Transcription Factor Binding Site Prediction with Convolutional Neural Networks

Keton Kakkar KKAKKAR1@SWARTHMORE.EDU

Swarthmore College, 500 College Avenue, Swarthmore, PA 19081

Katherine Kwok KKWOK@SWARTHMORE.EDU

Swarthmore College, 500 College Avenue, Swarthmore, PA 19081

Ameet Soni SONI@CS.SWARTHMORE.EDU

Swarthmore College, 500 College Avenue, Swarthmore, PA 19081

Abstract

Datasets of in-vivo transcription factor binding sites are becoming more available, yet the capacity to gather data using biological approaches remains limited. Supervised machine learning methods have been shown to work well on the task of predicting a binary classification using labeled data. Here we implement a multi-layered convolutional neural network (CNN) on data from the ENCODE-DREAM challenge and show that CNNs yield comparable results to feed forward neural networks for predicting transcription factor binding sites within and across cell-types.

1. Introduction

Transcription factors (TFs) are regulatory proteins that induce or repress transcription of genes. The task of identifying where transcription factors bind onto DNA is important for the field of systems biology. Chromatin immunoprecipitation followed by sequencing (ChIP-seq) is the most common technique for experimentally determining TF-DNA binding maps; though there exist datasets for *in-vitro* binding patterns, it is not possible at present to conduct ChIP-seq assays for all TFs on every cell type for all possible conditions. This motivates the need for a computational model to predict TF binding sites given existing data in order to supplement experimental results. Supervised machine learning techniques show good performance on binary classification tasks with labeled data. This paper presents results from running a convolutional neural network (CNN) on ChIP-seq data from the 2016 ENCODE-DREAM *in-vivo* transcription factor binding site prediction challenge on both within and across cell-type data. The multi-layered one-dimensional convolution model we implemented shows on average similar performance to that of a standard feed forward neural network (NN). The paper is structured as follows: an description of our CNN design, an explanation of data processing and formatting, a discussion of results, and an overview of limitations and directions for further research.

CPSC 93 Deep Learning Final Report Proceedings, Swarthmore College, Spring 2018.

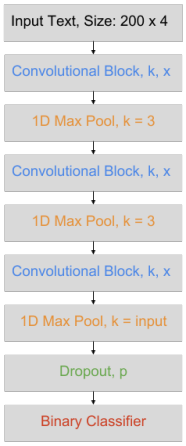

Figure 1. Model Architecture.

2. Motivation and Model

Supervised machine learning tasks are suited to binary classification problems in which there are large amounts of data. CNNs, though traditionally used on images, have been shown to work well on text analysis problems (Conneau et al., 2016). The task of labeling sequences which comprise of genetic base pairs can be thought of as similar to the task of labeling sentences which comprise of letters. In fact, Alipinahi, et al. and Chen et al. have shown that CNNs work well for TF binding site prediction (Alipanahi et al., 2015) (Chen et al., 2017). The motivation for using CNN is their ability to capture motifs of varying length and be invariant to translations in the data.

The data is fed in as a one-hot matrix with alphabet-length of four, corresponding to the four base pairs, and a base-

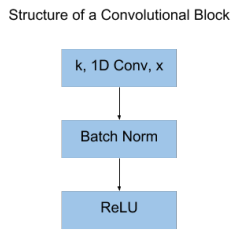

Figure 2. Convolutional block

pair length of 200, which is roughly how many base pairs are labeled during a ChIP-seq assay. This input matrix of [200, 4] is fed into a CNN, the architecture of which is outlined in Figure 1. It consists of 3 convolutional blocks (Figure 2) each followed by a pooling layer. Each convolutional layer in a convolutional block has a kernel size of length k, which is adjusted as a hyperparameter to the model. Taking intuition from applications of CNNs to text analysis, each convolutional layer is one-dimensional and convolves over the axis that represents base-pairs; it is not convolving over the axis which represents the alphabet and later contains the feature maps. The output of each convolutional layer is number of filters x, which is also a hyperparameter that is tuned. Each convolutional layer is followed by a batch normalization layer for regularization, and a rectified linear activation layer (ReLU) for a sparse, non-linear activation function. The first two max pooling layers have a kernel size of 3 and zero padding to maintain the dimensions of the data being passed through the network; the small kernel size of the pooling layers enable the extraction of local motifs produced by the convolutional block while amalgamating longer motifs by striding through the whole input. The final pooling layer performs a global pool over the non-feature map dimension to flatten the input and reduce dimensionality. Subsequently, there is a dropout layer for regularization and a logits layer for binary classification.

2.1. Hyperparameters

Large kernel sizes tend to be needed for text applications, so the choice of k size ranged from 5 to 7. Stacked convolutional layers with smaller kernel sizes as used in Deep CNN Text Analysis led to degenerate results for this network. The number of filters, also the number of feature maps in the output of the convolutional layers, were randomly tuned among the options 32, 64, 128, 256. The dropout keep probability and the learning rate were randomly tuned in a range, as recommended by Goodfellow, Bengio, and Courville (Goodfellow et al., 2016), and varied between [.5, .75] and [.001, .01], respectively. Zero-padding for the convolutional layer was also a hyperparameter, with the options being either same or valid padding. The batch size was set to be the square root of the total size of the data set, and the network was allowed to train for as many epochs as need with an early stopping patience of 20 epochs. The network was optimized using the Adam optimizer.

3. Experimental Design

ChIP-seq experiments sequence roughly 200 base pairs and classify them as bound, unbound, and ambiguous. The challenge data was presented as 200 base pair sequences sliding every 50 base pairs. In processing the data, we ignored ambiguous labels and compared each 200 base pair region against the provided peak information to yield non-overlapping sequences. The raw number of unbound sequences for each TF cell type pair greatly outnumbered the bound sequences. We randomly sub-sampled the unbound sequences to be roughly the same size as the number of bound sequences. 80% of the data was split into a training set and the other 20% became a test set that was held aside for evaluation purposes. 20% of the training set was used for validation of the model during training. We conducted two discrete experiments. The first was to judge classification accuracy of a model within cell types. In other words, the CNN was trained and evaluated on specific TF cell type pairs. The second experiment, in contrast, was to evaluate predictions across cell types. The dataset for the second experiment was gathered by amalgamating all the TF-cell type pairs for each TF.

4. Results and Discussion

In this section we compare performance between the CNN model outlined above and the feed forward network implemented by Kwok et al (Kwok et al., 2017). Additionally we compare results from the within-cell type and across-cell type experiments.

Code is available here:
https://github.swarthmore.edu/SoniGroup/ENCODE
Results are available here: https://bit.ly/2rVpd3l

The CNN model we implemented achieved similar results on the within cell-type experiment to that of the feed forward network implemented by Kwok on the same data (Kwok et al., 2017). Figure 3 depicts a comparison of the test accuracies of both the CNN and the feed forward NN averaged over 89 TF/cell-type pairs. Contrary to expectations, the CNN did not show significant improvement to the standard model on this problem.

On the second experiment we ran, using data from across cell types, we see similarly comparable performance (see

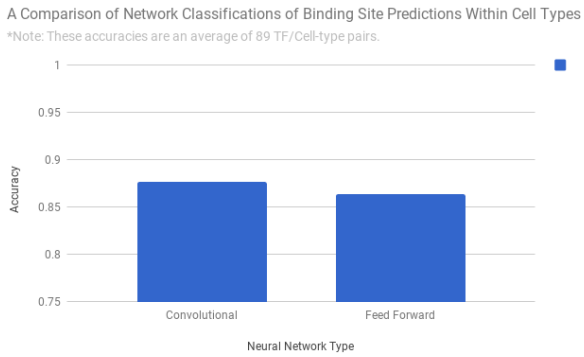

Figure 3. Model Within cell-type accuracies.

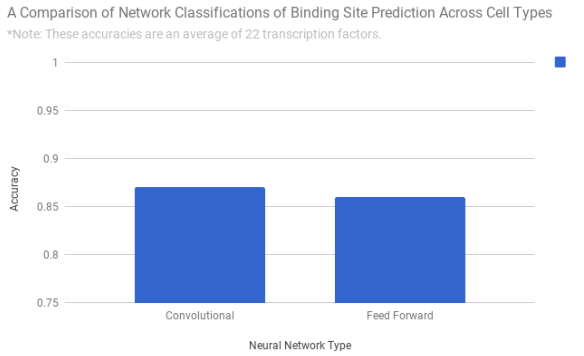

Figure 4. Model Across cell-type accuracies.

figure 4). Averaged over 22 transcription factors, we compare the test accuracies of the CNN and the feed forward NN to find that there is no significant improvement when using the CNN model as outlined above. We expected that the CNN's capacity for representing motifs of variable length would result in more accurate classification on this task than the feed forward network, given that the data is not cell type specific. This was not the case.

Figure 5 compares the CNN's performance on within cell-type and across cell-type binding site prediction. The across cell-type accuracy is calculated by averaging the accuracies of each TF/cell-type pair for a given transcription factor. Both bars, red and blue, represent the model's performance on the combined set of data for a particular transcription factor. The difference is that the blue, within cell-type bar shows the output of a model that was trained with the data of all the cell types combined, where as the red bar shows that of one trained just on one cell type. Intuitively, the accuracy of the across cell-type data ought to be significantly higher; more relevant training examples ought to result in higher accuracy and more 'learning.' This, however, does not seem to be the case. The CNN model outlined above seems to be unable to leverage the combined data,

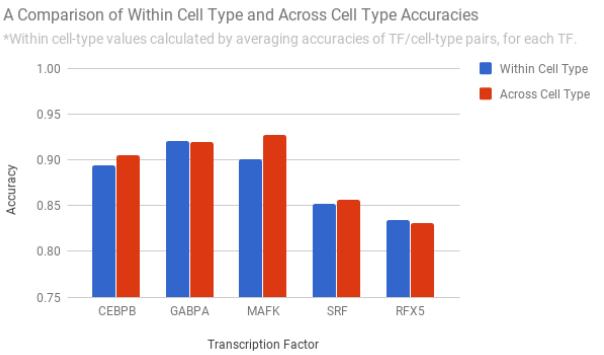

Figure 5. Model A comparison of within and across cell-type accuracies

and instead it appears as though there's a cap on learning at a certain point. One potential reason for the learning block could be the stark feature reduction of the global pooling layer at the end of the CNN (figure 1).

4.1. DNA Accessibility

Chromatin accessibility data (DNase-seq), also provided by the ENCODE-DREAM challenge, was incorporated into the network by a variety of methods, none of which seemed to be effective for the CNN. In contrast Kwok's results indicate that incorporating DNase-seq data into a feed forward neural network grants a significant boost in accuracy. The challenge for the CNN seemed to be how best to represent the accessibility data in the network. Encoding the DNase-seq information as a binary '5th character' in the one-hot matrix is unintuitive and yielded poor results. Adding the DNase-seq data as an extra input to the final dense layer did not affect the results of the CNN significantly either positively or negatively. Perhaps feeding the DN-ase data through a separate, feed-forward network and then tying the output of that network to the output of the CNN would yield benefits.

5. Further Research

Convolutional neural networks work for biological sequence analysis but run into limitations. Both Alipinahi et al. and Chen et al. incorporate CNNs into their TF binding site prediction models, but each add an additional element to their models. Alipinahi et al. feed their CNN into a feed forward network, and Chen et al. use a convolutional kernel model (Alipanahi et al., 2015) (Chen et al., 2017). This paper demonstrates how a standard CNN model performs on this sequence classification task. Following suit from Alipinahi and Chen, a direction for further research is to complicate the model to move past the limitations of the standard CNN. Insights from recent research indicate that a

temporal CNN works for text analysis, outperforming traditional text analysis models like LSTMs (Bai et al., 2018). Additionally, incorporating ResNet-like skip connections has been shown to allow for deeper CNNs while preventing vanishing gradients and network degradation (Conneau et al., 2016). Though the temporal convolution is not a function implemented in TensorFlow presently, I plan to rerun these experiments on a temporal CNN implemented in Torch.

An additional avenue for further research is to find ways to effectively incorporate ancillary data such as gene expression data and in-vitro TF binding site data, as well as to find an appropriate representation for the DNase-seq data.

Given that the goal of predicting TF binding sites is to further our understanding of systems biology, another area where the CNN model encounters problems on this task is interpretability. Deep learning models are difficult to interpret, and using them on a biological sequence analysis task, where the input is also relatively opaque to human understanding, raises concerns. Another area for future research is to interpret how the network is learning and to verify the prioritized sequence motifs against known influential motifs.

Acknowledgments

I would like to thank Professor Ameet Soni for advising this project, Katherine Kwok for work on data processing, the results from the feed forward network, and patient assistance with this project, and William Colgan for contributing input.

References

Alipanahi, B., Delong, A., Weirauch, M. T., and Frey, B. J. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat. Biotechnol.*, 33 (8):831–838, Aug 2015.

Bai, Shaojie, Kolter, J. Zico, and Koltun, Vladlen. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *CoRR*, abs/1803.01271, 2018. URL http://arxiv.org/abs/1803.01271.

Chen, Dexiong, Jacob, Laurent, and Mairal, Julien. Predicting transcription factor binding sites with convolutional kernel networks. *bioRxiv*, 2017. doi: 10.1101/217257. URL https://www.biorxiv.org/content/early/2017/11/10/217257.

Conneau, Alexis, Schwenk, Holger, Barrault, Loïc, and LeCun, Yann. Very deep convolutional networks for natural language processing. *CoRR*, abs/1606.01781, 2016. URL http://arxiv.org/abs/1606.01781.

Goodfellow, Ian, Bengio, Yoshua, and Courville, Aaron. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Kwok, Katherine, Soni, Ameet, and Kakkar, Keton. Ongoing research: Transcription factor binding site prediction with neural networks, 2017.