Hide & Seq

- Input
  - Rather than a one-hot for each nucleobase, could have a different channel for each nucleotide.
- Hidden Layers
  - CNN
    - Tinker with smaller kernel sizes (27 seems like it'd be a lot)
    - Pooling layers are a relic of the past, better to use convolution layers with modified strides such that the condensation method of the feature-maps is also a learnable parameter.
    - Batch Normalization and kernel initializers for properly creating weights and avoiding covariate shift.
  - BiLSTM
    - Utilize global average pooling as opposed to a flatten layer to prepare the result of the BiLSTM for the FCNN input layer. The incredibly significant benefit of Global Average Pooling is in the number of parameters. Consider the following:

      $$\begin{gathered} Let\ M_{BiLSTM} = the\ output\ of\ the\ BiLSTM\ s.t.\ M_{BiLSTM}\ is\ a\ rank\ N\ tensor \\ i.e.\ dim(M_{BiLSTM})\ = D_1 \times D_2 \times\ \ldots\ \times D_N \\ Let\ flatten(M_{BiLSTM}) = \vec{F}_{BiLSTM} \\ M_{BiLSTM}\ has\ \prod_{i=1}^{N} D_i\ \#\ of\ elements = dim(\vec{F}_{BiLSTM}) = D \\ \implies\ FCNN\ relies\ on\ D\ features \end{gathered}$$

      However, if we choose to use Global Average Pooling, we find the following:

      $$\begin{gathered} GlobalAveragePooling(M_{BiLSTM}) = O_{BiLSTM}\ s.t.\ O_{BiLSTM}\ is\ a\ 1 \times 1\ \times\ \ldots\ \times D_n\ Tensor \\ \implies\ FCNN\ only\ relies\ on\ D_n\ features \end{gathered}$$

      which is significantly less parameters. In turn, we have much less of a probability of over-fitting with our FCNN, and can thus build a more complex BiLSTM or CNN.
  - FCNN
- Output
  - Soft-max yielding

    $$\begin{gathered} Pr(p_i)\ s.t.\ 0 \leq\ i \leq n-1\ where \\ n = the\ number\ of\ nucleobases\ in\ the\ sequence \\ p_i = promoter\ at\ the\ i_{th}\ nucleobase \end{gathered}$$

    and the model would then select

    $$p_j\ s.t.\ Pr(p_j) \geq Pr(p_k)\ \forall j,k$$