 20.

We first compare the accuracy when hidden neurons in one layer $l$ are gradually ablated. We evaluate the accuracy on test set with

several baselines for ablating neurons as well as our approach. Ablation of a neuron is implemented by setting the activation value of this neuron to zero, as convention in previous studies [Dhamdhere et al. 2018; Shrikumar et al. 2018].

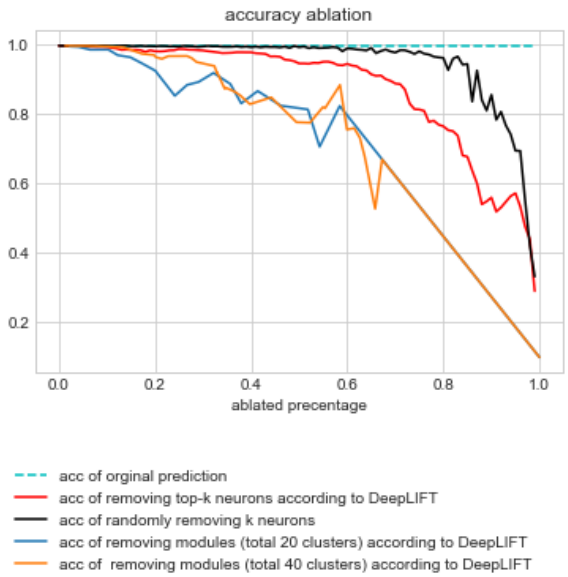



**Figure 6:** ***Trend of accuracy drop when neurons are ablated compared methods.***

- randomly remove hidden neurons.
- rank every hidden neurons with their average attribution score on validation set and greedily remove neurons with largest attribution score
- Our methods: notice that the co-adaptive modules we found do not have a order. We use the ground truth label of validation set to compute negative class entropy and thus obtain an order for ablation.

As shown in Figure 6, ablation by module achieves the most significant accuracy drop compared with greedy removing top-k neurons (all use DeepLIFT score). It demonstrate that these co-adaptive modules found by spectral coclustering do play an important role in discriminating data samples.

We also observe an interesting result that while the metric of precision generally is not heavily decreased by ablating co-adaptive moduleson compared with other methods, recall does drop down very heavily. As showin in Figure 7, the precision drop of compared ablation method is very close and almost identical, the recall drop is most significant for our approach (module ablation).

This also indicates that our found co-adpative modules find and detect very complex feature patterns and removing these modules would damage its predicting function.

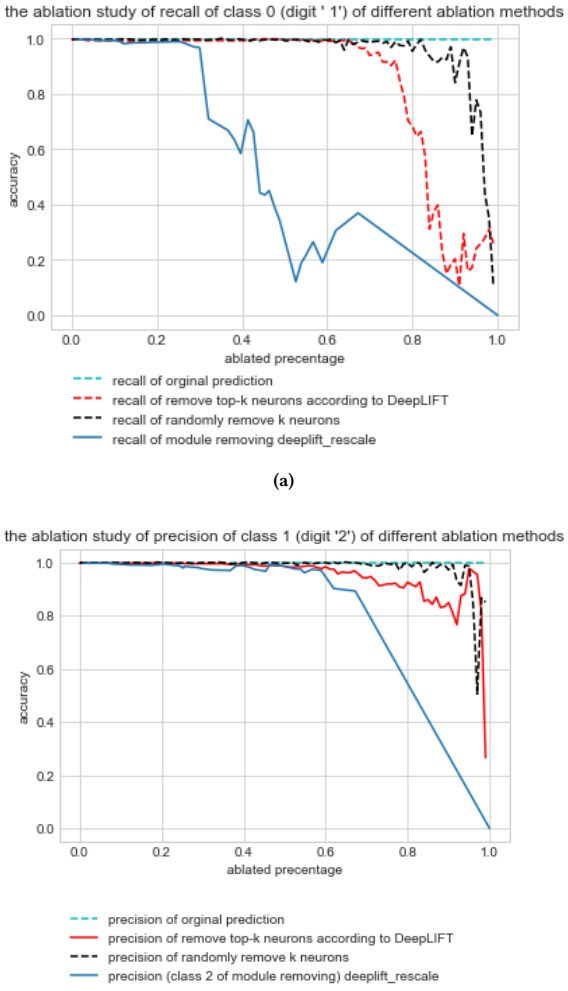
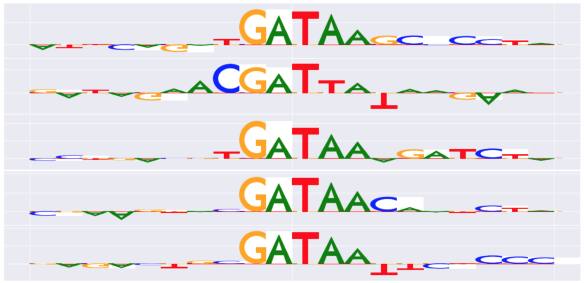



**Figure 7:** ***Trend of recall and precision drop when neurons are ablated (for class '1')***

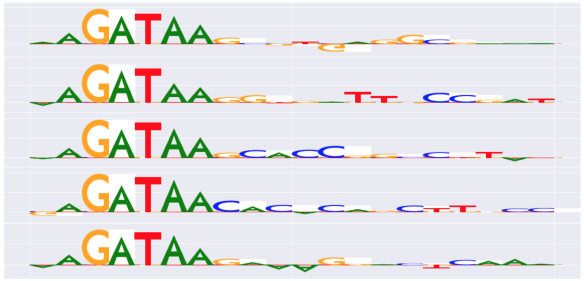
## 4.4 An application for transcription factor binding site motif discovery

We demonstrated how the co-adaptive modules could be used to discover feature patterns in a biological task.

Knowing the sequence specificities of DNA- and RNA-binding proteins is essential for developing models of the regulatory processes in biological systems and for identifying causal disease variants. Recently there are application of using convolutional neural networks to sequence-based problems in genomics signals the advent of the deep learning era in computational biology. Two recent methods, DeepBind [Alipanahi et al. 2015] and DeepSEA [Zhou and Troyanskaya 2015] , successfully applied deep learning to modeling the sequence specificity of protein binding with a performance superior to the best existing conventional learning methods.



(a)



(b)



(c)



(d) known motif from Jaspar

**Figure 8: *different modules find different feature patterns* The same procedure is applied to a DeepBind model where the known motif for this transcription factor from Jaspar is GATA. Each sample gene sequence is visualized using deeplift score (note that here we use deepLIFT not only for constructing the neuron-sample matrix, but to visualize the importance of each base-pair).**

However, how to interprete what is learned by these DNN is still not clear. DeepBind [Alipanahi et al. 2015] uses a weighted ensemble of position weight matrices to visualize the specificity sequences determined by DeepBind network.

We here demonstrate how our approach could be applied to find representative DNA specificities. We use one pretrained model from DeepBind (more than 600 models are in DeepBind for different tasks) and apply our approach based on this pretrained model.

It turns out that our approach is not only able to find very important motifs, but also even more complicated types of feature patterns, such as motif occuring at the left end of the sequence, motif occuring at the right end of the sequence and even motif occuring twice in the sequence (see Figure 8). Note that samples (DNA sequences) with these patterns are also divided into different groups, we are getting very important information out of the DNN's uninterpretable parameters. This appoach learns very fine-grained subtypes of feature patterns.

## 5 LIMITATIONS AND FUTURE WORK

We develop an powerful approach to parcelate a hidden layer into functionally related groups which we call co-adaptive module. We apply a spectral coclustering approach based on the attribution scores of hidden neurons to find closely related neuron-sample bicluster that predictions on these samples are highly related to these neurons.

We find the co-adaptive module we find play an important role in discriminating data samples and we also show how co-adaptive module could be used for feature pattern discovery.

There are also limitations. One major limitation of our proposed co-adaptive module is that we do not know how to correctly visualize co-adaptive module other than showing samples in the same bicluster. It also needs more experiment to ensure that if different attribution method make any difference. Although in our experiments the choice of attribution methods (DeepLIFT, total conductance, guided-prop) do not general differ.

Our approach to find co-adaptive modules has many potential applications. The first one naturally goes to efficient automatic pattern discovery from the dark knowledge inside the hightly complex DNN. It is been widely acknowledged that deep neural networks are very good at classification on huge datasets, but in domains like genomics, as feature patterns are very important but generally not easy for human to understand, we wish to understand the dark knowledge inside complex non-linear transformations in DNN. It could also be used for many other applications such as network compression and multi-task learning. Since our co-adaptive module finds fine-grained subsets of neurons that are very important for a subset of data samples, we could prune irrelevant neurons, take a instance, if we want to compress a 1000-class classification DNN into a 10-class classification DNN. In terms of multi-task learning, we could also find important neurons that are especially related for different tasks and only updates the weights on relevant neurons during training.

## REFERENCES

Babak Alipanahi, Andrew Delong, Matthew T Weirauch, and Brendan J Frey. 2015. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nature Biotechnology* 33, 8 (jul 2015), 831–838.

Kedar Dhamdhere, Mukund Sundararajan, and Qiqi Yan. 2018. How Important Is a Neuron? *arXiv.org* (May 2018). arXiv:1805.12233v1

G E Hinton, N Srivastava, and A Krizhevsky arXiv. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arxiv.org* (2012).

Klas Leino, Linyi Li, Shayak Sen, Anupam Datta, and Matt Fredrikson. 2018. Influence-Directed Explanations for Deep Convolutional Networks. *CoRR* (2018).

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning Important Features Through Propagating Activation Differences. *arXiv.org* (April 2017). arXiv:1704.02685v1

Avanti Shrikumar, Jocelin Su, and Anshul Kundaje. 2018. Computationally Efficient Measures of Internal Neuron Importance. *arXiv.org* (July 2018). arXiv:1807.09946v1

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout - a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* (2014).

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic Attribution for Deep Networks. *CoRR* abs/1703.01365 (2017). arXiv:1703.01365 http://arxiv.org/abs/1703.01365

Jian Zhou and Olga G Troyanskaya. 2015. Predicting effects of noncoding variants with deep learning–based sequence model. *Nature Methods* 12, 10 (aug 2015), 931–934.