Lab2 report: Using ANNs to predict Protein Secondary Structure

Friday, February 17, 2017 8:51 PM

Team Member: Zirui Tao, Hongyi Wang

Description:

Our experiment using ANN to predict protein structure achieved overall above 0.6 accuracy. Also, some additional back propagation techniques were experimented during the training procedure and our team provided subsequent analysis and comments on these techniques for our specific topic of interest. The detailed specification is below:

- **Topology of the neural network:**

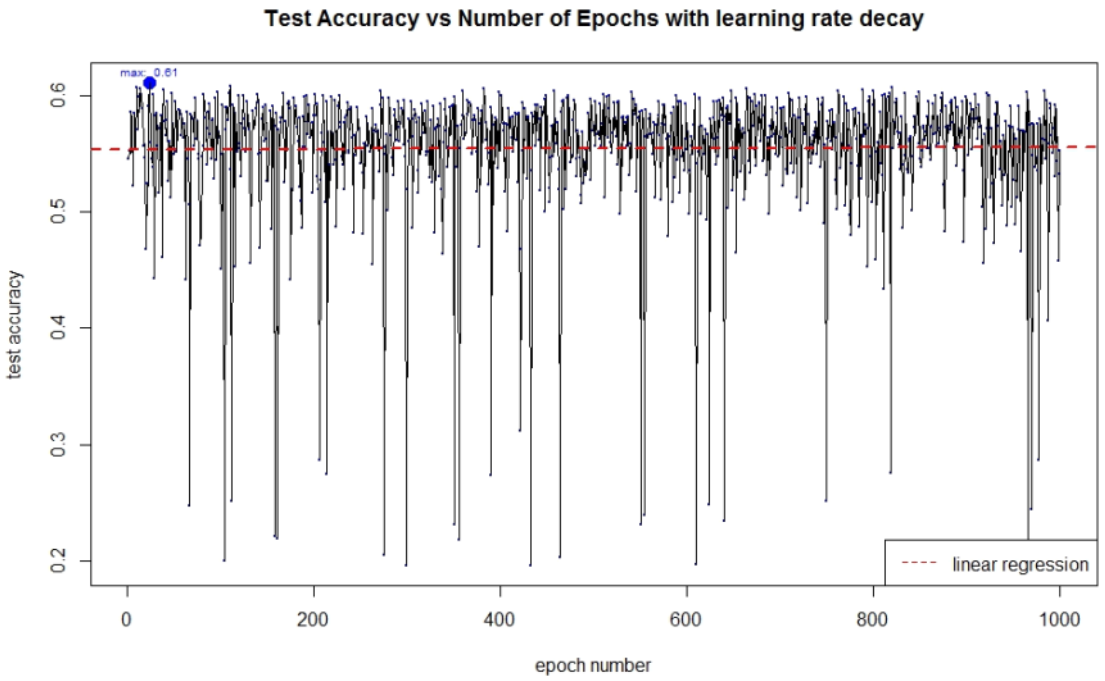
Our ANN consists 3 output perceptron, 5 hidden perceptron and 21*17 input perceptron. The number of output layer and input is fixed through the entire test stage since the domain knowledge of protein structure has "pre-defined" nature for these two layers: The sliding window is pre-defined as 17 and the selection of number of output obeys the fact that there are in total three different structures: $alpha('\_')$, $beta - helix('h')$ and $others$ $('e')$. For the hidden layer, we tried different numbers of perceptron numbers and found that perceptron number between 4-6 for the hidden layer generally obtain higher accuracy than other numbers between 1 and 10. One hot encoding was used for input vectors. There are 21 different amino acid values (20 possible inputs plus solvent for padding) thus the input dimension is 21 with only one being "1" and the rest being "0".

- **Activation functions:**

For activation function we adopted the suggestion in the lecture: RELU function for hidden nodes and sigmoid function for output nodes Activation function for Input layer is left out.

- **Number of Epochs:**

We selected 1000 as our number of maximum training epoch. However, from the observation of our test set accuracy. There is no apparent relationship between these two variables. We found that our plot of test accuracy vs number of epochs fluctuates, which we think is due to too drastic weight change in average. Below is the trending plot:



- **Testing methodology: Train, Tune and Test set and early stopping:**

Tune set accuracy is evaluated after each epoch and early stopping is implemented using different tune set accuracy. Each sample in tune set is unique and independent of others in train set and test set. Our team followed the suggested sampling strategy mentioned in the lecture: picking every other 5 protein strain as tune and every other 6 protein strain as test.

Note that at the start of each training step we shuffled the training to eliminate the biased sample in each set and the ordering effect on our learning model. Such shuffling strategy overall provided unbiased samples and guaranteed the randomness of sample.

- **Back-Propagation methodology and performance evaluation:**

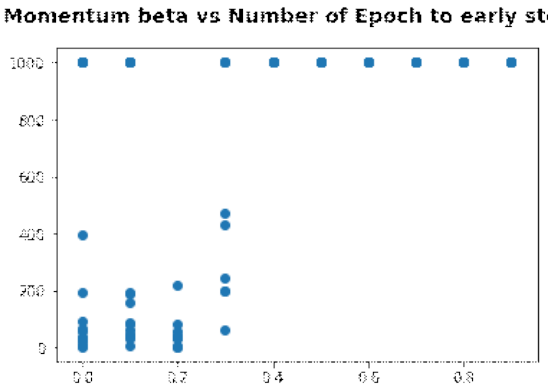
By performing Stochastic Gradient Descent (SGD) optimization, each time the weights change is associated with its inputs and connected perceptron derivative in next layer plus the current learning rate. After each forward propagation to output result, it is compared to the label, both in numeric vector format, and the difference is taken as an input to the back propagation. Each weight has its associated value to update dependent on the derivative of activation of perceptron's net input value it is connecting to. Therefore, it is also related to derivative of all perceptron and weight values ahead of itself since the propagation follows from output layer, hidden layer to input layer.

Choosing SGD is motivated by high cost of update after training the whole dataset. For this specific topic, our team think it is not even appropriate to have batch update (batch size more than 1). This is due to the fact that each input is an "one hot" encoded input, and it is hard to select representative input value when updating the weight value of input layer from a batch input feature vectors. Typical methodology of averaging all the inputs is not feasible and might lead to overfitting. It is also worth noting that updating according to non-standard input (not in "one-hot" format) contributes little to the model learning in the process of training.

- **Addition to back propagation: Momentum , Dropout, Weight Decay and comments about these three strategies:**

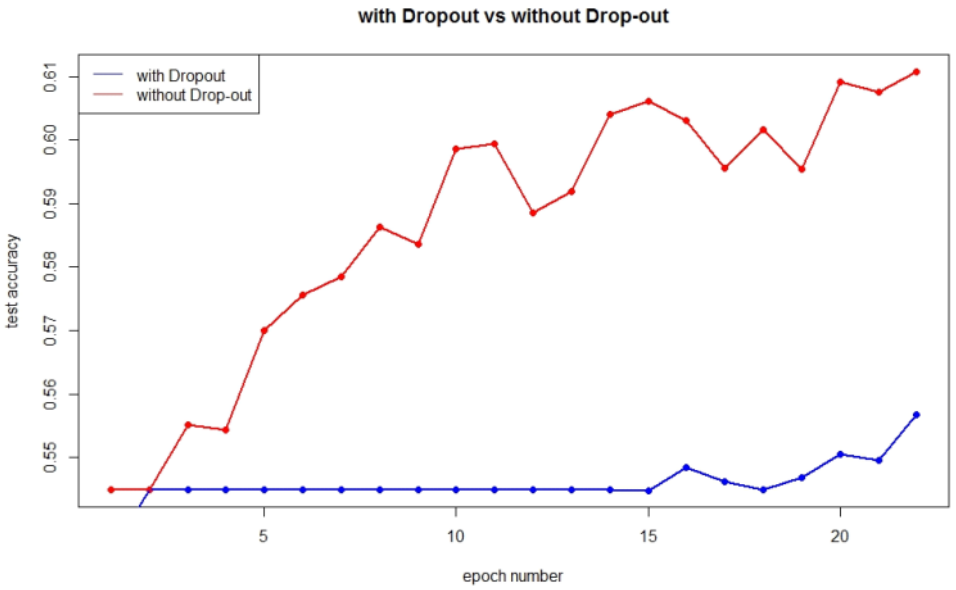
In addition to apply standard SGD, we tried momentum, term, dropout and weight decay in order to increase the speed and quality of training Momentum is summarized as combining the current Δw_{ij} to $\Delta w_{ij}'$ that was evaluaed at last training time. This helps faster convergence since the net weight update is the vector sum of these two vectors, giving large update at each time.
However, in order to reduce the overfitting, Weight Decay and Weight Dropout are used.
Weight decay is to incorporate the factored current weight term into Δw_{ij} as a partial factor for updating the current weight. The idea is to prevent convergence caused by overfittig, which makes each update trivial after certain number of training epoches.
Weight drop out is to change the topology of original network to prevent machine being to

o smart at leverage the trainning set. The use of Momentum facilitates the weight updates and thus stabalization of tune accuracy. Weight Decay in our samples also does little contributiton as we found that the



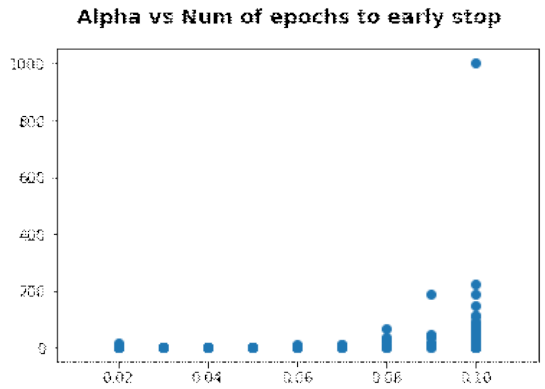
mean accuracy without weight decay	mean accuracy with weight decay 1
0.54	0.56

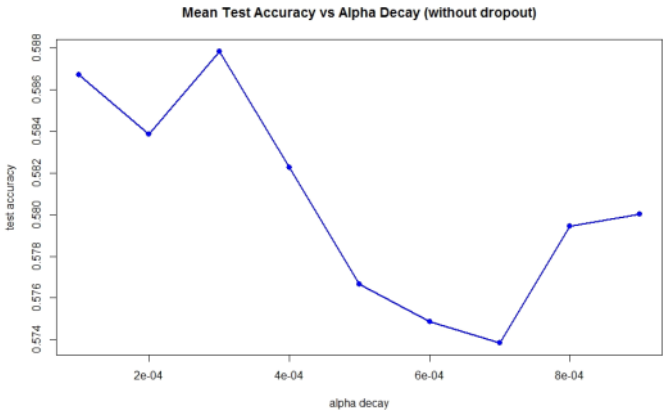
Noticed that Weight Dropout is also for preventing overfitting, but in different approach. As explained by Hinton's Group, it randomly turn off a fraction of the input and hidden units, before each sample being passed and the weights gets error back-propagated. In tune set and test, all of weights will be scaled by a factor that sums to fraction to 1, in order to normalize the expected weighted sum. Whereas Weight Dropout as an ensemble approach training an collection of ANNs, with weights getting dropped randomly among input units and hidden units, with unique, subset (or mini-batch) of training sample, an form of bagging strategy, and voting the results for predicting tune sets and test sets accuracy. It is equivalent to sotre $O(2^n)$ network in 1 collection. However, during our experimentation with Dropout strategy, the result is not ideal for either of methods. Both strategies downgraded our test accuracy and tune accuracy behaves unstable during the whole training. Therefore we concluded that standard Weight Dropout is not useful for this pretein prediction task. It is likely, however, to customize an Weight Dropout strategy that would improve our training performance (e.g with some specific domain knowledge).



- **Learning Rate annealing:**

Except back propagation methodology mentioned above, we also implemented the learning rate annealing as a function of training epochs. We found that our current version of learning rate annealing do slightly contribute (around ~2% in average) after some comparison testing.
More detailed implementation can be set such as only applying the learning rate change when the accuracy converges (changes between previous tune accuracy and current accuracy is below a small threshold), which generally gives good local optimum. We adopted alpha as 1/(1+decay* epoch_num), where decay is a small constant





- **Ensemble:**
  Our team also tried to utilize a network of ANNS to achieve better accuracy.
  However, even though we used excessive number of ANNs, 20 of them, we did not see huge improve on the test set accuracy thus we don't over emphasize the topic in this report.

- **Summary:**

By combining the domain knowledge and properly implementing back propagation algorithm with appropriately trying additional techniques. Our team has achieved 0.635 as out highest test accuracy during the whole experimentation and report overall trend of the effect of each strategies to test accuracy. We expected that higher accuracy could be achieved if we experimented more cases and parameter values or refine our detail implementation of algorithms such as learning rate annealing and weight dropout. Other strategies and algorithms are also welcomed to test for improving the overall test accuracy of this laboratory task.