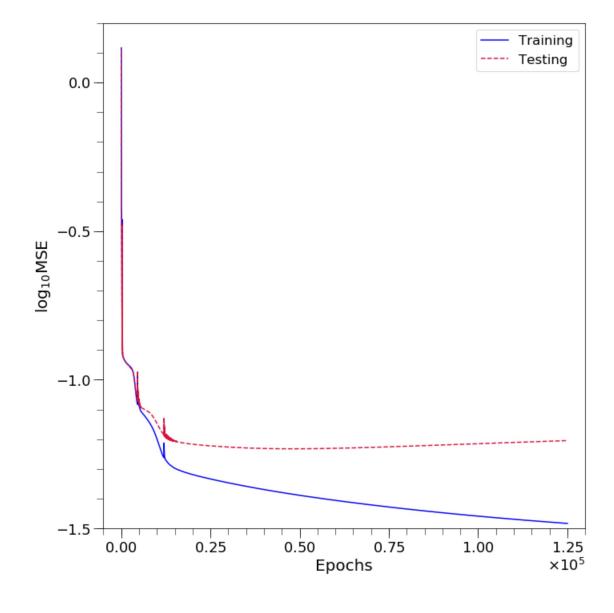
```
In [10]: # -*- coding: utf-8 -*-
         11 11 11
         Created on Tue Nov 12 18:49:03 2019
         @author: jorge
         import numpy as np
         from requiredFunctions.trainMLP import trainMLP
         from requiredFunctions.MLP import MLP
         from requiredFunctions.doubleMoon import doubleMoon
         import matplotlib as mpl
         import matplotlib.pyplot as plt
         import matplotlib.ticker as mtick
         height = 10
         width = 10
         mpl.rcParams['figure.figsize'] = (width, height)
         mpl.rcParams['font.size'] = 20
         mpl.rcParams['figure.titlesize'] = 'small'
         mpl.rcParams['legend.fontsize'] = 'small'
         mpl.rcParams['xtick.major.size'] = 12
         mpl.rcParams['xtick.minor.size'] = 8
         mpl.rcParams['xtick.labelsize'] = 18
         mpl.rcParams['ytick.major.size'] = 12
         mpl.rcParams['ytick.minor.size'] = 8
         mpl.rcParams['ytick.labelsize'] = 18
         # Data parameters
         N train = 300
         N \, val = 3000
         r = 1
         w = 0.6
         d = -0.5
         # Model parameters
         learning rate = 1.25e-3
         alpha = 0
         epochs = 1000*125
         # Init train data
         train data = doubleMoon(N train, w, r, d, seed=0)
         x_train, y_train = train_data[:,:2], train_data[:,2]
         x train, y train = x train.T, y train.reshape(1, len(y train))
         # Init validation data
         val_data = doubleMoon(N_val, w, r, d, seed=100)
         x_val, y_val = val_data[:,:2], val_data[:,2]
         x_val, y_val = x_val. y_val. reshape(1, len(y_val))
         wh, wo, mse, mse val = trainMLP(x train, y train, [5], learning rate, alpha,
                                          epochs, verbose=False, X_val=x_val, D_val=y_val)
         train_pred = MLP(x_train, wh, wo).flatten()
         train_pred[train_pred > 0] = 1
         train\_pred[train\_pred < 0] = -1
         val_pred = MLP(x_val, wh, wo).flatten()
         val pred[val pred > 0] = 1
         val pred[val pred < 0] = -1
         train_acc = (train_pred == y_train).sum() / N_train
         val_acc = (val_pred == y_val).sum() / N_val
```

1 of 2 11/19/2019, 12:43 AM

```
In [11]: epoch_grid = np.arange(0, epochs) + 1
          print('Early Stopping Point at', epoch_grid[np.argmin(mse_val)], 'epochs')
          print('Final Accuracies')
          print('Training: {}%'.format(train_acc*100))
          print('Testing: {}%'.format(val acc*100))
          fig, ax = plt.subplots()
          ax.plot(epoch_grid, np.log10(mse.flatten()), color='blue', markeredgewidth=2, label='Trainin
          g')
          ax.plot(epoch_grid, np.log10(mse_val.flatten()), '--', color='crimson', markeredgewidth=2, 1
          abel='Testing')
          ax.set_xlabel('Epochs')
          ax.set_ylabel(r'$\log_{10}$MSE')
          ax.xaxis.set_major_locator(mtick.MultipleLocator(25000))
ax.xaxis.set_minor_locator(mtick.MultipleLocator(5000))
          ax.yaxis.set_major_locator(mtick.MultipleLocator(0.5))
          ax.yaxis.set_minor_locator(mtick.MultipleLocator(0.1))
          ax.ticklabel_format(axis='x', style='sci', scilimits=(3,3), useMathText=True)
          ax.set xlim(-5000, epochs+5000)
          ax.set_ylim(-1.5,0.2)
          ax.legend()
          fig.tight layout(pad=0.5)
          #fig.savefig('../prob3a.eps', dpi=500)
```

Early Stopping Point at 47864 epochs Final Accuracies Training: 100.0% Testing: 98.8%



2 of 2 11/19/2019, 12:43 AM