Homework 5 – Deep Neural Networks (CS525 191N, Whitehill, Spring 2017)

You may complete this homework assignment either individually or in teams up to 2 people.

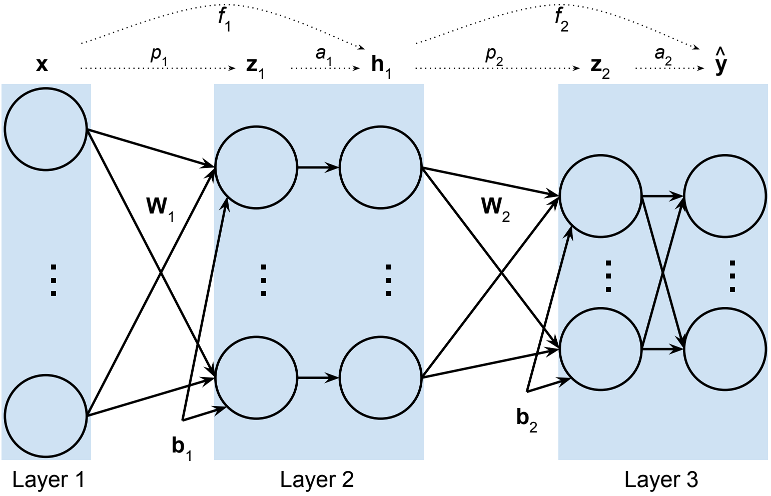
1. **3-layer neural network** [40 points]: In this problem you will train a 3-layer neural network to classify images of hand-written digits from the MNIST dataset. Similarly to Homework 4, the input to the network will be a 28 × 28-pixel image (converted into a 784-dimensional vector); the output will be a vector of 10 probabilities (one for each digit). Specifically, the network you create should implement a function $f : \mathbb{R}^{784} \rightarrow \mathbb{R}^{10}$, where:

$$\begin{aligned}
\mathbf{z}_1 &= p_1(\mathbf{x}) = \mathbf{W}_1\mathbf{x} + \mathbf{b}_1 \\
\mathbf{z}_2 &= p_2(\mathbf{h}_1) = \mathbf{W}_2\mathbf{h}_1 + \mathbf{b}_2 \\
\mathbf{h}_1 &= f_1(\mathbf{x}) = a_1(p_1(\mathbf{x})) \\
\hat{\mathbf{y}} &= f_2(\mathbf{h}_1) = a_2(p_2(\mathbf{h}_1)) \\
f(\mathbf{x}) &= f_2(f_1(\mathbf{x})) = a_2(p_2(a_1(p_1(\mathbf{x}))))
\end{aligned}$$

For the activation functions a_1, a_2 in your network, use:

$$\begin{aligned}
a_1(\mathbf{z}_1) &= \text{relu}(\mathbf{z}_1) \\
a_2(\mathbf{z}_2) &= \text{softmax}(\mathbf{z}_2)
\end{aligned}$$

The network specified above is shown in the figure below:



As usual, the cross-entropy cost function should be

$$J(\mathbf{W}_1, \mathbf{b}_1, \mathbf{W}_2, \mathbf{b}_2) = -\frac{1}{m}\sum_{j=1}^{m}\sum_{k=1}^{10} y_k^{(j)} \log \hat{y}_k^{(j)}$$

where m is the number of examples.

Hyperparameter tuning: In this problem, there are several different hyperparameters that will impact the network's performance:

- Number of units in the hidden layer (suggestions: $\{30, 40, 50\}$)
- Learning rate (suggestions: $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.5\}$)

- Minibatch size (suggestions: $16, 32, 64, 128, 256$)
- Number of epochs
- Regularization strength

In order not to "cheat" – and thus overestimate the performance of the network – it is crucial to optimize the hyperparameters **only** on the **validation** set; do **not** use the test set. (The training set would be ok but typically leads to worse performance.) Download the validation data here:

- `https://s3.amazonaws.com/jrwprojects/mnist_validation_images.npy`
- `https://s3.amazonaws.com/jrwprojects/mnist_validation_labels.npy`

Your task: Use stochastic gradient descent to minimize the cross-entropy with respect to $\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1$, and $\mathbf{b}_2$. Specifically:

(a) Implement stochastic gradient descent (SGD; see Section 5.9 and Algorithm 6.4 in the *Deep Learning* textbook) for the 3-layer neural network shown above. [25 points]

(b) Optimize the hyperparameters by training on the **training** set and selecting the parameter settings that optimize performance on the **validation** set. **You should *systematically* (i.e., in code) try at least 10 (in total, not for each hyperparameter) different hyperparameter settings**; accordingly, make sure there is a method called `findBestHyperparameters` (and please name it as such to help us during grading) [10 points]. **Include a screenshot** showing the progress and final output (selected hyperparameter values) of your hyperparameter optimization.

(c) After you have optimized your hyperparameters, then run your trained network on the **test set** and report (1) the cross-entropy and (2) the accuracy (percent correctly classified images). **Include a screenshot** showing both these values during the last 20 epochs of SGD. **The (unregularized) cross-entropy cost on the test set should be less than** 0.16**, and the accuracy (percentage correctly classified test images) should be at least** 95%**.** [5 points]

In addition to your Python code (`homework5_WPIUSERNAME1.py` or `homework5_WPIUSERNAME1_WPIUSERNAME2.py` for teams), create a PDF file (`homework5_WPIUSERNAME1.pdf` or `homework5_WPIUSERNAME1_WPIUSERNAME2.pdf` for teams) containing the screenshots described above. **Please submit both the PDF and Python files in a single Zip file.**