GEORGE MASON UNIVERSITY
Systems Engineering and Operations Research
OR750: Deep Learning , Fall Semester 2018: Homework Assignment 2. Due: Oct 29 (before class)

1. Organic DL Without using a deep learning framework, code a one-hidden layer neural network model (multi-layer perceptron) for classification, i.e. the last layer is softmax and with cross-entropy loss function. Implement back-propagation. Use your code to classify MNIST. You can download MNIST data from `http://yann.lecun.com/exdb/mnist/` and read it as a numpy array using the following function.

```
import struct
import numpy as np

def read_idx(filename):
    with open(filename, 'rb') as f:
        zero, data_type, dims = struct.unpack('>HBB', f.read(4))
        shape = tuple(struct.unpack('>I', f.read(4))[0] for d in range(dims))
        return np.fromstring(f.read(), dtype=np.uint8).reshape(shape)
```

Run a few numerical experiments with different non-linear activation functions, different number of neurons and different learning rates.

2. Organic CNN Replace the fully connected layer from the first problem with the convolutional layer.

3. Regularization Add ℓ_2 regularization to your model. Compare the accuracy and training speed.

4. Initialization Implement four different initialization techniques and apply them to your CNN model

- Zeros initialization
- Random initialization. This initializes the weights to some random values.
- Xavier initialization, which scales the variance of the inputs to each layer are scaled to have variance of `sqrt(1./layers_dims[l-1])`.
- He initialization. This initializes the weights to random values scaled according to a paper by He et al., 2015. Similar to Xavier you need to scale the inputs so they have the variance `sqrt(2./layers_dims[l-1])`

5. Optimization Implement ADAM modification to the SGD and apply it to MNIST classification.

6. Dropout (extra credit) Implement forward and backward propagation with dropout. Apply to MNIST classification problem.