CS 577 s21 – Assignment 3

Due by 3/23/2021

- In this assignment you will explore the topics of: loss, optimization, and regularization. You will need to implement several Neural Networks on GPU using a singularity container to solve classification and regression problems. This assignment will use the Xsede GPU cluster. The implementation should generally be done in Keras but you may use TensorFlow or PyTorch If you are already familiar with them and would like to use them. Note that we may not be able to offer help with frameworks other than Keras.
- Follow the submission instructions of the first assignment.

Theoretical questions

Loss

1. Write the equations for L1, L2, Huber, and Log-cosh loss functions and compare them. Explain the advantage or purpose of each loss function.
2. Write the equation for cross-entropy loss and explain how it is derived using maximum log-likelihood. Explain the worst cross-entropy loss value you expect for random assignment.
3. Write the equation for softmax loss and describe when to use it.
4. Write the equation for Kullback-Liebler loss and explain its meaning. Explain the circumstances under which there is no difference between using cross-entropy or Kullback-Leibler to train the network.
5. Explain Hinge loss and squared Hinge loss. Describe the fundamental idea behind it and the worst value you expect for it before learning.
6. Compute the hinge loss for a 3-class classification problem with three examples $x,^{(1)} x,^{(2)} x^{(3)}$ having labels $y^{(1)} = 1, y^{(2)} = 2, y^{(3)} = 3$, with prediction scores $\hat{y}^{(1)} = (0.5, 0.4, 0.3), \hat{y}^{(2)} = (1.3, 0.8, -0.6), \hat{y}^{(3)} = (1.4, -0.4, 2.7)$ representing distance from 3 decision boundaries (one-against-all-others).
7. Explain the purpose of adding a regularization term to the loss function. Explain the difference between L1 and L2 regularization and how they affect the weight distribution in the network. Explain the way to choose the regularization term coefficient.

8. Explain how L1 and L2 loss terms affect gradients in the network.
9. Explain the difference between kernel, bias, and activity regularization.

Optimization

1. Explain the advantage of back propagation over direct numerical computation of gradients (using the definition of partial derivatives). What is a possible use of direct numerical computation of gradients?
2. Explain the difference between stochastic gradient descent (SGD) and gradient descent (GD). Which one is expected to converge faster and why?
3. Explain the tradeoff in selecting the batch size for SGD. Explain the 4 main problems with SGD.
4. Explain how SGD with momentum addresses poor conditioning, minimum/saddle points, and noisy gradients.
5. Explain the advantage of the Nesterov Accelerated Gradient (NAG) momentum vs simple momentum. Explain the term "accelerated" in NAG.
6. Explain possible strategies for learning rate decay.
7. Explain how the learning rate is computed using Newton's method and explain the meaning of the Hessian matrix.
8. Explain the condition number and what happens when there is poor conditioning.
9. Explain the way in which the AdaGrad algorithm approximates the inverse of the Hessian.
10. Explain the problem with AdaGrad and how RMSProp addresses it.
11. Explain how the Adam algorithm combines RMSProp with momentum. Explain the need for a bias corrected term.
12. Explain gradient descent with line search and the bracketing algorithm. Describe an alternative to bracketing and the advantages/disadvantages of bracketing compared with this alternative.
13. Explain quasi-newton methods. What is the advantage of the BFGS algorithm over Newton. What are the advantages/disadvantage of BFGS compared with Adam.

Regularization

1. Explain how weight decay is related to adding a regularization term to the loss function.
2. Explain how early stopping to prevent overfitting is performed. Explain the strategies to reuse the validation data.
3. Explain how data augmentation is performed and how it assists in preventing overfitting.

4. Explain how dropout is performed. What are the advantages/disadvantages of droupout?

5. Explain how the expected value of all combinations of dropped out networks can be approximated efficiently during testing.

6. Explain how batch normalization is performed during training and during testing. In what way doe batch normalization introduces randomness into training?

7. Explain the purpose of scale and shift parameters in batch normalization. What are the values of scale and shift parameters that will cause the normalization to be canceled? Explain how the scale and shift parameters can be learned and what is a good initial value for them.

8. Explain how ensemble classifiers can assist with overfitting. Describe are possible strategies for producing ensemble classifiers.

Programming questions

1. Identify a data set from the UCI repository[1] that you will use for multi-class classification and another data set that you will use for single output regression.

2. Load the data sets, clean them as necessary, vectorize them, and split into train/val/test sets.

3. Evaluate different loss functions on your two data sets. Make sure that the loss function you choose is appropriate for the task at hand. Perform hyper parameter search as needed.

4. Evaluate different oprimizers on the data sets you chose. Measure the number of epochs and time needed to converge without overfitting using each of the optimizers and compare the results you obtain.

5. Evaluate the effect of different regularization measure on the performance of the classifiers on test data. Include the following regularization measures: weight decay, dropout, batch normalization, and some version of an ensemble classifier other than a dropout equivalent.

6. Summarize the results you obtain in each experiment and draw conclusions based on your observation. Try to summarize results in a concise manner without repeating unnecessary details.

[1] https://archive.ics.uci.edu/ml/datasets/