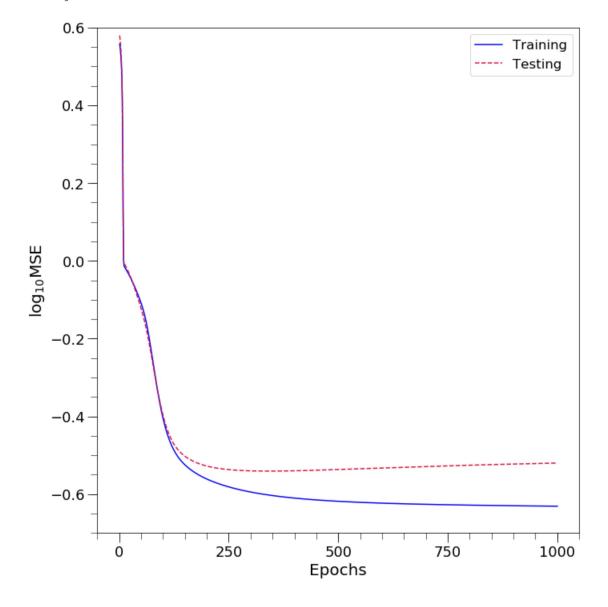
```
In [12]: | #!/usr/bin/env python3
         # -*- coding: utf-8 -*-
         Created on Sun Nov 17 16:17:27 2019
         @author: jorgeagr
         import numpy as np
         from requiredFunctions.trainMLP import trainMLP
         from requiredFunctions.MLP import MLP
         from requiredFunctions.gaussX import gaussX
         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
         var = 1
         # Model parameters
         learning rate = 1e-3
         alpha = 0
         epochs = 1000
         # Init train data
         train_data = gaussX(N_train, var, 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 = gaussX(N_val, var, seed=100)
         x_val, y_val = val_data[:,:2], val_data[:,2]
         x_val, y_val = x_val.T, 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</pre>
         train_acc = (train_pred == y_train).sum() / N_train
         val_acc = (val_pred == y_val).sum() / N_val
```

1 of 2 11/19/2019, 12:44 AM

```
In [13]: 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(250))
ax.xaxis.set_minor_locator(mtick.MultipleLocator(50))
          ax.yaxis.set_major_locator(mtick.MultipleLocator(0.2))
          ax.yaxis.set_minor_locator(mtick.MultipleLocator(0.05))
          ax.set_xlim(-50,epochs+50)
          ax.set_ylim(-0.7,0.6)
          ax.legend()
          fig.tight layout(pad=0.5)
          #fig.savefig('.../prob3b.eps', dpi=500)
```



2 of 2 11/19/2019, 12:44 AM