Machine Learning	Homework 3
CSCI 5622 Fall 2017	Due Time Oct 13, 2017

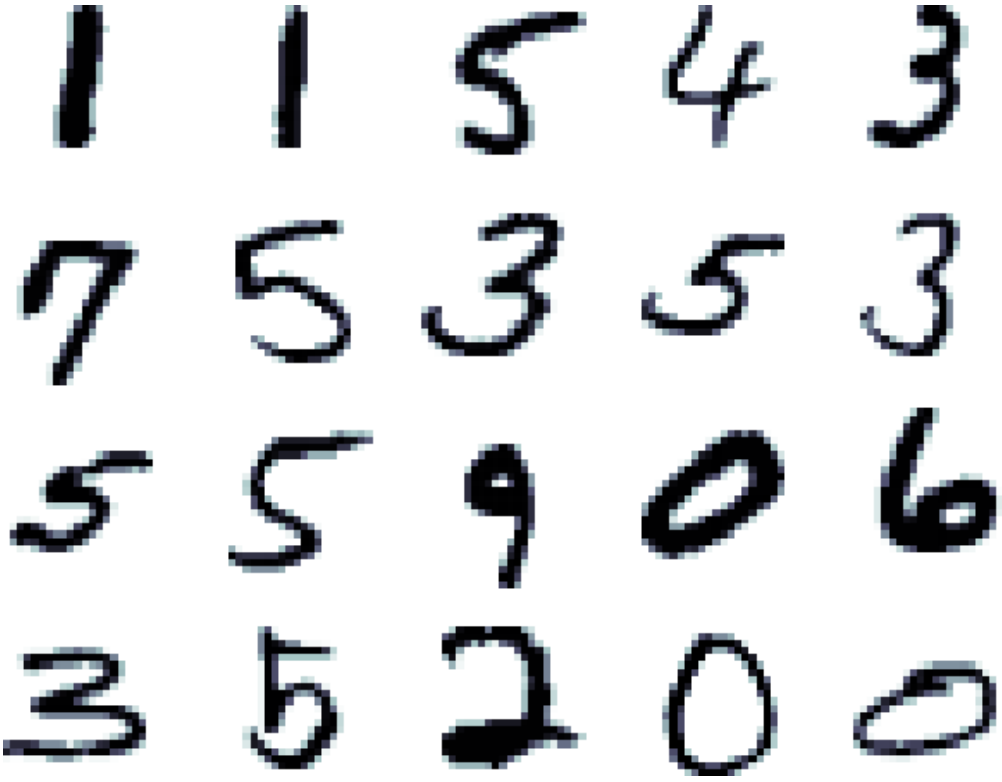
Submission Guidelines

- In order to download files required for the homework, clone `https://github.com/BoulderDS/csci_5622_hws`.
- For programming questions, submit python source files in a zip file.
- For other questions, submit a PDF file of no more than 4 pages.

All homework submissions are done through Moodle.

1 Back Propagation (35pts)

In this homework you'll implement a feed-forward neural network for classifying handwritten digits. Your tasks will be to implement back propagation to compute the parameter derivatives for SGD and also do L2 regularization for SGD. First, make sure your code works on a small dataset(tinyTOY.pkl.gz) before moving on to lower-resolution version of MNIST(tinyMNIST.pkl.gz).



1.1 Programming questions (20 pts)

Finish `nn.py`.

1. Finish *back_prop* function to compute the weights and biases.
2. Finish *SGD_train* function to do L2 regularization.
3. Add code to test on the *tinyMNIST* dataset.

1.2 Analysis (15 points)

1. What is the structure of your neural network (for both tinyTOY and tinyMNIST dataset)? Show the dimensions of the input layer, hidden layer and output layer.
2. What the role of the size of the hidden layer on train and test accuracy (plot accuracy vs. size of hidden layer using tinyMNIST dataset)?
3. How does the number of epochs affect train and test accuracy (plot accuracy vs. epochs using tinyMINST dataset)?

2 Keras CNN (35pts)

Here, you will use the Conv2D layer in Keras to build a Convolutional Neural Network for the MNIST dataset. The input dataset is the same as the MNIST dataset in HW1, so you need to reshape the vector of each image into matrix for the use of Conv2D. And you need to build your model using the layers provided by Keras and achieve an accuracy higher than 98.5%.

2.1 Programming questions (20pts)

Finish the `CNN.py` to build a CNN model, train and improve your model to achieve 98.5% accuracy on MNIST dataset. (Hint: use one hot encoding for label, input for the final Dense layer need to be flattened, try Dropout layer to improve your model and don't give up).

1. Reshape your MNIST data.
2. Finish *__init__* function to construct your model.
3. Finish *train* function and fit to your training data.

2.2 Analysis (15pts)

1. Point out at least three layer types you used in your model. Explain what are they used for.
2. How did you improve your model for higher accuracy?
3. Try different activation functions and batch sizes. Show the corresponding accuracy.

3 Keras RNN (30pts)

Here you will use Keras to build a RNN model for sentiment analysis. You should use word embeddings and LSTM to finish LSTM.py. You will test your model on the IMDB dataset. And you are expected to achieve an accuracy higher than 90%.

3.1 Programming questions (15pts)

Finish the `LSTM.py` to build an RNN model. Use word embeddings as the first layer and use LSTM for sequential prediction.

1. Preprocess data for LSTM (require data of the same length).
2. Finish *__init__* function to construct your model.
3. Finish *train* function and fit to your training data.

3.2 Analysis (15pts)

1. What is the purpose of the embedding layer? (Hint: think about the input and the output).
2. What is the effect of the hidden dimension size in LSTM?
3. Replace LSTM with GRU and compare their performance.

Extra credits (5pts) Try to use pretrained word embeddings to initialize the embedding layer and see how that changes the performance.