Final: Deep NLP

Alex Shah

12/7/17

1 Strategy

My strategy to determine which parameters to change was to research and recall the effects of each parameter on the data. For example I knew batch size would have little effect on the end results but may speed up the processing of the dataset, depending on the hardware. Since time was not of the essence to train a few epochs, I decided to focus my attention on other parameters. In this model, the only parameters which have an affect other than epochs, are the hidden units, number of recurrent steps, number of layers, and keep probability.

2 Most Influential Parameters

I knew right away that some of the most influential parameters would be the number of hidden units and number of layers. Increasing these parameters would increase the overall computation space therefore increasing training time, but would most likely improve perplexity (ie decrease the value). I did not expect the best results to be from increasing the number of steps to 50. In retrospect it does make sense that increasing the number of recurrence steps would improve results. By combining the three most influential parameters the results improved even more dramatically. I also did not expect that the number of layers made nearly no difference to the training, validation, and test perplexities. The only change made by increasing the number of layers was lowering the initial perplexity which quickly fell into step with the default run.

3 Least Influential Parameters

Most likely this code was not optimized for changing the dropout layer. It was intially set to 1 and the results of decreasing the keep probability were adverse at worst and roughly the same at best. It would not effect results to change some of the other paramaters, other than causing vanishing/exploding gradient or other related problems. Initial weight scale would not change the results. IIncreasing the initial learning rate would make results worse until the learning rate begins to drop, and starting low would adversely affect the model. The max gradient norm just prevents against exploding gradient. Epoch related parameters involve the number of epochs at the initial training rate, then the amount of training epochs. As with any mode, more epochs would approach a better perplexity at the cost of time. Batch size, vocab size, etc, are related to processing the data and would not affect results. So the least influential of the parameters which affect perplexity was keep probability, which was only 2.

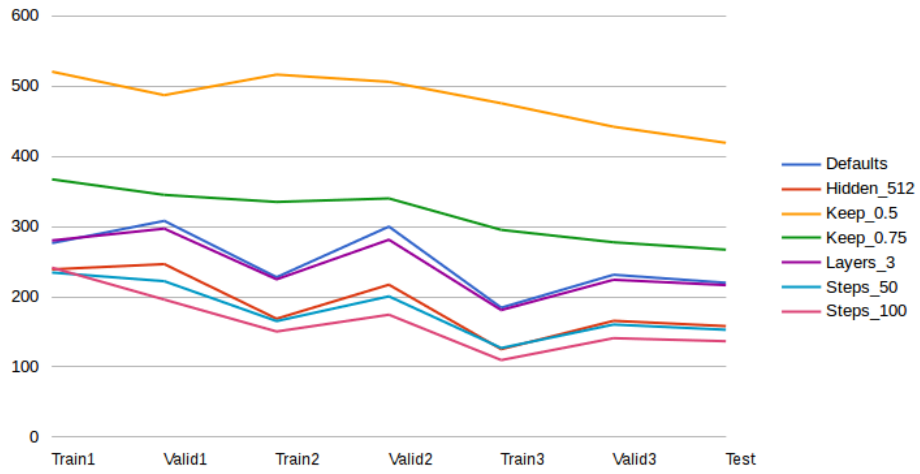

Figure 1: Test parameters over third epoch

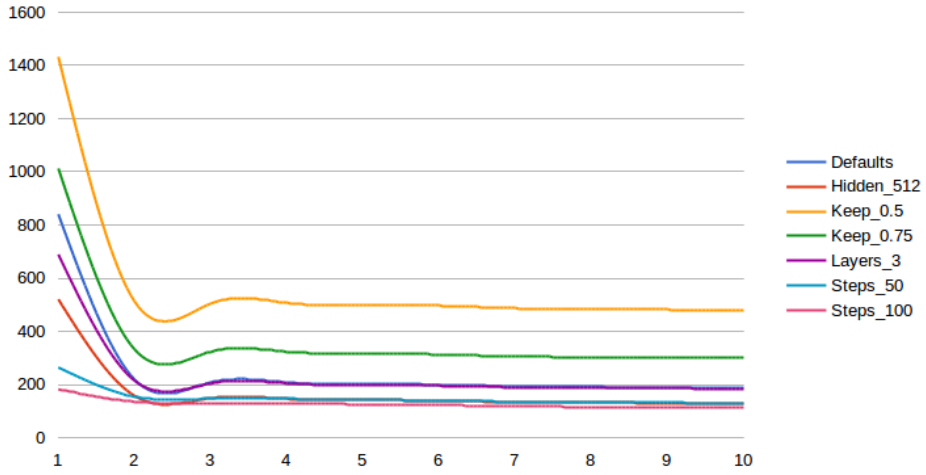

Figure 2: Training, Validation, and Test Perplexity

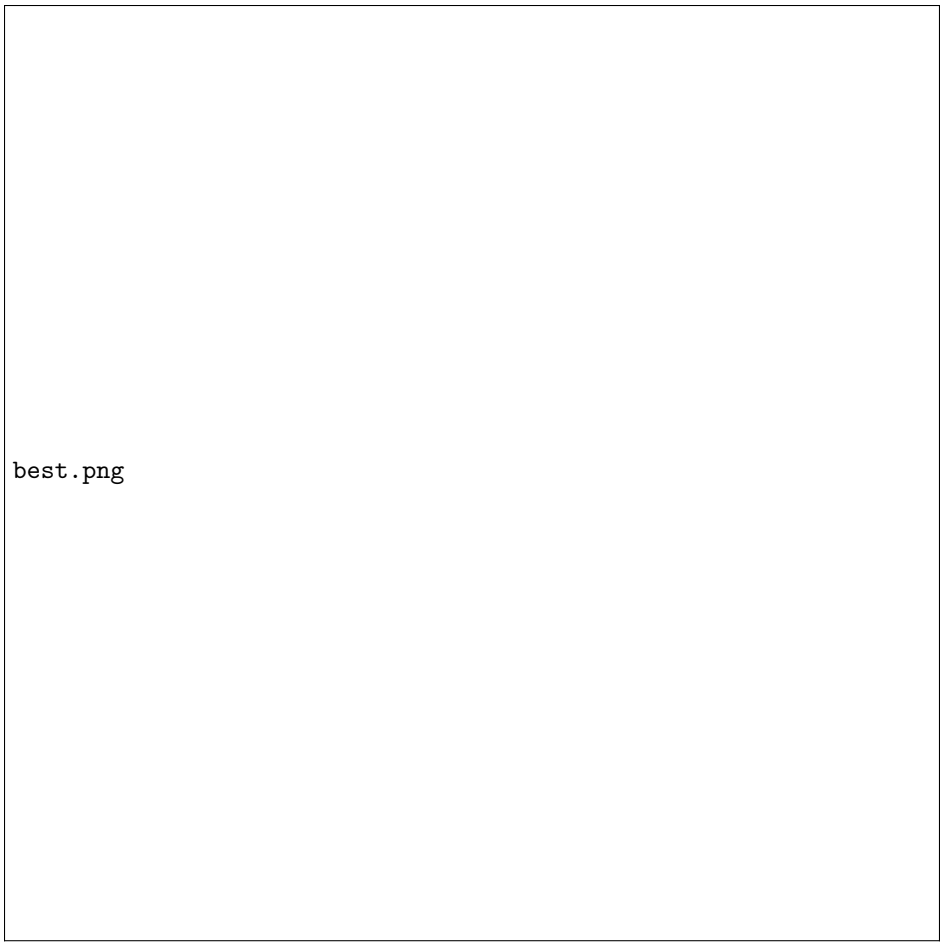

Figure 3: Combining Best hyperparameter changes

4 Best Runs Combined

5 Sentences

The default settings produced the following output:

Epoch 1 : Learning rate: 1.000 Epoch 1 : Train Perplexity: 273.428 Epoch 1 : Valid Perplexity: 311.307 Sample sentence: the company will be used in a ¡unk¿ test in a ¡unk¿ test in a ¡unk¿ test in a ¡unk¿ test Epoch 2 : Learning rate: 1.000 Epoch 2 : Train Perplexity: 224.126 Epoch 2 : Valid Perplexity: 294.604 Sample sentence: the company said ¡eos¿ ¡unk¿ said it will be used in a ¡unk¿ test in a ¡unk¿ test in a ¡unk¿ Epoch 3 : Learning rate: 0.500 Epoch 3 : Train Perplexity: 181.681 Epoch 3 : Valid Perplexity: 231.615 Sample sentence: the company said ¡eos¿ ¡unk¿ said it will be a ¡unk¿ in a ¡unk¿ ¡unk¿ in a ¡unk¿ ¡unk¿ in a Test Perplexity: 220.851

The sentences the default settings generated seemed to get stuck in a loop. Maybe from the limited number of epochs it was unable to capture significant information to make 20 words worth of text. The initial 3 to 5 words were more sentence-like, but quickly reverted to repetition.