Chris Winsor
UMass Lowell COMP 5300
Natural Language Processing
2/27/2023

Homework 4 - Torch Text Classifier

WandB runs

WandB runs are at https://wandb.ai/metrowest/hw4_nn_text_classifier.
Best run is https://wandb.ai/metrowest/hw4_nn_text_classifier/runs/a9d8vxjl

Summary:

Architecture was linear input and output with (n) hidden layers each having Batch Norm, Linear, ReLu and Dropout in that order. With experimentation our best performing model reached 88.4% validation and 84.44% test accuracy at 8000 global_steps.

- TfidfVectorizer with unigram, limited to 30,000 terms.
- 3 num_hidden_layers of size 30.
- Batch Norm and Dropout
- Xavier initialization for weights
- Adam optimizer with learning rate of 1e-5

Takeaways:

- Overfitting was apparent even in models with few hidden layers.
- Xavier weight init worked well, where random weight init resulted in diminishing gradients
- Batch normalization seemed to stabilize the training somewhat
- Language model, namely dictionary size remains important

Experiment Sequence

We experimented on various areas of the model.

Language model was the first area of focus. For these tests the model was 3 hidden layers of size 30, no batch_norm, Adam, LR 1e-5

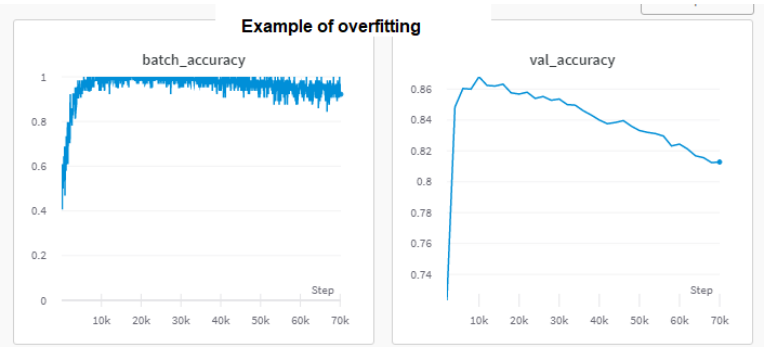
- Unigram 500 words (82%)
- Unigram 2500 words (85%)
- Unigram 5000 words (84%) with a noticeable slowdown in performance
- 4-gram 5000 words (84%) - here the preprocessing time gets long

We concluded (and later revisited this below) that larger dictionary and multi-grams did help.

Hidden Layers were the next area of experiments - namely the number of hidden layers and their size. All tests with unigram 2500 words, Admm, LR 1E-5, no batch_norm.

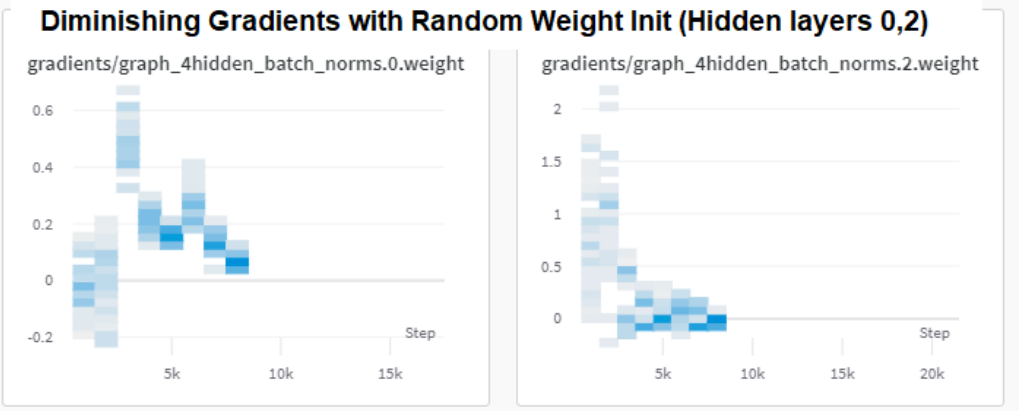
- #Hidden Layers, Size Val Accuracy (%)
- 0 hidden layers 84.20
- 1 hidden layer size 10 83.76
- 1 hidden layer size 30 84.28 (starts overfitting at 28K batches)
- 1 hidden layer size 70 84.56 (starts overfitting at 28K batches)
- 3 hidden layers size 30 84.16

Overfitting occurred in nearly all models. We conclude the model is very powerful relative to the dataset.



We explored **Learning Rates** finding above 1e-3 caused our Adam optimizer to not converge. We experimented with **SGD optimizer** and observed the default **momentum** of 0.9 needed to be smaller to get that to converge (we used 0.2).

Random weight initialization (instead of Xavier) resulted in diminishing gradients in the hidden layers.



We added **Batch Norm** while using same settings as above, observing a modest stability learning rate.

- #Hidden Layers Val Accuracy
- 1 hidden layer size 10 82.50
- 1 hidden layer size 30 84.60
- 3 hidden layers size 30 84.56

We weren't making progress on accuracy and felt our model was not challenged bv the 2500 word vocabulary. So we increased the vocabulary to 10k, then 30K. We ran 3 hidden layers of size 30 still using Batch Norm, Admm, LR 1E-5.
This resulted in our best result to date at 86.8%

- #Configuration Val Accuracy
- 1 hidden layer size 10 85.4%
- 3 hidden layers size 30 86.8% <--- run 93
- No stop-words, unigram, 10000 words, 3 hidden layers size 30 86.6%
- **No stop-words, unigram, 30000 words, 3 hidden layers size 30 88.4% <--- Run 96**

With this larger dictionary and combination of parameters we proceeded to the test dataset using the configuration from Run 96. **Test accuracy was 84.44%.**