Homework 8: Due Tue 10-09-2018
Total Points (20 pts)

Fully-Connected Networks

(a) (5 pts) Load and preprocess the `notMNIST_train_AB.npz` dataset. Determine a baseline accuracy rate.

```
# load training notMNIST data
notMNIST = np.load('../Data/notMNIST/notMNIST_train_AB.npz')
images_train = notMNIST['images_train']
labels_train = notMNIST['labels_train']
label_names  = notMNIST['label_names']
print(images_train.shape)
print(labels_train.shape)
```

(b) (5 pts) Train a logistic regression network using the Adam's optimizer. Select a batch size (1, 10, 100, or 1000) that minimizes the time to complete one epoch. Then use early stopping with `patience = 10`, 100 epochs and an 80%–20% train-validation dataset split to regularize the network. What is your validation accuracy rate for your lowest validation loss?

```
from time import time

time_start = time()
hist = model.fit(. . .,batch_size=BS,. . . )
time_stop = time()
time_elapsed = time_stop - time_start
```

(c) (5 pts) Add a 256 node layer. Using the batch size determined in part (b), repeat part (b). What is your validation accuracy rate for your lowest validation loss?

(d) (5 pts) Load `notMNIST_test_AB.npz`. Use your trained network from part (c) to make predictions using the test data. Compute the test error rate. Plot the incorrectly classified images and determine a human level accuracy rate. (For simplicity, assume that you can correctly classify all images correctly classified by the network. Just test yourself on the incorrectly classified images.)

```
Ph_test = model.predict(X_test)
ix_errors = (Ph_test.argmax(axis=1) != P_test.argmax(axis=1))
images_errors = images_test[ix_errors,:,:]
labels_errors = labels_test[ix_errors]
acc_test = 1 - len(images_errors)/len(images_test)

N = 8
M = 7
plt.figure(figsize=(10,10))
for i in range(len(images_errors)):
    plt.subplot(M, N, i+1)
    plt.imshow(images_errors[i,:,:], cmap='Greys_r')
    plt.title(label_names[labels_errors[i]])
    plt.show()
```