Discovery of Transcription Factor Binding Sites with Deep Convolutional Neural Networks

Reesab Pathak, Stanford University

Introduction

Since the Human Genome Project, biologists have become interested in gene regulators, including transcription factors. These proteins, which both directly and indirectly modulate expression of genes, can be identified through biochemical and genetic techniques. However, due to the density of these proteins across the genome, it is increasingly difficult to identify true transcription factor binding sites in a consistent fashion. As a result, computational biologists have focused on statistical and machine learning techniques to predict transcription factor binding sites in the genome.

A key instance of this problem includes the **multi-task classification** of novel and known transcription factor binding sites. Though traditional machine learning techniques, including decision trees, random forests, and support vector machines have been applied to the problem, we turn our attention to deep learning architechtures and investigate the feasibility of learning binding sites with deep convolutional neural networks.

Convolutional Neural Networks

Deep convolutional neural networks (also called CNNs, or ConvNets) are related to regular neural networks; they take data in an array as input and return a prediction after a series of hidden layers of so-called neurons.

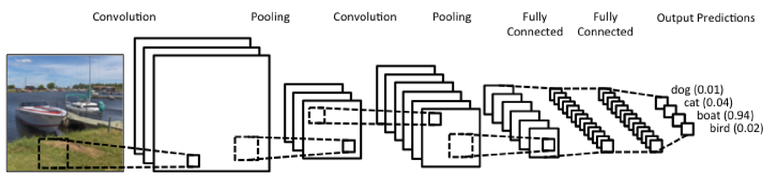

Figure 1: Schematic of a convolutional nueral network in the context of image classification.

Convolutional neural networks have three features: (1) convolutional layers, (2) pooling or “subsampling” layers, and (3) fully connected layers. The convolutional layer conducts an affine transformation over its input. Multiple activation maps are created by multiple convolutional filters. These pass to an activation function, which is a non-linearity such as a rectified linear unit (ReLU). The output is then passed to the pooling layer, which subsamples the convolutional output and takes the maximum entries within the domain of a subsampling unit (called “Max Pooling”). This process (convolution to max pool) is repeated. Unlike previous layers, the final layer is fully connected and a matrix-multiply is done to get output predictions. (Our architecture: CL-MP-CL-MP-CL-MP-FC) Convolutional neural networks like other feedforward archictures use backpropogation to update weights during each epoch during training.

Our data is simulated 1000 base-pair sequence, from which we embed a number of Encyclopedia of DNA Elements (ENCODE) motifs to get a positive class, and a set of negatively encoded sequences for a negative class. There may be multiple target vectors if multiple motifs are embedded. The embedded motifs are drawn from a Binomial distribution based on the probabilities given in the position weight matrix which is empirically determined by the transcription factor. The embedded sequence is one-hot encoded into an image, based on the four channels of the genomic alphabet (A, C, T, G). This image is passed as input into a deep CNN implemented using Keras, and is run on a GPU compute node (Sherlock).

Results

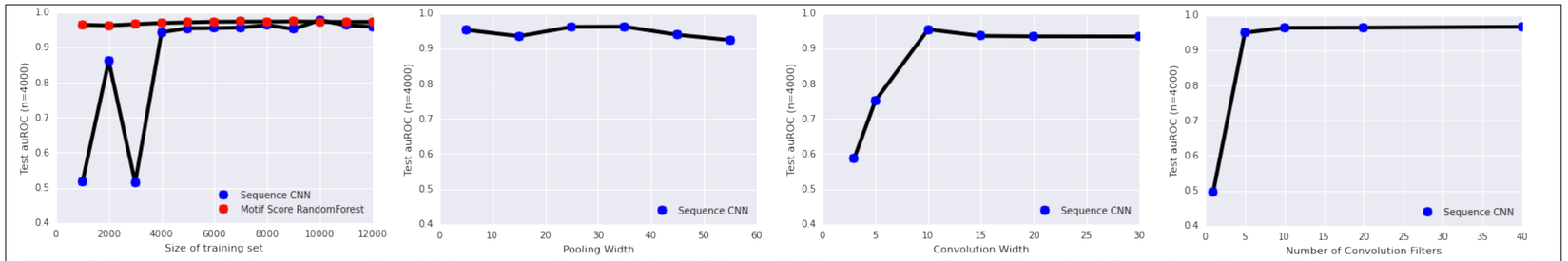

Figure 2: Hyperparameter tuning and sufficient data are essential to high (auROC) classification accuracy. (From left to right) (a) Comparison of amount training examples necessary to learn a single motif, plotted in a CNN versus a classical Random Forest model given ideal decision boundaries. (b) Max pooling width versus auROC for single motif learning, (c) Convolution filter width versus testing auROC, (d) Convolution layer size ersus test auROC. Motif tested is TAL1, an immunologically relevant transcription factor.

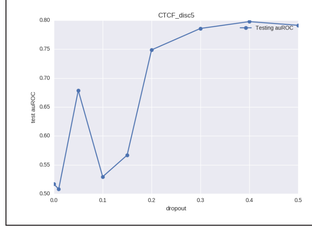

Figure 3: Dropout enhances testing accuracy. A method commonly used in CNNs is to dropout some neurons from one layer to the next. This plot is dropout percentage versus auROC for the single motif classification problem

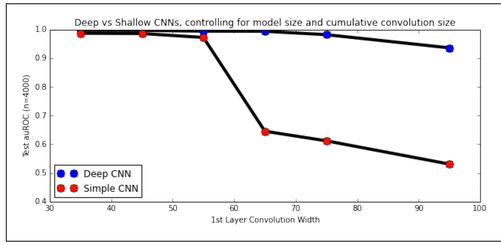

Figure 4: Multiple convolutional layers drastically increase model performance. Multiple motif embedding problem. Comparison of model performance, keeping output to max-pooling layer fixed. Changing convolution filter width shows dramatic change in accuracy in deep vs. shallow CNNs.

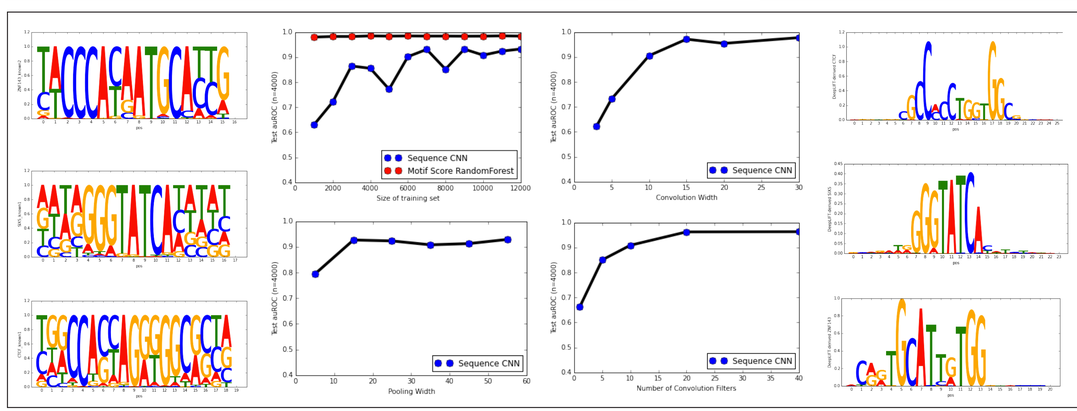

Figure 5: Convolutional neural networks can learn multiple transcription factor binding motifs simultaneously. (From left to right) (a) Three Encyclopedia of DNA elements (ENCODE) project transcription factor motifs and their representative consensus sequences. (b) Hyperparameter tuning learning curves for the first motif, (c) Feature importance using DeepLIFT, a linear feature importance tracker that identifies input features with greatest weight. The three motifs are known to co-bind, and are:

1. CTCF, known motif 1. CTCF is responsible for transcription regulation, V(D)J recombination, and chromatin architecture regulation
2. ZNF143, known motif 2. ZNF143 is a transcription activator
3. SIX5, known motif 1. SIX5 recruits DNA-binding proteins.

Literature Cited

1. Leung, M. K. K., Delong, A., Alipanahi, B. & Frey, B. J. Machine Learning in Genomic Medicine: A Review of Computational Problems and Data Sets. Proc. IEEE 104, 176–197 (2016).
2. Alipanahi, B., Delong, A., Weirauch, M. T. & Frey, B. J. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. Nat Biotechnol 33, 831–838 (2015).
3. Li, Y., Chen, C.-Y., Kaye, A. M. & Wasserman, W. W. The identification of cis-regulatory elements: A review from a machine learning perspective. Biosystems. 138, 6–17 (2015).
4. Consortium, R. E. et al. Integrative analysis of 111 reference human epigenomes. Nature 518, 317–330 (2015).

Future Questions & Conclusions

This work shows that it is indeed possible to learn transcription factor binding sites based on sequencing data, after sufficient data and hyperparameter tuning. We’re left with the following questions:

1. How does the optimal convolution filter width scale with the entropy of the position weight matrix (PWM) of the transcription factor motif?
2. How do hyperparameters change when transcription factor density is sparse versus localized?
3. How will CNN-based learning perform on Illumina genome sequencing data?
4. How do regularization and dropout affect CNN-based learning of TF motifs?