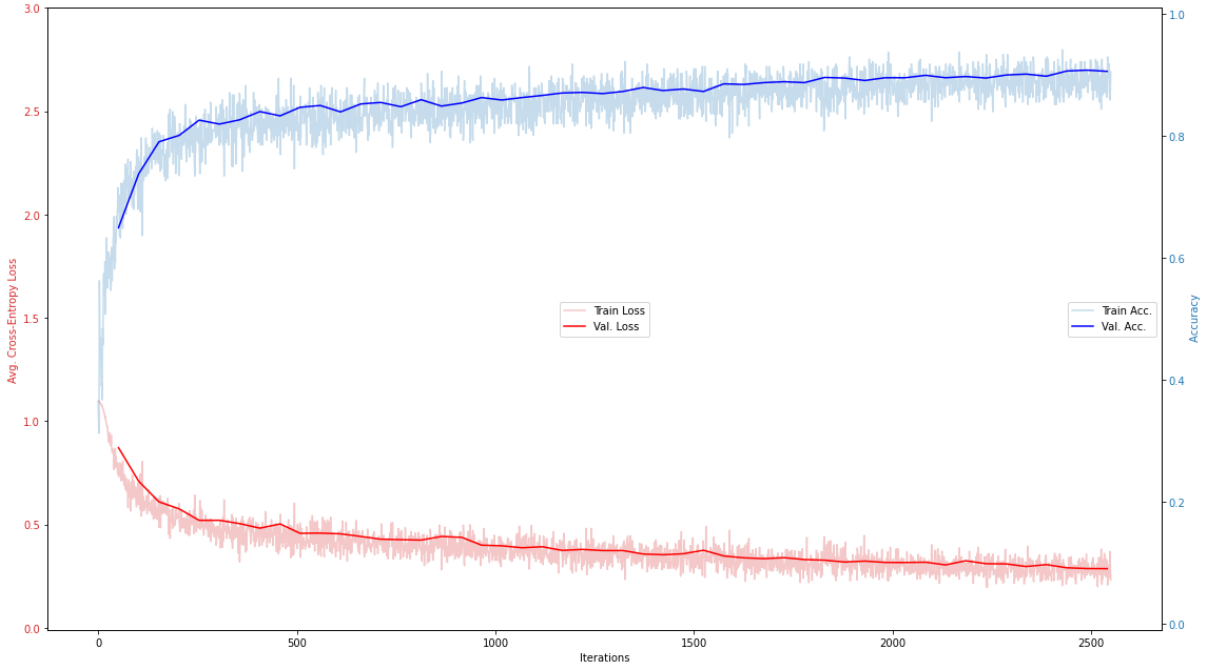
AI535 - Report of Homework 3

Bikram Pandit

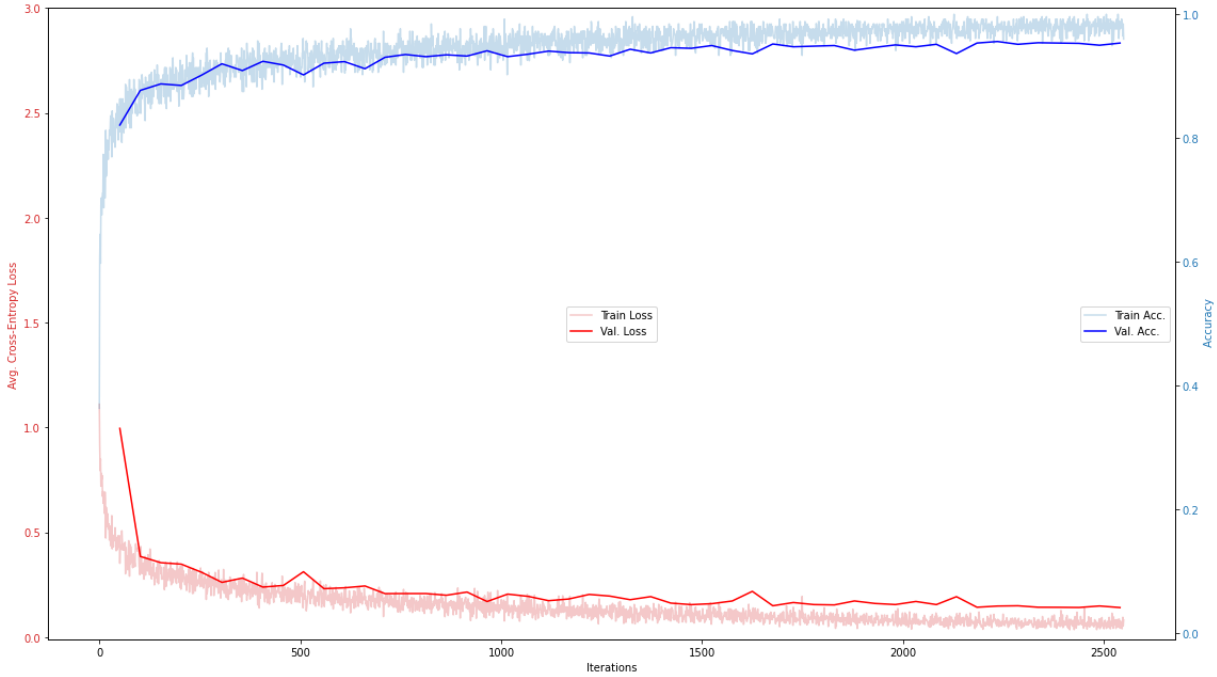
I. Convolution neural network

Question 2:

Convolution neural network **without batch normalization**

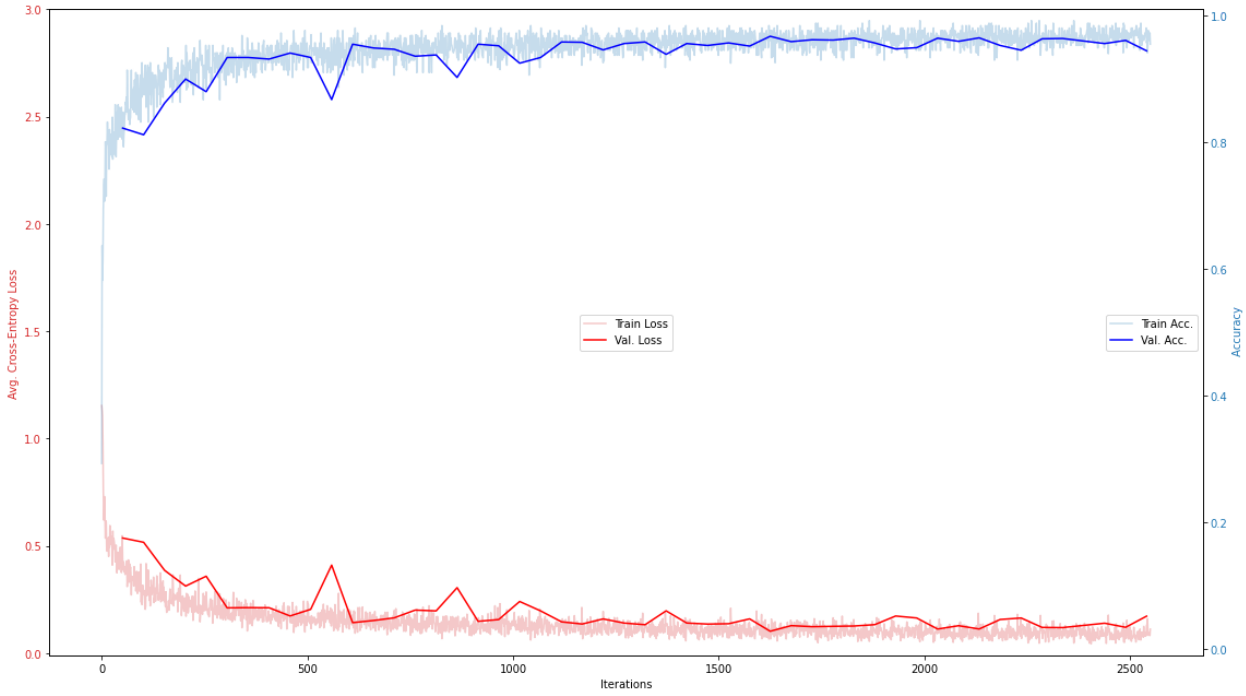


Convolution neural network **with batch normalization**



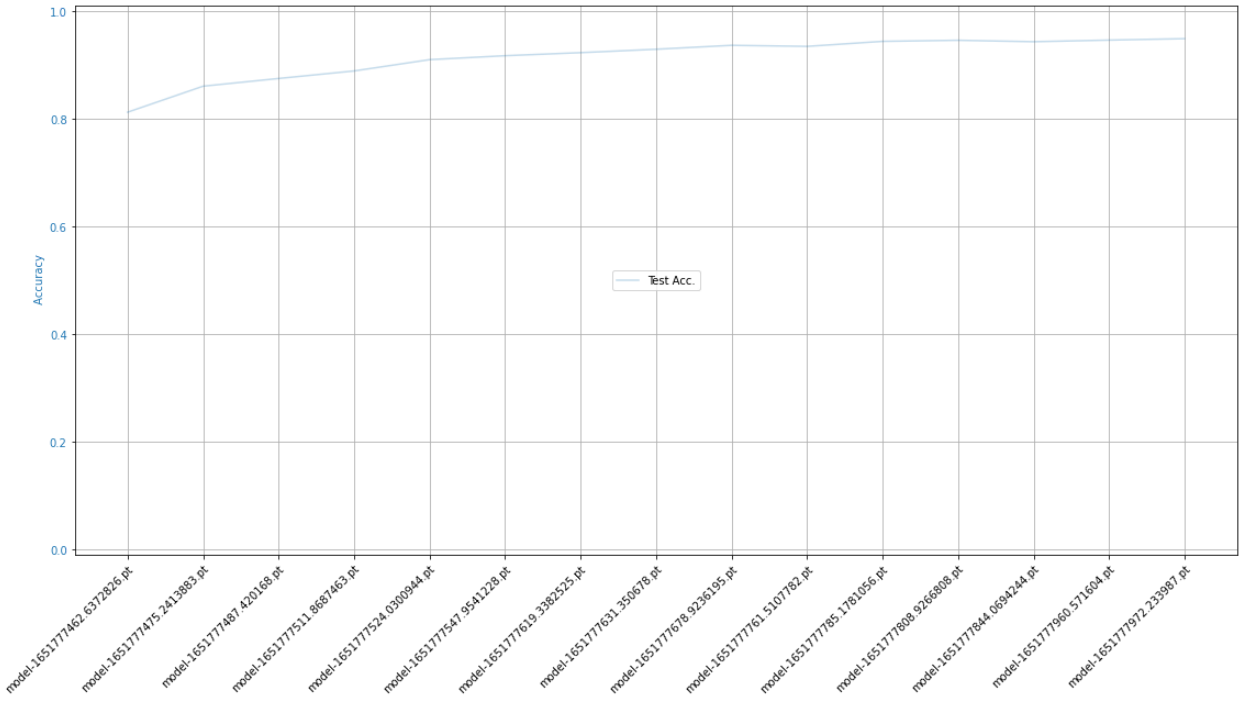
The first difference we observe is that with batch normalization we have significantly improved the validation accuracy. Batch normalization tends to make solve covariate shift problems and help to train a model easier and faster. On top of that, we may increase the learning rate and still get decent results when using batch normalization. The following graph shows accuracy when the learning rate is reduced to 0.003 from 0.0003.

Batch normalization and the learning rate is increased



Question 3:

The following plot shows the accuracy of a test set over the epochs evaluated with saved models. These models are saved whenever validation accuracy is higher than the last saved model.



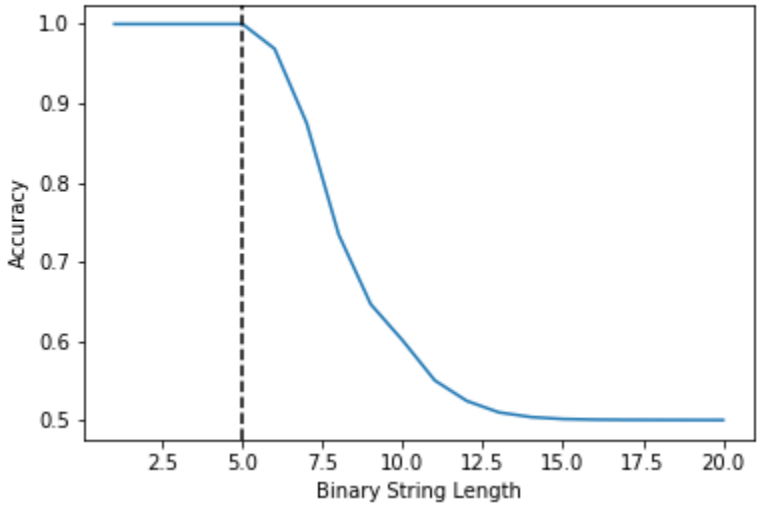
I. Simple Sequence Models for Parity

Question 5

[Default settings]
Maximum_training_sequence_length = 5
Layers = 2
Hidden_size = 64
Learning rate = 0.003
Weight decay=0.00001

As indicated above, I used the default hidden dimension and number of layers to train a model. This model produced 100% train accuracy and 100% validation accuracy until the maximum training sequence length. After that, we see a gradual decrease in the accuracy of our validation set when as sequence length exceeds. Further experiments were performed (Question 7) to reduce the model complexity yet produce better results.

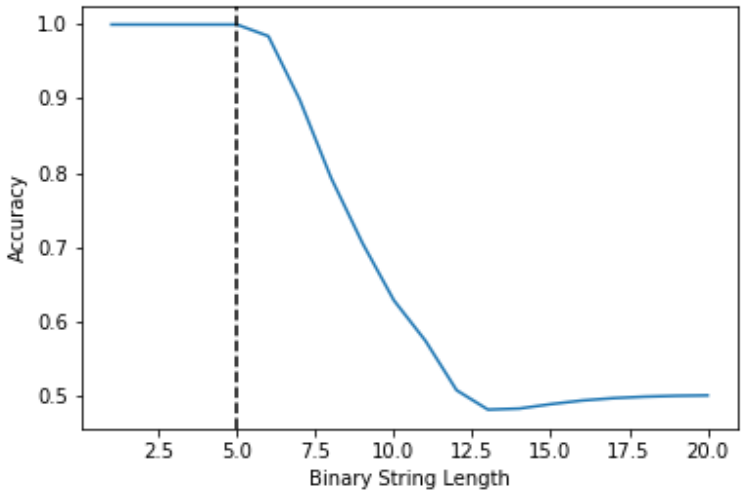
Question 6



It is easy to think why we achieved 100% validation accuracy for sequences length less than or equal to the maximum sequence length. This is because our training set contains exactly the same data as our validation set up to that length. However, once the length of the validation set exceeds, it didn't do a good job. We can say generalization was poor. Next, we perform different experiments to see if we can improve our accuracy.

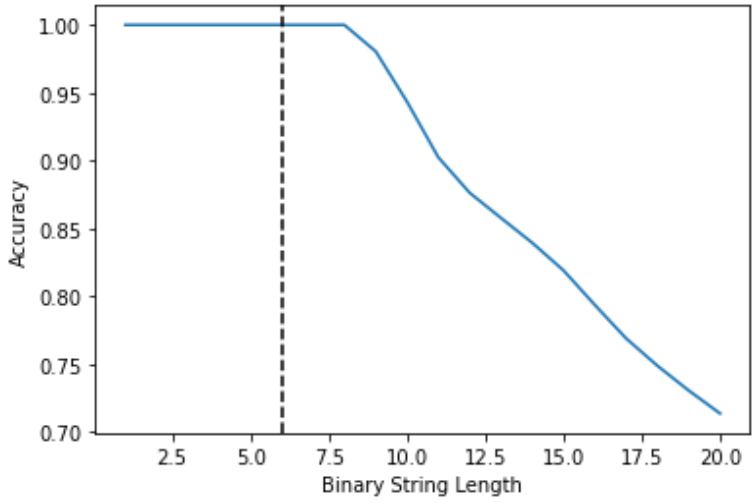
Question 7
Experiment 1 (No improvement)

Maximum_training_sequence_length = 5
Layers = 2
Hidden_size = **128 (changed)**
Learning rate = 0.003
Weight decay=0.00001



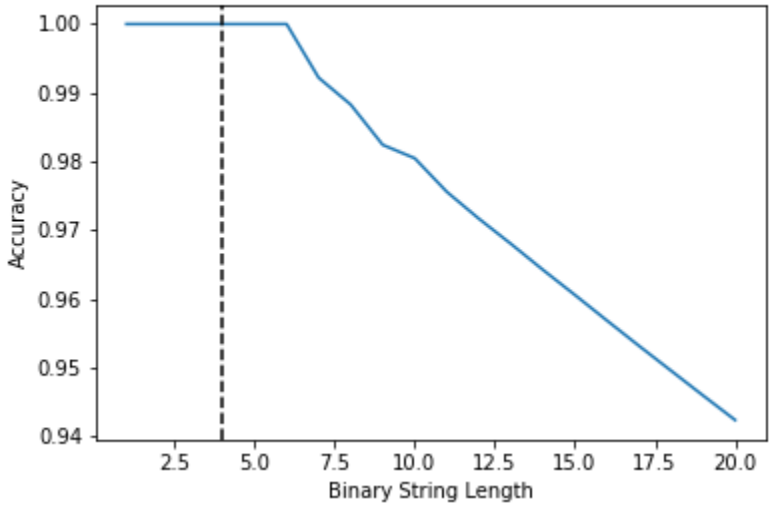
Experiment 2 (Improvement but feels cheating)

Maximum_training_sequence_length = **6 (changed)**
Layers = 2
Hidden_size = 64
Learning rate = 0.003
Weight decay=0.00001



Experiment 3 (Significant improvement)

Maximum_training_sequence_length = **4 (changed)**
Layers = 2
Hidden_size = **1 (changed)**
Learning rate = 0.003
Weight decay=0.00001



We can see in **experiment 1** that increasing the hidden dimension didn't help in generalization.

In **experiment 2**, we can see it did slightly a better job, however that is always true to happen if we increase the maximum sequence length for training. For example, if the maximum sequence length was 20, validation accuracy would be 100% but it doesn't mean that the model is good.

However, in **experiment 3**, we can see with a simple model, we achieved almost generalization. This could be analogous to the network we've created by hand where for parity check, a dense network is not required. With a smaller training sequence length, our model might've learned only to consider bits in narrow window size which is actually true for parity check. However, this is my rough hypothesis.

According to all experiments performed, the smallest size for hidden state size is 1 for which the training accuracy converges at 100% and validation accuracy is higher for all string lengths as compared with other experiments.

II. Debriefing

1. Approximately how many hours did you spend on this assignment?
    a. I think I spent around 8 hours.
2. Would you rate it as easy, moderate, or difficult?
    a. I think it was moderate.
3. Did you work on it mostly alone or did you discuss the problems with others?
    a. I did this assignment alone.
4. How deeply do you feel you understand the material it covers (0%–100%)?
    a. I think 95% would be my rough guess.
5. Any other comments?
    a. No comments.