Predicting DNA-Protein Binding with Deep Convolutional Neural Networks

Andrew Wiens • Computer Science & Computational Biology • Cornell University

adw223@cornell.edu

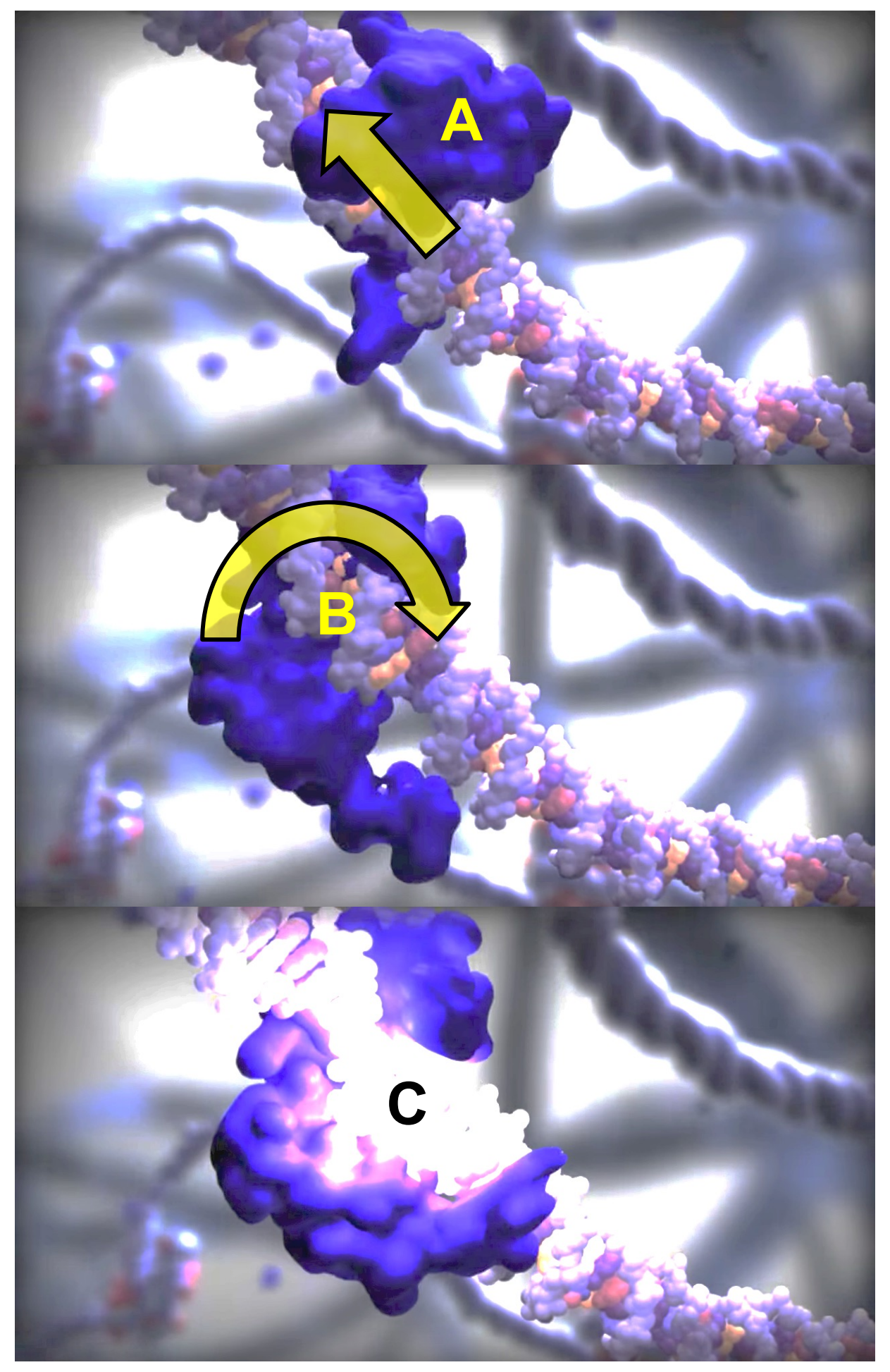
This project is on GitHub:
https://git.io/v1sAy

Cornell University

Question

Which deep neural network architecture(s) should biologists use to predict whether a transcription factor protein will bind to a particular DNA sequence?

Background



A. A **transcription factor (TF)** protein slides unbound along chromatin, the particular form of the DNA double helix that exists inside a cell's nucleus

B. The TF slides along the chromatin in a spinning motion around the double helix until a **binding site** is encountered

C. If the TF reaches an *accessible* location on the chromatin containing a **motif**, a short DNA sequence that is compatible with the TF, the TF changes shape (conformation) and **binds** to the DNA. This causes interactions with other proteins which affects the **expression** of genes

Images from ref. [4]

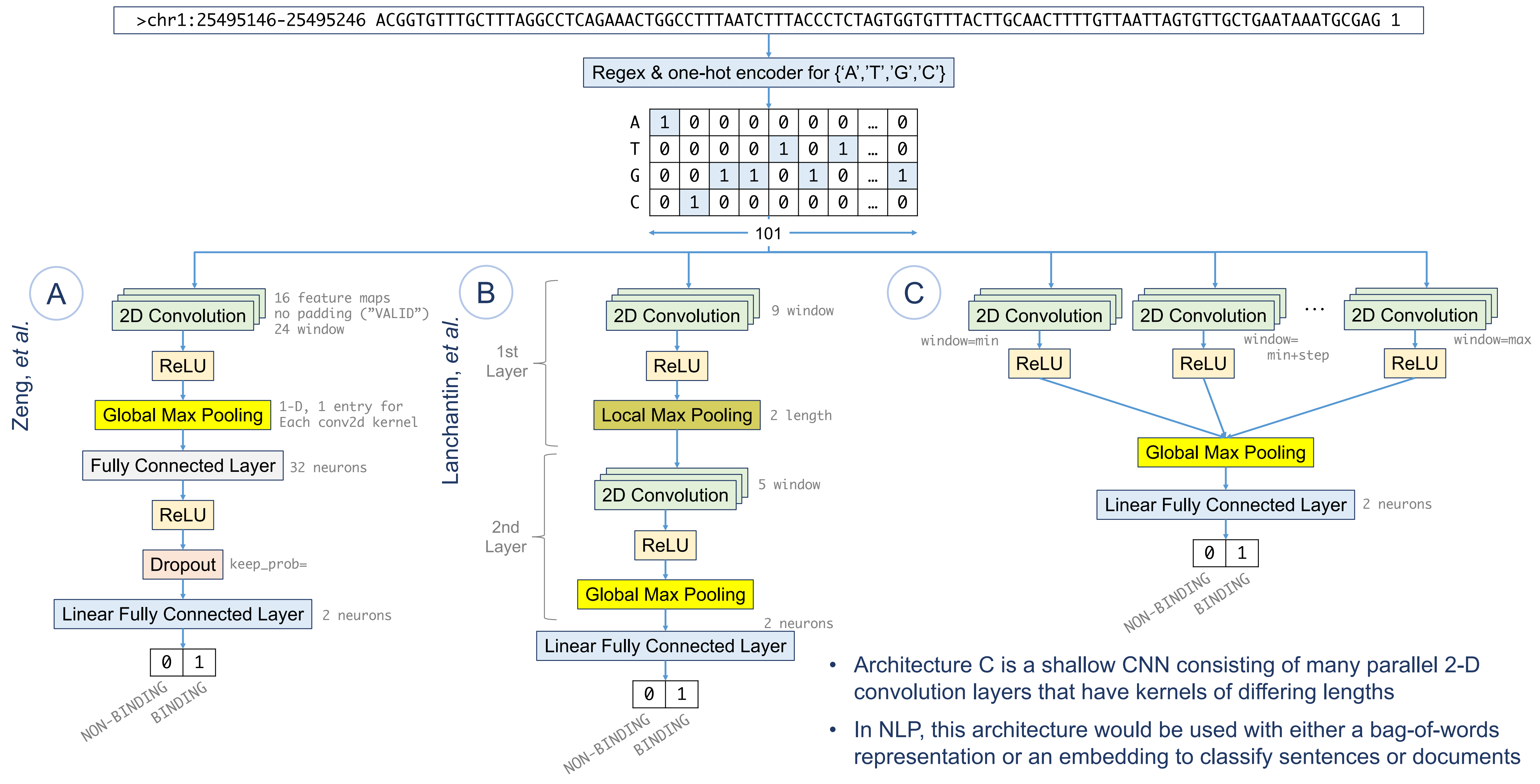
- Modeling DNA sequence protein-binding specificity is analogous to the computer vision task of binary (two-class) image classification
- Instead of processing 2-D images with three color channels (R,G,B), consider a genome sequence as a fixed length 1-D sequence window with four channels (A,C,G,T)
- Advantage of convolutional neural networks (CNN) for genomics is the ability to detect a motif anywhere is in the DNA sequence window
- Two tasks have been explored with deep neural networks:
  1. *Motif discovery* classifies sequences that are bound by a transcription factor from negative sequences that are di-nucleotide shuffles of the positively bound sequences
     - This is a relatively easier classification task
  2. ***Motif occupancy*** discriminates genomic motif instances that are bound by a transcription factor (positive set) from motif instances that are not bound by the same transcription factor (negative set) in the same cell type
     - Tested in this work
- Previous work used Theano (Quang *et al.*), Torch (Lanchantin, *et al.*), and Caffe (Zeng *et al.*)
  - To our knowledge, TensorFlow has not been used yet

Materials

- TensorFlow + Nvidia GeForce GTX 970 GPU
- 108 datasets from the Encyclopedia of DNA Elements (ENCODE) for motif occupancy classification of different transcription factors was used. (Datasets were compiled by Zeng et. al.)
  - DNA sequences of 101 base-pairs from K562 cell line labeled as either **1** (TF bound) or **0** (TF unbound)

Methods

Three architectures were tested:



```
>chr1:25495146-25495246 ACGGTGTTTGCTTTAGGCCTCAGAAACTGGCCTTTAATCTTTACCCTCTAGTGGTGTTTACTTGCAACTTTTGTTAATTAGTGTTGCTGAATAAATGCGAG 1
```

Regex & one-hot encoder for {'A','T','G','C'}

A	1	0	0	0	0	0	0	…	0
T	0	0	0	0	1	0	1	…	0
G	0	0	1	1	0	1	0	…	1
C	0	1	0	0	0	0	0	…	0

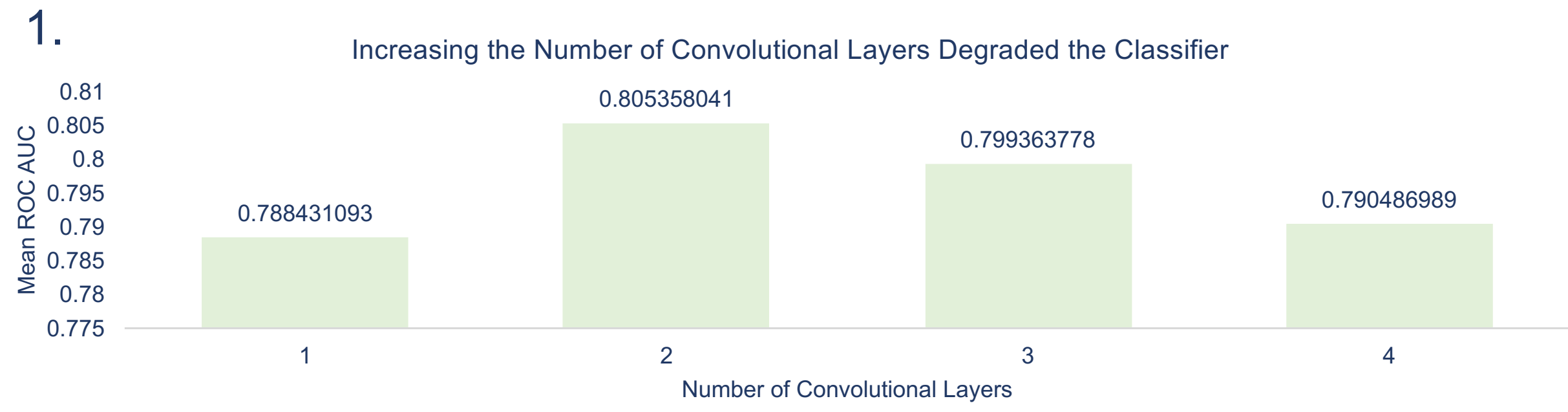
101

- Architecture C is a shallow CNN consisting of many parallel 2-D convolution layers that have kernels of differing lengths
- In NLP, this architecture would be used with either a bag-of-words representation or an embedding to classify sentences or documents

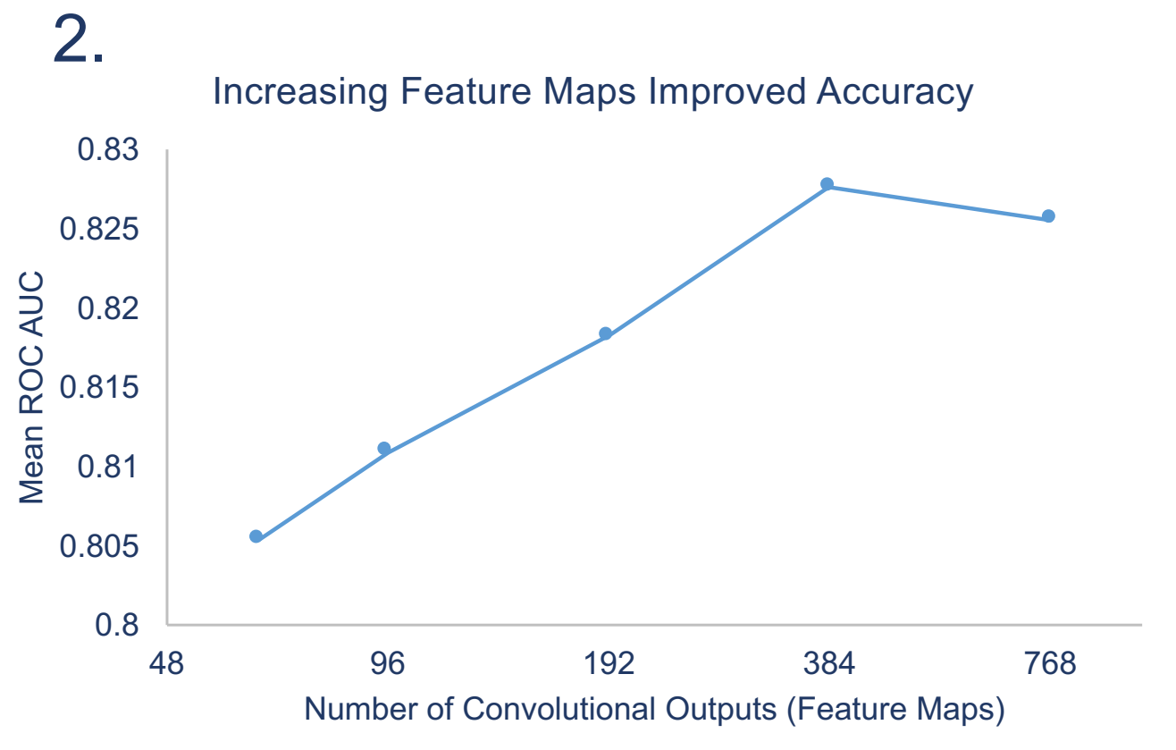
Results

Architecture B

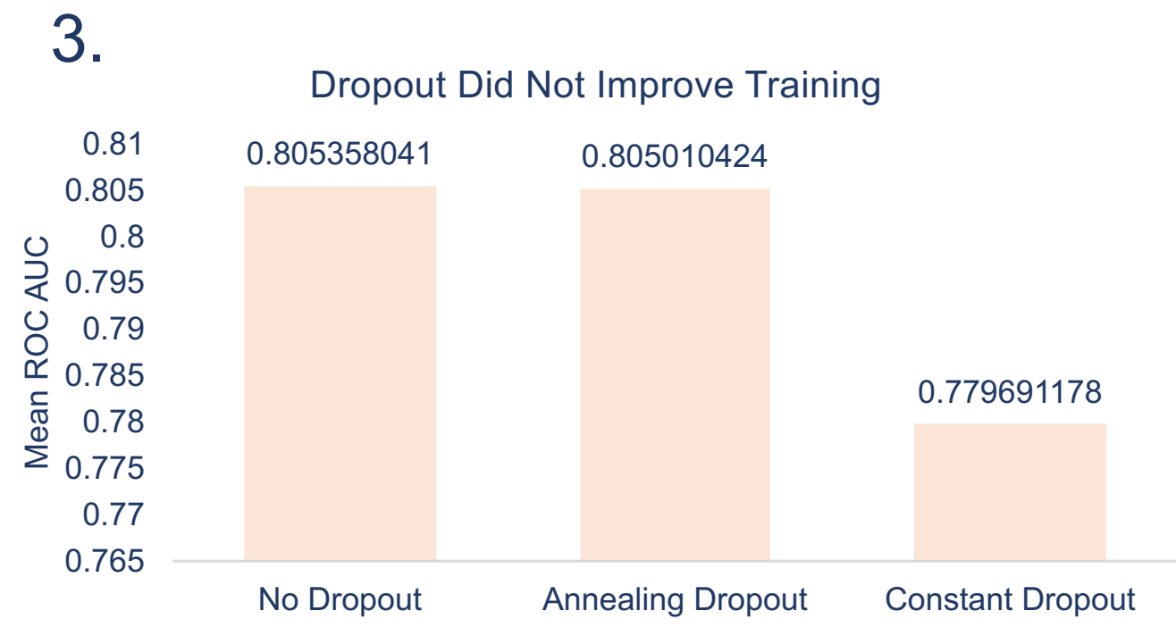
1.


Increasing the Number of Convolutional Layers Degraded the Classifier

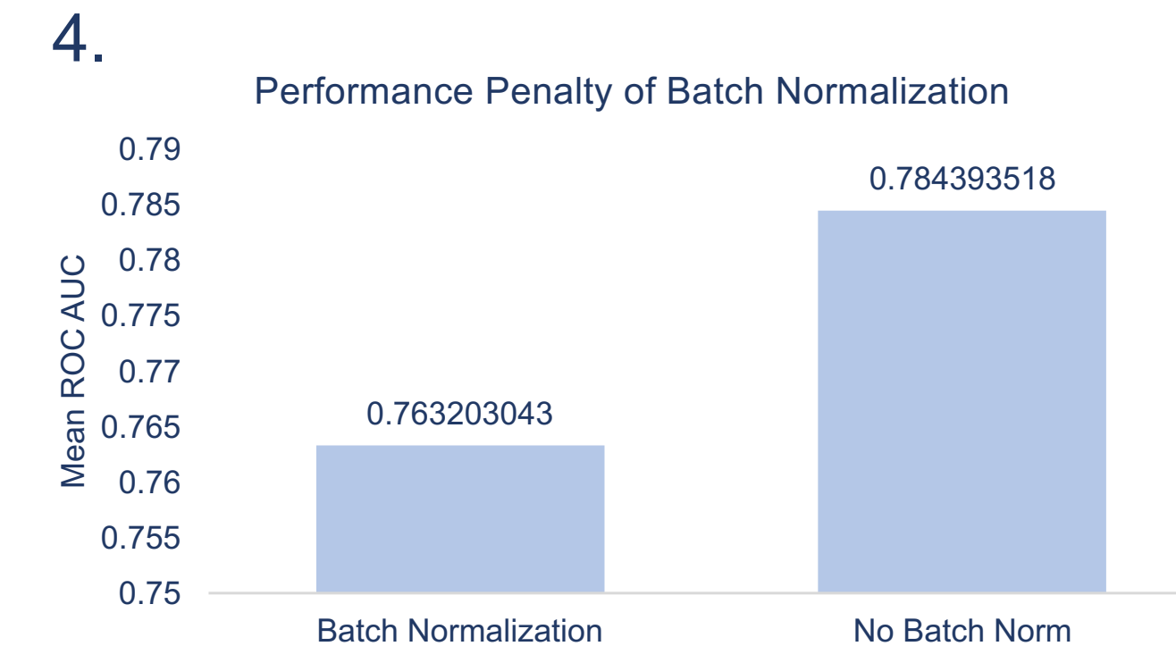
2.


Increasing Feature Maps Improved Accuracy

3.


Dropout Did Not Improve Training

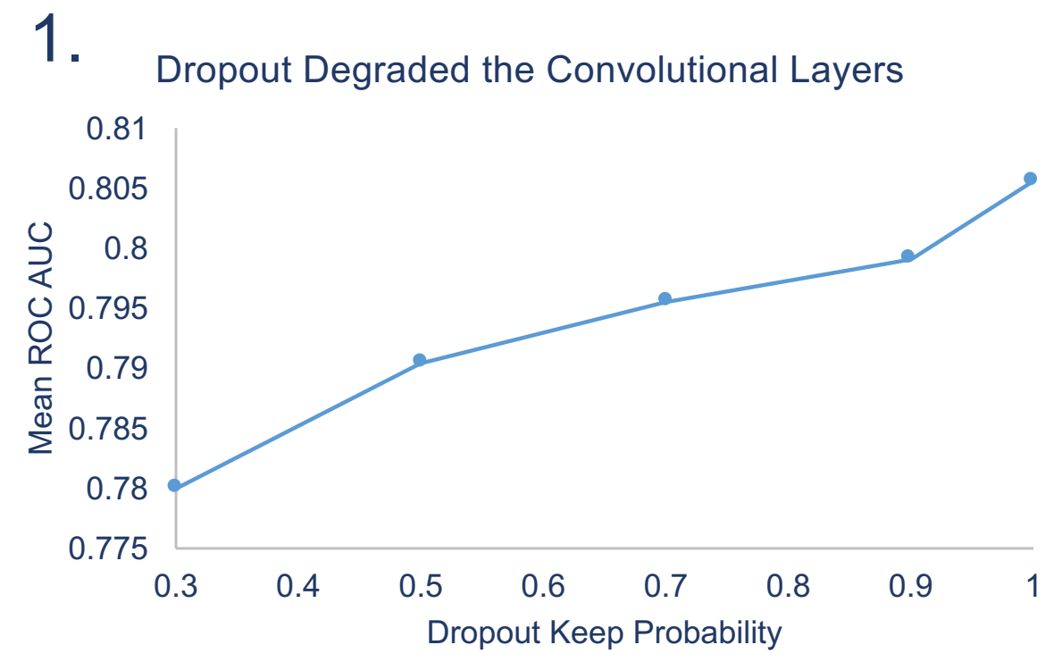
4.


Performance Penalty of Batch Normalization

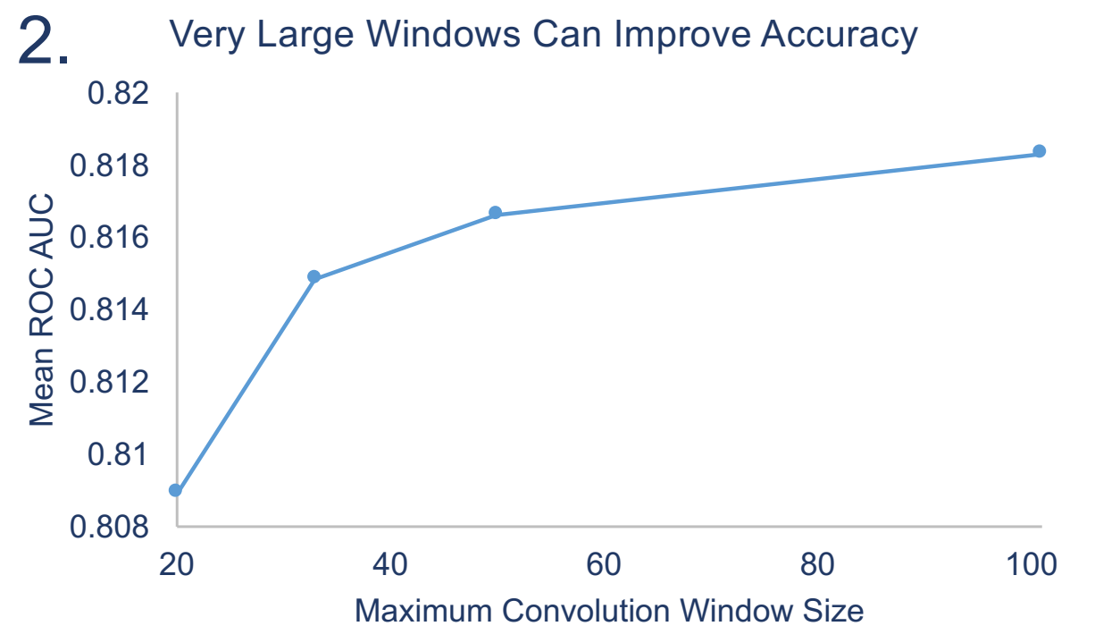
1) Increasing network depth did not always increase performance. The optimal depth was two convolutional layers
2) Increasing the number of feature maps in Architecture B exhibited diminishing returns in AUC.
   - Linear relationship between AUC and log(number of feature maps) in most of the tests
3) As with convolutional nets over 2-D images, applying dropout to the convolutional layers did not improve AUC; if anything, AUC was degraded. A scheme where dropout's keep probability was annealed to 1.0 during training was tested but failed to surpass the no-dropout tests
4) Batch normalization had a strong negative affect on AUC

Architecture C

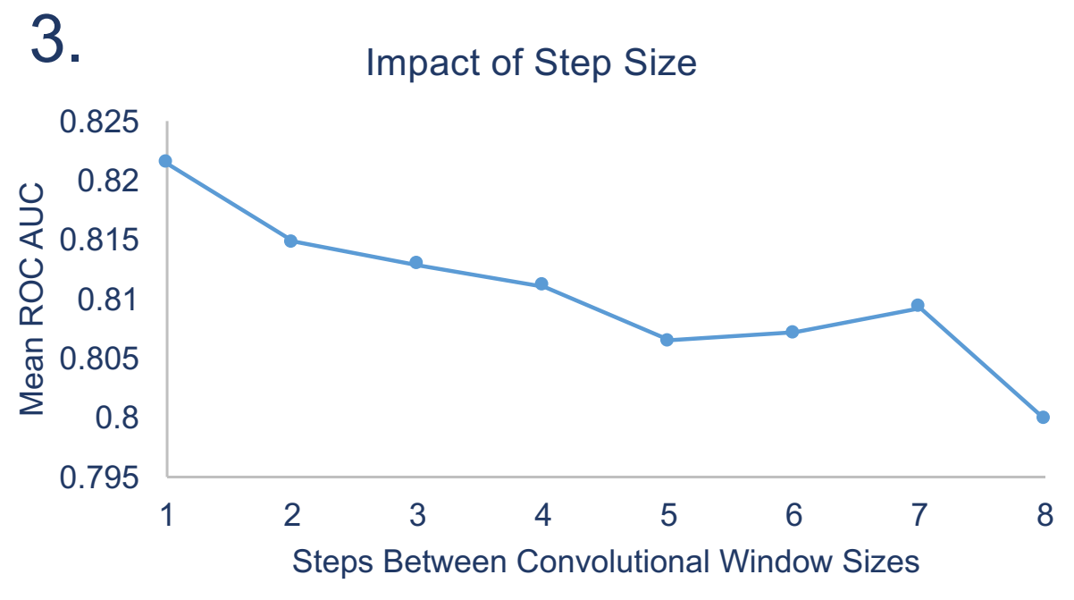
1.


Dropout Degraded the Convolutional Layers

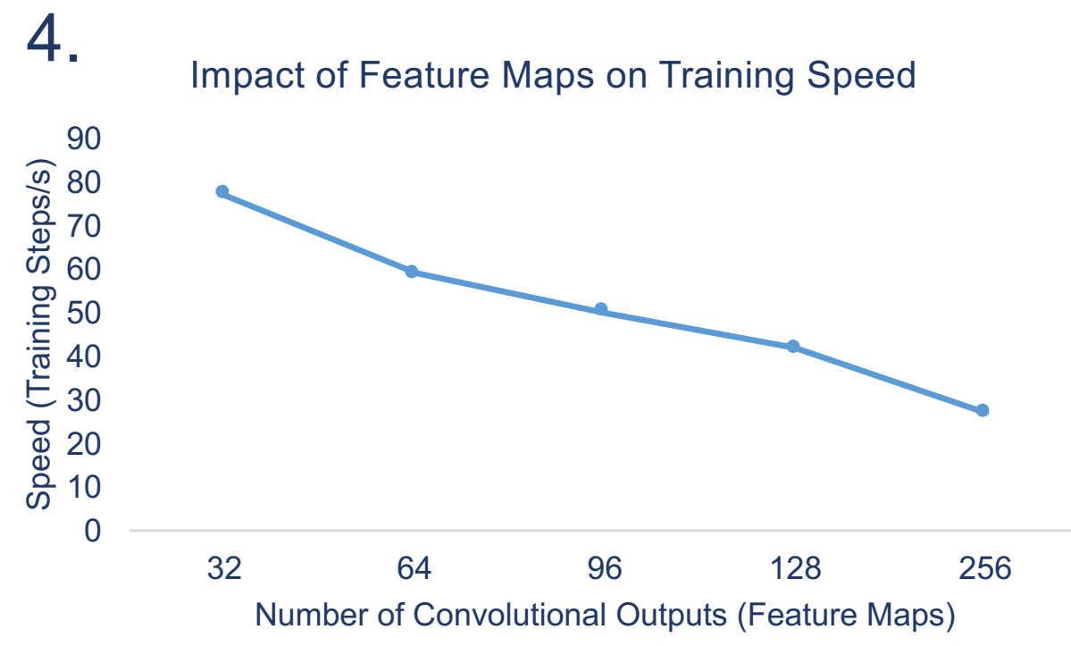
2.


Very Large Windows Can Improve Accuracy

3.

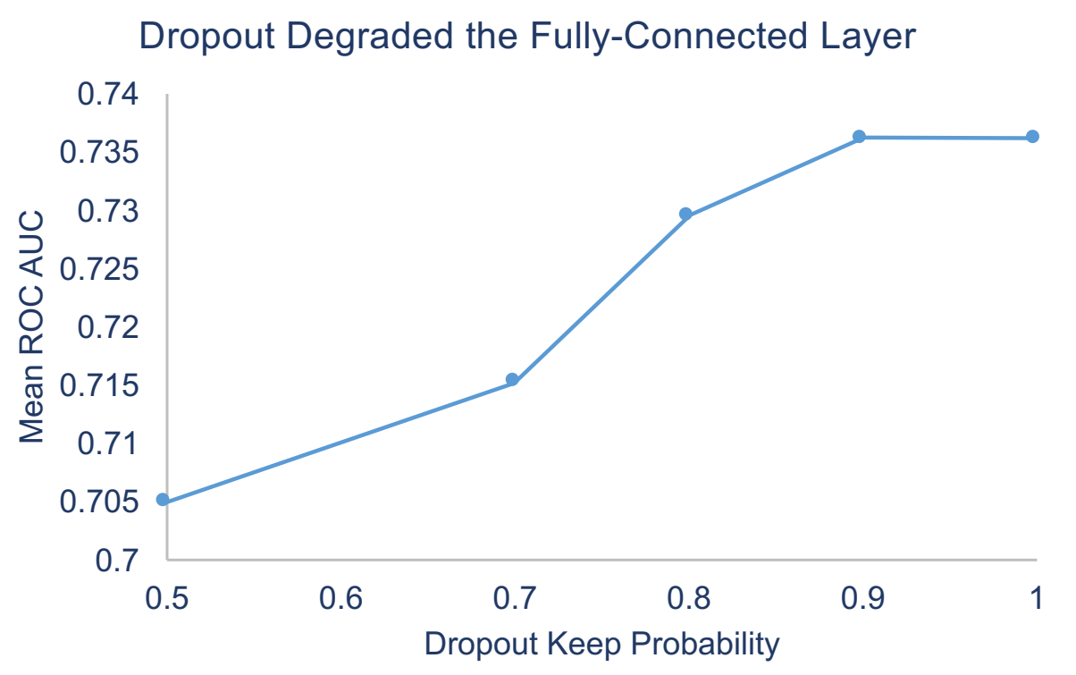

Impact of Step Size

4.


Impact of Feature Maps on Training Speed

1) Regularizing the convolutional layer with dropout strongly degraded the classifier performance
2) Surprisingly, increasing the max convolutional window size up to the whole length of the DNA sequence (101 bp) continued to improved classifier performance even though TFs bind to shorter sequences of <15 bp
3) Decreasing the window step size improved the classifier (but increased training time)
4) Training time also increased linearly with log(number of feature maps)

Architecture A


Dropout Degraded the Fully-Connected Layer

- Applying dropout to regularize the FC layer degraded classifier performance; the best results were obtained without dropout
- As with the other architectures, dropout also reduced AUC when applied to the convolutional layer

Results cont'd

Overall best results obtained with each architecture:

	Mean ROC AUC
Arch. A (Zeng, *et al.*)	0.7302
Arch. B (Lanchantin, *et al.*)	0.8276
Arch. C	**0.8297**

A. 1 convolutional layer with 128 feature maps and one fully-connected layer
   AUC *decreased* with more feature maps (AUC = 0.7020)

B. 2 convolutional layers with 768 feature maps each
   15 training steps per second on GTX 970

C. 99 convolution layers with windows from 3 to 101 in steps of 1 and 96 feature maps each
   But slower to train (< 6 training steps per second)

Conclusions

1. Global max pooling is the most important component of convolutional nets for FC binding prediction. Preliminary trials with standard MNIST-type convolutional nets were very poor (< 0.7 AUC) because they used the more typical local pooling method. *A bad network can be improved dramatically by switching from local to global max pooling!*
2. Omit fully connected layers. Despite the fact that Architecture A (Zeng, *et al*.) improved results over DNNs studied earlier in 2015, it performed much worse than a newer architecture that eliminated the FC layer (Lanchantin, *et al.*). Increasing beyond 1 FC layer did not improve AUC, either.
3. Drop dropout. The authors of architectures A and B both used dropout to reduce overfitting. Surprisingly, dropout in the FC layer of architecture A decreased AUC; less surprisingly, dropout also decreased AUC when applied to the convolutional layers of B and C as it does in many CNNs for images. Holdout sets were used
4. Use more feature maps. ROC AUC increased with greater numbers of feature maps in architectures B and C, however training time increased linearly; A was the only architecture hindered by additional feature maps
5. Use fewer than three convolutional layers of depth. It may be possible to improve results with more than two conv layers of depth, but not with these specific architectures
6. Use good coverage of all convolutional window sizes. Previous work has focused on windows < 24 bp. Evidence here suggests benefits from even very long windows (101 bp) relative to the length of TF binding sites (<15 bp)
7. Choose convolutional window sizes ≥ 3 bp. Window sizes of 1 and 2 base pairs did not improve classification
8. Generally safe to stop training at 7000 iterations
9. Also, this is the first work to test these methods for TF binding prediction in TensorFlow, to the author's knowledge ☺

Future Work

1. Try RNNs like LSTM for TF binding
2. Steal from NLP. Genome + TF binding is a language

References

1. Zeng, H., Edwards, M. D., Liu, G., & Gifford, D. K. (2016). Convolutional neural network architectures for predicting DNA-protein binding. *Bioinformatics*, *32*(12), i121–i127. https://doi.org/10.1093/bioinformatics/btw255
2. Quang, D., & Xie, X. (2016). DanQ: A hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. Nucleic Acids Research, 44(11), 1–6. https://doi.org/10.1093/nar/gkw226
3. Lanchantin, Jack, Ritambhara Singh, Beilun Wang, and Yanjun Qi. "Deep Motif Dashboard: Visualizing and Understanding Genomic Sequences Using Deep Neural Networks." arXiv preprint arXiv:1608.03644 (2016).
4. How Genes are Regulated: Transcription Factors. https://www.youtube.com/watch?v=MkUgkDLp2iE

www.PosterPresentations.com