```
In [2]:  from google.colab import drive
         drive.mount('/content/gdrive', force_remount=True)
         import os
         os.chdir("/content/gdrive/My Drive/Colab Notebooks/Startpkg_A2")
```

Mounted at /content/gdrive

```
In [3]:  import numpy as np
         import matplotlib.pyplot as plt

         from utils.data_process import get_CIFAR10_data
         from models.neural_net import NeuralNetwork
         from kaggle_submission import output_submission_csv

         ######### If not using Colab, you may skip these setup #########
         %matplotlib inline
         plt.rcParams['figure.figsize'] = (10.0, 8.0) # set default size of plots
         plt.rcParams['image.interpolation'] = 'nearest'
         plt.rcParams['image.cmap'] = 'gray'

         # for auto-reloading external modules
         # see http://stackoverflow.com/questions/1907993/autoreload-of-modules-in-ipyt
         hon
         %load_ext autoreload
         %autoreload 2
         #################### End of setup ###########################
```

Loading CIFAR-10

Now that you have implemented a neural network that passes gradient checks and works on toy data, you will
test your network on the CIFAR-10 dataset.

```
In [5]:  # You can change these numbers for experimentation
         # For submission be sure they are set to the default values
         TRAIN_IMAGES = 49000
         VAL_IMAGES = 1000
         TEST_IMAGES = 5000   # Default is 5000, do not modify this for your submission.

         data = get_CIFAR10_data(TRAIN_IMAGES, VAL_IMAGES, TEST_IMAGES)
         X_train, y_train = data['X_train'], data['y_train']
         X_val, y_val = data['X_val'], data['y_val']
         X_test, y_test = data['X_test'], data['y_test']
```

Train a network

To train our network we will use SGD. In addition, we will adjust the learning rate with an exponential learning rate schedule as optimization proceeds; after each epoch, we will reduce the learning rate by multiplying it by a decay rate.

You can try different numbers of layers and also the different activation functions that you implemented on the CIFAR-10 dataset below.

```
In [6]: input_size = 32 * 32 * 3
        num_layers = 3
        hidden_size = 120
        hidden_sizes = [hidden_size]*(num_layers-1)
        num_classes = 10
        learning_rate = 0.05
        learning_rate_decay = 0.95
        net = NeuralNetwork(input_size, hidden_sizes, num_classes, num_layers, nonline
        arity='sigmoid')

        # Train the network
        stats = net.train(X_train, y_train, X_val, y_val,
                    num_iters=1000, batch_size=200,
                    learning_rate=learning_rate, learning_rate_decay=learning_rate_dec
        ay,
                    reg=0.00, verbose=True)

        # Predict on the validation set
        val_acc = (net.predict(X_val) == y_val).mean()
        print('Validation accuracy: ', val_acc)
```

```
iteration 0 / 1000: loss 2.360263
iteration 100 / 1000: loss 2.171381
iteration 200 / 1000: loss 2.022783
iteration 300 / 1000: loss 2.017632
iteration 400 / 1000: loss 1.983689
iteration 500 / 1000: loss 1.900510
iteration 600 / 1000: loss 1.885977
iteration 700 / 1000: loss 1.860290
iteration 800 / 1000: loss 1.836469
iteration 900 / 1000: loss 1.893836
Validation accuracy:  0.375
```

Graph loss and train/val accuracies

Examining the loss graph along with the train and val accuracy graphs should help you gain some intuition for the hyperparameters you should try in the hyperparameter tuning below. It should also help with debugging any issues you might have with your network.

```python
# Plot the loss function and train / validation accuracies
plt.subplot(2, 1, 1)
plt.plot(stats['loss_history'])
plt.title('Loss history')
plt.xlabel('Iteration')
plt.ylabel('Loss')

plt.subplot(2, 1, 2)
plt.plot(stats['train_acc_history'], label='train')
plt.plot(stats['val_acc_history'], label='val')
plt.title('Classification accuracy history')
plt.xlabel('Epoch')
plt.ylabel('Clasification accuracy')
plt.legend()
plt.show()
```
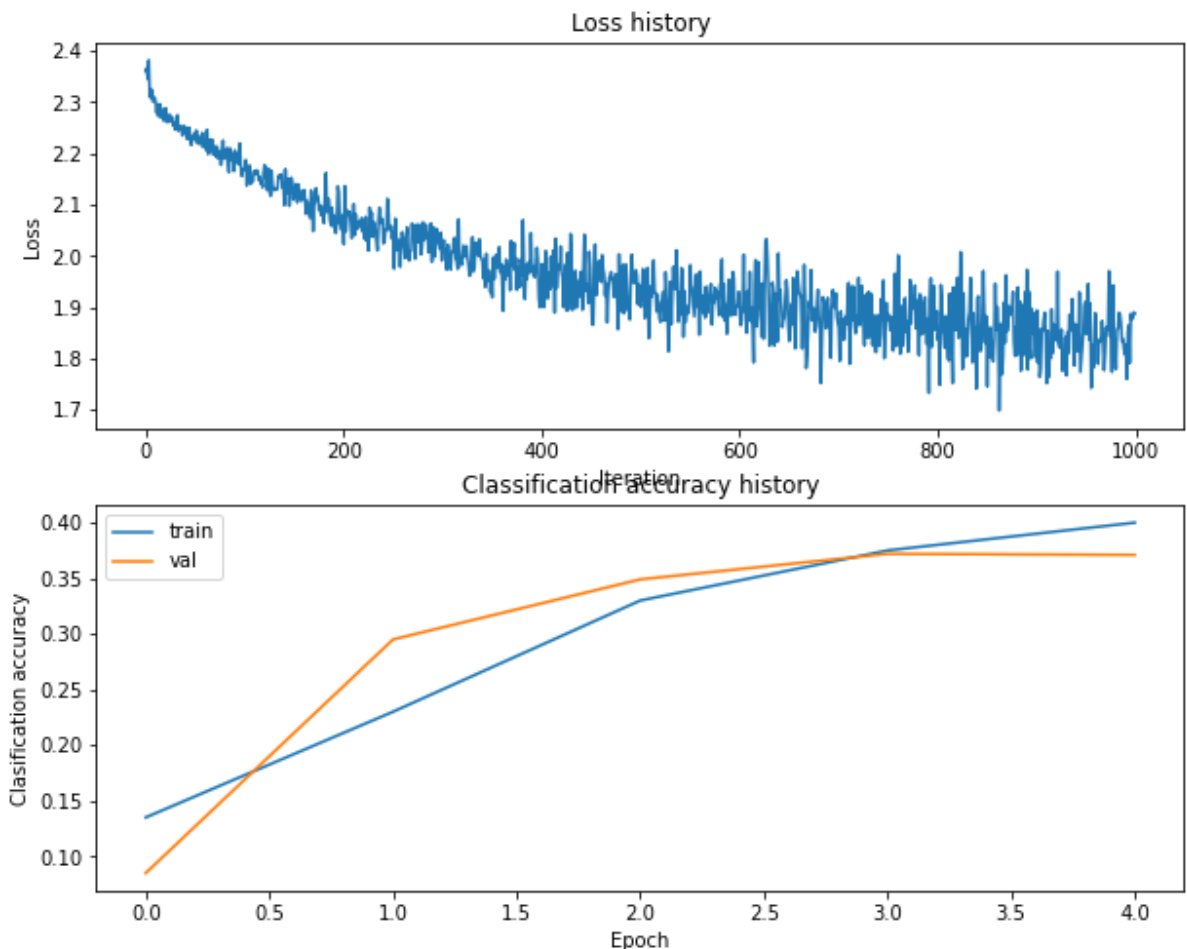
Hyperparameter tuning

Once you have successfully trained a network you can tune your hyparameters to increase your accuracy.

Based on the graphs of the loss function above you should be able to develop some intuition about what hyperparameter adjustments may be necessary. A very noisy loss implies that the learning rate might be too high, while a linearly decreasing loss would suggest that the learning rate may be too low. A large gap between training and validation accuracy would suggest overfitting due to large model without much regularization. No gap between training and validation accuracy would indicate low model capacity (low model complexity).

You will compare networks of two and three layers using the different activation functions you implemented.

The different hyperparameters you can experiment with are:

- **Batch size**: We recommend you leave this at 200 initially which is the batch size we used.
- **Number of iterations**: You can gain an intuition for how many iterations to run by checking when the validation accuracy plateaus in your train/val accuracy graph.
- **Initialization** Weight initialization is very important for neural networks. We used the initialization `W = np.random.randn(n) / sqrt(n)` where `n` is the input dimension for layer corresponding to `W`. We recommend you stick with the given initializations, but you may explore modifying these. Typical initialization practices: http://cs231n.github.io/neural-networks-2/#init (http://cs231n.github.io/neural-networks-2/#init)
- **Learning rate**: Generally from around 1e-4 to 1e-1 is a good range to explore according to our implementation.
- **Learning rate decay**: We recommend a 0.95 decay to start.
- **Hidden layer size**: You should explore up to around 120 units per layer. For three-layer network, we fixed the two hidden layers to be the same size when obtaining the target numbers. However, you may experiment with having different size hidden layers.
- **Regularization coefficient**: We recommend trying values in the range 0 to 0.1.

Hints:

- After getting a sense of the parameters by trying a few values yourself, you will likely want to write a few for loops to traverse over a set of hyperparameters.
- If you find that your train loss is decreasing, but your train and val accuracy start to decrease rather than increase, your model likely started minimizing the regularization term. To prevent this you will need to decrease the regularization coefficient.

DEFINING MATRIX OF PARAMETERS (ADDED BY OR)

```
##################################################################################
###
#                         DEFINING THE MATRIX OF PARAMETERS
#
#             THIS MATRIX WILL BE USED FOR OPTIMIZATION OF ALL THE MODELS
#
##################################################################################
###

learning_rate_values = [0.1, 0.05, 0.03]
hidden_size_values = [20, 50, 80, 100, 120]
batch_size_values = [50, 100, 200]

#All of this is to build the parameter matrix

#Initializing the parameters matrix

parameters_matrix = np.zeros((len(learning_rate_values)*len(hidden_size_values
)*len(batch_size_values),3))

#Entering the learning rate values

parameters_matrix[0:len(hidden_size_values)*len(batch_size_values)][:,0] = lea
rning_rate_values[0]
parameters_matrix[len(hidden_size_values)*len(batch_size_values):2*len(hidden_
size_values)*len(batch_size_values)][:,0] = learning_rate_values[1]
parameters_matrix[2*len(hidden_size_values)*len(batch_size_values):3*len(hidde
n_size_values)*len(batch_size_values)][:,0] = learning_rate_values[2]

#Entering the hidden size values

parameters_matrix[0:len(parameters_matrix):5,1] = hidden_size_values[0]
parameters_matrix[1:len(parameters_matrix):5,1] = hidden_size_values[1]
parameters_matrix[2:len(parameters_matrix):5,1] = hidden_size_values[2]
parameters_matrix[3:len(parameters_matrix):5,1] = hidden_size_values[3]
parameters_matrix[4:len(parameters_matrix):5,1] = hidden_size_values[4]

#Entering the hidden size values

parameters_matrix[0:len(parameters_matrix):3,2] = batch_size_values[0]
parameters_matrix[1:len(parameters_matrix):3,2] = batch_size_values[1]
parameters_matrix[2:len(parameters_matrix):3,2] = batch_size_values[2]
```

Two-layer Relu Activation Network

```
In [9]:  best_2layer_relu = None # store the best model into this

         #################################################################################
         ###
         # TODO: Tune hyperparameters using the validation set. Store your best trained
         #
         # model in best_2layer_relu.
         #
         #################################################################################
         ###

         #Initialize "best" validation accuracy

         best_val_acc = 0

         #Initialize vector to store all validation accuracies

         val_acc_vec = np.zeros(len(parameters_matrix))

         #Cycle through the matrix of parameters

         for i in range(len(parameters_matrix)):

             input_size = 32 * 32 * 3
             num_layers = 2
             hidden_size = int(parameters_matrix[i,1])
             hidden_sizes = [hidden_size]*(num_layers-1)
             num_classes = 10
             learning_rate = parameters_matrix[i,0]
             learning_rate_decay = 0.95
             batch_size = int(parameters_matrix[i,2])
             net = NeuralNetwork(input_size, hidden_sizes, num_classes, num_layers, nonli
         nearity='relu')

             # Train the network
             stats = net.train(X_train, y_train, X_val, y_val,
                         num_iters=1000, batch_size=batch_size,
                         learning_rate=learning_rate, learning_rate_decay=learning_rate_d
         ecay,
                         reg=0.0, verbose=True)

             # Predict on the validation set
             val_acc = (net.predict(X_val) == y_val).mean()

             print('For parameters: ', parameters_matrix[i])
             print('Validation accuracy: ', val_acc)

             #Store the validaction accuracies

             val_acc_vec[i] = val_acc

             #Save the best model and stats

             if val_acc > best_val_acc:

                 best_val_acc = val_acc
```

```python
    best_2layer_relu = net
    best_stats = stats

#Print the best combination of trial parameters:

ind = np.argmax(val_acc_vec)

print('The best combination of parameters is: learning rate = ',parameters_mat
rix[ind,0], ', hidden layer size = ',int(parameters_matrix[ind,1]), ', batch s
ize = ',int(parameters_matrix[ind,2]))
print('For this combination, the validation accuracy is ', val_acc_vec[ind])
```

```
iteration 0 / 1000: loss 2.715286
iteration 100 / 1000: loss 1.866864
iteration 200 / 1000: loss 1.913203
iteration 300 / 1000: loss 1.814135
iteration 400 / 1000: loss 1.981856
iteration 500 / 1000: loss 1.978752
iteration 600 / 1000: loss 1.811956
iteration 700 / 1000: loss 1.678494
iteration 800 / 1000: loss 1.764641
iteration 900 / 1000: loss 1.839433
For parameters:  [ 0.1 20.   50. ]
Validation accuracy:  0.398
iteration 0 / 1000: loss 2.542021
iteration 100 / 1000: loss 1.683022
iteration 200 / 1000: loss 1.664625
iteration 300 / 1000: loss 1.557218
iteration 400 / 1000: loss 1.844410
iteration 500 / 1000: loss 1.750250
iteration 600 / 1000: loss 1.544880
iteration 700 / 1000: loss 1.627553
iteration 800 / 1000: loss 1.524133
iteration 900 / 1000: loss 1.440771
For parameters:  [  0.1  50.  100. ]
Validation accuracy:  0.473
iteration 0 / 1000: loss 2.428241
iteration 100 / 1000: loss 1.796744
iteration 200 / 1000: loss 1.546376
iteration 300 / 1000: loss 1.577306
iteration 400 / 1000: loss 1.571245
iteration 500 / 1000: loss 1.298537
iteration 600 / 1000: loss 1.566067
iteration 700 / 1000: loss 1.336600
iteration 800 / 1000: loss 1.361602
iteration 900 / 1000: loss 1.421368
For parameters:  [1.e-01 8.e+01 2.e+02]
Validation accuracy:  0.489
iteration 0 / 1000: loss 2.434387
iteration 100 / 1000: loss 1.897614
iteration 200 / 1000: loss 1.790517
iteration 300 / 1000: loss 1.549533
iteration 400 / 1000: loss 1.756254
iteration 500 / 1000: loss 1.785943
iteration 600 / 1000: loss 1.959229
iteration 700 / 1000: loss 1.622413
iteration 800 / 1000: loss 1.623365
iteration 900 / 1000: loss 2.029529
For parameters:  [  0.1 100.   50. ]
Validation accuracy:  0.414
iteration 0 / 1000: loss 2.376951
iteration 100 / 1000: loss 1.788000
iteration 200 / 1000: loss 1.559161
iteration 300 / 1000: loss 1.560658
iteration 400 / 1000: loss 1.558631
iteration 500 / 1000: loss 1.715426
iteration 600 / 1000: loss 1.543630
iteration 700 / 1000: loss 1.522090
iteration 800 / 1000: loss 1.422984
```

```
iteration 900 / 1000: loss 1.484674
For parameters:  [1.0e-01 1.2e+02 1.0e+02]
Validation accuracy:  0.468
iteration 0 / 1000: loss 2.426119
iteration 100 / 1000: loss 1.717044
iteration 200 / 1000: loss 1.629928
iteration 300 / 1000: loss 1.697360
iteration 400 / 1000: loss 1.646700
iteration 500 / 1000: loss 1.741680
iteration 600 / 1000: loss 1.713300
iteration 700 / 1000: loss 1.547610
iteration 800 / 1000: loss 1.525743
iteration 900 / 1000: loss 1.465556
For parameters:  [1.e-01 2.e+01 2.e+02]
Validation accuracy:  0.441
iteration 0 / 1000: loss 2.662212
iteration 100 / 1000: loss 1.804719
iteration 200 / 1000: loss 1.825623
iteration 300 / 1000: loss 1.827037
iteration 400 / 1000: loss 1.991980
iteration 500 / 1000: loss 1.731884
iteration 600 / 1000: loss 1.457862
iteration 700 / 1000: loss 1.559146
iteration 800 / 1000: loss 1.554897
iteration 900 / 1000: loss 1.445868
For parameters:  [ 0.1 50.   50. ]
Validation accuracy:  0.36
iteration 0 / 1000: loss 2.518125
iteration 100 / 1000: loss 1.738500
iteration 200 / 1000: loss 1.639498
iteration 300 / 1000: loss 1.994568
iteration 400 / 1000: loss 1.681102
iteration 500 / 1000: loss 1.672262
iteration 600 / 1000: loss 1.432016
iteration 700 / 1000: loss 1.425032
iteration 800 / 1000: loss 1.553365
iteration 900 / 1000: loss 1.344180
For parameters:  [  0.1  80.   100. ]
Validation accuracy:  0.418
iteration 0 / 1000: loss 2.380153
iteration 100 / 1000: loss 1.728520
iteration 200 / 1000: loss 1.534835
iteration 300 / 1000: loss 1.449737
iteration 400 / 1000: loss 1.362949
iteration 500 / 1000: loss 1.485556
iteration 600 / 1000: loss 1.462125
iteration 700 / 1000: loss 1.338245
iteration 800 / 1000: loss 1.258395
iteration 900 / 1000: loss 1.283329
For parameters:  [1.e-01 1.e+02 2.e+02]
Validation accuracy:  0.5
iteration 0 / 1000: loss 2.525483
iteration 100 / 1000: loss 1.878190
iteration 200 / 1000: loss 1.749712
iteration 300 / 1000: loss 1.666091
iteration 400 / 1000: loss 1.970726
iteration 500 / 1000: loss 2.062984
```

```
iteration 600 / 1000: loss 1.614487
iteration 700 / 1000: loss 1.638533
iteration 800 / 1000: loss 1.766345
iteration 900 / 1000: loss 1.767084
For parameters:  [1.0e-01 1.2e+02 5.0e+01]
Validation accuracy:  0.418
iteration 0 / 1000: loss 2.606611
iteration 100 / 1000: loss 1.730961
iteration 200 / 1000: loss 1.743571
iteration 300 / 1000: loss 1.783963
iteration 400 / 1000: loss 1.847133
iteration 500 / 1000: loss 1.416942
iteration 600 / 1000: loss 1.683316
iteration 700 / 1000: loss 1.667880
iteration 800 / 1000: loss 1.747269
iteration 900 / 1000: loss 1.620577
For parameters:  [  0.1  20.  100. ]
Validation accuracy:  0.43
iteration 0 / 1000: loss 2.498759
iteration 100 / 1000: loss 1.711566
iteration 200 / 1000: loss 1.572389
iteration 300 / 1000: loss 1.591275
iteration 400 / 1000: loss 1.555117
iteration 500 / 1000: loss 1.541600
iteration 600 / 1000: loss 1.518847
iteration 700 / 1000: loss 1.381872
iteration 800 / 1000: loss 1.422076
iteration 900 / 1000: loss 1.315779
For parameters:  [1.e-01 5.e+01 2.e+02]
Validation accuracy:  0.477
iteration 0 / 1000: loss 2.480604
iteration 100 / 1000: loss 1.962881
iteration 200 / 1000: loss 1.836400
iteration 300 / 1000: loss 1.554923
iteration 400 / 1000: loss 1.772844
iteration 500 / 1000: loss 1.915968
iteration 600 / 1000: loss 1.679191
iteration 700 / 1000: loss 1.792873
iteration 800 / 1000: loss 1.380780
iteration 900 / 1000: loss 1.774523
For parameters:  [ 0.1 80.   50. ]
Validation accuracy:  0.44
iteration 0 / 1000: loss 2.475962
iteration 100 / 1000: loss 1.697423
iteration 200 / 1000: loss 1.684923
iteration 300 / 1000: loss 1.464603
iteration 400 / 1000: loss 1.566533
iteration 500 / 1000: loss 1.393598
iteration 600 / 1000: loss 1.483803
iteration 700 / 1000: loss 1.518905
iteration 800 / 1000: loss 1.502741
iteration 900 / 1000: loss 1.444068
For parameters:  [  0.1 100.  100. ]
Validation accuracy:  0.466
iteration 0 / 1000: loss 2.507818
iteration 100 / 1000: loss 1.661972
iteration 200 / 1000: loss 1.693131
```

```
iteration 300 / 1000: loss 1.489109
iteration 400 / 1000: loss 1.571926
iteration 500 / 1000: loss 1.349370
iteration 600 / 1000: loss 1.234844
iteration 700 / 1000: loss 1.226016
iteration 800 / 1000: loss 1.545961
iteration 900 / 1000: loss 1.250473
For parameters:  [1.0e-01 1.2e+02 2.0e+02]
Validation accuracy:  0.524
iteration 0 / 1000: loss 2.649751
iteration 100 / 1000: loss 2.082025
iteration 200 / 1000: loss 1.839678
iteration 300 / 1000: loss 1.873448
iteration 400 / 1000: loss 1.777536
iteration 500 / 1000: loss 1.778407
iteration 600 / 1000: loss 1.748325
iteration 700 / 1000: loss 1.654586
iteration 800 / 1000: loss 1.835584
iteration 900 / 1000: loss 1.731624
For parameters:  [ 0.05 20.   50.  ]
Validation accuracy:  0.399
iteration 0 / 1000: loss 2.438073
iteration 100 / 1000: loss 1.774453
iteration 200 / 1000: loss 1.587868
iteration 300 / 1000: loss 1.613218
iteration 400 / 1000: loss 1.685128
iteration 500 / 1000: loss 1.481976
iteration 600 / 1000: loss 1.693441
iteration 700 / 1000: loss 1.522465
iteration 800 / 1000: loss 1.260546
iteration 900 / 1000: loss 1.456370
For parameters:  [5.e-02 5.e+01 1.e+02]
Validation accuracy:  0.473
iteration 0 / 1000: loss 2.609454
iteration 100 / 1000: loss 1.655263
iteration 200 / 1000: loss 1.566539
iteration 300 / 1000: loss 1.617915
iteration 400 / 1000: loss 1.508518
iteration 500 / 1000: loss 1.480823
iteration 600 / 1000: loss 1.466878
iteration 700 / 1000: loss 1.452852
iteration 800 / 1000: loss 1.281365
iteration 900 / 1000: loss 1.332153
For parameters:  [5.e-02 8.e+01 2.e+02]
Validation accuracy:  0.498
iteration 0 / 1000: loss 2.340921
iteration 100 / 1000: loss 1.618954
iteration 200 / 1000: loss 1.785193
iteration 300 / 1000: loss 1.815763
iteration 400 / 1000: loss 1.767731
iteration 500 / 1000: loss 1.689865
iteration 600 / 1000: loss 1.502381
iteration 700 / 1000: loss 1.792459
iteration 800 / 1000: loss 1.373051
iteration 900 / 1000: loss 1.661950
For parameters:  [5.e-02 1.e+02 5.e+01]
Validation accuracy:  0.458
```

```
iteration 0 / 1000: loss 2.420423
iteration 100 / 1000: loss 1.847190
iteration 200 / 1000: loss 1.671877
iteration 300 / 1000: loss 1.777128
iteration 400 / 1000: loss 1.563582
iteration 500 / 1000: loss 1.404980
iteration 600 / 1000: loss 1.405071
iteration 700 / 1000: loss 1.259767
iteration 800 / 1000: loss 1.368514
iteration 900 / 1000: loss 1.461776
For parameters:  [5.0e-02 1.2e+02 1.0e+02]
Validation accuracy:  0.483
iteration 0 / 1000: loss 2.461378
iteration 100 / 1000: loss 1.787870
iteration 200 / 1000: loss 1.709388
iteration 300 / 1000: loss 1.716957
iteration 400 / 1000: loss 1.573373
iteration 500 / 1000: loss 1.588788
iteration 600 / 1000: loss 1.542064
iteration 700 / 1000: loss 1.532098
iteration 800 / 1000: loss 1.496796
iteration 900 / 1000: loss 1.462122
For parameters:  [5.e-02 2.e+01 2.e+02]
Validation accuracy:  0.445
iteration 0 / 1000: loss 2.407305
iteration 100 / 1000: loss 1.643106
iteration 200 / 1000: loss 2.024314
iteration 300 / 1000: loss 1.613249
iteration 400 / 1000: loss 1.446963
iteration 500 / 1000: loss 1.643224
iteration 600 / 1000: loss 1.396599
iteration 700 / 1000: loss 1.382745
iteration 800 / 1000: loss 1.682422
iteration 900 / 1000: loss 1.584512
For parameters:  [ 0.05 50.   50.  ]
Validation accuracy:  0.443
iteration 0 / 1000: loss 2.412643
iteration 100 / 1000: loss 1.810660
iteration 200 / 1000: loss 1.513533
iteration 300 / 1000: loss 1.638817
iteration 400 / 1000: loss 1.532174
iteration 500 / 1000: loss 1.508489
iteration 600 / 1000: loss 1.628086
iteration 700 / 1000: loss 1.277978
iteration 800 / 1000: loss 1.360869
iteration 900 / 1000: loss 1.420019
For parameters:  [5.e-02 8.e+01 1.e+02]
Validation accuracy:  0.469
iteration 0 / 1000: loss 2.624492
iteration 100 / 1000: loss 1.628949
iteration 200 / 1000: loss 1.541047
iteration 300 / 1000: loss 1.472769
iteration 400 / 1000: loss 1.457646
iteration 500 / 1000: loss 1.513345
iteration 600 / 1000: loss 1.428303
iteration 700 / 1000: loss 1.321445
iteration 800 / 1000: loss 1.296052
```

```
iteration 900 / 1000: loss 1.409009
For parameters:  [5.e-02 1.e+02 2.e+02]
Validation accuracy:  0.5
iteration 0 / 1000: loss 2.735369
iteration 100 / 1000: loss 1.821706
iteration 200 / 1000: loss 1.535711
iteration 300 / 1000: loss 1.557933
iteration 400 / 1000: loss 1.609216
iteration 500 / 1000: loss 1.458676
iteration 600 / 1000: loss 1.513089
iteration 700 / 1000: loss 1.433682
iteration 800 / 1000: loss 1.759217
iteration 900 / 1000: loss 1.533537
For parameters:  [5.0e-02 1.2e+02 5.0e+01]
Validation accuracy:  0.448
iteration 0 / 1000: loss 2.558711
iteration 100 / 1000: loss 1.891120
iteration 200 / 1000: loss 1.714737
iteration 300 / 1000: loss 1.686663
iteration 400 / 1000: loss 1.608266
iteration 500 / 1000: loss 1.538979
iteration 600 / 1000: loss 1.469570
iteration 700 / 1000: loss 1.824436
iteration 800 / 1000: loss 1.551256
iteration 900 / 1000: loss 1.894040
For parameters:  [5.e-02 2.e+01 1.e+02]
Validation accuracy:  0.43
iteration 0 / 1000: loss 2.450733
iteration 100 / 1000: loss 1.867561
iteration 200 / 1000: loss 1.568302
iteration 300 / 1000: loss 1.579026
iteration 400 / 1000: loss 1.488592
iteration 500 / 1000: loss 1.438244
iteration 600 / 1000: loss 1.514497
iteration 700 / 1000: loss 1.348599
iteration 800 / 1000: loss 1.471637
iteration 900 / 1000: loss 1.403929
For parameters:  [5.e-02 5.e+01 2.e+02]
Validation accuracy:  0.483
iteration 0 / 1000: loss 2.804274
iteration 100 / 1000: loss 1.446387
iteration 200 / 1000: loss 1.714435
iteration 300 / 1000: loss 1.557163
iteration 400 / 1000: loss 1.360716
iteration 500 / 1000: loss 1.639995
iteration 600 / 1000: loss 1.620307
iteration 700 / 1000: loss 1.712601
iteration 800 / 1000: loss 1.598090
iteration 900 / 1000: loss 1.427387
For parameters:  [5.e-02 8.e+01 5.e+01]
Validation accuracy:  0.451
iteration 0 / 1000: loss 2.474303
iteration 100 / 1000: loss 1.678994
iteration 200 / 1000: loss 1.629488
iteration 300 / 1000: loss 1.396176
iteration 400 / 1000: loss 1.691628
iteration 500 / 1000: loss 1.627583
```

```
iteration 600 / 1000: loss 1.309060
iteration 700 / 1000: loss 1.322883
iteration 800 / 1000: loss 1.460409
iteration 900 / 1000: loss 1.629806
For parameters:  [5.e-02 1.e+02 1.e+02]
Validation accuracy:  0.472
iteration 0 / 1000: loss 2.661310
iteration 100 / 1000: loss 1.739933
iteration 200 / 1000: loss 1.568916
iteration 300 / 1000: loss 1.407349
iteration 400 / 1000: loss 1.550511
iteration 500 / 1000: loss 1.470718
iteration 600 / 1000: loss 1.301667
iteration 700 / 1000: loss 1.432915
iteration 800 / 1000: loss 1.474231
iteration 900 / 1000: loss 1.419195
For parameters:  [5.0e-02 1.2e+02 2.0e+02]
Validation accuracy:  0.507
iteration 0 / 1000: loss 2.478321
iteration 100 / 1000: loss 1.991663
iteration 200 / 1000: loss 1.918174
iteration 300 / 1000: loss 1.721801
iteration 400 / 1000: loss 1.792469
iteration 500 / 1000: loss 1.754125
iteration 600 / 1000: loss 2.038703
iteration 700 / 1000: loss 1.731625
iteration 800 / 1000: loss 1.648066
iteration 900 / 1000: loss 1.600500
For parameters:  [3.e-02 2.e+01 5.e+01]
Validation accuracy:  0.411
iteration 0 / 1000: loss 2.555676
iteration 100 / 1000: loss 1.851040
iteration 200 / 1000: loss 1.783851
iteration 300 / 1000: loss 1.622707
iteration 400 / 1000: loss 1.575700
iteration 500 / 1000: loss 1.636633
iteration 600 / 1000: loss 1.540888
iteration 700 / 1000: loss 1.532103
iteration 800 / 1000: loss 1.642933
iteration 900 / 1000: loss 1.487687
For parameters:  [3.e-02 5.e+01 1.e+02]
Validation accuracy:  0.479
iteration 0 / 1000: loss 2.516289
iteration 100 / 1000: loss 1.631177
iteration 200 / 1000: loss 1.585127
iteration 300 / 1000: loss 1.475524
iteration 400 / 1000: loss 1.555140
iteration 500 / 1000: loss 1.549211
iteration 600 / 1000: loss 1.479004
iteration 700 / 1000: loss 1.484843
iteration 800 / 1000: loss 1.455360
iteration 900 / 1000: loss 1.289475
For parameters:  [3.e-02 8.e+01 2.e+02]
Validation accuracy:  0.491
iteration 0 / 1000: loss 2.526589
iteration 100 / 1000: loss 1.678577
iteration 200 / 1000: loss 1.616626
```

```
iteration 300 / 1000: loss 1.706297
iteration 400 / 1000: loss 1.816006
iteration 500 / 1000: loss 1.996626
iteration 600 / 1000: loss 1.174053
iteration 700 / 1000: loss 1.475161
iteration 800 / 1000: loss 1.351806
iteration 900 / 1000: loss 1.625959
For parameters:  [3.e-02 1.e+02 5.e+01]
Validation accuracy:  0.464
iteration 0 / 1000: loss 2.451267
iteration 100 / 1000: loss 1.872902
iteration 200 / 1000: loss 1.718976
iteration 300 / 1000: loss 1.510867
iteration 400 / 1000: loss 1.567572
iteration 500 / 1000: loss 1.536055
iteration 600 / 1000: loss 1.454384
iteration 700 / 1000: loss 1.620166
iteration 800 / 1000: loss 1.520594
iteration 900 / 1000: loss 1.520137
For parameters:  [3.0e-02 1.2e+02 1.0e+02]
Validation accuracy:  0.475
iteration 0 / 1000: loss 2.491889
iteration 100 / 1000: loss 1.755961
iteration 200 / 1000: loss 1.769272
iteration 300 / 1000: loss 1.588391
iteration 400 / 1000: loss 1.688738
iteration 500 / 1000: loss 1.702200
iteration 600 / 1000: loss 1.547895
iteration 700 / 1000: loss 1.578232
iteration 800 / 1000: loss 1.568111
iteration 900 / 1000: loss 1.724036
For parameters:  [3.e-02 2.e+01 2.e+02]
Validation accuracy:  0.416
iteration 0 / 1000: loss 2.665119
iteration 100 / 1000: loss 2.003268
iteration 200 / 1000: loss 1.749592
iteration 300 / 1000: loss 1.840942
iteration 400 / 1000: loss 2.000071
iteration 500 / 1000: loss 1.545833
iteration 600 / 1000: loss 1.693547
iteration 700 / 1000: loss 1.438741
iteration 800 / 1000: loss 1.539980
iteration 900 / 1000: loss 1.552777
For parameters:  [3.e-02 5.e+01 5.e+01]
Validation accuracy:  0.433
iteration 0 / 1000: loss 2.631301
iteration 100 / 1000: loss 1.838058
iteration 200 / 1000: loss 1.506151
iteration 300 / 1000: loss 1.747571
iteration 400 / 1000: loss 1.568485
iteration 500 / 1000: loss 1.609352
iteration 600 / 1000: loss 1.480039
iteration 700 / 1000: loss 1.310527
iteration 800 / 1000: loss 1.458614
iteration 900 / 1000: loss 1.369625
For parameters:  [3.e-02 8.e+01 1.e+02]
Validation accuracy:  0.472
```

```
iteration 0 / 1000: loss 2.661046
iteration 100 / 1000: loss 1.756782
iteration 200 / 1000: loss 1.621254
iteration 300 / 1000: loss 1.510133
iteration 400 / 1000: loss 1.586507
iteration 500 / 1000: loss 1.560481
iteration 600 / 1000: loss 1.540028
iteration 700 / 1000: loss 1.459255
iteration 800 / 1000: loss 1.304648
iteration 900 / 1000: loss 1.386875
For parameters:  [3.e-02 1.e+02 2.e+02]
Validation accuracy:  0.491
iteration 0 / 1000: loss 2.574094
iteration 100 / 1000: loss 1.983784
iteration 200 / 1000: loss 1.645728
iteration 300 / 1000: loss 1.448059
iteration 400 / 1000: loss 1.729623
iteration 500 / 1000: loss 1.657289
iteration 600 / 1000: loss 1.760682
iteration 700 / 1000: loss 1.624786
iteration 800 / 1000: loss 1.543331
iteration 900 / 1000: loss 1.804036
For parameters:  [3.0e-02 1.2e+02 5.0e+01]
Validation accuracy:  0.451
iteration 0 / 1000: loss 2.360071
iteration 100 / 1000: loss 2.012974
iteration 200 / 1000: loss 1.779618
iteration 300 / 1000: loss 1.784505
iteration 400 / 1000: loss 1.615778
iteration 500 / 1000: loss 1.592882
iteration 600 / 1000: loss 1.653272
iteration 700 / 1000: loss 1.398708
iteration 800 / 1000: loss 1.820599
iteration 900 / 1000: loss 1.541349
For parameters:  [3.e-02 2.e+01 1.e+02]
Validation accuracy:  0.426
iteration 0 / 1000: loss 2.670196
iteration 100 / 1000: loss 1.696313
iteration 200 / 1000: loss 1.694602
iteration 300 / 1000: loss 1.589026
iteration 400 / 1000: loss 1.552249
iteration 500 / 1000: loss 1.499694
iteration 600 / 1000: loss 1.536549
iteration 700 / 1000: loss 1.542548
iteration 800 / 1000: loss 1.389237
iteration 900 / 1000: loss 1.401183
For parameters:  [3.e-02 5.e+01 2.e+02]
Validation accuracy:  0.485
iteration 0 / 1000: loss 2.539125
iteration 100 / 1000: loss 2.057166
iteration 200 / 1000: loss 1.845248
iteration 300 / 1000: loss 1.528448
iteration 400 / 1000: loss 1.707525
iteration 500 / 1000: loss 1.816928
iteration 600 / 1000: loss 1.528792
iteration 700 / 1000: loss 1.604526
iteration 800 / 1000: loss 1.640242
```

```
iteration 900 / 1000: loss 1.621079
For parameters:  [3.e-02 8.e+01 5.e+01]
Validation accuracy:  0.455
iteration 0 / 1000: loss 2.432269
iteration 100 / 1000: loss 1.923649
iteration 200 / 1000: loss 1.557253
iteration 300 / 1000: loss 1.481281
iteration 400 / 1000: loss 1.744673
iteration 500 / 1000: loss 1.654912
iteration 600 / 1000: loss 1.789358
iteration 700 / 1000: loss 1.453594
iteration 800 / 1000: loss 1.354064
iteration 900 / 1000: loss 1.573829
For parameters:  [3.e-02 1.e+02 1.e+02]
Validation accuracy:  0.464
iteration 0 / 1000: loss 2.622189
iteration 100 / 1000: loss 1.698678
iteration 200 / 1000: loss 1.612404
iteration 300 / 1000: loss 1.483058
iteration 400 / 1000: loss 1.544175
iteration 500 / 1000: loss 1.466680
iteration 600 / 1000: loss 1.372916
iteration 700 / 1000: loss 1.443018
iteration 800 / 1000: loss 1.327923
iteration 900 / 1000: loss 1.355483
For parameters:  [3.0e-02 1.2e+02 2.0e+02]
Validation accuracy:  0.493
The best combination of parameters is: learning rate =  0.1 , hidden layer si
ze =  120 , batch size =  200
For this combination, the validation accuracy is  0.524
```

Two-layer Sigmoid Activation Network

```
In [13]: best_2layer_sigmoid = None # store the best model into this

         ##################################################################
         ###
         # TODO: Tune hyperparameters using the validation set. Store your best trained
         #
         # model in best_2layer_sigmoid.
         #
         ##################################################################
         ###

         #Initialize "best" validation accuracy

         best_val_acc = 0

         #Initialize vector to store all validation accuracies

         val_acc_vec = np.zeros(len(parameters_matrix))

         #Cycle through the matrix of parameters

         for i in range(len(parameters_matrix)):

             input_size = 32 * 32 * 3
             num_layers = 2
             hidden_size = int(parameters_matrix[i,1])
             hidden_sizes = [hidden_size]*(num_layers-1)
             num_classes = 10
             learning_rate = parameters_matrix[i,0]
             learning_rate_decay = 0.95
             batch_size = int(parameters_matrix[i,2])
             net = NeuralNetwork(input_size, hidden_sizes, num_classes, num_layers, nonli
         nearity='sigmoid')

             # Train the network
             stats = net.train(X_train, y_train, X_val, y_val,
                         num_iters=1000, batch_size=batch_size,
                         learning_rate=learning_rate, learning_rate_decay=learning_rate_d
         ecay,
                         reg=0.0, verbose=True)

             # Predict on the validation set
             val_acc = (net.predict(X_val) == y_val).mean()

             print('For parameters: ', parameters_matrix[i])
             print('Validation accuracy: ', val_acc)

             #Store the validaction accuracies

             val_acc_vec[i] = val_acc

             #Save the best model and stats

             if val_acc > best_val_acc:

                 best_val_acc = val_acc
```

```
        best_2layer_sigmoid = net
        best_stats = stats

#Print the best combination of trial parameters:

ind = np.argmax(val_acc_vec)

print('The best combination of parameters is: learning rate = ',parameters_mat
rix[ind,0], ', hidden layer size = ',int(parameters_matrix[ind,1]), ', batch s
ize = ',int(parameters_matrix[ind,2]))
print('For this combination, the validation accuracy is ', val_acc_vec[ind])
```

```
iteration 0 / 1000: loss 2.394454
iteration 100 / 1000: loss 1.964292
iteration 200 / 1000: loss 1.812530
iteration 300 / 1000: loss 1.871928
iteration 400 / 1000: loss 1.753414
iteration 500 / 1000: loss 2.034388
iteration 600 / 1000: loss 1.926047
iteration 700 / 1000: loss 1.773440
iteration 800 / 1000: loss 1.973519
iteration 900 / 1000: loss 1.611529
For parameters:  [ 0.1 20.   50. ]
Validation accuracy:  0.401
iteration 0 / 1000: loss 2.349226
iteration 100 / 1000: loss 1.873230
iteration 200 / 1000: loss 1.881829
iteration 300 / 1000: loss 1.686167
iteration 400 / 1000: loss 1.657714
iteration 500 / 1000: loss 1.769798
iteration 600 / 1000: loss 1.679894
iteration 700 / 1000: loss 1.534242
iteration 800 / 1000: loss 1.637482
iteration 900 / 1000: loss 1.567340
For parameters:  [  0.1  50.  100. ]
Validation accuracy:  0.42
iteration 0 / 1000: loss 2.570874
iteration 100 / 1000: loss 1.810058
iteration 200 / 1000: loss 1.679592
iteration 300 / 1000: loss 1.707853
iteration 400 / 1000: loss 1.604382
iteration 500 / 1000: loss 1.711257
iteration 600 / 1000: loss 1.710577
iteration 700 / 1000: loss 1.637946
iteration 800 / 1000: loss 1.621874
iteration 900 / 1000: loss 1.653794
For parameters:  [1.e-01 8.e+01 2.e+02]
Validation accuracy:  0.436
iteration 0 / 1000: loss 2.465977
iteration 100 / 1000: loss 1.940992
iteration 200 / 1000: loss 1.906685
iteration 300 / 1000: loss 1.810134
iteration 400 / 1000: loss 1.861575
iteration 500 / 1000: loss 2.028587
iteration 600 / 1000: loss 1.566310
iteration 700 / 1000: loss 1.476782
iteration 800 / 1000: loss 1.578780
iteration 900 / 1000: loss 1.679164
For parameters:  [  0.1 100.   50. ]
Validation accuracy:  0.428
iteration 0 / 1000: loss 2.328111
iteration 100 / 1000: loss 1.752327
iteration 200 / 1000: loss 1.696437
iteration 300 / 1000: loss 1.724829
iteration 400 / 1000: loss 1.835532
iteration 500 / 1000: loss 1.592002
iteration 600 / 1000: loss 1.707046
iteration 700 / 1000: loss 1.815531
iteration 800 / 1000: loss 1.791658
```

```
iteration 900 / 1000: loss 1.658150
For parameters:  [1.0e-01 1.2e+02 1.0e+02]
Validation accuracy:  0.416
iteration 0 / 1000: loss 2.536106
iteration 100 / 1000: loss 1.886215
iteration 200 / 1000: loss 1.863925
iteration 300 / 1000: loss 1.769263
iteration 400 / 1000: loss 1.754592
iteration 500 / 1000: loss 1.670776
iteration 600 / 1000: loss 1.803482
iteration 700 / 1000: loss 1.722791
iteration 800 / 1000: loss 1.567201
iteration 900 / 1000: loss 1.625860
For parameters:  [1.e-01 2.e+01 2.e+02]
Validation accuracy:  0.402
iteration 0 / 1000: loss 2.420396
iteration 100 / 1000: loss 1.794949
iteration 200 / 1000: loss 1.744375
iteration 300 / 1000: loss 1.741356
iteration 400 / 1000: loss 1.634575
iteration 500 / 1000: loss 1.591375
iteration 600 / 1000: loss 1.441955
iteration 700 / 1000: loss 1.763750
iteration 800 / 1000: loss 1.647008
iteration 900 / 1000: loss 1.649860
For parameters:  [ 0.1 50.   50. ]
Validation accuracy:  0.403
iteration 0 / 1000: loss 2.323143
iteration 100 / 1000: loss 1.797190
iteration 200 / 1000: loss 1.817047
iteration 300 / 1000: loss 1.858731
iteration 400 / 1000: loss 1.613066
iteration 500 / 1000: loss 1.749125
iteration 600 / 1000: loss 1.773327
iteration 700 / 1000: loss 1.738443
iteration 800 / 1000: loss 1.649997
iteration 900 / 1000: loss 1.555992
For parameters:  [  0.1  80.  100. ]
Validation accuracy:  0.423
iteration 0 / 1000: loss 2.488553
iteration 100 / 1000: loss 1.757147
iteration 200 / 1000: loss 1.788353
iteration 300 / 1000: loss 1.831604
iteration 400 / 1000: loss 1.739132
iteration 500 / 1000: loss 1.644011
iteration 600 / 1000: loss 1.656923
iteration 700 / 1000: loss 1.537302
iteration 800 / 1000: loss 1.562717
iteration 900 / 1000: loss 1.671345
For parameters:  [1.e-01 1.e+02 2.e+02]
Validation accuracy:  0.415
iteration 0 / 1000: loss 2.459715
iteration 100 / 1000: loss 1.703319
iteration 200 / 1000: loss 2.061390
iteration 300 / 1000: loss 1.766120
iteration 400 / 1000: loss 1.917015
iteration 500 / 1000: loss 1.632332
```

```
iteration 600 / 1000: loss 1.933932
iteration 700 / 1000: loss 1.596196
iteration 800 / 1000: loss 1.711755
iteration 900 / 1000: loss 1.656928
For parameters:  [1.0e-01 1.2e+02 5.0e+01]
Validation accuracy:  0.413
iteration 0 / 1000: loss 2.487347
iteration 100 / 1000: loss 1.942118
iteration 200 / 1000: loss 1.911423
iteration 300 / 1000: loss 1.845727
iteration 400 / 1000: loss 1.856512
iteration 500 / 1000: loss 1.780575
iteration 600 / 1000: loss 1.678178
iteration 700 / 1000: loss 1.733185
iteration 800 / 1000: loss 1.649580
iteration 900 / 1000: loss 1.754960
For parameters:  [  0.1  20.  100. ]
Validation accuracy:  0.424
iteration 0 / 1000: loss 2.406264
iteration 100 / 1000: loss 1.820504
iteration 200 / 1000: loss 1.799362
iteration 300 / 1000: loss 1.744249
iteration 400 / 1000: loss 1.750998
iteration 500 / 1000: loss 1.575253
iteration 600 / 1000: loss 1.722688
iteration 700 / 1000: loss 1.661904
iteration 800 / 1000: loss 1.515969
iteration 900 / 1000: loss 1.555201
For parameters:  [1.e-01 5.e+01 2.e+02]
Validation accuracy:  0.422
iteration 0 / 1000: loss 2.420010
iteration 100 / 1000: loss 2.016294
iteration 200 / 1000: loss 1.645791
iteration 300 / 1000: loss 1.920713
iteration 400 / 1000: loss 1.582625
iteration 500 / 1000: loss 1.853434
iteration 600 / 1000: loss 1.717103
iteration 700 / 1000: loss 1.844395
iteration 800 / 1000: loss 1.650734
iteration 900 / 1000: loss 1.821369
For parameters:  [ 0.1 80.   50. ]
Validation accuracy:  0.426
iteration 0 / 1000: loss 2.487527
iteration 100 / 1000: loss 1.918807
iteration 200 / 1000: loss 1.560309
iteration 300 / 1000: loss 1.634781
iteration 400 / 1000: loss 1.819056
iteration 500 / 1000: loss 1.778763
iteration 600 / 1000: loss 1.759877
iteration 700 / 1000: loss 1.539222
iteration 800 / 1000: loss 1.625968
iteration 900 / 1000: loss 1.633689
For parameters:  [  0.1 100.  100. ]
Validation accuracy:  0.436
iteration 0 / 1000: loss 2.385669
iteration 100 / 1000: loss 1.783753
iteration 200 / 1000: loss 1.838120
```

```
iteration 300 / 1000: loss 1.616081
iteration 400 / 1000: loss 1.638918
iteration 500 / 1000: loss 1.682697
iteration 600 / 1000: loss 1.565685
iteration 700 / 1000: loss 1.740723
iteration 800 / 1000: loss 1.699835
iteration 900 / 1000: loss 1.649610
For parameters:  [1.0e-01 1.2e+02 2.0e+02]
Validation accuracy:  0.424
iteration 0 / 1000: loss 2.455716
iteration 100 / 1000: loss 2.065960
iteration 200 / 1000: loss 1.855749
iteration 300 / 1000: loss 2.001173
iteration 400 / 1000: loss 1.777433
iteration 500 / 1000: loss 1.789087
iteration 600 / 1000: loss 1.730583
iteration 700 / 1000: loss 1.703707
iteration 800 / 1000: loss 1.752186
iteration 900 / 1000: loss 1.734850
For parameters:  [ 0.05 20.   50.  ]
Validation accuracy:  0.408
iteration 0 / 1000: loss 2.420198
iteration 100 / 1000: loss 1.806616
iteration 200 / 1000: loss 1.746601
iteration 300 / 1000: loss 1.839837
iteration 400 / 1000: loss 1.718981
iteration 500 / 1000: loss 1.633449
iteration 600 / 1000: loss 1.711333
iteration 700 / 1000: loss 1.647152
iteration 800 / 1000: loss 1.836232
iteration 900 / 1000: loss 1.853690
For parameters:  [5.e-02 5.e+01 1.e+02]
Validation accuracy:  0.418
iteration 0 / 1000: loss 2.373002
iteration 100 / 1000: loss 1.904264
iteration 200 / 1000: loss 1.760397
iteration 300 / 1000: loss 1.734230
iteration 400 / 1000: loss 1.643912
iteration 500 / 1000: loss 1.779889
iteration 600 / 1000: loss 1.701976
iteration 700 / 1000: loss 1.780676
iteration 800 / 1000: loss 1.657758
iteration 900 / 1000: loss 1.732420
For parameters:  [5.e-02 8.e+01 2.e+02]
Validation accuracy:  0.412
iteration 0 / 1000: loss 2.282326
iteration 100 / 1000: loss 1.935968
iteration 200 / 1000: loss 1.755951
iteration 300 / 1000: loss 1.693249
iteration 400 / 1000: loss 1.778067
iteration 500 / 1000: loss 1.790226
iteration 600 / 1000: loss 1.809796
iteration 700 / 1000: loss 1.933012
iteration 800 / 1000: loss 1.809176
iteration 900 / 1000: loss 1.881093
For parameters:  [5.e-02 1.e+02 5.e+01]
Validation accuracy:  0.423
```

```
iteration 0 / 1000: loss 2.400095
iteration 100 / 1000: loss 1.831414
iteration 200 / 1000: loss 1.800341
iteration 300 / 1000: loss 1.647824
iteration 400 / 1000: loss 1.650170
iteration 500 / 1000: loss 1.744460
iteration 600 / 1000: loss 1.689547
iteration 700 / 1000: loss 1.907491
iteration 800 / 1000: loss 1.786880
iteration 900 / 1000: loss 1.669094
For parameters:  [5.0e-02 1.2e+02 1.0e+02]
Validation accuracy:  0.418
iteration 0 / 1000: loss 2.476107
iteration 100 / 1000: loss 1.960507
iteration 200 / 1000: loss 1.858606
iteration 300 / 1000: loss 1.862437
iteration 400 / 1000: loss 1.893980
iteration 500 / 1000: loss 1.803977
iteration 600 / 1000: loss 1.747360
iteration 700 / 1000: loss 1.792101
iteration 800 / 1000: loss 1.754027
iteration 900 / 1000: loss 1.756797
For parameters:  [5.e-02 2.e+01 2.e+02]
Validation accuracy:  0.394
iteration 0 / 1000: loss 2.399920
iteration 100 / 1000: loss 1.914639
iteration 200 / 1000: loss 1.911489
iteration 300 / 1000: loss 1.784970
iteration 400 / 1000: loss 1.713829
iteration 500 / 1000: loss 1.696032
iteration 600 / 1000: loss 1.729354
iteration 700 / 1000: loss 1.627333
iteration 800 / 1000: loss 1.813409
iteration 900 / 1000: loss 1.592761
For parameters:  [ 0.05 50.   50.  ]
Validation accuracy:  0.428
iteration 0 / 1000: loss 2.353592
iteration 100 / 1000: loss 2.019339
iteration 200 / 1000: loss 1.909803
iteration 300 / 1000: loss 1.701646
iteration 400 / 1000: loss 1.786880
iteration 500 / 1000: loss 1.758090
iteration 600 / 1000: loss 1.703448
iteration 700 / 1000: loss 1.595363
iteration 800 / 1000: loss 1.641676
iteration 900 / 1000: loss 1.738780
For parameters:  [5.e-02 8.e+01 1.e+02]
Validation accuracy:  0.42
iteration 0 / 1000: loss 2.345795
iteration 100 / 1000: loss 1.815835
iteration 200 / 1000: loss 1.846279
iteration 300 / 1000: loss 1.794705
iteration 400 / 1000: loss 1.733719
iteration 500 / 1000: loss 1.798558
iteration 600 / 1000: loss 1.620289
iteration 700 / 1000: loss 1.760604
iteration 800 / 1000: loss 1.761495
```

```
iteration 900 / 1000: loss 1.677075
For parameters:  [5.e-02 1.e+02 2.e+02]
Validation accuracy:  0.414
iteration 0 / 1000: loss 2.469060
iteration 100 / 1000: loss 1.930331
iteration 200 / 1000: loss 1.770706
iteration 300 / 1000: loss 1.733105
iteration 400 / 1000: loss 1.754774
iteration 500 / 1000: loss 1.604754
iteration 600 / 1000: loss 1.831471
iteration 700 / 1000: loss 1.644048
iteration 800 / 1000: loss 1.546032
iteration 900 / 1000: loss 1.649707
For parameters:  [5.0e-02 1.2e+02 5.0e+01]
Validation accuracy:  0.402
iteration 0 / 1000: loss 2.479579
iteration 100 / 1000: loss 1.944570
iteration 200 / 1000: loss 1.934310
iteration 300 / 1000: loss 1.799130
iteration 400 / 1000: loss 1.735172
iteration 500 / 1000: loss 1.762588
iteration 600 / 1000: loss 1.790943
iteration 700 / 1000: loss 1.840967
iteration 800 / 1000: loss 1.715145
iteration 900 / 1000: loss 1.762889
For parameters:  [5.e-02 2.e+01 1.e+02]
Validation accuracy:  0.41
iteration 0 / 1000: loss 2.416241
iteration 100 / 1000: loss 1.945121
iteration 200 / 1000: loss 1.776174
iteration 300 / 1000: loss 1.901165
iteration 400 / 1000: loss 1.715005
iteration 500 / 1000: loss 1.770175
iteration 600 / 1000: loss 1.740622
iteration 700 / 1000: loss 1.701591
iteration 800 / 1000: loss 1.671792
iteration 900 / 1000: loss 1.693164
For parameters:  [5.e-02 5.e+01 2.e+02]
Validation accuracy:  0.42
iteration 0 / 1000: loss 2.375107
iteration 100 / 1000: loss 2.043450
iteration 200 / 1000: loss 1.928399
iteration 300 / 1000: loss 1.785400
iteration 400 / 1000: loss 1.687530
iteration 500 / 1000: loss 1.859464
iteration 600 / 1000: loss 1.617911
iteration 700 / 1000: loss 1.607465
iteration 800 / 1000: loss 1.714931
iteration 900 / 1000: loss 1.572047
For parameters:  [5.e-02 8.e+01 5.e+01]
Validation accuracy:  0.416
iteration 0 / 1000: loss 2.374696
iteration 100 / 1000: loss 1.871647
iteration 200 / 1000: loss 1.857009
iteration 300 / 1000: loss 1.818499
iteration 400 / 1000: loss 1.694697
iteration 500 / 1000: loss 1.813534
```

```
iteration 600 / 1000: loss 1.755415
iteration 700 / 1000: loss 1.688975
iteration 800 / 1000: loss 1.814380
iteration 900 / 1000: loss 1.769124
For parameters:  [5.e-02 1.e+02 1.e+02]
Validation accuracy:  0.406
iteration 0 / 1000: loss 2.412279
iteration 100 / 1000: loss 1.871401
iteration 200 / 1000: loss 1.867153
iteration 300 / 1000: loss 1.773904
iteration 400 / 1000: loss 1.752046
iteration 500 / 1000: loss 1.672935
iteration 600 / 1000: loss 1.754616
iteration 700 / 1000: loss 1.687278
iteration 800 / 1000: loss 1.712759
iteration 900 / 1000: loss 1.603706
For parameters:  [5.0e-02 1.2e+02 2.0e+02]
Validation accuracy:  0.424
iteration 0 / 1000: loss 2.350478
iteration 100 / 1000: loss 2.026968
iteration 200 / 1000: loss 1.971370
iteration 300 / 1000: loss 1.808695
iteration 400 / 1000: loss 1.902631
iteration 500 / 1000: loss 1.901005
iteration 600 / 1000: loss 1.766567
iteration 700 / 1000: loss 1.691466
iteration 800 / 1000: loss 1.998961
iteration 900 / 1000: loss 1.698829
For parameters:  [3.e-02 2.e+01 5.e+01]
Validation accuracy:  0.386
iteration 0 / 1000: loss 2.340535
iteration 100 / 1000: loss 1.998176
iteration 200 / 1000: loss 1.885393
iteration 300 / 1000: loss 1.939431
iteration 400 / 1000: loss 1.816331
iteration 500 / 1000: loss 1.869912
iteration 600 / 1000: loss 1.743681
iteration 700 / 1000: loss 1.927129
iteration 800 / 1000: loss 1.694740
iteration 900 / 1000: loss 1.799152
For parameters:  [3.e-02 5.e+01 1.e+02]
Validation accuracy:  0.399
iteration 0 / 1000: loss 2.482252
iteration 100 / 1000: loss 1.994098
iteration 200 / 1000: loss 1.873706
iteration 300 / 1000: loss 1.834685
iteration 400 / 1000: loss 1.867152
iteration 500 / 1000: loss 1.679867
iteration 600 / 1000: loss 1.844308
iteration 700 / 1000: loss 1.619347
iteration 800 / 1000: loss 1.776429
iteration 900 / 1000: loss 1.610273
For parameters:  [3.e-02 8.e+01 2.e+02]
Validation accuracy:  0.416
iteration 0 / 1000: loss 2.486651
iteration 100 / 1000: loss 1.908359
iteration 200 / 1000: loss 1.848283
```

```
iteration 300 / 1000: loss 1.599898
iteration 400 / 1000: loss 1.836110
iteration 500 / 1000: loss 1.751169
iteration 600 / 1000: loss 1.821119
iteration 700 / 1000: loss 1.906609
iteration 800 / 1000: loss 1.587478
iteration 900 / 1000: loss 1.576083
For parameters:  [3.e-02 1.e+02 5.e+01]
Validation accuracy:  0.409
iteration 0 / 1000: loss 2.347113
iteration 100 / 1000: loss 1.822641
iteration 200 / 1000: loss 1.899611
iteration 300 / 1000: loss 1.982349
iteration 400 / 1000: loss 1.871965
iteration 500 / 1000: loss 1.705308
iteration 600 / 1000: loss 1.883409
iteration 700 / 1000: loss 1.724040
iteration 800 / 1000: loss 1.695064
iteration 900 / 1000: loss 1.711206
For parameters:  [3.0e-02 1.2e+02 1.0e+02]
Validation accuracy:  0.406
iteration 0 / 1000: loss 2.395162
iteration 100 / 1000: loss 2.006948
iteration 200 / 1000: loss 1.956431
iteration 300 / 1000: loss 1.987345
iteration 400 / 1000: loss 1.903888
iteration 500 / 1000: loss 1.850027
iteration 600 / 1000: loss 1.930595
iteration 700 / 1000: loss 1.767411
iteration 800 / 1000: loss 1.826727
iteration 900 / 1000: loss 1.897840
For parameters:  [3.e-02 2.e+01 2.e+02]
Validation accuracy:  0.398
iteration 0 / 1000: loss 2.312706
iteration 100 / 1000: loss 1.915991
iteration 200 / 1000: loss 1.988096
iteration 300 / 1000: loss 1.854926
iteration 400 / 1000: loss 1.940358
iteration 500 / 1000: loss 1.912949
iteration 600 / 1000: loss 1.756509
iteration 700 / 1000: loss 1.827021
iteration 800 / 1000: loss 1.795509
iteration 900 / 1000: loss 1.707773
For parameters:  [3.e-02 5.e+01 5.e+01]
Validation accuracy:  0.407
iteration 0 / 1000: loss 2.464060
iteration 100 / 1000: loss 2.008343
iteration 200 / 1000: loss 1.703615
iteration 300 / 1000: loss 1.906412
iteration 400 / 1000: loss 1.775457
iteration 500 / 1000: loss 1.727726
iteration 600 / 1000: loss 1.742025
iteration 700 / 1000: loss 1.818305
iteration 800 / 1000: loss 1.837705
iteration 900 / 1000: loss 1.795398
For parameters:  [3.e-02 8.e+01 1.e+02]
Validation accuracy:  0.415
```

```
iteration 0 / 1000: loss 2.453507
iteration 100 / 1000: loss 1.888267
iteration 200 / 1000: loss 1.846556
iteration 300 / 1000: loss 1.869506
iteration 400 / 1000: loss 1.778286
iteration 500 / 1000: loss 1.812916
iteration 600 / 1000: loss 1.840019
iteration 700 / 1000: loss 1.590447
iteration 800 / 1000: loss 1.676140
iteration 900 / 1000: loss 1.559915
For parameters:  [3.e-02 1.e+02 2.e+02]
Validation accuracy:  0.402
iteration 0 / 1000: loss 2.496868
iteration 100 / 1000: loss 1.960003
iteration 200 / 1000: loss 1.817180
iteration 300 / 1000: loss 2.041408
iteration 400 / 1000: loss 1.822522
iteration 500 / 1000: loss 1.620768
iteration 600 / 1000: loss 1.958609
iteration 700 / 1000: loss 1.641498
iteration 800 / 1000: loss 1.776475
iteration 900 / 1000: loss 1.828980
For parameters:  [3.0e-02 1.2e+02 5.0e+01]
Validation accuracy:  0.393
iteration 0 / 1000: loss 2.409908
iteration 100 / 1000: loss 2.035423
iteration 200 / 1000: loss 1.920146
iteration 300 / 1000: loss 2.013233
iteration 400 / 1000: loss 1.809811
iteration 500 / 1000: loss 1.958199
iteration 600 / 1000: loss 1.734700
iteration 700 / 1000: loss 1.779192
iteration 800 / 1000: loss 1.697347
iteration 900 / 1000: loss 1.835212
For parameters:  [3.e-02 2.e+01 1.e+02]
Validation accuracy:  0.394
iteration 0 / 1000: loss 2.430125
iteration 100 / 1000: loss 1.980472
iteration 200 / 1000: loss 1.894741
iteration 300 / 1000: loss 1.876238
iteration 400 / 1000: loss 1.790473
iteration 500 / 1000: loss 1.768539
iteration 600 / 1000: loss 1.786717
iteration 700 / 1000: loss 1.751591
iteration 800 / 1000: loss 1.620372
iteration 900 / 1000: loss 1.717104
For parameters:  [3.e-02 5.e+01 2.e+02]
Validation accuracy:  0.408
iteration 0 / 1000: loss 2.329865
iteration 100 / 1000: loss 1.922829
iteration 200 / 1000: loss 1.866684
iteration 300 / 1000: loss 1.862252
iteration 400 / 1000: loss 1.809355
iteration 500 / 1000: loss 1.688395
iteration 600 / 1000: loss 1.758873
iteration 700 / 1000: loss 1.601702
iteration 800 / 1000: loss 1.975365
```

```
iteration 900 / 1000: loss 1.793599
For parameters:  [3.e-02 8.e+01 5.e+01]
Validation accuracy:  0.411
iteration 0 / 1000: loss 2.532448
iteration 100 / 1000: loss 1.974171
iteration 200 / 1000: loss 1.714697
iteration 300 / 1000: loss 1.844785
iteration 400 / 1000: loss 1.886980
iteration 500 / 1000: loss 1.673030
iteration 600 / 1000: loss 1.871183
iteration 700 / 1000: loss 1.729487
iteration 800 / 1000: loss 1.915398
iteration 900 / 1000: loss 1.631230
For parameters:  [3.e-02 1.e+02 1.e+02]
Validation accuracy:  0.415
iteration 0 / 1000: loss 2.345802
iteration 100 / 1000: loss 1.854969
iteration 200 / 1000: loss 1.849685
iteration 300 / 1000: loss 1.862382
iteration 400 / 1000: loss 1.733506
iteration 500 / 1000: loss 1.847152
iteration 600 / 1000: loss 1.769355
iteration 700 / 1000: loss 1.759110
iteration 800 / 1000: loss 1.783592
iteration 900 / 1000: loss 1.725988
For parameters:  [3.0e-02 1.2e+02 2.0e+02]
Validation accuracy:  0.406
The best combination of parameters is: learning rate =  0.1 , hidden layer si
ze =  80 , batch size =  200
For this combination, the validation accuracy is  0.436
```

Three-layer Relu Activation Network

```
In [16]: best_3layer_relu = None # store the best model into this

         #################################################################################
         ###
         # TODO: Tune hyperparameters using the validation set. Store your best trained
         #
         # model in best_3layer_relu.
         #
         #################################################################################
         ###

         #Initialize "best" validation accuracy

         best_val_acc = 0

         #Initialize vector to store all validation accuracies

         val_acc_vec = np.zeros(len(parameters_matrix))

         #Cycle through the matrix of parameters

         for i in range(len(parameters_matrix)):

             input_size = 32 * 32 * 3
             num_layers = 3
             hidden_size = int(parameters_matrix[i,1])
             hidden_sizes = [hidden_size]*(num_layers-1)
             num_classes = 10
             learning_rate = parameters_matrix[i,0]
             learning_rate_decay = 0.95
             batch_size = int(parameters_matrix[i,2])
             net = NeuralNetwork(input_size, hidden_sizes, num_classes, num_layers, nonli
         nearity='relu')

             # Train the network
             stats = net.train(X_train, y_train, X_val, y_val,
                       num_iters=1000, batch_size=batch_size,
                       learning_rate=learning_rate, learning_rate_decay=learning_rate_d
         ecay,
                       reg=0.0, verbose=True)

             # Predict on the validation set
             val_acc = (net.predict(X_val) == y_val).mean()

             print('For parameters: ', parameters_matrix[i])
             print('Validation accuracy: ', val_acc)

             #Store the validaction accuracies

             val_acc_vec[i] = val_acc

             #Save the best model and stats

             if val_acc > best_val_acc:

                 best_val_acc = val_acc
```

```python
        best_3layer_relu = net
        best_stats = stats

#Print the best combination of trial parameters:

ind = np.argmax(val_acc_vec)

print('The best combination of parameters is: learning rate = ',parameters_mat
rix[ind,0], ', hidden layer size = ',int(parameters_matrix[ind,1]), ', batch s
ize = ',int(parameters_matrix[ind,2]))
print('For this combination, the validation accuracy is ', val_acc_vec[ind])
```

```
iteration 0 / 1000: loss 2.488548
iteration 100 / 1000: loss 2.149827
iteration 200 / 1000: loss 1.861412
iteration 300 / 1000: loss 1.573523
iteration 400 / 1000: loss 1.660306
iteration 500 / 1000: loss 1.895783
iteration 600 / 1000: loss 1.484960
iteration 700 / 1000: loss 1.507817
iteration 800 / 1000: loss 1.812805
iteration 900 / 1000: loss 1.652591
For parameters:  [ 0.1 20.   50. ]
Validation accuracy:  0.398
iteration 0 / 1000: loss 2.506757
iteration 100 / 1000: loss 1.790607
iteration 200 / 1000: loss 1.535027
iteration 300 / 1000: loss 1.468571
iteration 400 / 1000: loss 1.512914
iteration 500 / 1000: loss 1.590604
iteration 600 / 1000: loss 1.384771
iteration 700 / 1000: loss 1.651912
iteration 800 / 1000: loss 1.385913
iteration 900 / 1000: loss 1.587206
For parameters:  [  0.1  50.  100. ]
Validation accuracy:  0.459
iteration 0 / 1000: loss 2.356532
iteration 100 / 1000: loss 1.686930
iteration 200 / 1000: loss 1.563930
iteration 300 / 1000: loss 1.420197
iteration 400 / 1000: loss 1.643392
iteration 500 / 1000: loss 1.403180
iteration 600 / 1000: loss 1.398474
iteration 700 / 1000: loss 1.289954
iteration 800 / 1000: loss 1.273592
iteration 900 / 1000: loss 1.416335
For parameters:  [1.e-01 8.e+01 2.e+02]
Validation accuracy:  0.49
iteration 0 / 1000: loss 2.427977
iteration 100 / 1000: loss 1.795480
iteration 200 / 1000: loss 1.683010
iteration 300 / 1000: loss 1.726709
iteration 400 / 1000: loss 1.568488
iteration 500 / 1000: loss 1.400907
iteration 600 / 1000: loss 1.580139
iteration 700 / 1000: loss 1.256653
iteration 800 / 1000: loss 1.615023
iteration 900 / 1000: loss 1.370663
For parameters:  [  0.1 100.   50. ]
Validation accuracy:  0.447
iteration 0 / 1000: loss 2.368715
iteration 100 / 1000: loss 1.690701
iteration 200 / 1000: loss 1.571183
iteration 300 / 1000: loss 1.525240
iteration 400 / 1000: loss 1.494508
iteration 500 / 1000: loss 1.591508
iteration 600 / 1000: loss 1.668100
iteration 700 / 1000: loss 1.352411
iteration 800 / 1000: loss 1.368516
```

```
iteration 900 / 1000: loss 1.499359
For parameters:  [1.0e-01 1.2e+02 1.0e+02]
Validation accuracy:  0.503
iteration 0 / 1000: loss 2.366797
iteration 100 / 1000: loss 1.871385
iteration 200 / 1000: loss 1.643839
iteration 300 / 1000: loss 1.508153
iteration 400 / 1000: loss 1.581833
iteration 500 / 1000: loss 1.568563
iteration 600 / 1000: loss 1.560668
iteration 700 / 1000: loss 1.669056
iteration 800 / 1000: loss 1.464973
iteration 900 / 1000: loss 1.552902
For parameters:  [1.e-01 2.e+01 2.e+02]
Validation accuracy:  0.441
iteration 0 / 1000: loss 2.410853
iteration 100 / 1000: loss 1.771586
iteration 200 / 1000: loss 2.127667
iteration 300 / 1000: loss 1.591124
iteration 400 / 1000: loss 1.640894
iteration 500 / 1000: loss 1.707135
iteration 600 / 1000: loss 1.651658
iteration 700 / 1000: loss 1.807850
iteration 800 / 1000: loss 1.491906
iteration 900 / 1000: loss 1.324043
For parameters:  [ 0.1 50.   50. ]
Validation accuracy:  0.437
iteration 0 / 1000: loss 2.396920
iteration 100 / 1000: loss 1.857339
iteration 200 / 1000: loss 1.536795
iteration 300 / 1000: loss 1.698601
iteration 400 / 1000: loss 1.466450
iteration 500 / 1000: loss 1.408033
iteration 600 / 1000: loss 1.547902
iteration 700 / 1000: loss 1.691444
iteration 800 / 1000: loss 1.418001
iteration 900 / 1000: loss 1.277043
For parameters:  [  0.1  80.   100. ]
Validation accuracy:  0.465
iteration 0 / 1000: loss 2.403956
iteration 100 / 1000: loss 1.732292
iteration 200 / 1000: loss 1.667944
iteration 300 / 1000: loss 1.454914
iteration 400 / 1000: loss 1.429880
iteration 500 / 1000: loss 1.418885
iteration 600 / 1000: loss 1.542091
iteration 700 / 1000: loss 1.366346
iteration 800 / 1000: loss 1.290044
iteration 900 / 1000: loss 1.229839
For parameters:  [1.e-01 1.e+02 2.e+02]
Validation accuracy:  0.507
iteration 0 / 1000: loss 2.387579
iteration 100 / 1000: loss 1.828161
iteration 200 / 1000: loss 1.648334
iteration 300 / 1000: loss 1.788637
iteration 400 / 1000: loss 1.592732
iteration 500 / 1000: loss 1.225307
```

```
iteration 600 / 1000: loss 1.482539
iteration 700 / 1000: loss 1.486220
iteration 800 / 1000: loss 1.460764
iteration 900 / 1000: loss 1.558186
For parameters:  [1.0e-01 1.2e+02 5.0e+01]
Validation accuracy:  0.45
iteration 0 / 1000: loss 2.341265
iteration 100 / 1000: loss 1.834678
iteration 200 / 1000: loss 1.897337
iteration 300 / 1000: loss 1.619981
iteration 400 / 1000: loss 1.547761
iteration 500 / 1000: loss 1.729191
iteration 600 / 1000: loss 1.734293
iteration 700 / 1000: loss 1.533691
iteration 800 / 1000: loss 1.534677
iteration 900 / 1000: loss 1.280966
For parameters:  [  0.1  20.   100. ]
Validation accuracy:  0.441
iteration 0 / 1000: loss 2.379973
iteration 100 / 1000: loss 1.711375
iteration 200 / 1000: loss 1.580873
iteration 300 / 1000: loss 1.512723
iteration 400 / 1000: loss 1.486065
iteration 500 / 1000: loss 1.475074
iteration 600 / 1000: loss 1.567917
iteration 700 / 1000: loss 1.465371
iteration 800 / 1000: loss 1.338320
iteration 900 / 1000: loss 1.419387
For parameters:  [1.e-01 5.e+01 2.e+02]
Validation accuracy:  0.492
iteration 0 / 1000: loss 2.474014
iteration 100 / 1000: loss 1.787571
iteration 200 / 1000: loss 1.740740
iteration 300 / 1000: loss 1.780470
iteration 400 / 1000: loss 1.746044
iteration 500 / 1000: loss 1.864854
iteration 600 / 1000: loss 1.773115
iteration 700 / 1000: loss 1.789275
iteration 800 / 1000: loss 1.453697
iteration 900 / 1000: loss 1.689362
For parameters:  [ 0.1 80.   50. ]
Validation accuracy:  0.44
iteration 0 / 1000: loss 2.496058
iteration 100 / 1000: loss 1.863945
iteration 200 / 1000: loss 1.682928
iteration 300 / 1000: loss 1.536935
iteration 400 / 1000: loss 1.511178
iteration 500 / 1000: loss 1.430720
iteration 600 / 1000: loss 1.507224
iteration 700 / 1000: loss 1.529159
iteration 800 / 1000: loss 1.497969
iteration 900 / 1000: loss 1.380143
For parameters:  [  0.1 100.   100. ]
Validation accuracy:  0.475
iteration 0 / 1000: loss 2.435805
iteration 100 / 1000: loss 1.635913
iteration 200 / 1000: loss 1.724216
```

```
iteration 300 / 1000: loss 1.564232
iteration 400 / 1000: loss 1.522211
iteration 500 / 1000: loss 1.524997
iteration 600 / 1000: loss 1.324900
iteration 700 / 1000: loss 1.413647
iteration 800 / 1000: loss 1.249312
iteration 900 / 1000: loss 1.269321
For parameters:  [1.0e-01 1.2e+02 2.0e+02]
Validation accuracy:  0.477
iteration 0 / 1000: loss 2.483989
iteration 100 / 1000: loss 1.850517
iteration 200 / 1000: loss 1.908296
iteration 300 / 1000: loss 1.808306
iteration 400 / 1000: loss 1.622488
iteration 500 / 1000: loss 1.725570
iteration 600 / 1000: loss 1.549689
iteration 700 / 1000: loss 1.715273
iteration 800 / 1000: loss 1.739351
iteration 900 / 1000: loss 1.736076
For parameters:  [ 0.05 20.   50.  ]
Validation accuracy:  0.422
iteration 0 / 1000: loss 2.478430
iteration 100 / 1000: loss 1.807153
iteration 200 / 1000: loss 1.873974
iteration 300 / 1000: loss 1.626945
iteration 400 / 1000: loss 1.464768
iteration 500 / 1000: loss 1.438573
iteration 600 / 1000: loss 1.444805
iteration 700 / 1000: loss 1.695742
iteration 800 / 1000: loss 1.380290
iteration 900 / 1000: loss 1.434087
For parameters:  [5.e-02 5.e+01 1.e+02]
Validation accuracy:  0.466
iteration 0 / 1000: loss 2.467992
iteration 100 / 1000: loss 1.655150
iteration 200 / 1000: loss 1.604073
iteration 300 / 1000: loss 1.459405
iteration 400 / 1000: loss 1.435752
iteration 500 / 1000: loss 1.382560
iteration 600 / 1000: loss 1.352078
iteration 700 / 1000: loss 1.266220
iteration 800 / 1000: loss 1.358554
iteration 900 / 1000: loss 1.434934
For parameters:  [5.e-02 8.e+01 2.e+02]
Validation accuracy:  0.481
iteration 0 / 1000: loss 2.389696
iteration 100 / 1000: loss 2.028125
iteration 200 / 1000: loss 1.573349
iteration 300 / 1000: loss 1.424485
iteration 400 / 1000: loss 1.756657
iteration 500 / 1000: loss 1.604768
iteration 600 / 1000: loss 1.816790
iteration 700 / 1000: loss 1.598568
iteration 800 / 1000: loss 1.417245
iteration 900 / 1000: loss 1.768143
For parameters:  [5.e-02 1.e+02 5.e+01]
Validation accuracy:  0.458
```

```
iteration 0 / 1000: loss 2.477916
iteration 100 / 1000: loss 1.628520
iteration 200 / 1000: loss 1.738098
iteration 300 / 1000: loss 1.715109
iteration 400 / 1000: loss 1.606572
iteration 500 / 1000: loss 1.390299
iteration 600 / 1000: loss 1.504065
iteration 700 / 1000: loss 1.485633
iteration 800 / 1000: loss 1.484221
iteration 900 / 1000: loss 1.408944
For parameters:  [5.0e-02 1.2e+02 1.0e+02]
Validation accuracy:  0.46
iteration 0 / 1000: loss 2.378349
iteration 100 / 1000: loss 1.798541
iteration 200 / 1000: loss 1.741227
iteration 300 / 1000: loss 1.744580
iteration 400 / 1000: loss 1.682288
iteration 500 / 1000: loss 1.612828
iteration 600 / 1000: loss 1.647541
iteration 700 / 1000: loss 1.564386
iteration 800 / 1000: loss 1.521308
iteration 900 / 1000: loss 1.496103
For parameters:  [5.e-02 2.e+01 2.e+02]
Validation accuracy:  0.435
iteration 0 / 1000: loss 2.403335
iteration 100 / 1000: loss 1.760945
iteration 200 / 1000: loss 1.788317
iteration 300 / 1000: loss 1.673200
iteration 400 / 1000: loss 1.759980
iteration 500 / 1000: loss 1.639813
iteration 600 / 1000: loss 1.630916
iteration 700 / 1000: loss 1.753673
iteration 800 / 1000: loss 1.867768
iteration 900 / 1000: loss 1.692217
For parameters:  [ 0.05 50.   50.  ]
Validation accuracy:  0.447
iteration 0 / 1000: loss 2.461821
iteration 100 / 1000: loss 1.652415
iteration 200 / 1000: loss 1.618583
iteration 300 / 1000: loss 1.562309
iteration 400 / 1000: loss 1.530729
iteration 500 / 1000: loss 1.476320
iteration 600 / 1000: loss 1.678424
iteration 700 / 1000: loss 1.483633
iteration 800 / 1000: loss 1.443606
iteration 900 / 1000: loss 1.396224
For parameters:  [5.e-02 8.e+01 1.e+02]
Validation accuracy:  0.477
iteration 0 / 1000: loss 2.352702
iteration 100 / 1000: loss 1.752401
iteration 200 / 1000: loss 1.647341
iteration 300 / 1000: loss 1.613936
iteration 400 / 1000: loss 1.540483
iteration 500 / 1000: loss 1.379023
iteration 600 / 1000: loss 1.421652
iteration 700 / 1000: loss 1.408681
iteration 800 / 1000: loss 1.390706
```

```
iteration 900 / 1000: loss 1.262250
For parameters:  [5.e-02 1.e+02 2.e+02]
Validation accuracy:  0.492
iteration 0 / 1000: loss 2.531880
iteration 100 / 1000: loss 1.759208
iteration 200 / 1000: loss 1.777803
iteration 300 / 1000: loss 1.403432
iteration 400 / 1000: loss 1.411589
iteration 500 / 1000: loss 1.783714
iteration 600 / 1000: loss 1.325228
iteration 700 / 1000: loss 1.551734
iteration 800 / 1000: loss 1.339371
iteration 900 / 1000: loss 1.531675
For parameters:  [5.0e-02 1.2e+02 5.0e+01]
Validation accuracy:  0.461
iteration 0 / 1000: loss 2.417943
iteration 100 / 1000: loss 1.767447
iteration 200 / 1000: loss 1.631187
iteration 300 / 1000: loss 1.584551
iteration 400 / 1000: loss 1.649359
iteration 500 / 1000: loss 1.602876
iteration 600 / 1000: loss 1.657365
iteration 700 / 1000: loss 1.596195
iteration 800 / 1000: loss 1.555987
iteration 900 / 1000: loss 1.501673
For parameters:  [5.e-02 2.e+01 1.e+02]
Validation accuracy:  0.411
iteration 0 / 1000: loss 2.462928
iteration 100 / 1000: loss 1.800741
iteration 200 / 1000: loss 1.657112
iteration 300 / 1000: loss 1.605263
iteration 400 / 1000: loss 1.610468
iteration 500 / 1000: loss 1.592227
iteration 600 / 1000: loss 1.465059
iteration 700 / 1000: loss 1.428843
iteration 800 / 1000: loss 1.554932
iteration 900 / 1000: loss 1.430861
For parameters:  [5.e-02 5.e+01 2.e+02]
Validation accuracy:  0.463
iteration 0 / 1000: loss 2.319795
iteration 100 / 1000: loss 1.750372
iteration 200 / 1000: loss 1.892024
iteration 300 / 1000: loss 1.643956
iteration 400 / 1000: loss 1.566700
iteration 500 / 1000: loss 1.775872
iteration 600 / 1000: loss 1.392110
iteration 700 / 1000: loss 1.513182
iteration 800 / 1000: loss 1.620749
iteration 900 / 1000: loss 1.560164
For parameters:  [5.e-02 8.e+01 5.e+01]
Validation accuracy:  0.451
iteration 0 / 1000: loss 2.454141
iteration 100 / 1000: loss 1.579955
iteration 200 / 1000: loss 1.642354
iteration 300 / 1000: loss 1.650825
iteration 400 / 1000: loss 1.715080
iteration 500 / 1000: loss 1.457042
```

```
iteration 600 / 1000: loss 1.522399
iteration 700 / 1000: loss 1.478631
iteration 800 / 1000: loss 1.516866
iteration 900 / 1000: loss 1.260067
For parameters:  [5.e-02 1.e+02 1.e+02]
Validation accuracy:  0.495
iteration 0 / 1000: loss 2.386968
iteration 100 / 1000: loss 1.596986
iteration 200 / 1000: loss 1.554853
iteration 300 / 1000: loss 1.583574
iteration 400 / 1000: loss 1.402773
iteration 500 / 1000: loss 1.378430
iteration 600 / 1000: loss 1.375337
iteration 700 / 1000: loss 1.315856
iteration 800 / 1000: loss 1.275970
iteration 900 / 1000: loss 1.365006
For parameters:  [5.0e-02 1.2e+02 2.0e+02]
Validation accuracy:  0.501
iteration 0 / 1000: loss 2.448250
iteration 100 / 1000: loss 2.110432
iteration 200 / 1000: loss 1.665533
iteration 300 / 1000: loss 1.843127
iteration 400 / 1000: loss 1.807560
iteration 500 / 1000: loss 1.773654
iteration 600 / 1000: loss 1.761658
iteration 700 / 1000: loss 1.285798
iteration 800 / 1000: loss 1.898557
iteration 900 / 1000: loss 1.426165
For parameters:  [3.e-02 2.e+01 5.e+01]
Validation accuracy:  0.408
iteration 0 / 1000: loss 2.422844
iteration 100 / 1000: loss 1.886660
iteration 200 / 1000: loss 1.700300
iteration 300 / 1000: loss 1.896407
iteration 400 / 1000: loss 1.371108
iteration 500 / 1000: loss 1.518603
iteration 600 / 1000: loss 1.491910
iteration 700 / 1000: loss 1.587119
iteration 800 / 1000: loss 1.451214
iteration 900 / 1000: loss 1.448096
For parameters:  [3.e-02 5.e+01 1.e+02]
Validation accuracy:  0.461
iteration 0 / 1000: loss 2.432160
iteration 100 / 1000: loss 1.759717
iteration 200 / 1000: loss 1.637465
iteration 300 / 1000: loss 1.685979
iteration 400 / 1000: loss 1.483603
iteration 500 / 1000: loss 1.545669
iteration 600 / 1000: loss 1.441953
iteration 700 / 1000: loss 1.512083
iteration 800 / 1000: loss 1.470915
iteration 900 / 1000: loss 1.430387
For parameters:  [3.e-02 8.e+01 2.e+02]
Validation accuracy:  0.488
iteration 0 / 1000: loss 2.475252
iteration 100 / 1000: loss 1.811449
iteration 200 / 1000: loss 1.636489
```

```
iteration 300 / 1000: loss 1.640049
iteration 400 / 1000: loss 1.721024
iteration 500 / 1000: loss 1.778773
iteration 600 / 1000: loss 1.373376
iteration 700 / 1000: loss 1.557382
iteration 800 / 1000: loss 1.594996
iteration 900 / 1000: loss 1.253603
For parameters:  [3.e-02 1.e+02 5.e+01]
Validation accuracy:  0.452
iteration 0 / 1000: loss 2.435103
iteration 100 / 1000: loss 1.801877
iteration 200 / 1000: loss 1.673828
iteration 300 / 1000: loss 1.692316
iteration 400 / 1000: loss 1.661874
iteration 500 / 1000: loss 1.472946
iteration 600 / 1000: loss 1.600589
iteration 700 / 1000: loss 1.433157
iteration 800 / 1000: loss 1.437748
iteration 900 / 1000: loss 1.228969
For parameters:  [3.0e-02 1.2e+02 1.0e+02]
Validation accuracy:  0.479
iteration 0 / 1000: loss 2.347293
iteration 100 / 1000: loss 1.899067
iteration 200 / 1000: loss 1.753977
iteration 300 / 1000: loss 1.684895
iteration 400 / 1000: loss 1.737257
iteration 500 / 1000: loss 1.698016
iteration 600 / 1000: loss 1.694025
iteration 700 / 1000: loss 1.521366
iteration 800 / 1000: loss 1.605131
iteration 900 / 1000: loss 1.594005
For parameters:  [3.e-02 2.e+01 2.e+02]
Validation accuracy:  0.435
iteration 0 / 1000: loss 2.389172
iteration 100 / 1000: loss 1.943899
iteration 200 / 1000: loss 1.964574
iteration 300 / 1000: loss 1.655812
iteration 400 / 1000: loss 1.687209
iteration 500 / 1000: loss 1.422541
iteration 600 / 1000: loss 1.659894
iteration 700 / 1000: loss 1.809398
iteration 800 / 1000: loss 1.628091
iteration 900 / 1000: loss 1.548941
For parameters:  [3.e-02 5.e+01 5.e+01]
Validation accuracy:  0.447
iteration 0 / 1000: loss 2.413365
iteration 100 / 1000: loss 1.866576
iteration 200 / 1000: loss 1.730257
iteration 300 / 1000: loss 1.662296
iteration 400 / 1000: loss 1.563400
iteration 500 / 1000: loss 1.623761
iteration 600 / 1000: loss 1.692550
iteration 700 / 1000: loss 1.285304
iteration 800 / 1000: loss 1.429589
iteration 900 / 1000: loss 1.559710
For parameters:  [3.e-02 8.e+01 1.e+02]
Validation accuracy:  0.481
```

```
iteration 0 / 1000: loss 2.369549
iteration 100 / 1000: loss 1.744381
iteration 200 / 1000: loss 1.712059
iteration 300 / 1000: loss 1.589181
iteration 400 / 1000: loss 1.518529
iteration 500 / 1000: loss 1.573149
iteration 600 / 1000: loss 1.398003
iteration 700 / 1000: loss 1.294511
iteration 800 / 1000: loss 1.483313
iteration 900 / 1000: loss 1.467256
For parameters:  [3.e-02 1.e+02 2.e+02]
Validation accuracy:  0.48
iteration 0 / 1000: loss 2.433897
iteration 100 / 1000: loss 1.942873
iteration 200 / 1000: loss 1.849292
iteration 300 / 1000: loss 1.610350
iteration 400 / 1000: loss 1.602557
iteration 500 / 1000: loss 1.213208
iteration 600 / 1000: loss 1.624326
iteration 700 / 1000: loss 1.665147
iteration 800 / 1000: loss 1.373463
iteration 900 / 1000: loss 1.390888
For parameters:  [3.0e-02 1.2e+02 5.0e+01]
Validation accuracy:  0.459
iteration 0 / 1000: loss 2.549713
iteration 100 / 1000: loss 1.921282
iteration 200 / 1000: loss 1.726941
iteration 300 / 1000: loss 1.819238
iteration 400 / 1000: loss 1.521763
iteration 500 / 1000: loss 1.688051
iteration 600 / 1000: loss 1.847164
iteration 700 / 1000: loss 1.636834
iteration 800 / 1000: loss 1.468065
iteration 900 / 1000: loss 1.574748
For parameters:  [3.e-02 2.e+01 1.e+02]
Validation accuracy:  0.423
iteration 0 / 1000: loss 2.419260
iteration 100 / 1000: loss 1.816838
iteration 200 / 1000: loss 1.684026
iteration 300 / 1000: loss 1.623757
iteration 400 / 1000: loss 1.623373
iteration 500 / 1000: loss 1.496840
iteration 600 / 1000: loss 1.574236
iteration 700 / 1000: loss 1.358700
iteration 800 / 1000: loss 1.518648
iteration 900 / 1000: loss 1.544579
For parameters:  [3.e-02 5.e+01 2.e+02]
Validation accuracy:  0.485
iteration 0 / 1000: loss 2.389910
iteration 100 / 1000: loss 1.677383
iteration 200 / 1000: loss 1.815267
iteration 300 / 1000: loss 1.723192
iteration 400 / 1000: loss 1.675641
iteration 500 / 1000: loss 1.700449
iteration 600 / 1000: loss 1.626160
iteration 700 / 1000: loss 1.608528
iteration 800 / 1000: loss 1.574901
```

```
iteration 900 / 1000: loss 1.697080
For parameters:  [3.e-02 8.e+01 5.e+01]
Validation accuracy:  0.441
iteration 0 / 1000: loss 2.412420
iteration 100 / 1000: loss 1.898045
iteration 200 / 1000: loss 1.734733
iteration 300 / 1000: loss 1.649473
iteration 400 / 1000: loss 1.453475
iteration 500 / 1000: loss 1.713627
iteration 600 / 1000: loss 1.467008
iteration 700 / 1000: loss 1.527426
iteration 800 / 1000: loss 1.672172
iteration 900 / 1000: loss 1.361769
For parameters:  [3.e-02 1.e+02 1.e+02]
Validation accuracy:  0.475
iteration 0 / 1000: loss 2.315362
iteration 100 / 1000: loss 1.653302
iteration 200 / 1000: loss 1.606265
iteration 300 / 1000: loss 1.582040
iteration 400 / 1000: loss 1.599255
iteration 500 / 1000: loss 1.589134
iteration 600 / 1000: loss 1.472316
iteration 700 / 1000: loss 1.517007
iteration 800 / 1000: loss 1.286834
iteration 900 / 1000: loss 1.183258
For parameters:  [3.0e-02 1.2e+02 2.0e+02]
Validation accuracy:  0.478
The best combination of parameters is: learning rate =  0.1 , hidden layer si
ze =  100 , batch size =  200
For this combination, the validation accuracy is  0.507
```

Three-layer Sigmoid Activation Network

```
In [18]: best_3layer_sigmoid = None # store the best model into this

         ########################################################################
         ###
         # TODO: Tune hyperparameters using the validation set. Store your best trained
         #
         # model in best_3layer_sigmoid.
         #
         ########################################################################
         ###

         #Initialize "best" validation accuracy

         best_val_acc = 0

         #Initialize vector to store all validation accuracies

         val_acc_vec = np.zeros(len(parameters_matrix))

         #Cycle through the matrix of parameters

         for i in range(len(parameters_matrix)):

             input_size = 32 * 32 * 3
             num_layers = 3
             hidden_size = int(parameters_matrix[i,1])
             hidden_sizes = [hidden_size]*(num_layers-1)
             num_classes = 10
             learning_rate = parameters_matrix[i,0]
             learning_rate_decay = 0.95
             batch_size = int(parameters_matrix[i,2])
             net = NeuralNetwork(input_size, hidden_sizes, num_classes, num_layers, nonli
         nearity='sigmoid')

             # Train the network
             stats = net.train(X_train, y_train, X_val, y_val,
                         num_iters=1000, batch_size=batch_size,
                         learning_rate=learning_rate, learning_rate_decay=learning_rate_d
         ecay,
                         reg=0.0, verbose=True)

             # Predict on the validation set
             val_acc = (net.predict(X_val) == y_val).mean()

             print('For parameters: ', parameters_matrix[i])
             print('Validation accuracy: ', val_acc)

             #Store the validaction accuracies

             val_acc_vec[i] = val_acc

             #Save the best model and stats

             if val_acc > best_val_acc:

                 best_val_acc = val_acc
```

```
        best_3layer_sigmoid = net
        best_stats = stats

#Print the best combination of trial parameters:

ind = np.argmax(val_acc_vec)

print('The best combination of parameters is: learning rate = ',parameters_mat
rix[ind,0], ', hidden layer size = ',int(parameters_matrix[ind,1]), ', batch s
ize = ',int(parameters_matrix[ind,2]))
print('For this combination, the validation accuracy is ', val_acc_vec[ind])
```

```
iteration 0 / 1000: loss 2.341498
iteration 100 / 1000: loss 2.207852
iteration 200 / 1000: loss 2.139997
iteration 300 / 1000: loss 2.051488
iteration 400 / 1000: loss 2.049709
iteration 500 / 1000: loss 2.104036
iteration 600 / 1000: loss 2.050512
iteration 700 / 1000: loss 1.851410
iteration 800 / 1000: loss 1.940754
iteration 900 / 1000: loss 2.018969
For parameters:  [ 0.1 20.  50. ]
Validation accuracy:  0.357
iteration 0 / 1000: loss 2.448612
iteration 100 / 1000: loss 2.077613
iteration 200 / 1000: loss 2.039883
iteration 300 / 1000: loss 2.033507
iteration 400 / 1000: loss 1.895880
iteration 500 / 1000: loss 1.932328
iteration 600 / 1000: loss 1.925898
iteration 700 / 1000: loss 1.838447
iteration 800 / 1000: loss 1.856493
iteration 900 / 1000: loss 1.591386
For parameters:  [  0.1  50.  100. ]
Validation accuracy:  0.386
iteration 0 / 1000: loss 2.364265
iteration 100 / 1000: loss 2.081808
iteration 200 / 1000: loss 2.016848
iteration 300 / 1000: loss 1.931935
iteration 400 / 1000: loss 1.859091
iteration 500 / 1000: loss 1.844422
iteration 600 / 1000: loss 1.795662
iteration 700 / 1000: loss 1.825233
iteration 800 / 1000: loss 1.681379
iteration 900 / 1000: loss 1.788236
For parameters:  [1.e-01 8.e+01 2.e+02]
Validation accuracy:  0.414
iteration 0 / 1000: loss 2.413968
iteration 100 / 1000: loss 2.127262
iteration 200 / 1000: loss 2.028107
iteration 300 / 1000: loss 1.918618
iteration 400 / 1000: loss 1.834029
iteration 500 / 1000: loss 1.918868
iteration 600 / 1000: loss 1.710111
iteration 700 / 1000: loss 1.979391
iteration 800 / 1000: loss 1.606796
iteration 900 / 1000: loss 1.870395
For parameters:  [  0.1 100.   50. ]
Validation accuracy:  0.402
iteration 0 / 1000: loss 2.389077
iteration 100 / 1000: loss 2.102345
iteration 200 / 1000: loss 1.956554
iteration 300 / 1000: loss 1.985965
iteration 400 / 1000: loss 1.777410
iteration 500 / 1000: loss 1.974525
iteration 600 / 1000: loss 1.737353
iteration 700 / 1000: loss 1.668831
iteration 800 / 1000: loss 1.772797
```

```
iteration 900 / 1000: loss 1.746701
For parameters:  [1.0e-01 1.2e+02 1.0e+02]
Validation accuracy:  0.384
iteration 0 / 1000: loss 2.340397
iteration 100 / 1000: loss 2.210570
iteration 200 / 1000: loss 2.149262
iteration 300 / 1000: loss 2.050630
iteration 400 / 1000: loss 2.020849
iteration 500 / 1000: loss 1.944951
iteration 600 / 1000: loss 1.877223
iteration 700 / 1000: loss 1.891415
iteration 800 / 1000: loss 1.842731
iteration 900 / 1000: loss 1.780380
For parameters:  [1.e-01 2.e+01 2.e+02]
Validation accuracy:  0.364
iteration 0 / 1000: loss 2.370122
iteration 100 / 1000: loss 2.148976
iteration 200 / 1000: loss 2.010571
iteration 300 / 1000: loss 2.057767
iteration 400 / 1000: loss 1.919012
iteration 500 / 1000: loss 1.860668
iteration 600 / 1000: loss 1.922120
iteration 700 / 1000: loss 1.898974
iteration 800 / 1000: loss 1.886093
iteration 900 / 1000: loss 1.821627
For parameters:  [ 0.1 50.   50. ]
Validation accuracy:  0.374
iteration 0 / 1000: loss 2.479070
iteration 100 / 1000: loss 2.042379
iteration 200 / 1000: loss 2.066532
iteration 300 / 1000: loss 1.969494
iteration 400 / 1000: loss 1.981710
iteration 500 / 1000: loss 1.830800
iteration 600 / 1000: loss 1.884657
iteration 700 / 1000: loss 1.804319
iteration 800 / 1000: loss 1.784642
iteration 900 / 1000: loss 1.819560
For parameters:  [  0.1  80.   100. ]
Validation accuracy:  0.379
iteration 0 / 1000: loss 2.458014
iteration 100 / 1000: loss 2.117269
iteration 200 / 1000: loss 2.002774
iteration 300 / 1000: loss 1.930603
iteration 400 / 1000: loss 1.880399
iteration 500 / 1000: loss 1.828451
iteration 600 / 1000: loss 1.828094
iteration 700 / 1000: loss 1.844087
iteration 800 / 1000: loss 1.813307
iteration 900 / 1000: loss 1.770259
For parameters:  [1.e-01 1.e+02 2.e+02]
Validation accuracy:  0.371
iteration 0 / 1000: loss 2.481327
iteration 100 / 1000: loss 2.065398
iteration 200 / 1000: loss 1.981070
iteration 300 / 1000: loss 1.781368
iteration 400 / 1000: loss 1.890408
iteration 500 / 1000: loss 1.905495
```

```
iteration 600 / 1000: loss 1.839487
iteration 700 / 1000: loss 1.854714
iteration 800 / 1000: loss 1.843192
iteration 900 / 1000: loss 1.760606
For parameters:  [1.0e-01 1.2e+02 5.0e+01]
Validation accuracy:  0.406
iteration 0 / 1000: loss 2.421596
iteration 100 / 1000: loss 2.220491
iteration 200 / 1000: loss 2.189993
iteration 300 / 1000: loss 2.026762
iteration 400 / 1000: loss 2.038135
iteration 500 / 1000: loss 2.111229
iteration 600 / 1000: loss 1.932581
iteration 700 / 1000: loss 1.892606
iteration 800 / 1000: loss 1.821492
iteration 900 / 1000: loss 1.926973
For parameters:  [  0.1  20.   100. ]
Validation accuracy:  0.355
iteration 0 / 1000: loss 2.349100
iteration 100 / 1000: loss 2.099988
iteration 200 / 1000: loss 2.061398
iteration 300 / 1000: loss 1.875662
iteration 400 / 1000: loss 1.934079
iteration 500 / 1000: loss 1.871254
iteration 600 / 1000: loss 1.806076
iteration 700 / 1000: loss 1.941981
iteration 800 / 1000: loss 1.917081
iteration 900 / 1000: loss 1.861327
For parameters:  [1.e-01 5.e+01 2.e+02]
Validation accuracy:  0.382
iteration 0 / 1000: loss 2.370517
iteration 100 / 1000: loss 2.115029
iteration 200 / 1000: loss 2.169580
iteration 300 / 1000: loss 2.089761
iteration 400 / 1000: loss 1.956986
iteration 500 / 1000: loss 1.783870
iteration 600 / 1000: loss 1.788431
iteration 700 / 1000: loss 1.675544
iteration 800 / 1000: loss 1.800275
iteration 900 / 1000: loss 1.747724
For parameters:  [ 0.1 80.   50. ]
Validation accuracy:  0.389
iteration 0 / 1000: loss 2.424740
iteration 100 / 1000: loss 2.076541
iteration 200 / 1000: loss 1.857389
iteration 300 / 1000: loss 1.913411
iteration 400 / 1000: loss 1.768074
iteration 500 / 1000: loss 1.808753
iteration 600 / 1000: loss 1.815828
iteration 700 / 1000: loss 1.783271
iteration 800 / 1000: loss 1.657926
iteration 900 / 1000: loss 1.786941
For parameters:  [  0.1 100.   100. ]
Validation accuracy:  0.388
iteration 0 / 1000: loss 2.414725
iteration 100 / 1000: loss 2.075634
iteration 200 / 1000: loss 1.901399
```

```
iteration 300 / 1000: loss 1.885077
iteration 400 / 1000: loss 1.822682
iteration 500 / 1000: loss 1.898381
iteration 600 / 1000: loss 1.792772
iteration 700 / 1000: loss 1.765559
iteration 800 / 1000: loss 1.855042
iteration 900 / 1000: loss 1.734327
For parameters:  [1.0e-01 1.2e+02 2.0e+02]
Validation accuracy:  0.407
iteration 0 / 1000: loss 2.351040
iteration 100 / 1000: loss 2.266038
iteration 200 / 1000: loss 2.194641
iteration 300 / 1000: loss 2.204253
iteration 400 / 1000: loss 2.172737
iteration 500 / 1000: loss 2.078930
iteration 600 / 1000: loss 2.101879
iteration 700 / 1000: loss 1.967918
iteration 800 / 1000: loss 1.987544
iteration 900 / 1000: loss 1.879330
For parameters:  [ 0.05 20.   50.  ]
Validation accuracy:  0.329
iteration 0 / 1000: loss 2.370049
iteration 100 / 1000: loss 2.168831
iteration 200 / 1000: loss 2.093210
iteration 300 / 1000: loss 2.053266
iteration 400 / 1000: loss 2.101964
iteration 500 / 1000: loss 1.908137
iteration 600 / 1000: loss 1.999922
iteration 700 / 1000: loss 1.909557
iteration 800 / 1000: loss 1.847564
iteration 900 / 1000: loss 1.904145
For parameters:  [5.e-02 5.e+01 1.e+02]
Validation accuracy:  0.351
iteration 0 / 1000: loss 2.412317
iteration 100 / 1000: loss 2.198026
iteration 200 / 1000: loss 2.095225
iteration 300 / 1000: loss 1.976668
iteration 400 / 1000: loss 1.941050
iteration 500 / 1000: loss 1.910940
iteration 600 / 1000: loss 1.878348
iteration 700 / 1000: loss 1.938718
iteration 800 / 1000: loss 1.935688
iteration 900 / 1000: loss 1.807334
For parameters:  [5.e-02 8.e+01 2.e+02]
Validation accuracy:  0.352
iteration 0 / 1000: loss 2.486944
iteration 100 / 1000: loss 2.256717
iteration 200 / 1000: loss 2.028991
iteration 300 / 1000: loss 1.982665
iteration 400 / 1000: loss 2.100781
iteration 500 / 1000: loss 1.986462
iteration 600 / 1000: loss 1.942680
iteration 700 / 1000: loss 1.698867
iteration 800 / 1000: loss 1.729140
iteration 900 / 1000: loss 1.840234
For parameters:  [5.e-02 1.e+02 5.e+01]
Validation accuracy:  0.38
```

```
iteration 0 / 1000: loss 2.315558
iteration 100 / 1000: loss 2.128884
iteration 200 / 1000: loss 2.012528
iteration 300 / 1000: loss 2.068834
iteration 400 / 1000: loss 1.972363
iteration 500 / 1000: loss 1.995456
iteration 600 / 1000: loss 1.882271
iteration 700 / 1000: loss 1.860277
iteration 800 / 1000: loss 1.784486
iteration 900 / 1000: loss 1.893442
For parameters:  [5.0e-02 1.2e+02 1.0e+02]
Validation accuracy:  0.374
iteration 0 / 1000: loss 2.332475
iteration 100 / 1000: loss 2.225142
iteration 200 / 1000: loss 2.162154
iteration 300 / 1000: loss 2.135651
iteration 400 / 1000: loss 2.100617
iteration 500 / 1000: loss 2.048210
iteration 600 / 1000: loss 2.094557
iteration 700 / 1000: loss 1.985015
iteration 800 / 1000: loss 2.026365
iteration 900 / 1000: loss 1.992307
For parameters:  [5.e-02 2.e+01 2.e+02]
Validation accuracy:  0.293
iteration 0 / 1000: loss 2.334120
iteration 100 / 1000: loss 2.162681
iteration 200 / 1000: loss 2.091840
iteration 300 / 1000: loss 2.089505
iteration 400 / 1000: loss 1.941406
iteration 500 / 1000: loss 1.974974
iteration 600 / 1000: loss 1.919954
iteration 700 / 1000: loss 2.042265
iteration 800 / 1000: loss 1.930522
iteration 900 / 1000: loss 1.939204
For parameters:  [ 0.05 50.   50.  ]
Validation accuracy:  0.353
iteration 0 / 1000: loss 2.384247
iteration 100 / 1000: loss 2.216204
iteration 200 / 1000: loss 2.162714
iteration 300 / 1000: loss 1.987644
iteration 400 / 1000: loss 2.003775
iteration 500 / 1000: loss 1.977043
iteration 600 / 1000: loss 1.948022
iteration 700 / 1000: loss 1.917032
iteration 800 / 1000: loss 1.861673
iteration 900 / 1000: loss 1.855129
For parameters:  [5.e-02 8.e+01 1.e+02]
Validation accuracy:  0.358
iteration 0 / 1000: loss 2.456170
iteration 100 / 1000: loss 2.143607
iteration 200 / 1000: loss 2.061057
iteration 300 / 1000: loss 2.011322
iteration 400 / 1000: loss 1.956127
iteration 500 / 1000: loss 1.946235
iteration 600 / 1000: loss 1.872850
iteration 700 / 1000: loss 1.808731
iteration 800 / 1000: loss 1.926650
```

```
iteration 900 / 1000: loss 1.799540
For parameters:  [5.e-02 1.e+02 2.e+02]
Validation accuracy:  0.372
iteration 0 / 1000: loss 2.358424
iteration 100 / 1000: loss 2.158301
iteration 200 / 1000: loss 2.082831
iteration 300 / 1000: loss 1.996793
iteration 400 / 1000: loss 1.883716
iteration 500 / 1000: loss 2.097584
iteration 600 / 1000: loss 1.791184
iteration 700 / 1000: loss 1.875459
iteration 800 / 1000: loss 1.991684
iteration 900 / 1000: loss 1.684003
For parameters:  [5.0e-02 1.2e+02 5.0e+01]
Validation accuracy:  0.364
iteration 0 / 1000: loss 2.426748
iteration 100 / 1000: loss 2.212196
iteration 200 / 1000: loss 2.153461
iteration 300 / 1000: loss 2.140821
iteration 400 / 1000: loss 2.109336
iteration 500 / 1000: loss 2.098734
iteration 600 / 1000: loss 2.096055
iteration 700 / 1000: loss 2.009370
iteration 800 / 1000: loss 1.954278
iteration 900 / 1000: loss 2.041460
For parameters:  [5.e-02 2.e+01 1.e+02]
Validation accuracy:  0.309
iteration 0 / 1000: loss 2.488710
iteration 100 / 1000: loss 2.201617
iteration 200 / 1000: loss 2.121595
iteration 300 / 1000: loss 2.083578
iteration 400 / 1000: loss 2.024399
iteration 500 / 1000: loss 1.957256
iteration 600 / 1000: loss 1.956924
iteration 700 / 1000: loss 1.985621
iteration 800 / 1000: loss 1.922655
iteration 900 / 1000: loss 1.849824
For parameters:  [5.e-02 5.e+01 2.e+02]
Validation accuracy:  0.357
iteration 0 / 1000: loss 2.431763
iteration 100 / 1000: loss 2.192848
iteration 200 / 1000: loss 2.067292
iteration 300 / 1000: loss 2.047826
iteration 400 / 1000: loss 2.025818
iteration 500 / 1000: loss 2.066377
iteration 600 / 1000: loss 1.940036
iteration 700 / 1000: loss 1.756844
iteration 800 / 1000: loss 2.036926
iteration 900 / 1000: loss 2.043059
For parameters:  [5.e-02 8.e+01 5.e+01]
Validation accuracy:  0.369
iteration 0 / 1000: loss 2.426608
iteration 100 / 1000: loss 2.179206
iteration 200 / 1000: loss 2.135026
iteration 300 / 1000: loss 2.053648
iteration 400 / 1000: loss 1.964938
iteration 500 / 1000: loss 1.827273
```

```
iteration 600 / 1000: loss 1.957723
iteration 700 / 1000: loss 1.860583
iteration 800 / 1000: loss 1.865474
iteration 900 / 1000: loss 1.856841
For parameters:  [5.e-02 1.e+02 1.e+02]
Validation accuracy:  0.34
iteration 0 / 1000: loss 2.322078
iteration 100 / 1000: loss 2.165835
iteration 200 / 1000: loss 2.043643
iteration 300 / 1000: loss 2.044443
iteration 400 / 1000: loss 1.978318
iteration 500 / 1000: loss 1.970712
iteration 600 / 1000: loss 1.925278
iteration 700 / 1000: loss 1.862245
iteration 800 / 1000: loss 1.861564
iteration 900 / 1000: loss 1.921277
For parameters:  [5.0e-02 1.2e+02 2.0e+02]
Validation accuracy:  0.37
iteration 0 / 1000: loss 2.410755
iteration 100 / 1000: loss 2.254633
iteration 200 / 1000: loss 2.202615
iteration 300 / 1000: loss 2.233503
iteration 400 / 1000: loss 2.187810
iteration 500 / 1000: loss 2.161803
iteration 600 / 1000: loss 2.165336
iteration 700 / 1000: loss 2.112059
iteration 800 / 1000: loss 2.169692
iteration 900 / 1000: loss 2.045631
For parameters:  [3.e-02 2.e+01 5.e+01]
Validation accuracy:  0.296
iteration 0 / 1000: loss 2.517343
iteration 100 / 1000: loss 2.243674
iteration 200 / 1000: loss 2.225400
iteration 300 / 1000: loss 2.159400
iteration 400 / 1000: loss 2.138322
iteration 500 / 1000: loss 2.140139
iteration 600 / 1000: loss 2.076396
iteration 700 / 1000: loss 2.086987
iteration 800 / 1000: loss 2.005066
iteration 900 / 1000: loss 1.967004
For parameters:  [3.e-02 5.e+01 1.e+02]
Validation accuracy:  0.327
iteration 0 / 1000: loss 2.584331
iteration 100 / 1000: loss 2.214143
iteration 200 / 1000: loss 2.159015
iteration 300 / 1000: loss 2.138768
iteration 400 / 1000: loss 2.062011
iteration 500 / 1000: loss 1.975079
iteration 600 / 1000: loss 1.978128
iteration 700 / 1000: loss 2.004669
iteration 800 / 1000: loss 1.929064
iteration 900 / 1000: loss 1.901959
For parameters:  [3.e-02 8.e+01 2.e+02]
Validation accuracy:  0.324
iteration 0 / 1000: loss 2.355964
iteration 100 / 1000: loss 2.176187
iteration 200 / 1000: loss 2.122950
```

```
iteration 300 / 1000: loss 2.063921
iteration 400 / 1000: loss 1.967118
iteration 500 / 1000: loss 2.067608
iteration 600 / 1000: loss 1.831401
iteration 700 / 1000: loss 1.958748
iteration 800 / 1000: loss 1.961925
iteration 900 / 1000: loss 1.914812
For parameters:  [3.e-02 1.e+02 5.e+01]
Validation accuracy:  0.342
iteration 0 / 1000: loss 2.432296
iteration 100 / 1000: loss 2.212569
iteration 200 / 1000: loss 2.073832
iteration 300 / 1000: loss 2.054449
iteration 400 / 1000: loss 2.021154
iteration 500 / 1000: loss 1.895779
iteration 600 / 1000: loss 2.001518
iteration 700 / 1000: loss 2.004096
iteration 800 / 1000: loss 1.989022
iteration 900 / 1000: loss 1.958868
For parameters:  [3.0e-02 1.2e+02 1.0e+02]
Validation accuracy:  0.347
iteration 0 / 1000: loss 2.410058
iteration 100 / 1000: loss 2.245920
iteration 200 / 1000: loss 2.223053
iteration 300 / 1000: loss 2.203342
iteration 400 / 1000: loss 2.159479
iteration 500 / 1000: loss 2.133978
iteration 600 / 1000: loss 2.130835
iteration 700 / 1000: loss 2.098385
iteration 800 / 1000: loss 2.140647
iteration 900 / 1000: loss 2.082727
For parameters:  [3.e-02 2.e+01 2.e+02]
Validation accuracy:  0.274
iteration 0 / 1000: loss 2.278432
iteration 100 / 1000: loss 2.213406
iteration 200 / 1000: loss 2.208855
iteration 300 / 1000: loss 2.066700
iteration 400 / 1000: loss 2.051404
iteration 500 / 1000: loss 2.077448
iteration 600 / 1000: loss 2.030642
iteration 700 / 1000: loss 1.890243
iteration 800 / 1000: loss 2.013143
iteration 900 / 1000: loss 1.944813
For parameters:  [3.e-02 5.e+01 5.e+01]
Validation accuracy:  0.323
iteration 0 / 1000: loss 2.359923
iteration 100 / 1000: loss 2.203286
iteration 200 / 1000: loss 2.120148
iteration 300 / 1000: loss 2.125394
iteration 400 / 1000: loss 2.053189
iteration 500 / 1000: loss 1.995812
iteration 600 / 1000: loss 2.018539
iteration 700 / 1000: loss 2.050029
iteration 800 / 1000: loss 1.940142
iteration 900 / 1000: loss 1.933720
For parameters:  [3.e-02 8.e+01 1.e+02]
Validation accuracy:  0.334
```

```
iteration 0 / 1000: loss 2.405881
iteration 100 / 1000: loss 2.213258
iteration 200 / 1000: loss 2.137062
iteration 300 / 1000: loss 2.156366
iteration 400 / 1000: loss 2.055444
iteration 500 / 1000: loss 1.980912
iteration 600 / 1000: loss 1.989552
iteration 700 / 1000: loss 2.006236
iteration 800 / 1000: loss 1.967418
iteration 900 / 1000: loss 1.890400
For parameters:  [3.e-02 1.e+02 2.e+02]
Validation accuracy:  0.342
iteration 0 / 1000: loss 2.374897
iteration 100 / 1000: loss 2.138628
iteration 200 / 1000: loss 2.068674
iteration 300 / 1000: loss 2.125135
iteration 400 / 1000: loss 2.040380
iteration 500 / 1000: loss 1.963412
iteration 600 / 1000: loss 1.904206
iteration 700 / 1000: loss 1.920922
iteration 800 / 1000: loss 1.934204
iteration 900 / 1000: loss 1.865893
For parameters:  [3.0e-02 1.2e+02 5.0e+01]
Validation accuracy:  0.355
iteration 0 / 1000: loss 2.496609
iteration 100 / 1000: loss 2.246303
iteration 200 / 1000: loss 2.158703
iteration 300 / 1000: loss 2.165596
iteration 400 / 1000: loss 2.142796
iteration 500 / 1000: loss 2.118159
iteration 600 / 1000: loss 2.097155
iteration 700 / 1000: loss 2.030318
iteration 800 / 1000: loss 2.074157
iteration 900 / 1000: loss 2.019326
For parameters:  [3.e-02 2.e+01 1.e+02]
Validation accuracy:  0.306
iteration 0 / 1000: loss 2.427046
iteration 100 / 1000: loss 2.260295
iteration 200 / 1000: loss 2.188190
iteration 300 / 1000: loss 2.178611
iteration 400 / 1000: loss 2.136770
iteration 500 / 1000: loss 2.090595
iteration 600 / 1000: loss 2.073835
iteration 700 / 1000: loss 2.018430
iteration 800 / 1000: loss 2.010283
iteration 900 / 1000: loss 2.006048
For parameters:  [3.e-02 5.e+01 2.e+02]
Validation accuracy:  0.327
iteration 0 / 1000: loss 2.552201
iteration 100 / 1000: loss 2.187638
iteration 200 / 1000: loss 2.124987
iteration 300 / 1000: loss 2.150177
iteration 400 / 1000: loss 2.011614
iteration 500 / 1000: loss 2.083815
iteration 600 / 1000: loss 1.995669
iteration 700 / 1000: loss 2.041478
iteration 800 / 1000: loss 1.878195
```

```
iteration 900 / 1000: loss 1.901249
For parameters:  [3.e-02 8.e+01 5.e+01]
Validation accuracy:  0.346
iteration 0 / 1000: loss 2.352675
iteration 100 / 1000: loss 2.162777
iteration 200 / 1000: loss 2.103423
iteration 300 / 1000: loss 2.114047
iteration 400 / 1000: loss 2.057731
iteration 500 / 1000: loss 1.995899
iteration 600 / 1000: loss 2.036591
iteration 700 / 1000: loss 1.996298
iteration 800 / 1000: loss 1.986193
iteration 900 / 1000: loss 1.962722
For parameters:  [3.e-02 1.e+02 1.e+02]
Validation accuracy:  0.354
iteration 0 / 1000: loss 2.339715
iteration 100 / 1000: loss 2.198587
iteration 200 / 1000: loss 2.154298
iteration 300 / 1000: loss 2.091658
iteration 400 / 1000: loss 2.037219
iteration 500 / 1000: loss 2.012031
iteration 600 / 1000: loss 1.972614
iteration 700 / 1000: loss 1.979341
iteration 800 / 1000: loss 1.894481
iteration 900 / 1000: loss 1.896944
For parameters:  [3.0e-02 1.2e+02 2.0e+02]
Validation accuracy:  0.334
The best combination of parameters is: learning rate =  0.1 , hidden layer si
ze =  80 , batch size =  200
For this combination, the validation accuracy is  0.414
```

Run on the test set

When you are done experimenting, you should evaluate your final trained networks on the test set.

```
In [10]: print('Two-layer relu')
         test_acc = (best_2layer_relu.predict(X_test) == y_test).mean()
         print('Test accuracy: ', test_acc)
```

```
Two-layer relu
Test accuracy:  0.5038
```

```
In [14]: print('Two-layer sigmoid')
         test_acc = (best_2layer_sigmoid.predict(X_test) == y_test).mean()
         print('Test accuracy: ', test_acc)
```

```
Two-layer sigmoid
Test accuracy:  0.4354
```

```
In [17]: print('Three-layer relu')
         test_acc = (best_3layer_relu.predict(X_test) == y_test).mean()
         print('Test accuracy: ', test_acc)

         Three-layer relu
         Test accuracy:  0.505
```

```
In [19]: print('Three-layer sigmoid')
         test_acc = (best_3layer_sigmoid.predict(X_test) == y_test).mean()
         print('Test accuracy: ', test_acc)

         Three-layer sigmoid
         Test accuracy:  0.3964
```

Kaggle output

Once you are satisfied with your solution and test accuracy output a file to submit your test set predictions to the Kaggle for Assignment 2 Neural Network. Use the following code to do so:

```
In [12]: output_submission_csv('nn_2layer_relu_submission.csv', best_2layer_relu.predic
         t(X_test))
```

```
In [15]: output_submission_csv('nn_2layer_sigmoid_submission.csv', best_2layer_sigmoid.
         predict(X_test))
```

```
In [17]: output_submission_csv('nn_3layer_relu_submission.csv', best_3layer_relu.predic
         t(X_test))
```

```
In [18]: output_submission_csv('nn_3layer_sigmoid_submission.csv', best_3layer_sigmoid.
         predict(X_test))
```