HW 4: Neural Networks
Advanced Machine Learning

Prof: Yannet Interian

Due: February 20th 2018

For this homework you will submit write up with your solutions to github as well as your code. Your code should be written using the deep learning library Pytorch. I am providing some sample code.

1 Neural Network by hand (5 points)

Given the following specification of a neural network. Compute prediction you will give to $x = (-1, -1)$. (Show me the computation.)

$b_1^{(1)}$	$w_{11}^{(1)}$	$w_{12}^{(1)}$	$b_2^{(1)}$	$w_{21}^{(1)}$	$w_{22}^{(1)}$	$b^{(2)}$	$w_1^{(2)}$	$w_2^{(2)}$
-30	20	20	10	-20	-20	-10	20	20

Assume that

$$\sigma(a) = h(a) = \begin{cases} 1 & \text{if } a > 0 \\ 0 & \text{otherwise.} \end{cases}$$

2 Back-propagation equations (5 points)

Derive the back-propagation equations for 2-layer neural network for binary classification. Use the cross-entropy loss function and Relu ($max(0, x)$) as the activation function.

3 Neural networks for handwritten digit recognition (15 points)

Neural networks are among the most powerful machine learning models, which can represent complex, nonlinear functions. However, due to a high number of hyperparameters, neural networks are extremely difficult to tune for a particular task. In this exercise, we will apply neural networks for the task of handwritten digit recognition.

Data: you are given samples of 28x28 pixel images containing a handwritten digit with value 0-255. Our goal is to predict the digit 0 to 9 being drawn. Specifically, for this assignment, we will examine how the choice of different parameters affect the performance of a model. Find here `http://yann.lecun.com/exdb/mnist/` some results on this dataset. We will use *Adam* optimizer for all experiments.

1. (3 points) The sample code contains the code required to train a neural network. You will train a 2 layers neural network. The hidden layer has M=300 neurons. We are using Relu as the activation of the hidden layer. We are using softmax after the second layer ($F.cross_entropy$ computes the softwax). Use the training and validation set find a good learning rate. Report a table of validation accuracy for the following values of learning rate 1, 0.1, 0.01, 0.001, 0.0001, 0.00001. Interpolate between the best two values. Train for *nb_epoch* = 10. (You need to restart the initial weights by calling the function *get_model*() on every experiment.

2. (3 points) Examine how the size of a hidden layer affect the model's performance. Use demo code to train the model with hidden layer size parameters of 10, 50, 100, 300, 1000, 2000. Use learning rate 0.01 and train for *nb_epoch* = 10. Report the validation accuracy achieved by

each of these models. Which hidden layer size parameter achieves the best performance? Are some of these model overfitting? How can you tell that?

3. (3 points) Train a neural network that employs L_2 weight decay and compare it with the network without regularization. Train the model with weight decay parameters $0, 0.0001, 0.001, 0.01, 0.1, 0.3$. Set $M = 300$ and $lr = 0.001$ and 20 epochs. Report the train loss, validation loss and validation accuracy.

4. (3 points) Dropout is a very popular technique in deep learning community that enforces regularization in the neural network. The basic idea behind dropout is simple: during the training a selected fraction of neurons are zeroed out. Explore how the choice of a dropout parameter affect the performance of the model. Train the model with various dropout parameters ranging for 0 to 1. Take $M = 300$ and $lr = 0.001$ and 20 epochs. Report the train loss, validation loss and validation accuracy. Which parameter value achieves the best performance? Why? Does the dropout help to increase testing accuracy compared to the model without a dropout? Compare results with L2 regularization. You can experiment with lowering the learning rate after 10 epochs.

5. (3 points) Explore building a 3-layer network. Experiment with different parameters and regularization. Add your code to the write up and summarize your findings. For some ideas look at these results: `http://yann.lecun.com/exdb/mnist/`