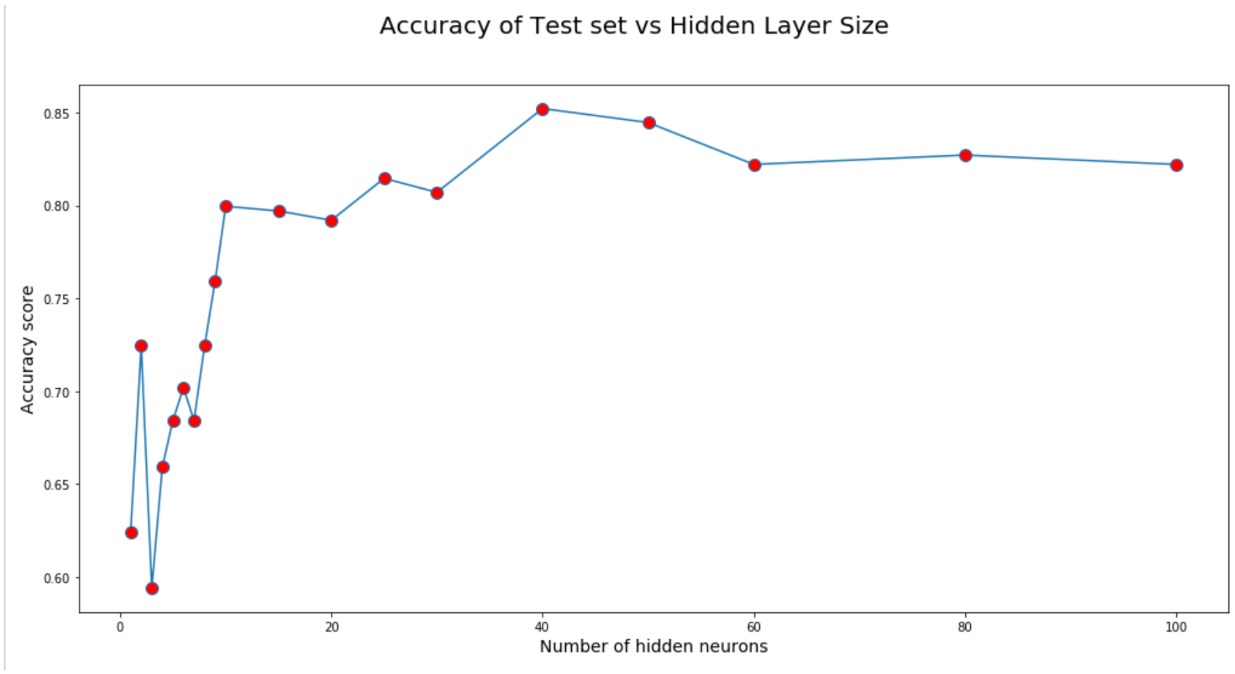
# train1.csv

## 1. How does the number of hidden units impact the results?

With fewer hidden layer neurons, the model performs badly, but as more neurons are added it starts to perform better. However, increasing even more results in the same performance. It is likely that network starts to overfit to the training set and as a result the test accuracy suffers. Increasing the number of neurons in the hidden layer increases accuracy until it starts to overfit.

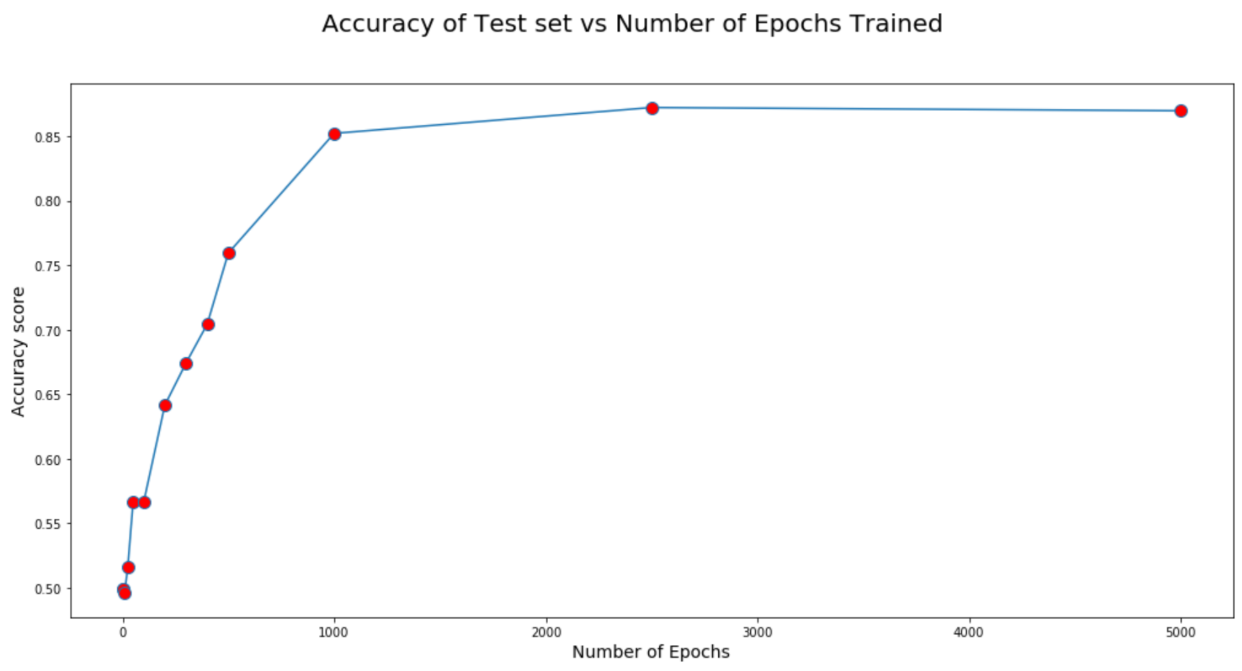


Hidden layer sizes 1,2,3,4,5,6,7,8,9,10,15,20,25,30,40,50,60,80,100

1000 epochs, learning rate of 0.01

## 2. How does the training time impact the results?

As the training time (number of epochs trained) increases, the accuracy in the test set increases as well up until a point. Increasing the number of epochs didn’t affect accuracy and it plateaued beyond that. The amount of compute time goes up at that point with no significant gains in performance. Increasing training time improves results but it will plateau and become more computationally expensive.

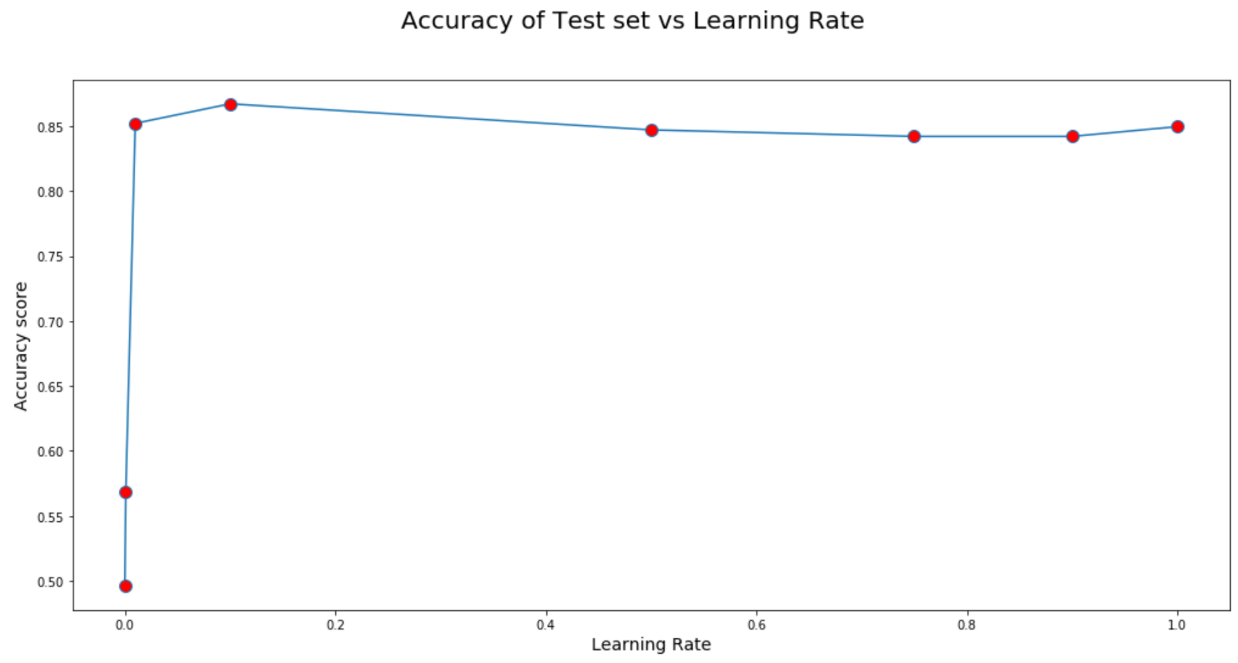


Trained on epoch_sizes of 1, 5, 10, 25, 50, 100, 200, 300, 400, 500, 1000, 2500, 5000

Learning Rate of 0.01, 40 hidden neurons

## 3. How does the learning rate impact the results?

As the learning rate increased, the test accuracy increased as well. I believe this is because a lower learning rate would be more accurate but requires more epochs to train. In this case since we have so many epochs, the lower learning rate has better accuracy and it starts to get worse as it gets higher.

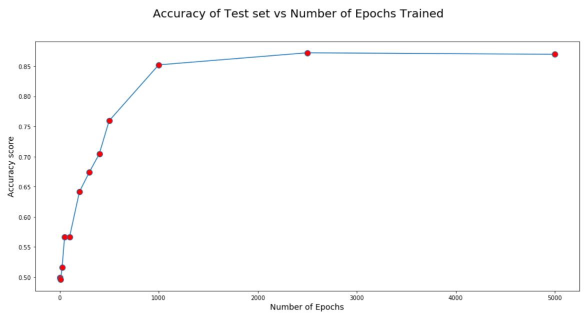


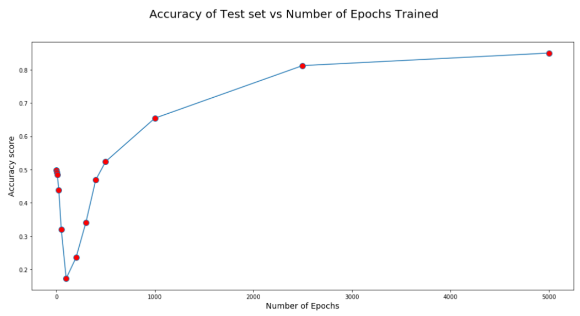
Conducted with 1000 epochs

Learning rate 0.1, Hidden layer neurons 64

## 4. What other critical parameters impacted the results?

I played with the final activation layer function and in both the training time and the learning rate graphs had some interesting changes. On the left is using softmax as the activation function for the output layer and the right is using sigmoid. Sigmoid decreased accuracy from 1 epoch to around 25, this could be because it learns decently but needs more epochs to get really accurate.

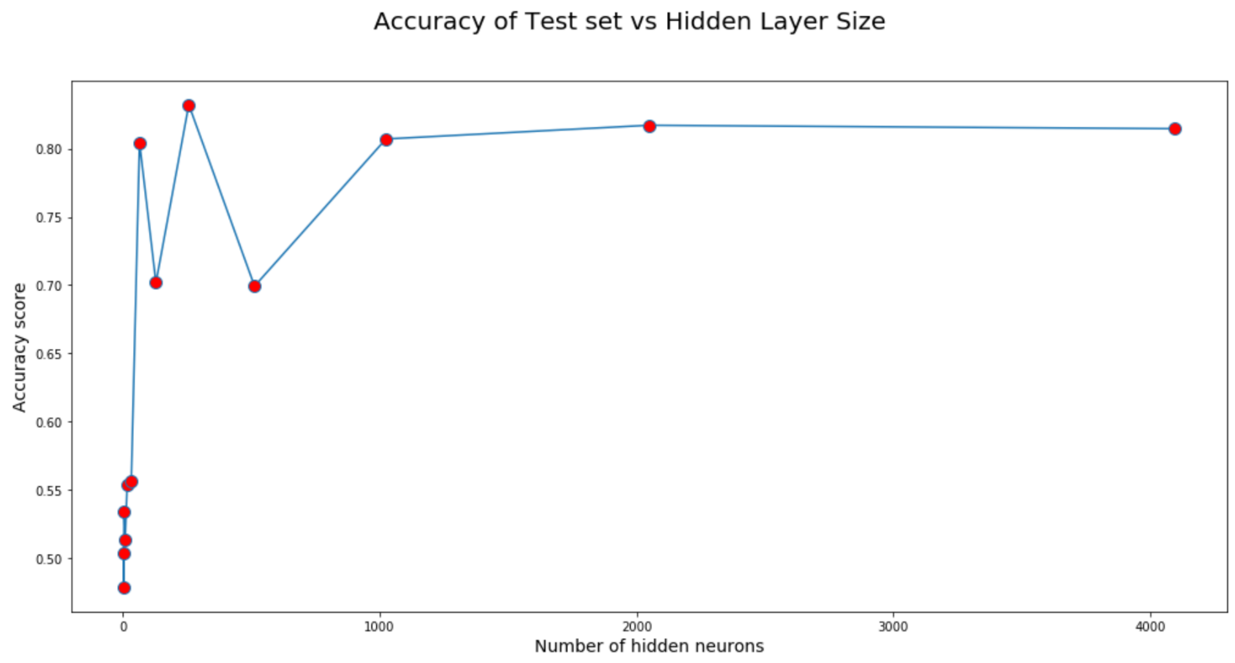




## train2.csv

### 1. How does the number of hidden units impact the results?

With fewer hidden layer neurons, the model performs badly, but as more neurons are added it starts to perform better. We need more hidden neurons than in the first test set. Increasing the number of neurons in the hidden layer increases accuracy until it starts to plateau.

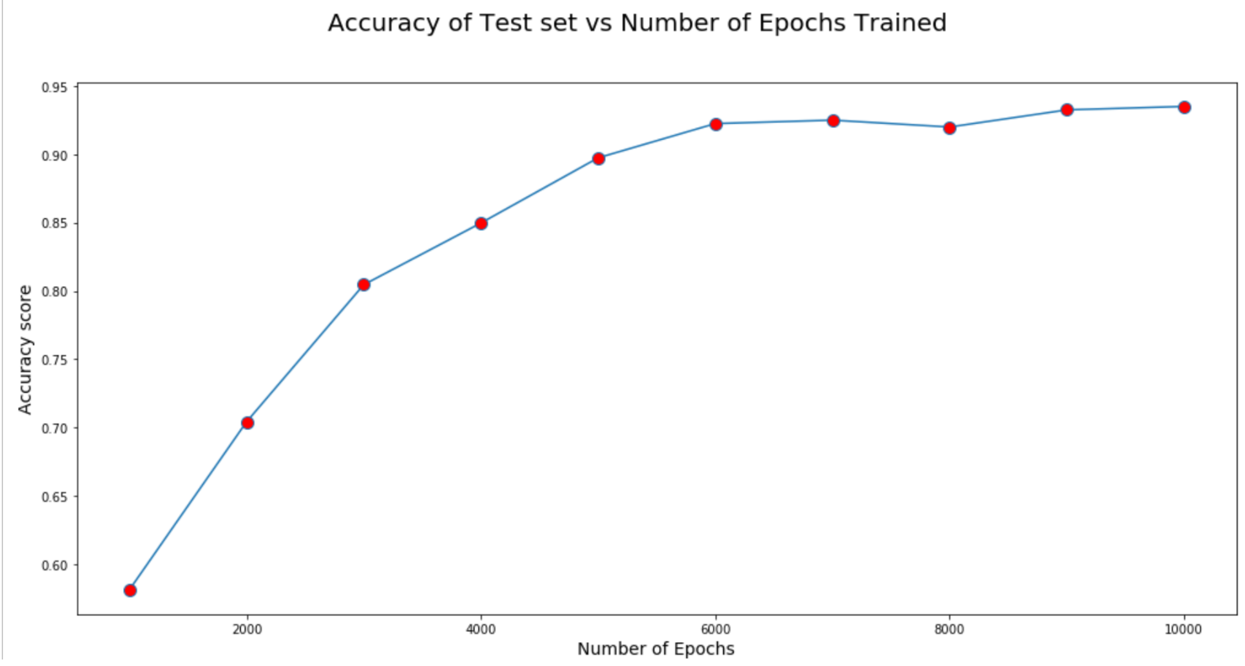


Number of hidden layer neurons 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096

3000 epochs, learning rate of 0.1

### 2. How does the training time impact the results?

As the training time (number of epochs trained) increases, the accuracy in the test set increases as well up until a point. In the experiment I conducted, the graph below shows that at 6000 epochs, increasing the number of epochs didn't affect accuracy and it plateaued beyond that. The amount of time of compute time goes up at that point with no significant gains in performance. Increasing training time improves results but it will plateau and become more computationally expensive.

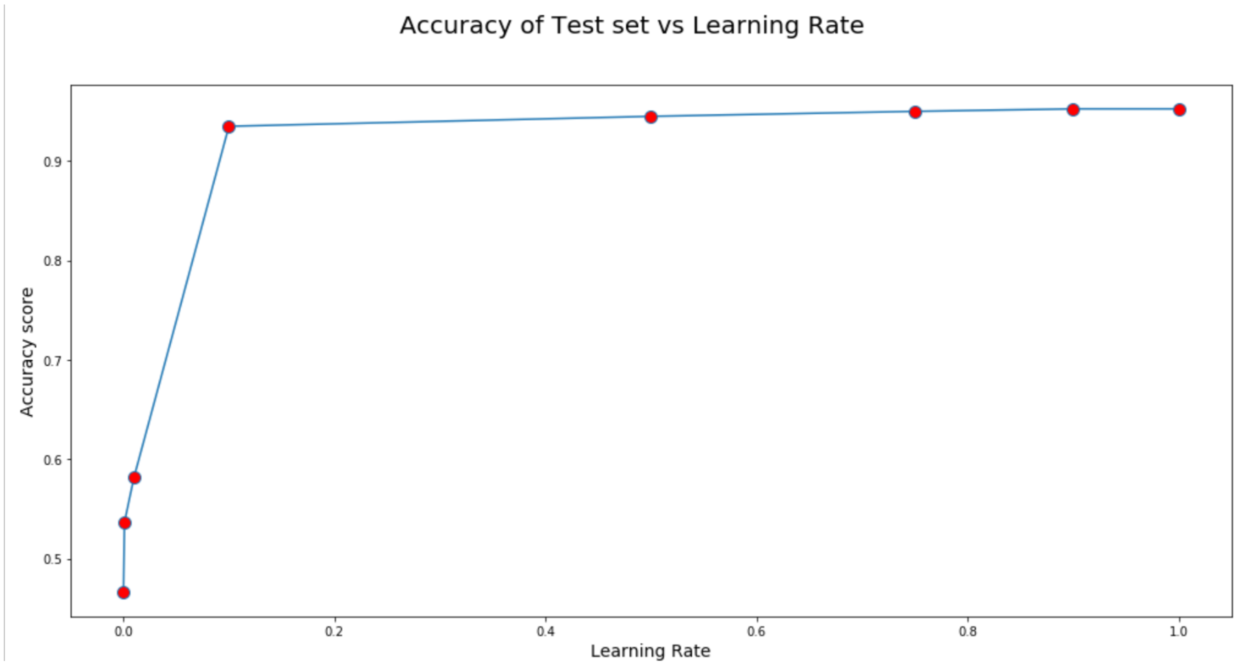


Trained on epoch_sizes of 1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000

Learning Rate of 0.1, 64 hidden neurons

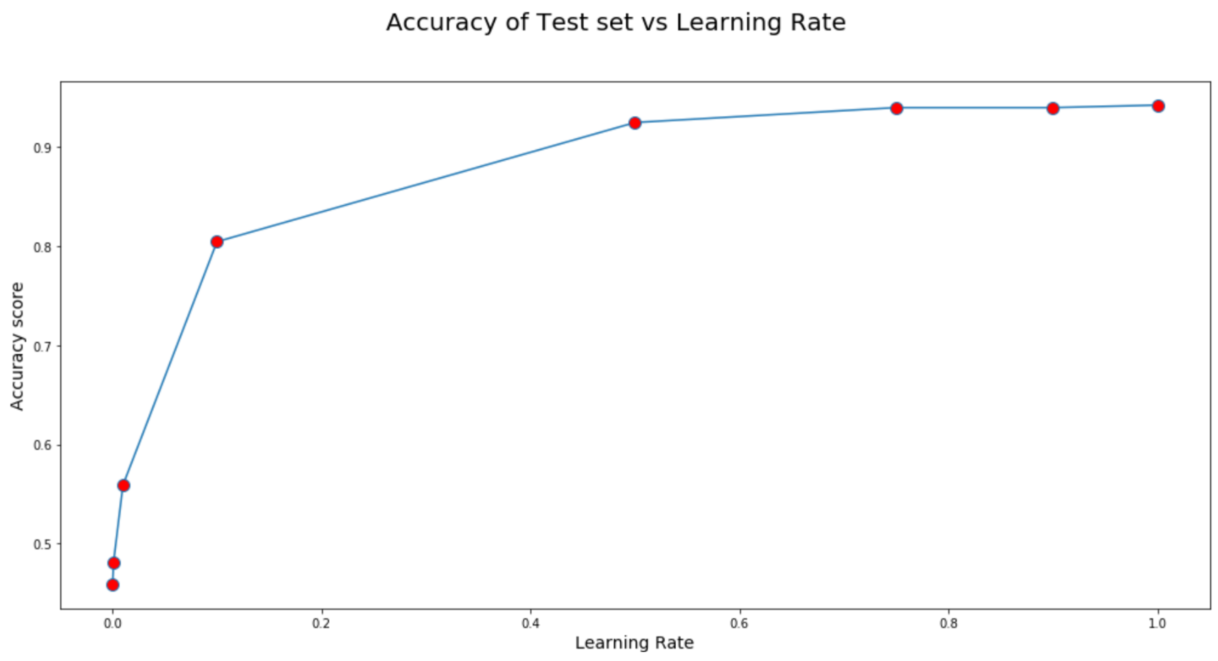
## 3. How does the learning rate impact the results?

As the learning rate increased, the test accuracy increased as well. I conducted the same test with 3000 and 10000 epochs and you can see that the lower learning rate ends up reaching the same amount of accuracy with more time.



Learning rates of 0.0001, 0.001, 0.01, 0.1, 0.5, 0.75, 0.9, 1

10000 epochs, 64 hidden layer neurons

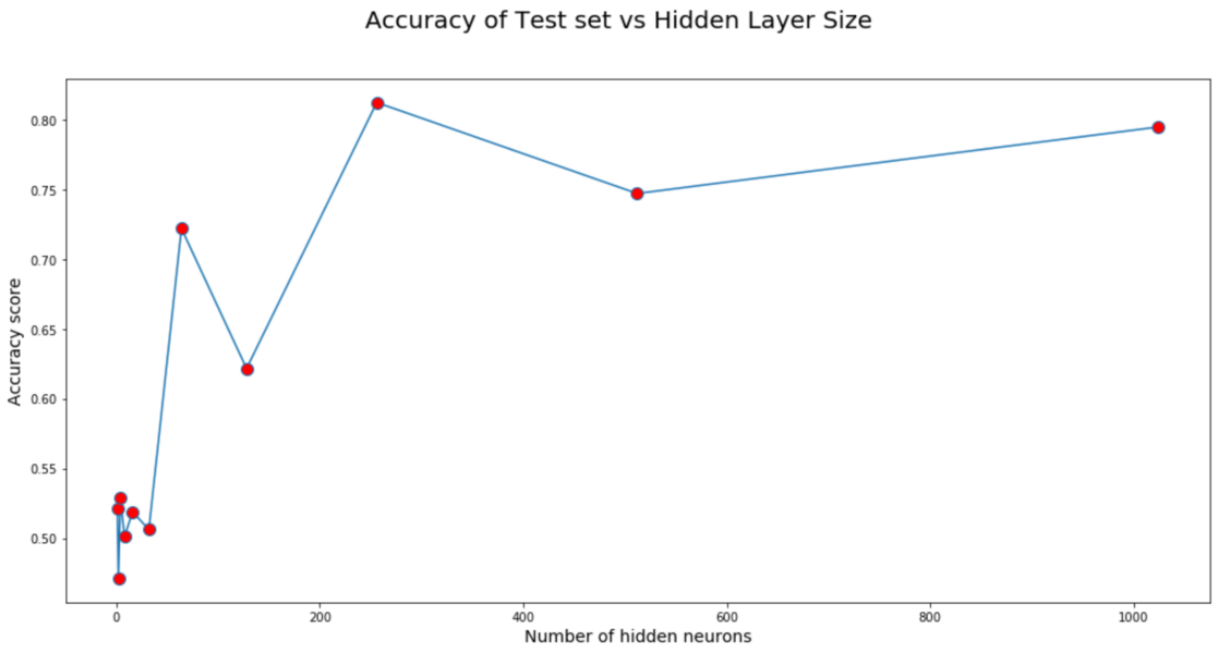


Learning rates of 0.0001, 0.001, 0.01, 0.1, 0.5, 0.75, 0.9, 1

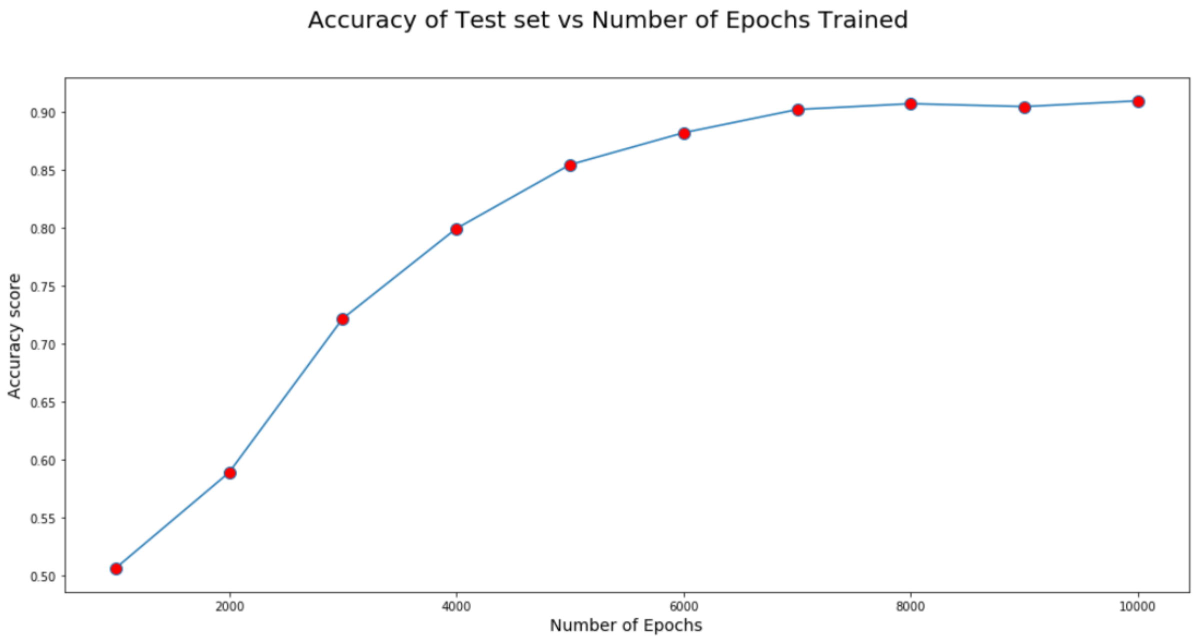
3000 epochs, 64 hidden layer neurons

## 4. What other critical parameters impacted the results?

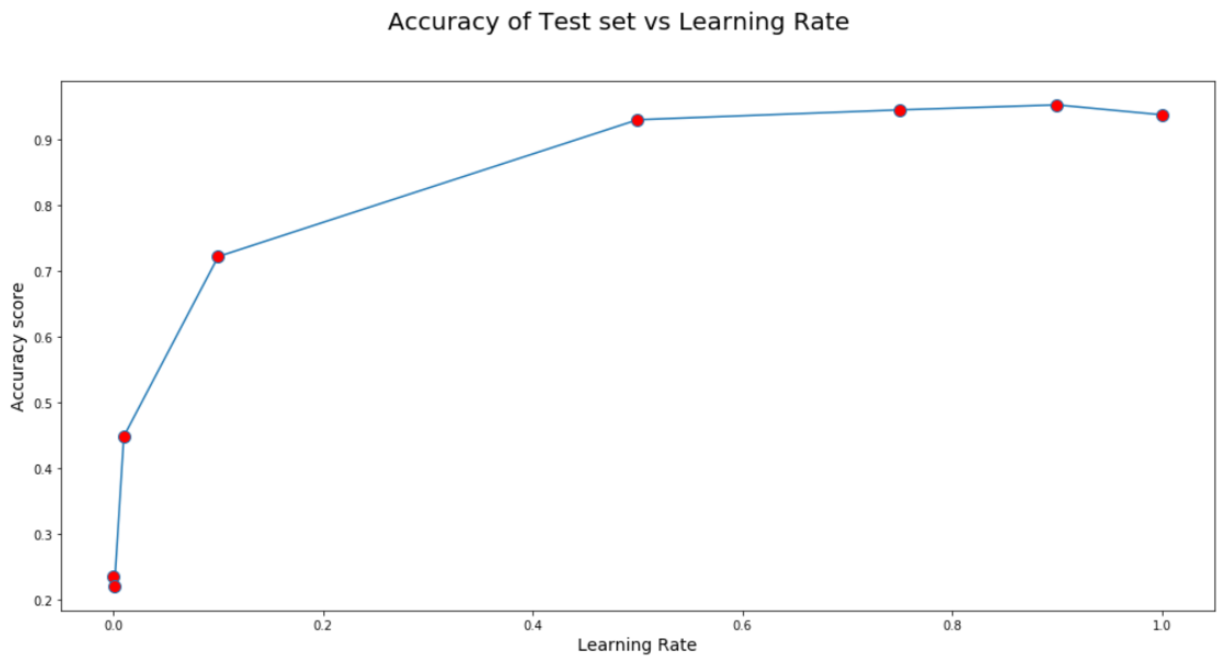
Once again I played with the final layer activations. In all the above graphs I used softmax as a final layer, but the below graphs use sigmoid. With sigmoid as a final layer, it appears to prefer fewer hidden neurons, more epochs, and a higher learning rate, with all other parameters kept the same from the above graphs.



Accuracy of Test set vs Hidden Layer Size
Accuracy score
Number of hidden neurons
0.80
0.75
0.70
0.65
0.60
0.55
0.50
0
200
400
600
800
1000



Accuracy of Test set vs Number of Epochs Trained
Accuracy score
Number of Epochs
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0.50
2000
4000
6000
8000
10000



## 5. What conclusions can you draw from your results? What do you think is causing the difference in performance?

The second set seems to consistently >90% test accuracy with tuned parameters but the first set seems to settle at around 85%. It's possible that the first set being simpler that the data isn't as varied so the test set performs worse but the second set provides more varied data which lets us train better. The first set could contain more outliers which are not close to the rest of the data. There could also be strange distributions where it's unequal. There could also be more features in the first set and therefore we would need more hidden layers to classify more accurately.

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
```

## Helpers

In [2]:
```
def sigmoid(z):
    return 1. / (1. + np.exp(-z))

def sigmoid_deriv(z):
    return z * (1. - z)

def forward_prop(X, w1,b1,w2,b2):
    Z1 = np.matmul(w1,X) + b1
    A1 = sigmoid(Z1) #sigmoid on hidden layer
    Z2 = np.matmul(w2,A1) + b2
    A2 = np.exp(Z2) / np.sum(np.exp(Z2), axis=0) #softmax on output

    return Z1, A1, Z2, A2

def backwards_prop(w1, w2, A1, A2, X, Y):
    m = X.shape[1]
    # this is the derivative of MSE
    dZ2 = A2-Y
    dw2 = (1./m) * np.matmul(dZ2, A1.T)
    db2 = (1./m) * np.sum(dZ2, axis=1, keepdims=True)
    dA1 = np.matmul(w2.T,dZ2)
    dZ1 = dA1 * sigmoid_deriv(A1)
    dw1 = (1./m) * np.matmul(dZ1, X.T)
    db1 = (1./m) * np.sum(dZ1, axis=1, keepdims=True)
    return dw1, dw2, db1, db2

def gradient_descent(w1,w2,b1,b2,dw1,dw2,db1,db2, learning_rate):
    w2 = w2 - learning_rate * dw2
    b2 = b2 - learning_rate * db2
    w1 = w1 - learning_rate * dw1
    b1 = b1 - learning_rate * db1
    return w2, b2, w1, b1
```

## Hyperparameters

In [3]:
```
#set 2
#learning_rate = 0.1
#epochs = 3000
#hidden_layer_size = 64
#epoch_sizes = [1000, 2000, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000]
#hidden_layer_sizes = [1,2,4,8,16,32,64,128,256,512,1024,2048,4096]
#learning_rates = [0.0001, 0.001, 0.01, 0.1, 0.5, 0.75, 0.9, 1]

#set 1
learning_rate = 0.01
epochs = 1000
hidden_layer_size = 40
epoch_sizes = [1, 5, 10, 25,50,100,200,300,400,500,1000,2500, 5000]
hidden_layer_sizes = [1,2,3,4,5,6,7,8,9,10,15,20,25,30,40,50,60,80,100]
learning_rates = [0.0001, 0.001, 0.01, 0.1, 0.5, 0.75, 0.9, 1]
```

## Data Preprocessing

In [4]:
```
train1 = np.loadtxt(open("train1.csv", "rb"), delimiter=",", skiprows=1)
# result is arranged x1, x2, x3, x4, x5, y1, y2
X_train = train1[:,:5].T
print(X_train.shape)
Y_train = train1[:, 5:].T
print(Y_train.shape)

test1 = np.loadtxt(open("test1.csv", "rb"), delimiter=",", skiprows=1)
X_test = test1[:,:5].T
print(X_test.shape)
Y_test = test1[:, 5:].T
print(Y_test.shape)
```

```
(5, 399)
(2, 399)
(5, 399)
(2, 399)
```

## Changing HIDDEN LAYER NEURONS

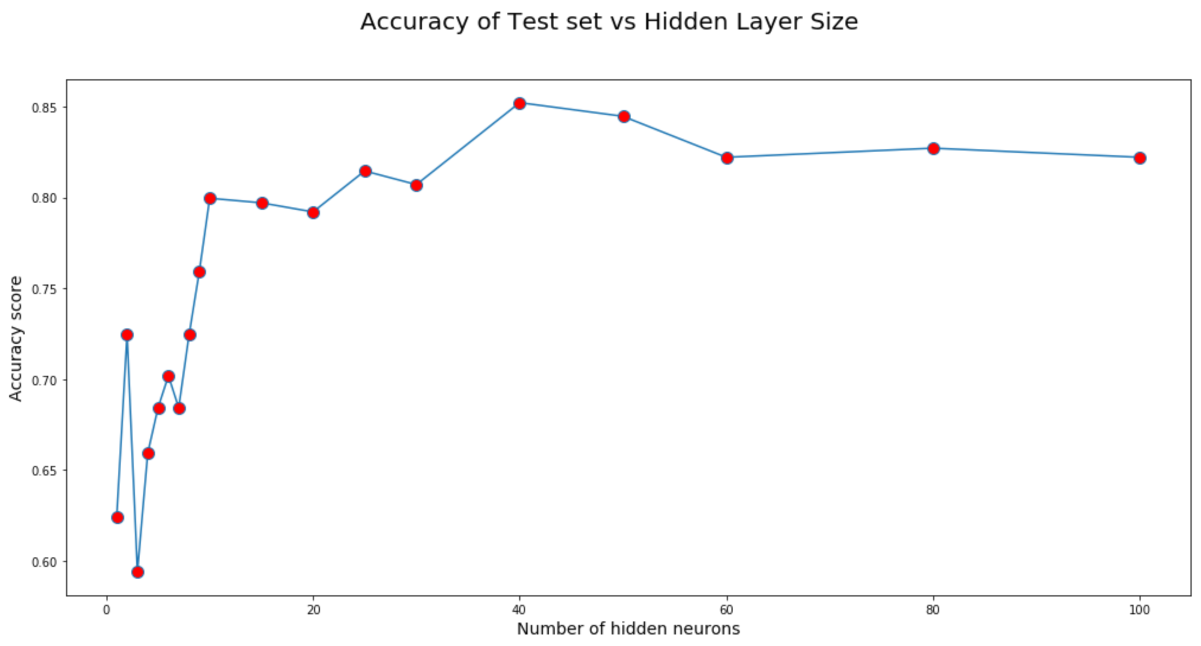
In [5]:
```
scores = []
costs = []
for hidden_layer_num in hidden_layer_sizes:
    print('hidden layer size', hidden_layer_num)
    input_nodes = X_train.shape[0]
    hidden_nodes = hidden_layer_num
    output_nodes = Y_train.shape[0]

    np.random.seed(68)
    w1 = np.random.randn(hidden_nodes,input_nodes)
    b1 = np.zeros((hidden_nodes,1))
    w2 = np.random.randn(output_nodes,hidden_nodes)
    b2 = np.zeros((output_nodes,1))

    for epoch in range(epochs):
        # forward propagation
        Z1, A1, Z2, A2 = forward_prop(X_train,w1,b1,w2,b2)

        # mse loss
        mse_loss = np.mean((Y_train - A2) ** 2)

        # backwards propagation
        dw1, dw2, db1, db2 = backwards_prop(w1, w2, A1, A2, X_train, Y_trair

        # gradient descent
        w2, b2, w1, b1 = gradient_descent(w1,w2,b1,b2,dw1,dw2,db1,db2, learr

        if (epoch % 100 == 0):
            print("Epoch", epoch, "cost: ", mse_loss)

    # calculate training accuracy
    _, _, _, A2_test = forward_prop(X_train,w1,b1,w2,b2)
    predictions_train = np.round(A2_test)
    correct_train = 0
    for j in range(predictions_train.shape[1]): # this is dumb but it works
        if (predictions_train[0][j] == Y_train[0][j] and predictions_train[1
            correct_train = correct_train + 1
    print('Accuracy Train: ', correct_train * 1.0 / predictions_train.shape[

    # calculate test accuracy
    _, _, _, A2_test = forward_prop(X_test,w1,b1,w2,b2)
    predictions_test = np.round(A2_test)
    correct_test = 0
    for j in range(predictions_test.shape[1]):
        if (predictions_test[0][j] == Y_test[0][j] and predictions_test[1][j
            correct_test = correct_test + 1
    score = correct_test * 1.0 / predictions_test.shape[1]
    scores.append(score)
    costs.append(mse_loss)
    print('Accuracy Test ', score)
```

```
hidden layer size 1
Epoch 0 cost:  0.25561552357657974
Epoch 100 cost:  0.25276561775055023
Epoch 200 cost:  0.250553860576344
Epoch 300 cost:  0.24864588651678374
Epoch 400 cost:  0.2468920301941872
Epoch 500 cost:  0.24521540385767948
Epoch 600 cost:  0.24357343251296598
Epoch 700 cost:  0.24194385381806482
Epoch 800 cost:  0.24031825917500663
Epoch 900 cost:  0.23869767608499629
Accuracy Train:  0.656641604010025
Accuracy Test  0.6240601503759399
hidden layer size 2
Epoch 0 cost:  0.2963953341470414
Epoch 100 cost:  0.23654164627732807
Epoch 200 cost:  0.22239221802592424
Epoch 300 cost:  0.21786603941442398
Epoch 400 cost:  0.21490117939141953
Epoch 500 cost:  0.21219230151616866
Epoch 600 cost:  0.2095432544073853
Epoch 700 cost:  0.2069275989087533
Epoch 800 cost:  0.20434214205586723
Epoch 900 cost:  0.20178482160373273
Accuracy Train:  0.7543859649122807
Accuracy Test  0.7243107769423559
hidden layer size 3
Epoch 0 cost:  0.4717032170127107
Epoch 100 cost:  0.3498595264718739
Epoch 200 cost:  0.2874791381155239
Epoch 300 cost:  0.27569483045706855
Epoch 400 cost:  0.26985976485798896
Epoch 500 cost:  0.26468862650511066
Epoch 600 cost:  0.2595394432913096
Epoch 700 cost:  0.2541058196899535
Epoch 800 cost:  0.24803130263641276
Epoch 900 cost:  0.24103631212893342
Accuracy Train:  0.6441102756892231
Accuracy Test  0.5939849624060151
hidden layer size 4
Epoch 0 cost:  0.34233291230416507
Epoch 100 cost:  0.31079660156445
Epoch 200 cost:  0.29432267188321054
Epoch 300 cost:  0.2786066357309047
Epoch 400 cost:  0.2633047854780563
Epoch 500 cost:  0.2491051107158358
Epoch 600 cost:  0.23653979067275255
Epoch 700 cost:  0.2258216098885623
Epoch 800 cost:  0.2168615772925637
Epoch 900 cost:  0.20940467478194152
Accuracy Train:  0.6967418546365914
Accuracy Test  0.6591478696741855
hidden layer size 5
Epoch 0 cost:  0.48304271822932143
Epoch 100 cost:  0.27653993458889825
Epoch 200 cost:  0.2301608150208923
Epoch 300 cost:  0.22135955547083164
Epoch 400 cost:  0.21496343909251736
Epoch 500 cost:  0.20960560054947155
Epoch 600 cost:  0.20497110350957962
Epoch 700 cost:  0.2008428134372524
Epoch 800 cost:  0.19706992654629232
Epoch 900 cost:  0.19355132997845753
Accuracy Train:  0.7343358395989975
Accuracy Test  0.6842105263157895
hidden layer size 6
Epoch 0 cost:  0.4360893817939204
Epoch 100 cost:  0.28817805013353304
Epoch 200 cost:  0.2547664413615944
Epoch 300 cost:  0.23551844526713517
Epoch 400 cost:  0.22447980714970336
Epoch 500 cost:  0.21727554418431555
Epoch 600 cost:  0.21177543439014157
Epoch 700 cost:  0.20709253134146013
Epoch 800 cost:  0.20285165833558083
Epoch 900 cost:  0.19888405996388
Accuracy Train:  0.7117794486215538
Accuracy Test  0.7017543859649122
hidden layer size 7
Epoch 0 cost:  0.3618360180435311
Epoch 100 cost:  0.3388491638282496
Epoch 200 cost:  0.3137983029187168
Epoch 300 cost:  0.28515799355015276
Epoch 400 cost:  0.25956399190343854
Epoch 500 cost:  0.24128597027456913
Epoch 600 cost:  0.22840111047218492
Epoch 700 cost:  0.21786511048761062
Epoch 800 cost:  0.20807968880687158
Epoch 900 cost:  0.1986423285162282
Accuracy Train:  0.7142857142857143
Accuracy Test  0.6842105263157895
hidden layer size 8
Epoch 0 cost:  0.34136936766222126
Epoch 100 cost:  0.23071635664676507
Epoch 200 cost:  0.22423961169608617
Epoch 300 cost:  0.21908549247703815
Epoch 400 cost:  0.2142244601799949
Epoch 500 cost:  0.20963223945732848
Epoch 600 cost:  0.2052906554476002
Epoch 700 cost:  0.20118058174453982
Epoch 800 cost:  0.1972820531891894
Epoch 900 cost:  0.19357474281084983
Accuracy Train:  0.7694235588972431
Accuracy Test  0.7243107769423559
hidden layer size 9
Epoch 0 cost:  0.26486159418464694
Epoch 100 cost:  0.21206690453999097
Epoch 200 cost:  0.2052047901338781
Epoch 300 cost:  0.1990579474361857
Epoch 400 cost:  0.1933892507367164
Epoch 500 cost:  0.18812244470517958
Epoch 600 cost:  0.18319657253124227
Epoch 700 cost:  0.1785647874365616
Epoch 800 cost:  0.17419159507274012
Epoch 900 cost:  0.170050388605227
Accuracy Train:  0.7969924812030075
Accuracy Test  0.7593984962406015
hidden layer size 10
```

```
Epoch 0 cost:  0.355272648975788
Epoch 100 cost:  0.22863515252375705
Epoch 200 cost:  0.2050635296918924
Epoch 300 cost:  0.1896706837657421
Epoch 400 cost:  0.17789913861073214
Epoch 500 cost:  0.16877854691138933
Epoch 600 cost:  0.1616497846288389
Epoch 700 cost:  0.15589414353727918
Epoch 800 cost:  0.15104841181919085
Epoch 900 cost:  0.14681667327905046
Accuracy Train:  0.8170426065162907
Accuracy Test  0.7994987468671679
hidden layer size 15
Epoch 0 cost:  0.4590139096992968
Epoch 100 cost:  0.27453864399043326
Epoch 200 cost:  0.2521759073955279
Epoch 300 cost:  0.23212238325300028
Epoch 400 cost:  0.21353404544037732
Epoch 500 cost:  0.19676125450580648
Epoch 600 cost:  0.1826132525494493
Epoch 700 cost:  0.1714222526810834
Epoch 800 cost:  0.16270605140794855
Epoch 900 cost:  0.15573685223843797
Accuracy Train:  0.8095238095238095
Accuracy Test  0.7969924812030075
hidden layer size 20
Epoch 0 cost:  0.24941779938775718
Epoch 100 cost:  0.22020112309117026
Epoch 200 cost:  0.19976824653671837
Epoch 300 cost:  0.1845201195674096
Epoch 400 cost:  0.1729921314841523
Epoch 500 cost:  0.16405569217562666
Epoch 600 cost:  0.15693076567124334
Epoch 700 cost:  0.15109900188174005
Epoch 800 cost:  0.14621635429613014
Epoch 900 cost:  0.14205035208736752
Accuracy Train:  0.8170426065162907
Accuracy Test  0.7919799498746867
hidden layer size 25
Epoch 0 cost:  0.41649762300166265
Epoch 100 cost:  0.32630143191720135
Epoch 200 cost:  0.26821768271837904
Epoch 300 cost:  0.22528991558162154
Epoch 400 cost:  0.19533344365809302
Epoch 500 cost:  0.17309936284815672
Epoch 600 cost:  0.15646576093573053
Epoch 700 cost:  0.14402769756786732
Epoch 800 cost:  0.1346321023048168
Epoch 900 cost:  0.1274011936887108
Accuracy Train:  0.8471177944862155
Accuracy Test  0.8145363408521303
hidden layer size 30
Epoch 0 cost:  0.28672185805511086
Epoch 100 cost:  0.2284605413112099
Epoch 200 cost:  0.19128132797967853
Epoch 300 cost:  0.16687707384575437
Epoch 400 cost:  0.1502567821792885
Epoch 500 cost:  0.13893847182574218
Epoch 600 cost:  0.1312187014788597
Epoch 700 cost:  0.12583910101486118
Epoch 800 cost:  0.12193652423663566
Epoch 900 cost:  0.11896326822988554
Accuracy Train:  0.8345864661654135
Accuracy Test  0.8070175438596491
hidden layer size 40
Epoch 0 cost:  0.494646843732235
Epoch 100 cost:  0.3306136424840891
Epoch 200 cost:  0.26992970075780653
Epoch 300 cost:  0.22167300488916736
Epoch 400 cost:  0.18628393651429304
Epoch 500 cost:  0.16154685857944162
Epoch 600 cost:  0.14450969320383228
Epoch 700 cost:  0.1326668376657647
Epoch 800 cost:  0.12424321995123708
Epoch 900 cost:  0.11807627113500971
Accuracy Train:  0.8571428571428571
Accuracy Test  0.8521303258145363
hidden layer size 50
Epoch 0 cost:  0.5146900645120666
Epoch 100 cost:  0.3461363245913695
Epoch 200 cost:  0.26620745198341966
Epoch 300 cost:  0.20664277752400262
Epoch 400 cost:  0.1706143613366958
Epoch 500 cost:  0.14932366138195788
Epoch 600 cost:  0.13585094697076294
Epoch 700 cost:  0.12667333376167425
Epoch 800 cost:  0.12005427371936674
Epoch 900 cost:  0.11507940592773207
Accuracy Train:  0.8546365914786967
Accuracy Test  0.8446115288220551
hidden layer size 60
Epoch 0 cost:  0.4753304758408258
Epoch 100 cost:  0.3143553725664616
Epoch 200 cost:  0.22550646082398257
Epoch 300 cost:  0.16990873679148807
Epoch 400 cost:  0.1411145277708284
Epoch 500 cost:  0.1262800445382529
Epoch 600 cost:  0.11826376806778062
Epoch 700 cost:  0.11350448136356134
Epoch 800 cost:  0.11031408260094745
Epoch 900 cost:  0.10792931273777277
Accuracy Train:  0.8571428571428571
Accuracy Test  0.8220551378446115
hidden layer size 80
Epoch 0 cost:  0.45601961180843137
Epoch 100 cost:  0.24647387156952313
Epoch 200 cost:  0.15784639198036368
Epoch 300 cost:  0.1319451496822779
Epoch 400 cost:  0.12240653525481845
Epoch 500 cost:  0.11725100557434988
Epoch 600 cost:  0.11372539000024119
Epoch 700 cost:  0.1110081457634849
Epoch 800 cost:  0.10877736583287598
Epoch 900 cost:  0.10687500384937428
Accuracy Train:  0.8671679197994987
Accuracy Test  0.8270676691729323
hidden layer size 100
Epoch 0 cost:  0.49861374032383143
Epoch 100 cost:  0.1479068936180449
Epoch 200 cost:  0.12592328884469242
Epoch 300 cost:  0.11771278844128401
Epoch 400 cost:  0.11382968339742296
Epoch 500 cost:  0.11142769748648666
Epoch 600 cost:  0.10960089780738515
Epoch 700 cost:  0.10803993175333435
Epoch 800 cost:  0.10663106084655143
Epoch 900 cost:  0.1053278981875611
Accuracy Train:  0.8671679197994987
Accuracy Test  0.8220551378446115
```

In [6]:
```
fig = plt.figure()
fig.suptitle('Accuracy of Test set vs Hidden Layer Size', fontsize = 20)
fig.set_figwidth(17)
fig.set_figheight(8)
ax = fig.add_subplot(111)
ax.plot(hidden_layer_sizes, scores, '-o', markersize = 10, markerfacecolor =
ax.set_xlabel('Number of hidden neurons', fontsize = 14)
ax.set_ylabel('Accuracy score', fontsize = 14)
```

Out[6]:
```
Text(0, 0.5, 'Accuracy score')
```



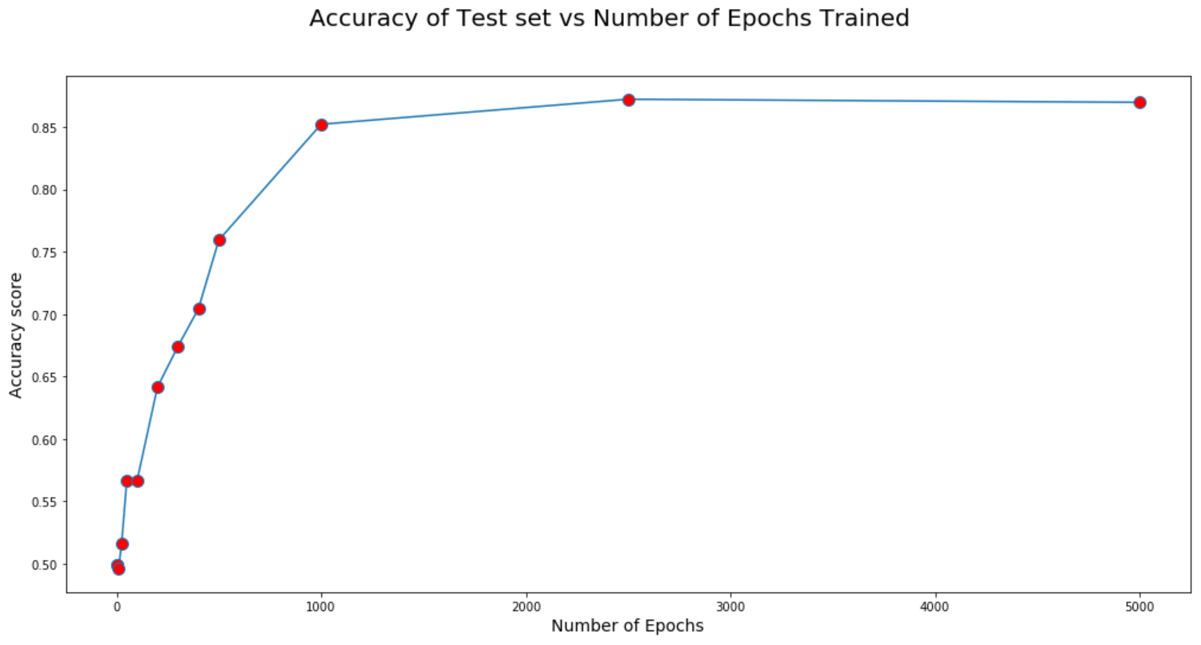
## Changing EPOCHS

```
In [7]: scores = []
        costs = []
        for num_epochs in epoch_sizes:
            print('epoch size', num_epochs)
            input_nodes = X_train.shape[0]
            hidden_nodes = hidden_layer_size
            output_nodes = Y_train.shape[0]

            np.random.seed(68)
            w1 = np.random.randn(hidden_nodes,input_nodes)
            b1 = np.zeros((hidden_nodes,1))
            w2 = np.random.randn(output_nodes,hidden_nodes)
            b2 = np.zeros((output_nodes,1))

            for epoch in range(num_epochs):
                # forward propagation
                Z1, A1, Z2, A2 = forward_prop(X_train,w1,b1,w2,b2)

                # mse loss
                mse_loss = np.mean((Y_train - A2) ** 2)

                # backwards propagation
                dw1, dw2, db1, db2 = backwards_prop(w1, w2, A1, A2, X_train, Y_trair

                # gradient descent
                w2, b2, w1, b1 = gradient_descent(w1,w2,b1,b2,dw1,dw2,db1,db2, learr

                if (epoch % 100 == 0):
                    print("Epoch", epoch, "cost: ", mse_loss)

            # calculate training accuracy
            _, _, _, A2_test = forward_prop(X_train,w1,b1,w2,b2)
            predictions_train = np.round(A2_test)
            correct_train = 0
            for j in range(predictions_train.shape[1]): # this is dumb but it works
                if (predictions_train[0][j] == Y_train[0][j] and predictions_train[1
                    correct_train = correct_train + 1
            print('Accuracy Train: ', correct_train * 1.0 / predictions_train.shape[

            # calculate test accuracy
            _, _, _, A2_test = forward_prop(X_test,w1,b1,w2,b2)
            predictions_test = np.round(A2_test)
            correct_test = 0
            for j in range(predictions_test.shape[1]):
                if (predictions_test[0][j] == Y_test[0][j] and predictions_test[1][j
                    correct_test = correct_test + 1
            score = correct_test * 1.0 / predictions_test.shape[1]
            scores.append(score)
            costs.append(mse_loss)
            print('Accuracy Test ', score)
```

```
epoch size 1
Epoch 0 cost:  0.494646843732235
Accuracy Train:  0.49874686716791977
Accuracy Test  0.49874686716791977
epoch size 5
Epoch 0 cost:  0.494646843732235
Accuracy Train:  0.49874686716791977
Accuracy Test  0.49874686716791977
epoch size 10
Epoch 0 cost:  0.494646843732235
Accuracy Train:  0.49874686716791977
Accuracy Test  0.49624060150375937
epoch size 25
Epoch 0 cost:  0.494646843732235
Accuracy Train:  0.5087719298245614
Accuracy Test  0.5162907268170426
epoch size 50
Epoch 0 cost:  0.494646843732235
Accuracy Train:  0.5238095238095238
Accuracy Test  0.5664160401002506
epoch size 100
Epoch 0 cost:  0.494646843732235
Accuracy Train:  0.5388471177944862
Accuracy Test  0.5664160401002506
epoch size 200
Epoch 0 cost:  0.494646843732235
Epoch 100 cost:  0.3306136424840891
Accuracy Train:  0.5989974937343359
Accuracy Test  0.6416040100250626
epoch size 300
Epoch 0 cost:  0.494646843732235
Epoch 100 cost:  0.3306136424840891
Epoch 200 cost:  0.26992970075780653
Accuracy Train:  0.6541353383458647
Accuracy Test  0.6741854636591479
epoch size 400
Epoch 0 cost:  0.494646843732235
Epoch 100 cost:  0.3306136424840891
Epoch 200 cost:  0.26992970075780653
Epoch 300 cost:  0.22167300488916736
Accuracy Train:  0.7243107769423559
Accuracy Test  0.7042606516290727
epoch size 500
Epoch 0 cost:  0.494646843732235
Epoch 100 cost:  0.3306136424840891
Epoch 200 cost:  0.26992970075780653
Epoch 300 cost:  0.22167300488916736
Epoch 400 cost:  0.18628393651429304
Accuracy Train:  0.7994987468671679
Accuracy Test  0.7593984962406015
epoch size 1000
Epoch 0 cost:  0.494646843732235
Epoch 100 cost:  0.3306136424840891
Epoch 200 cost:  0.26992970075780653
Epoch 300 cost:  0.22167300488916736
Epoch 400 cost:  0.18628393651429304
Epoch 500 cost:  0.16154685857944162
Epoch 600 cost:  0.14450969320383228
Epoch 700 cost:  0.1326668376657647
Epoch 800 cost:  0.12424321995123708
Epoch 900 cost:  0.11807627113500971
Accuracy Train:  0.8571428571428571
Accuracy Test  0.8521303258145363
epoch size 2500
Epoch 0 cost:  0.494646843732235
Epoch 100 cost:  0.3306136424840891
Epoch 200 cost:  0.26992970075780653
Epoch 300 cost:  0.22167300488916736
Epoch 400 cost:  0.18628393651429304
Epoch 500 cost:  0.16154685857944162
Epoch 600 cost:  0.14450969320383228
Epoch 700 cost:  0.1326668376657647
Epoch 800 cost:  0.12424321995123708
Epoch 900 cost:  0.11807627113500971
Epoch 1000 cost:  0.11342485793003629
Epoch 1100 cost:  0.10981628285915747
Epoch 1200 cost:  0.10694457972424809
Epoch 1300 cost:  0.1046073781598438
Epoch 1400 cost:  0.10266753343921746
Epoch 1500 cost:  0.10102978103042817
Epoch 1600 cost:  0.09962635555033823
Epoch 1700 cost:  0.09840797015724154
Epoch 1800 cost:  0.0973380396993897
Epoch 1900 cost:  0.09638890027645097
Epoch 2000 cost:  0.09553928157856673
Epoch 2100 cost:  0.09477258150332703
Epoch 2200 cost:  0.09407566510474828
Epoch 2300 cost:  0.0934380130671769
Epoch 2400 cost:  0.0928511076216159
Accuracy Train:  0.8897243107769424
Accuracy Test  0.8721804511278195
epoch size 5000
Epoch 0 cost:  0.494646843732235
Epoch 100 cost:  0.3306136424840891
Epoch 200 cost:  0.26992970075780653
Epoch 300 cost:  0.22167300488916736
Epoch 400 cost:  0.18628393651429304
Epoch 500 cost:  0.16154685857944162
Epoch 600 cost:  0.14450969320383228
Epoch 700 cost:  0.1326668376657647
Epoch 800 cost:  0.12424321995123708
Epoch 900 cost:  0.11807627113500971
Epoch 1000 cost:  0.11342485793003629
Epoch 1100 cost:  0.10981628285915747
Epoch 1200 cost:  0.10694457972424809
Epoch 1300 cost:  0.1046073781598438
Epoch 1400 cost:  0.10266753343921746
Epoch 1500 cost:  0.10102978103042817
Epoch 1600 cost:  0.09962635555033823
Epoch 1700 cost:  0.09840797015724154
Epoch 1800 cost:  0.0973380396993897
Epoch 1900 cost:  0.09638890027645097
Epoch 2000 cost:  0.09553928157856673
Epoch 2100 cost:  0.09477258150332703
Epoch 2200 cost:  0.09407566510474828
Epoch 2300 cost:  0.0934380130671769
Epoch 2400 cost:  0.0928511076216159
Epoch 2500 cost:  0.09230798266875737
Epoch 2600 cost:  0.09180288937757745
Epoch 2700 cost:  0.09133104426771294
Epoch 2800 cost:  0.09088843707168211
Epoch 2900 cost:  0.09047168251107933
Epoch 3000 cost:  0.09007790473883902
Epoch 3100 cost:  0.08970464636556179
Epoch 3200 cost:  0.08934979618927007
Epoch 3300 cost:  0.08901153129918009
Epoch 3400 cost:  0.08868827033091496
Epoch 3500 cost:  0.0883786354496742
Epoch 3600 cost:  0.0880814212211763
Epoch 3700 cost:  0.08779556896041488
Epoch 3800 cost:  0.08752014546868835
Epoch 3900 cost:  0.0872543253102034
Epoch 4000 cost:  0.08699737596214516
Epoch 4100 cost:  0.08674864531167141
Epoch 4200 cost:  0.08650755108078978
Epoch 4300 cost:  0.08627357184348262
Epoch 4400 cost:  0.08604623936460894
Epoch 4500 cost:  0.08582513204135617
Epoch 4600 cost:  0.08560986926856856
Epoch 4700 cost:  0.08540010658156096
Epoch 4800 cost:  0.08519553145587731
Epoch 4900 cost:  0.0849958596642631
Accuracy Train:  0.8947368421052632
Accuracy Test  0.8696741854636592
```

In [8]:
```python
fig = plt.figure()
fig.suptitle('Accuracy of Test set vs Number of Epochs Trained', fontsize =
fig.set_figwidth(17)
fig.set_figheight(8)
ax = fig.add_subplot(111)
ax.plot(epoch_sizes, scores, '-o', markersize = 10, markerfacecolor = 'r')
ax.set_xlabel('Number of Epochs', fontsize = 14)
ax.set_ylabel('Accuracy score', fontsize = 14)
```

Out[8]:
```
Text(0, 0.5, 'Accuracy score')
```



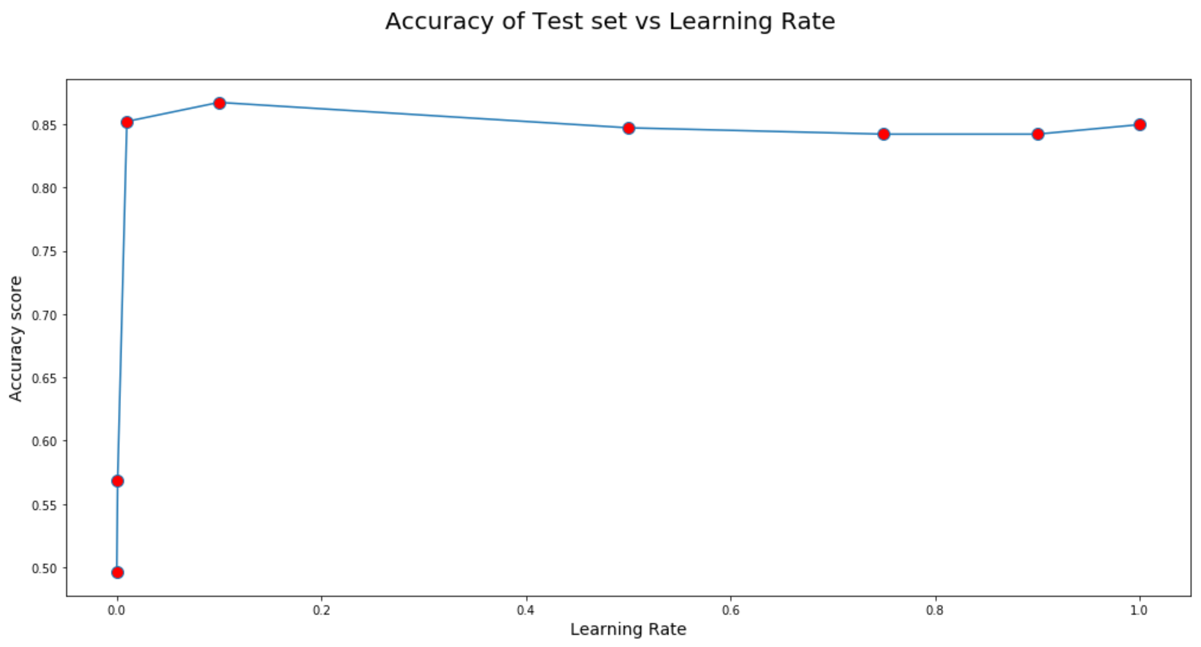
## Changing LEARNING RATES

In [9]:
```
scores = []
costs = []
for learning_rate_it in learning_rates:
    print('learning rate',learning_rate_it)
    input_nodes = X_train.shape[0]
    hidden_nodes = hidden_layer_size
    output_nodes = Y_train.shape[0]

    np.random.seed(68)
    w1 = np.random.randn(hidden_nodes,input_nodes)
    b1 = np.zeros((hidden_nodes,1))
    w2 = np.random.randn(output_nodes,hidden_nodes)
    b2 = np.zeros((output_nodes,1))

    for epoch in range(epochs):
        # forward propagation
        Z1, A1, Z2, A2 = forward_prop(X_train,w1,b1,w2,b2)

        # mse loss
        mse_loss = np.mean((Y_train - A2) ** 2)

        # backwards propagation
        dw1, dw2, db1, db2 = backwards_prop(w1, w2, A1, A2, X_train, Y_trair

        # gradient descent
        w2, b2, w1, b1 = gradient_descent(w1,w2,b1,b2,dw1,dw2,db1,db2, learr

        if (epoch % 100 == 0):
            print("Epoch", epoch, "cost: ", mse_loss)

    # calculate training accuracy
    _, _, _, A2_test = forward_prop(X_train,w1,b1,w2,b2)
    predictions_train = np.round(A2_test)
    correct_train = 0
    for j in range(predictions_train.shape[1]): # this is dumb but it works
        if (predictions_train[0][j] == Y_train[0][j] and predictions_train[1
            correct_train = correct_train + 1
    print('Accuracy Train: ', correct_train * 1.0 / predictions_train.shape[

    # calculate test accuracy
    _, _, _, A2_test = forward_prop(X_test,w1,b1,w2,b2)
    predictions_test = np.round(A2_test)
    correct_test = 0
    for j in range(predictions_test.shape[1]):
        if (predictions_test[0][j] == Y_test[0][j] and predictions_test[1][j
            correct_test = correct_test + 1
    score = correct_test * 1.0 / predictions_test.shape[1]
    scores.append(score)
    costs.append(mse_loss)
    print('Accuracy Test ', score)
```

```
learning rate 0.0001
Epoch 0 cost:  0.494646843732235
Epoch 100 cost:  0.49357466284575613
Epoch 200 cost:  0.4923439462326721
Epoch 300 cost:  0.4909370300921532
Epoch 400 cost:  0.48933622668002075
Epoch 500 cost:  0.48752450285372284
Epoch 600 cost:  0.4854863031273879
Epoch 700 cost:  0.4832084788138402
Epoch 800 cost:  0.48068125550516166
Epoch 900 cost:  0.47789914529038013
Accuracy Train:  0.49874686716791977
Accuracy Test  0.49624060150375937
learning rate 0.001
Epoch 0 cost:  0.494646843732235
Epoch 100 cost:  0.47485450479945457
Epoch 200 cost:  0.4339709207513641
Epoch 300 cost:  0.3940423196459815
Epoch 400 cost:  0.3718654369786363
Epoch 500 cost:  0.3616267968259137
Epoch 600 cost:  0.3552366705630814
Epoch 700 cost:  0.3493981093360265
Epoch 800 cost:  0.3433358441394869
Epoch 900 cost:  0.33702979723084264
Accuracy Train:  0.5388471177944862
Accuracy Test  0.568922305764411
learning rate 0.01
Epoch 0 cost:  0.494646843732235
Epoch 100 cost:  0.3306136424840891
Epoch 200 cost:  0.26992970075780653
Epoch 300 cost:  0.22167300488916736
Epoch 400 cost:  0.18628393651429304
Epoch 500 cost:  0.16154685857944162
Epoch 600 cost:  0.14450969320383228
Epoch 700 cost:  0.1326668376657647
Epoch 800 cost:  0.12424321995123708
Epoch 900 cost:  0.11807627113500971
Accuracy Train:  0.8571428571428571
Accuracy Test  0.8521303258145363
learning rate 0.1
Epoch 0 cost:  0.494646843732235
Epoch 100 cost:  0.11329925521671372
Epoch 200 cost:  0.09553951762636517
Epoch 300 cost:  0.09009591766916993
Epoch 400 cost:  0.08701982424809669
Epoch 500 cost:  0.0848244174930287
Epoch 600 cost:  0.08308588336887876
Epoch 700 cost:  0.0816387696145691
Epoch 800 cost:  0.080402145250826
Epoch 900 cost:  0.07932855502213247
Accuracy Train:  0.9072681704260651
Accuracy Test  0.8671679197994987
learning rate 0.5
Epoch 0 cost:  0.494646843732235
Epoch 100 cost:  0.08679463749881278
Epoch 200 cost:  0.0798407772926563
Epoch 300 cost:  0.07605737724700898
Epoch 400 cost:  0.0734824587979813
Epoch 500 cost:  0.0712887662681112
Epoch 600 cost:  0.07790449994634825
Epoch 700 cost:  0.0692240600814496
Epoch 800 cost:  0.06587161048918438
Epoch 900 cost:  0.06307190485957637
Accuracy Train:  0.9323308270676691
Accuracy Test  0.8471177944862155
learning rate 0.75
Epoch 0 cost:  0.494646843732235
Epoch 100 cost:  0.10746658287795516
Epoch 200 cost:  0.09246516955965538
Epoch 300 cost:  0.08634494312397246
Epoch 400 cost:  0.08177971684263098
Epoch 500 cost:  0.07705116247655752
Epoch 600 cost:  0.07256810743842615
Epoch 700 cost:  0.06862559824466898
Epoch 800 cost:  0.06524151533756911
Epoch 900 cost:  0.060960782140692796
Accuracy Train:  0.9273182957393483
Accuracy Test  0.8421052631578947
learning rate 0.9
Epoch 0 cost:  0.494646843732235
Epoch 100 cost:  0.10213746407023207
Epoch 200 cost:  0.09232545621049248
Epoch 300 cost:  0.0852600226922824
Epoch 400 cost:  0.079396553017153
Epoch 500 cost:  0.07478222485446999
Epoch 600 cost:  0.07095822583138625
Epoch 700 cost:  0.06728440206187511
Epoch 800 cost:  0.0637601780416217
Epoch 900 cost:  0.06042082049237882
Accuracy Train:  0.9273182957393483
Accuracy Test  0.8421052631578947
learning rate 1
Epoch 0 cost:  0.494646843732235
Epoch 100 cost:  0.10361663965998252
Epoch 200 cost:  0.09085447152478784
Epoch 300 cost:  0.08340311487679555
Epoch 400 cost:  0.07789689785422257
Epoch 500 cost:  0.07309632324329657
Epoch 600 cost:  0.06904826448743027
Epoch 700 cost:  0.06558138314135752
Epoch 800 cost:  0.062326183311260934
Epoch 900 cost:  0.05908187405982046
Accuracy Train:  0.9223057644110275
Accuracy Test  0.849624060150376
```

In [10]:
```python
fig = plt.figure()
fig.suptitle('Accuracy of Test set vs Learning Rate', fontsize = 20)
fig.set_figwidth(17)
fig.set_figheight(8)
ax = fig.add_subplot(111)
ax.plot(learning_rates, scores, '-o', markersize = 10, markerfacecolor = 'r'
ax.set_xlabel('Learning Rate', fontsize = 14)
ax.set_ylabel('Accuracy score', fontsize = 14)
```

Out[10]:
```
Text(0, 0.5, 'Accuracy score')
```


Accuracy of Test set vs Learning Rate
Accuracy score
Learning Rate
0.50
0.55
0.60
0.65
0.70
0.75
0.80
0.85
0.0
0.2
0.4
0.6
0.8
1.0

```
In [ ]:
```