

## ex4

April 19, 2017

```
In [ ]: import numpy as np
        from sklearn import neural_network
        from sklearn import metrics
        import matplotlib.pyplot as plt
        import sys
        import itertools

exp_train = np.genfromtxt('train_expression.csv', delimiter=',', skip_header=1)
exp_test = np.genfromtxt('test_expression.csv', delimiter=',', skip_header=1)
train_phenotype = (np.genfromtxt('train_phen.csv', delimiter=',', skip_header=1))
test_phenotype = (np.genfromtxt('test_phen.csv', delimiter=',', skip_header=1))

exp_train_mean = np.mean(exp_train, axis=1)
exp_train_std = np.std(exp_train, axis=1)
exp_test_mean = np.mean(exp_test, axis=1)
exp_test_std = np.std(exp_train, axis=1)

for i in range(exp_train.shape[0]):
    for j in range(exp_train.shape[1]):
        exp_train[i][j] = (exp_train[i][j] - exp_train_mean[i]) / exp_train_std[i]

for i in range(exp_test.shape[0]):
    for j in range(exp_test.shape[1]):
        exp_test[i][j] = (exp_test[i][j] - exp_test_mean[i]) / exp_test_std[i]

l_1 = [50 for i in range(1)]
l_2 = [50 for i in range(2)]
l_16 = np.full(16, 50, dtype=int)
l_32 = np.full(32, 50, dtype=int)

model_1r_layer = neural_network.MLPClassifier(hidden_layer_sizes=l_1, activation_fn='relu')
model_1_relu = model_1r_layer.fit(exp_train, train_phenotype)
predict_phen_1_relu = model_1_relu.predict(exp_test)
error_1_relu = np.abs(predict_phen_1_relu - test_phenotype).sum() / test_phenotype.sum()
print('% error for 1 layer RELU w/ 200 iterations: ' + str(error_1_relu))
```

```

model_2r_layer = neural_network.MLPClassifier(hidden_layer_sizes=1_2, activ
model_2_relu = model_2r_layer.fit(exp_train, train_phenotype)
predict_phen_2_relu = model_2_relu.predict(exp_test)
error_2_relu = np.abs(predict_phen_2_relu - test_phenotype).sum() / test_ph
print('% error for 2 layer RELU w/ 200 iterations: ' + str(error_2_relu))

model_16r_layer = neural_network.MLPClassifier(hidden_layer_sizes=1_16, act
model_16_relu = model_16r_layer.fit(exp_train, train_phenotype)
predict_phen_16_relu = model_16_relu.predict(exp_test)
error_16_relu = np.sum(np.abs(predict_phen_16_relu - test_phenotype)) / tes
print('% error for 16 layer RELU w/ 200 iterations: ' + str(error_16_relu))

model_32r_layer = neural_network.MLPClassifier(hidden_layer_sizes=1_32, act
model_32_relu = model_32r_layer.fit(exp_train, train_phenotype)
predict_phen_32_relu = model_32_relu.predict(exp_test)
error_32_relu = np.sum(np.abs(predict_phen_32_relu - test_phenotype)) / tes
print('% error for 32 layer RELU w/ 200 iterations: ' + str(error_32_relu))

model_1l_layer = neural_network.MLPClassifier(hidden_layer_sizes=1_1, activ
model_1_logistic = model_1l_layer.fit(exp_train, train_phenotype)
predict_phen_1_logistic = model_1_logistic.predict(exp_test)
error_1_logistic = np.abs(predict_phen_1_logistic - test_phenotype).sum()/t
print('% error for 1 layer logistic activation w/ 200 iterations: ' + str

model_2l_layer = neural_network.MLPClassifier(hidden_layer_sizes=1_2, activ
model_2_logistic = model_2l_layer.fit(exp_train, train_phenotype)
predict_phen_2_logistic = model_2_logistic.predict(exp_test)
error_2_logistic = np.abs(predict_phen_2_logistic - test_phenotype).sum()/t
print('% error for 2 layer logistic activation w/ 200 iterations: ' + str

model_16l_layer = neural_network.MLPClassifier(hidden_layer_sizes=1_16, act
model_16_logistic = model_16l_layer.fit(exp_train, train_phenotype)
predict_phen_16_logistic = model_16_logistic.predict(exp_test)
error_16_logistic = np.abs(predict_phen_16_logistic - test_phenotype).sum()
print('% error for 16 layer logistic activation w/ 200 iterations: ' + str

model_32l_layer = neural_network.MLPClassifier(hidden_layer_sizes=1_32, act
model_32_logistic = model_32l_layer.fit(exp_train, train_phenotype)
predict_phen_32_logistic = model_32_logistic.predict(exp_test)
error_32_logistic = np.abs(predict_phen_32_logistic - test_phenotype).sum()
print('% error for 32 layer logistic activation w/ 200 iterations: ' + str

% error for 1 layer RELU w/ 200 iterations: 72
% error for 2 layer RELU w/ 200 iterations: 68
% error for 16 layer RELU w/ 200 iterations: 69
% error for 32 layer RELU w/ 200 iterations: 69

```

C:\Users\tmelo1\Anaconda2\lib\site-packages\sklearn\neural\_network\multilayer\_perce

```

% (), ConvergenceWarning)

% error for 1 layer logistic activation w/ 200 iterations: 59
% error for 2 layer logistic activation w/ 200 iterations: 59
% error for 16 layer logistic activation w/ 200 iterations: 56
% error for 32 layer logistic activation w/ 200 iterations: 50

In [ ]: model_1r_layer = neural_network.MLPClassifier(hidden_layer_sizes=1_1, activ
model_1_relu = model_1r_layer.fit(exp_train, train_phenotype)
predict_phen_1_relu = model_1_relu.predict(exp_test)
error_1_relu = np.abs(predict_phen_1_relu - test_phenotype).sum() / test_ph
print('% error for 1 layer RELU w/ 10000 iterations: ' + str(error_1_relu))

model_2r_layer = neural_network.MLPClassifier(hidden_layer_sizes=1_2, activ
model_2_relu = model_2r_layer.fit(exp_train, train_phenotype)
predict_phen_2_relu = model_2_relu.predict(exp_test)
error_2_relu = np.abs(predict_phen_2_relu - test_phenotype).sum() / test_ph
print('% error for 2 layer RELU w/ 10000 iterations: ' + str(error_2_relu))

model_16r_layer = neural_network.MLPClassifier(hidden_layer_sizes=1_16, act
model_16_relu = model_16r_layer.fit(exp_train, train_phenotype)
predict_phen_16_relu = model_16_relu.predict(exp_test)
error_16_relu = np.sum(np.abs(predict_phen_16_relu - test_phenotype)) / tes
print('% error for 16 layer RELU w/ 10000 iterations: ' + str(error_16_relu))

model_32r_layer = neural_network.MLPClassifier(hidden_layer_sizes=1_32, act
model_32_relu = model_32r_layer.fit(exp_train, train_phenotype)
predict_phen_32_relu = model_32_relu.predict(exp_test)
error_32_relu = np.sum(np.abs(predict_phen_32_relu - test_phenotype)) / tes
print('% error for 32 layer RELU w/ 10000 iterations: ' + str(error_32_relu))

model_1l_layer = neural_network.MLPClassifier(hidden_layer_sizes=1_1, activ
model_1_logistic = model_1l_layer.fit(exp_train, train_phenotype)
predict_phen_1_logistic = model_1_logistic.predict(exp_test)
error_1_logistic = np.abs(predict_phen_1_logistic - test_phenotype).sum()/t
print('% error for 1 layer logistic activation w/ 1000 iterations: ' + str

model_2l_layer = neural_network.MLPClassifier(hidden_layer_sizes=1_2, activ
model_2_logistic = model_2l_layer.fit(exp_train, train_phenotype)
predict_phen_2_logistic = model_2_logistic.predict(exp_test)
error_2_logistic = np.abs(predict_phen_2_logistic - test_phenotype).sum()/t
print('% error for 2 layer logistic activation w/ 10000 iterations: ' + str

model_16l_layer = neural_network.MLPClassifier(hidden_layer_sizes=1_16, act
model_16_logistic = model_16l_layer.fit(exp_train, train_phenotype)
predict_phen_16_logistic = model_16_logistic.predict(exp_test)

```

```

error_16_logistic = np.abs(predict_phen_16_logistic - test_phenotype).sum()
print('% error for 16 layer logistic activation w/ 10000 iterations: ' + s

model_32l_layer = neural_network.MLPClassifier(hidden_layer_sizes=1_32, act
model_32_logistic = model_32l_layer.fit(exp_train, train_phenotype)
predict_phen_32_logistic = model_32_logistic.predict(exp_test)
error_32_logistic = np.abs(predict_phen_32_logistic - test_phenotype).sum()
print('% error for 32 layer logistic activation w/ 10000 iterations: ' + s

% error for 1 layer RELU w/ 10000 iterations: 72
% error for 2 layer RELU w/ 10000 iterations: 68
% error for 16 layer RELU w/ 10000 iterations: 69
% error for 32 layer RELU w/ 10000 iterations: 69

```

```
In [ ]:
```