

```
In [1]: # mount my Google Drive
from google.colab import drive
drive.mount("/content/drive")
```

Mounted at /content/drive

```
In [2]: # Change the working directory to hw5
import os
os.chdir('/content/drive/MyDrive/CSC578/hw5/hw5-files')
```

```
In [3]: # a helper function
def format_res(collection_L):
    """
    A function that takes a list and returns a formatted string
    """
    res_str = '['
    for i in range(len(train_error)):
        if i < (len(train_error) - 1):
            res_str += str(train_error[i]) + ',' + '\n'
        else:
            res_str += str(train_error[i]) + ']'
    return res_str
```

578hw5 -- Start-up application code

(0) Tiny experiment with One-training-One-test Iris data

[You can run the start-up code as is to generate this.]

```
In [4]: # import NN578_network2_cp as network2
import NN578_network2 as network2
import numpy as np

# Test with one-data Iris data

inst1 = (np.array([5.7, 3, 4.2, 1.2]), np.array([0., 1., 0.]))
x1 = np.reshape(inst1[0], (4, 1))
y1 = np.reshape(inst1[1], (3, 1))
sample1 = [(x1, y1)]
inst2 = (np.array([4.8, 3.4, 1.6, 0.2]), np.array([1., 0., 0.]))
x2 = np.reshape(inst2[0], (4, 1))
y2 = np.reshape(inst2[1], (3, 1))
sample2 = [(x2, y2)]

net4 = network2.load_network("iris-423.dat")
net4.set_parameters(cost=network2.QuadraticCost)

net4.SGD(sample1, 2, 1, 1.0, evaluation_data=sample2, monitor_evaluation_cost=True,
        monitor_evaluation_accuracy=True,
        monitor_training_cost=True,
        monitor_training_accuracy=True)
```

Epoch 0 training complete

Cost on training data: 0.26673128660052947

```
Accuracy on training data: 1 / 1
Cost on evaluation data: 0.3244002758397572
Accuracy on evaluation data: 0 / 1
```

```
Epoch 1 training complete
Cost on training data: 0.2107866577006649
Accuracy on training data: 1 / 1
Cost on evaluation data: 0.37647122809828154
Accuracy on evaluation data: 0 / 1
```

```
Out[4]: ([0.26673128660052947, 0.2107866577006649],
        [1, 1],
        [0.3244002758397572, 0.37647122809828154],
        [0, 0])
```

Load the iris_train, iris_test datasets

```
In [5]: # Load the iris train-test (separate) data files
def my_load_csv(fname, no_trainfeatures, no_testfeatures):
    ret = np.genfromtxt(fname, delimiter=',')
    data = np.array([(entry[:no_trainfeatures], entry[no_trainfeatures:]) for entry in ret])
    temp_inputs = [np.reshape(x, (no_trainfeatures, 1)) for x in data[:,0]]
    temp_results = [np.reshape(y, (no_testfeatures, 1)) for y in data[:,1]]
    dataset = list(zip(temp_inputs, temp_results))
    return dataset

iris_train = my_load_csv('iris-train-1.csv', 4, 3)
iris_test = my_load_csv('iris-test-1.csv', 4, 3)
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:4: VisibleDeprecationWarning: Creating an ndarray from ragged nested sequences (which is a list-or-tuple of lists-or-tuples-or ndarrays with different lengths or shapes) is deprecated. If you meant to do this, you must specify 'dtype=object' when creating the ndarray.
after removing the cwd from sys.path.
```

(1) Sigmoid + Sigmoid + QuadraticCost

[You can run the start-up code as is to generate this.]

```
In [6]: net2 = network2.load_network("iris-423.dat")

# Set hyper-parameter values individually after the network
net2.set_parameters(cost=network2.QuadraticCost, act_hidden=network2.Sigmoid, act_output=network2.Sigmoid)

train_error, train_correct, test_error, test_correct = net2.SGD(iris_train,
                                                                    15, 10, 1.0,
                                                                    evaluation_data=iris_test,
                                                                    monitor_evaluation_cost=True,
                                                                    monitor_evaluation_accuracy=True,
                                                                    monitor_training_cost=True,
                                                                    monitor_training_accuracy=True)

print('Training error: \n %s\n' %format_res(train_error))
print('Training correct: \n %s\n' %train_correct)
print('Test error: \n %s\n' %format_res(test_error))
print('Test correct: \n %s' %test_correct)
```

```
Epoch 0 training complete
Cost on training data: 0.3392795030659105
```

Accuracy on training data: 21 / 95
Cost on evaluation data: 0.34444833747414394
Accuracy on evaluation data: 15 / 55

Epoch 1 training complete
Cost on training data: 0.3332680387432046
Accuracy on training data: 33 / 95
Cost on evaluation data: 0.3357508215659017
Accuracy on evaluation data: 17 / 55

Epoch 2 training complete
Cost on training data: 0.33169864345910943
Accuracy on training data: 33 / 95
Cost on evaluation data: 0.33331819877313307
Accuracy on evaluation data: 17 / 55

Epoch 3 training complete
Cost on training data: 0.3280047701277544
Accuracy on training data: 33 / 95
Cost on evaluation data: 0.3293129769432788
Accuracy on evaluation data: 17 / 55

Epoch 4 training complete
Cost on training data: 0.3105024143021552
Accuracy on training data: 65 / 95
Cost on evaluation data: 0.311452504680719
Accuracy on evaluation data: 35 / 55

Epoch 5 training complete
Cost on training data: 0.2822226213778409
Accuracy on training data: 64 / 95
Cost on evaluation data: 0.28266644748693787
Accuracy on evaluation data: 39 / 55

Epoch 6 training complete
Cost on training data: 0.2618530851998504
Accuracy on training data: 65 / 95
Cost on evaluation data: 0.2622221163236103
Accuracy on evaluation data: 40 / 55

Epoch 7 training complete
Cost on training data: 0.24637620748622577
Accuracy on training data: 64 / 95
Cost on evaluation data: 0.24693122708939252
Accuracy on evaluation data: 40 / 55

Epoch 8 training complete
Cost on training data: 0.23397287354343002
Accuracy on training data: 65 / 95
Cost on evaluation data: 0.234842439155326
Accuracy on evaluation data: 41 / 55

Epoch 9 training complete
Cost on training data: 0.2240389321795399
Accuracy on training data: 67 / 95
Cost on evaluation data: 0.22523745295521982
Accuracy on evaluation data: 41 / 55

Epoch 10 training complete
Cost on training data: 0.21604989548513004
Accuracy on training data: 70 / 95
Cost on evaluation data: 0.2175515310927702
Accuracy on evaluation data: 44 / 55

Epoch 11 training complete
Cost on training data: 0.20957620155325693

Accuracy on training data: 73 / 95
Cost on evaluation data: 0.21134700807259035
Accuracy on evaluation data: 47 / 55

Epoch 12 training complete
Cost on training data: 0.2042808031846783
Accuracy on training data: 81 / 95
Cost on evaluation data: 0.20628846293823358
Accuracy on evaluation data: 52 / 55

Epoch 13 training complete
Cost on training data: 0.19990418595499365
Accuracy on training data: 94 / 95
Cost on evaluation data: 0.20212044282632813
Accuracy on evaluation data: 54 / 55

Epoch 14 training complete
Cost on training data: 0.19624754621498341
Accuracy on training data: 65 / 95
Cost on evaluation data: 0.19864854335473608
Accuracy on evaluation data: 35 / 55

Training error:
[0.3392795030659105,
0.3332680387432046,
0.33169864345910943,
0.3280047701277544,
0.3105024143021552,
0.2822226213778409,
0.2618530851998504,
0.24637620748622577,
0.23397287354343002,
0.2240389321795399,
0.21604989548513004,
0.20957620155325693,
0.2042808031846783,
0.19990418595499365,
0.19624754621498341]

Training correct:
[21, 33, 33, 33, 65, 64, 65, 64, 65, 67, 70, 73, 81, 94, 65]

Test error:
[0.3392795030659105,
0.3332680387432046,
0.33169864345910943,
0.3280047701277544,
0.3105024143021552,
0.2822226213778409,
0.2618530851998504,
0.24637620748622577,
0.23397287354343002,
0.2240389321795399,
0.21604989548513004,
0.20957620155325693,
0.2042808031846783,
0.19990418595499365,
0.19624754621498341]

Test correct:
[15, 17, 17, 17, 35, 39, 40, 40, 41, 41, 44, 47, 52, 54, 35]

(2) Sigmoid + Sigmoid + CrossEntropy

```
In [7]: net2 = network2.load_network("iris-423.dat")
# Set hyper-parameter values individually after the network
net2.set_parameters(cost=network2.CrossEntropyCost, act_hidden=network2.Sigmoid, act_out=network2.Sigmoid,
train_error, train_correct, test_error, test_correct = net2.SGD(iris_train,
                                                                    15, 10, 1.0,
                                                                    evaluation_data=iris_test,
                                                                    monitor_evaluation_cost=network2.Cost,
                                                                    monitor_evaluation_accuracy=network2.Accuracy,
                                                                    monitor_training_cost=network2.Cost,
                                                                    monitor_training_accuracy=network2.Accuracy)

print('Training error: \n %s\n' %format_res(train_error))
print('Training correct: \n %s\n' %train_correct)
print('Test error: \n %s\n' %format_res(test_error))
print('Test correct: \n %s' %test_correct)
```

```
Epoch 0 training complete
Cost on training data: 1.8603802674198153
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.86801026041802
Accuracy on evaluation data: 18 / 55
```

```
Epoch 1 training complete
Cost on training data: 1.460606534019205
Accuracy on training data: 69 / 95
Cost on evaluation data: 1.466055535637722
Accuracy on evaluation data: 44 / 55
```

```
Epoch 2 training complete
Cost on training data: 1.2667163241805766
Accuracy on training data: 92 / 95
Cost on evaluation data: 1.2774861545892018
Accuracy on evaluation data: 54 / 55
```

```
Epoch 3 training complete
Cost on training data: 1.157146339882345
Accuracy on training data: 65 / 95
Cost on evaluation data: 1.1718471307186422
Accuracy on evaluation data: 35 / 55
```

```
Epoch 4 training complete
Cost on training data: 1.0668798505712163
Accuracy on training data: 65 / 95
Cost on evaluation data: 1.086639159620854
Accuracy on evaluation data: 35 / 55
```

```
Epoch 5 training complete
Cost on training data: 1.0269316288572254
Accuracy on training data: 65 / 95
Cost on evaluation data: 1.0468525231446604
Accuracy on evaluation data: 35 / 55
```

```
Epoch 6 training complete
Cost on training data: 1.0030159043215203
Accuracy on training data: 65 / 95
Cost on evaluation data: 1.0230817418856184
Accuracy on evaluation data: 35 / 55
```

```
Epoch 7 training complete
Cost on training data: 0.9876051818657172
Accuracy on training data: 65 / 95
```

Cost on evaluation data: 1.008345450466028
Accuracy on evaluation data: 35 / 55

Epoch 8 training complete
Cost on training data: 0.9772276017636308
Accuracy on training data: 65 / 95
Cost on evaluation data: 0.9990397749812812
Accuracy on evaluation data: 35 / 55

Epoch 9 training complete
Cost on training data: 0.9690604086884842
Accuracy on training data: 65 / 95
Cost on evaluation data: 0.9916782036368461
Accuracy on evaluation data: 35 / 55

Epoch 10 training complete
Cost on training data: 0.9623999454604477
Accuracy on training data: 64 / 95
Cost on evaluation data: 0.9856464279019421
Accuracy on evaluation data: 35 / 55

Epoch 11 training complete
Cost on training data: 0.956444321822222
Accuracy on training data: 64 / 95
Cost on evaluation data: 0.9800468983199271
Accuracy on evaluation data: 35 / 55

Epoch 12 training complete
Cost on training data: 0.9519759891060907
Accuracy on training data: 64 / 95
Cost on evaluation data: 0.9761371328834567
Accuracy on evaluation data: 34 / 55

Epoch 13 training complete
Cost on training data: 0.9459190082319839
Accuracy on training data: 64 / 95
Cost on evaluation data: 0.9696831841725915
Accuracy on evaluation data: 35 / 55

Epoch 14 training complete
Cost on training data: 0.9566747295346814
Accuracy on training data: 64 / 95
Cost on evaluation data: 0.9846220534440293
Accuracy on evaluation data: 34 / 55

Training error:
[1.8603802674198153,
1.460606534019205,
1.2667163241805766,
1.157146339882345,
1.0668798505712163,
1.0269316288572254,
1.0030159043215203,
0.9876051818657172,
0.9772276017636308,
0.9690604086884842,
0.9623999454604477,
0.956444321822222,
0.9519759891060907,
0.9459190082319839,
0.9566747295346814]

Training correct:
[33, 69, 92, 65, 65, 65, 65, 65, 65, 65, 64, 64, 64, 64, 64]

Test error:

```
[1.8603802674198153,  
1.460606534019205,  
1.2667163241805766,  
1.157146339882345,  
1.0668798505712163,  
1.0269316288572254,  
1.0030159043215203,  
0.9876051818657172,  
0.9772276017636308,  
0.9690604086884842,  
0.9623999454604477,  
0.9564443218222222,  
0.9519759891060907,  
0.9459190082319839,  
0.9566747295346814]
```

Test correct:

```
[18, 44, 54, 35, 35, 35, 35, 35, 35, 35, 35, 35, 34, 35, 34]
```

(3) Sigmoid + Softmax + CrossEntropy

```
In [8]: net2 = network2.load_network("iris-423.dat")  
# Set hyper-parameter values individually after the network  
net2.set_parameters(cost=network2.CrossEntropyCost, act_hidden=network2.Sigmoid, act_out=  
  
train_error, train_correct, test_error, test_correct = net2.SGD(iris_train,  
                                                                15, 10, 1.0,  
                                                                evaluation_data=iris_te  
                                                                monitor_evaluation_cost  
                                                                monitor_evaluation_accur  
                                                                monitor_training_cost=Tr  
                                                                monitor_training_accuracy=T  
  
print('Training error: \n %s\n' %format_res(train_error))  
print('Training correct: \n %s\n' %train_correct)  
print('Test error: \n %s\n' %format_res(test_error))  
print('Test correct: \n %s' %test_correct)
```

```
Epoch 0 training complete  
Cost on training data: 1.0725541578143791  
Accuracy on training data: 33 / 95  
Cost on evaluation data: 1.0757367096860873  
Accuracy on evaluation data: 17 / 55
```

```
Epoch 1 training complete  
Cost on training data: 0.7394112289313796  
Accuracy on training data: 65 / 95  
Cost on evaluation data: 0.7407176979301714  
Accuracy on evaluation data: 40 / 55
```

```
Epoch 2 training complete  
Cost on training data: 0.6205618473893993  
Accuracy on training data: 85 / 95  
Cost on evaluation data: 0.6257845728763475  
Accuracy on evaluation data: 52 / 55
```

```
Epoch 3 training complete  
Cost on training data: 0.5670736844143868  
Accuracy on training data: 93 / 95  
Cost on evaluation data: 0.5731784036177967
```

Accuracy on evaluation data: 54 / 55

Epoch 4 training complete

Cost on training data: 0.5333770349417168

Accuracy on training data: 88 / 95

Cost on evaluation data: 0.5400095734834603

Accuracy on evaluation data: 54 / 55

Epoch 5 training complete

Cost on training data: 0.5023152528399535

Accuracy on training data: 81 / 95

Cost on evaluation data: 0.5103342513102356

Accuracy on evaluation data: 51 / 55

Epoch 6 training complete

Cost on training data: 0.49562244982507964

Accuracy on training data: 83 / 95

Cost on evaluation data: 0.5032223921305511

Accuracy on evaluation data: 51 / 55

Epoch 7 training complete

Cost on training data: 1.209107753252144

Accuracy on training data: 30 / 95

Cost on evaluation data: 1.1812599586020067

Accuracy on evaluation data: 20 / 55

Epoch 8 training complete

Cost on training data: 0.4959209670421647

Accuracy on training data: 65 / 95

Cost on evaluation data: 0.5043047769715578

Accuracy on evaluation data: 35 / 55

Epoch 9 training complete

Cost on training data: 0.48173436665899527

Accuracy on training data: 65 / 95

Cost on evaluation data: 0.48915877340088626

Accuracy on evaluation data: 35 / 55

Epoch 10 training complete

Cost on training data: 0.4730909670850905

Accuracy on training data: 65 / 95

Cost on evaluation data: 0.4792400201766988

Accuracy on evaluation data: 35 / 55

Epoch 11 training complete

Cost on training data: 0.4770244676850298

Accuracy on training data: 71 / 95

Cost on evaluation data: 0.48383236398826346

Accuracy on evaluation data: 46 / 55

Epoch 12 training complete

Cost on training data: 0.471738288694459

Accuracy on training data: 65 / 95

Cost on evaluation data: 0.4783861410469212

Accuracy on evaluation data: 35 / 55

Epoch 13 training complete

Cost on training data: 0.6019085510144767

Accuracy on training data: 51 / 95

Cost on evaluation data: 0.5645536612751829

Accuracy on evaluation data: 29 / 55

Epoch 14 training complete

Cost on training data: 0.7069201485054665

Accuracy on training data: 64 / 95

Cost on evaluation data: 0.7042809323432099

Training error:

```
[1.0725541578143791,
0.7394112289313796,
0.6205618473893993,
0.5670736844143868,
0.5333770349417168,
0.5023152528399535,
0.49562244982507964,
1.209107753252144,
0.4959209670421647,
0.48173436665899527,
0.4730909670850905,
0.4770244676850298,
0.471738288694459,
0.6019085510144767,
0.7069201485054665]
```

Training correct:

```
[33, 65, 85, 93, 88, 81, 83, 30, 65, 65, 65, 71, 65, 51, 64]
```

Test error:

```
[1.0725541578143791,
0.7394112289313796,
0.6205618473893993,
0.5670736844143868,
0.5333770349417168,
0.5023152528399535,
0.49562244982507964,
1.209107753252144,
0.4959209670421647,
0.48173436665899527,
0.4730909670850905,
0.4770244676850298,
0.471738288694459,
0.6019085510144767,
0.7069201485054665]
```

Test correct:

```
[17, 40, 52, 54, 54, 51, 51, 20, 35, 35, 35, 46, 35, 29, 40]
```

(4) Sigmoid + Softmax + LogLikelihood

```
In [9]: net2 = network2.load_network("iris-423.dat")
# Set hyper-parameter values individually after the network
net2.set_parameters(cost=network2.LogLikelihood, act_hidden=network2.Sigmoid, act_output

train_error, train_correct, test_error, test_correct = net2.SGD(iris_train,
                                                                15, 10, 1.0,
                                                                evaluation_data=iris_te
                                                                monitor_evaluation_cost
                                                                monitor_evaluation_accur
                                                                monitor_training_cost=Tr
                                                                monitor_training_accuracy=T

print('Training error: \n %s\n' %format_res(train_error))
print('Training correct: \n %s\n' %train_correct)
print('Test error: \n %s\n' %format_res(test_error))
print('Test correct: \n %s' %test_correct)
```

Epoch 0 training complete
Cost on training data: 1.0725541578143791
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.0757367096860873
Accuracy on evaluation data: 17 / 55

Epoch 1 training complete
Cost on training data: 0.7394112289313796
Accuracy on training data: 65 / 95
Cost on evaluation data: 0.7407176979301714
Accuracy on evaluation data: 40 / 55

Epoch 2 training complete
Cost on training data: 0.6205618473893993
Accuracy on training data: 85 / 95
Cost on evaluation data: 0.6257845728763475
Accuracy on evaluation data: 52 / 55

Epoch 3 training complete
Cost on training data: 0.5670736844143868
Accuracy on training data: 93 / 95
Cost on evaluation data: 0.5731784036177967
Accuracy on evaluation data: 54 / 55

Epoch 4 training complete
Cost on training data: 0.5333770349417168
Accuracy on training data: 88 / 95
Cost on evaluation data: 0.5400095734834603
Accuracy on evaluation data: 54 / 55

Epoch 5 training complete
Cost on training data: 0.5023152528399535
Accuracy on training data: 81 / 95
Cost on evaluation data: 0.5103342513102356
Accuracy on evaluation data: 51 / 55

Epoch 6 training complete
Cost on training data: 0.49562244982507964
Accuracy on training data: 83 / 95
Cost on evaluation data: 0.5032223921305511
Accuracy on evaluation data: 51 / 55

Epoch 7 training complete
Cost on training data: 1.209107753252144
Accuracy on training data: 30 / 95
Cost on evaluation data: 1.1812599586020067
Accuracy on evaluation data: 20 / 55

Epoch 8 training complete
Cost on training data: 0.4959209670421647
Accuracy on training data: 65 / 95
Cost on evaluation data: 0.5043047769715578
Accuracy on evaluation data: 35 / 55

Epoch 9 training complete
Cost on training data: 0.48173436665899527
Accuracy on training data: 65 / 95
Cost on evaluation data: 0.48915877340088626
Accuracy on evaluation data: 35 / 55

Epoch 10 training complete
Cost on training data: 0.4730909670850905
Accuracy on training data: 65 / 95
Cost on evaluation data: 0.4792400201766988
Accuracy on evaluation data: 35 / 55

Epoch 11 training complete
Cost on training data: 0.4770244676850298
Accuracy on training data: 71 / 95
Cost on evaluation data: 0.48383236398826346
Accuracy on evaluation data: 46 / 55

Epoch 12 training complete
Cost on training data: 0.471738288694459
Accuracy on training data: 65 / 95
Cost on evaluation data: 0.4783861410469212
Accuracy on evaluation data: 35 / 55

Epoch 13 training complete
Cost on training data: 0.6019085510144767
Accuracy on training data: 51 / 95
Cost on evaluation data: 0.5645536612751829
Accuracy on evaluation data: 29 / 55

Epoch 14 training complete
Cost on training data: 0.7069201485054665
Accuracy on training data: 64 / 95
Cost on evaluation data: 0.7042809323432099
Accuracy on evaluation data: 40 / 55

Training error:
[1.0725541578143791,
0.7394112289313796,
0.6205618473893993,
0.5670736844143868,
0.5333770349417168,
0.5023152528399535,
0.49562244982507964,
1.209107753252144,
0.4959209670421647,
0.48173436665899527,
0.4730909670850905,
0.4770244676850298,
0.471738288694459,
0.6019085510144767,
0.7069201485054665]

Training correct:
[33, 65, 85, 93, 88, 81, 83, 30, 65, 65, 65, 71, 65, 51, 64]

Test error:
[1.0725541578143791,
0.7394112289313796,
0.6205618473893993,
0.5670736844143868,
0.5333770349417168,
0.5023152528399535,
0.49562244982507964,
1.209107753252144,
0.4959209670421647,
0.48173436665899527,
0.4730909670850905,
0.4770244676850298,
0.471738288694459,
0.6019085510144767,
0.7069201485054665]

Test correct:
[17, 40, 52, 54, 54, 51, 51, 20, 35, 35, 35, 46, 35, 29, 40]

(5) ReLU + Softmax + CrossEntropy

```
In [10]: net2 = network2.load_network("iris-423.dat")
# Set hyper-parameter values individually after the network
net2.set_parameters(cost=network2.CrossEntropyCost, act_hidden=network2.ReLU, act_output=network2.ReLU)

train_error, train_correct, test_error, test_correct = net2.SGD(iris_train,
                                                                15, 10, 1.0,
                                                                evaluation_data=iris_test,
                                                                monitor_evaluation_cost=True,
                                                                monitor_evaluation_accuracy=True,
                                                                monitor_training_cost=True,
                                                                monitor_training_accuracy=True)

print('Training error: \n %s\n' %format_res(train_error))
print('Training correct: \n %s\n' %train_correct)
print('Test error: \n %s\n' %format_res(test_error))
print('Test correct: \n %s' %test_correct)
```

Epoch 0 training complete
Cost on training data: 1.0978603945164458
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.1018835710978243
Accuracy on evaluation data: 17 / 55

Epoch 1 training complete
Cost on training data: 1.0978897492226267
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.1031078538772434
Accuracy on evaluation data: 17 / 55

Epoch 2 training complete
Cost on training data: 1.097891568467868
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.1031313957236202
Accuracy on evaluation data: 17 / 55

Epoch 3 training complete
Cost on training data: 1.097891605448383
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.1031318358327993
Accuracy on evaluation data: 17 / 55

Epoch 4 training complete
Cost on training data: 1.0978916061942783
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.1031318441811317
Accuracy on evaluation data: 17 / 55

Epoch 5 training complete
Cost on training data: 1.0978916062093553
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.1031318443416307
Accuracy on evaluation data: 17 / 55

Epoch 6 training complete
Cost on training data: 1.0978916062096606
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.1031318443447522
Accuracy on evaluation data: 17 / 55

Epoch 7 training complete

Cost on training data: 1.0978916062096673
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.1031318443448124
Accuracy on evaluation data: 17 / 55

Epoch 8 training complete
Cost on training data: 1.097891606209667
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.1031318443448146
Accuracy on evaluation data: 17 / 55

Epoch 9 training complete
Cost on training data: 1.0978916062096673
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.1031318443448146
Accuracy on evaluation data: 17 / 55

Epoch 10 training complete
Cost on training data: 1.0978916062096673
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.1031318443448146
Accuracy on evaluation data: 17 / 55

Epoch 11 training complete
Cost on training data: 1.0978916062096673
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.1031318443448146
Accuracy on evaluation data: 17 / 55

Epoch 12 training complete
Cost on training data: 1.0978916062096673
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.1031318443448146
Accuracy on evaluation data: 17 / 55

Epoch 13 training complete
Cost on training data: 1.0978916062096673
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.1031318443448146
Accuracy on evaluation data: 17 / 55

Epoch 14 training complete
Cost on training data: 1.0978916062096673
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.1031318443448146
Accuracy on evaluation data: 17 / 55

Training error:
[1.0978603945164458,
1.0978897492226267,
1.097891568467868,
1.097891605448383,
1.0978916061942783,
1.0978916062093553,
1.0978916062096606,
1.0978916062096673,
1.097891606209667,
1.0978916062096673,
1.0978916062096673,
1.0978916062096673,
1.0978916062096673,
1.0978916062096673,
1.0978916062096673,
1.0978916062096673]

Training correct:
[33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33]

```
Test error:
[1.0978603945164458,
1.0978897492226267,
1.097891568467868,
1.097891605448383,
1.0978916061942783,
1.0978916062093553,
1.0978916062096606,
1.0978916062096673,
1.097891606209667,
1.0978916062096673,
1.0978916062096673,
1.0978916062096673,
1.0978916062096673,
1.0978916062096673,
1.0978916062096673,
1.0978916062096673]
```

```
Test correct:
[17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17]
```

(6) ReLU + Softmax + LogLikelihood

```
In [11]: net2 = network2.load_network("iris-423.dat")
# Set hyper-parameter values individually after the network
net2.set_parameters(cost=network2.LogLikelihood, act_hidden=network2.ReLU, act_output=ne

train_error, train_correct, test_error, test_correct = net2.SGD(iris_train,
                                                                15, 10, 1.0,
                                                                evaluation_data=iris_te
                                                                monitor_evaluation_cost
                                                                monitor_evaluation_accur
                                                                monitor_training_cost=Tr
                                                                monitor_training_accuracy=T

print('Training error: \n %s\n' %format_res(train_error))
print('Training correct: \n %s\n' %train_correct)
print('Test error: \n %s\n' %format_res(test_error))
print('Test correct: \n %s' %test_correct)
```

```
Epoch 0 training complete
Cost on training data: 1.0978603945164458
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.1018835710978243
Accuracy on evaluation data: 17 / 55
```

```
Epoch 1 training complete
Cost on training data: 1.0978897492226267
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.1031078538772434
Accuracy on evaluation data: 17 / 55
```

```
Epoch 2 training complete
Cost on training data: 1.097891568467868
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.1031313957236202
Accuracy on evaluation data: 17 / 55
```

```
Epoch 3 training complete
Cost on training data: 1.097891605448383
Accuracy on training data: 33 / 95
```

Cost on evaluation data: 1.1031318358327993
Accuracy on evaluation data: 17 / 55

Epoch 4 training complete
Cost on training data: 1.0978916061942783
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.1031318441811317
Accuracy on evaluation data: 17 / 55

Epoch 5 training complete
Cost on training data: 1.0978916062093553
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.1031318443416307
Accuracy on evaluation data: 17 / 55

Epoch 6 training complete
Cost on training data: 1.0978916062096606
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.1031318443447522
Accuracy on evaluation data: 17 / 55

Epoch 7 training complete
Cost on training data: 1.0978916062096673
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.1031318443448124
Accuracy on evaluation data: 17 / 55

Epoch 8 training complete
Cost on training data: 1.097891606209667
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.1031318443448146
Accuracy on evaluation data: 17 / 55

Epoch 9 training complete
Cost on training data: 1.0978916062096673
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.1031318443448146
Accuracy on evaluation data: 17 / 55

Epoch 10 training complete
Cost on training data: 1.0978916062096673
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.1031318443448146
Accuracy on evaluation data: 17 / 55

Epoch 11 training complete
Cost on training data: 1.0978916062096673
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.1031318443448146
Accuracy on evaluation data: 17 / 55

Epoch 12 training complete
Cost on training data: 1.0978916062096673
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.1031318443448146
Accuracy on evaluation data: 17 / 55

Epoch 13 training complete
Cost on training data: 1.0978916062096673
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.1031318443448146
Accuracy on evaluation data: 17 / 55

Epoch 14 training complete
Cost on training data: 1.0978916062096673
Accuracy on training data: 33 / 95

Cost on evaluation data: 1.1031318443448146
Accuracy on evaluation data: 17 / 55

Training error:

```
[1.0978603945164458,  
1.0978897492226267,  
1.097891568467868,  
1.097891605448383,  
1.0978916061942783,  
1.0978916062093553,  
1.0978916062096606,  
1.0978916062096673,  
1.097891606209667,  
1.0978916062096673,  
1.0978916062096673,  
1.0978916062096673,  
1.0978916062096673,  
1.0978916062096673,  
1.0978916062096673,  
1.0978916062096673]
```

Training correct:

```
[33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33]
```

Test error:

```
[1.0978603945164458,  
1.0978897492226267,  
1.097891568467868,  
1.097891605448383,  
1.0978916061942783,  
1.0978916062093553,  
1.0978916062096606,  
1.0978916062096673,  
1.097891606209667,  
1.0978916062096673,  
1.0978916062096673,  
1.0978916062096673,  
1.0978916062096673,  
1.0978916062096673,  
1.0978916062096673,  
1.0978916062096673]
```

Test correct:

```
[17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17]
```

(7) Tanh + Sigmoid + Quadratic

```
In [12]: net2 = network2.load_network("iris-423.dat")  
# Set hyper-parameter values individually after the network  
net2.set_parameters(cost=network2.QuadraticCost, act_hidden=network2.Tanh, act_output=ne  
  
train_error, train_correct, test_error, test_correct = net2.SGD(iris_train,  
                                                                15, 10, 1.0,  
                                                                evaluation_data=iris_te  
                                                                monitor_evaluation_cost  
                                                                monitor_evaluation_accur  
                                                                monitor_training_cost=Tr  
                                                                monitor_training_accuracy=T  
  
print('Training error: \n %s\n' %format_res(train_error))  
print('Training correct: \n %s\n' %train_correct)  
print('Test error: \n %s\n' %format_res(test_error))  
print('Test correct: \n %s' %test_correct)
```


Epoch 0 training complete
Cost on training data: 0.3344311654023177
Accuracy on training data: 32 / 95
Cost on evaluation data: 0.3363579316611686
Accuracy on evaluation data: 18 / 55

Epoch 1 training complete
Cost on training data: 0.33175799332119593
Accuracy on training data: 33 / 95
Cost on evaluation data: 0.33326213908032815
Accuracy on evaluation data: 17 / 55

Epoch 2 training complete
Cost on training data: 0.2888319510914468
Accuracy on training data: 62 / 95
Cost on evaluation data: 0.28986158907272574
Accuracy on evaluation data: 38 / 55

Epoch 3 training complete
Cost on training data: 0.21879450487789684
Accuracy on training data: 62 / 95
Cost on evaluation data: 0.21994496188727214
Accuracy on evaluation data: 38 / 55

Epoch 4 training complete
Cost on training data: 0.19532043187167233
Accuracy on training data: 65 / 95
Cost on evaluation data: 0.1975753300018062
Accuracy on evaluation data: 35 / 55

Epoch 5 training complete
Cost on training data: 0.18552971480253466
Accuracy on training data: 65 / 95
Cost on evaluation data: 0.18851326412153177
Accuracy on evaluation data: 35 / 55

Epoch 6 training complete
Cost on training data: 0.18041560683139668
Accuracy on training data: 65 / 95
Cost on evaluation data: 0.18380157993229201
Accuracy on evaluation data: 35 / 55

Epoch 7 training complete
Cost on training data: 0.1773085579325556
Accuracy on training data: 65 / 95
Cost on evaluation data: 0.18092380021642968
Accuracy on evaluation data: 35 / 55

Epoch 8 training complete
Cost on training data: 0.17523125989906965
Accuracy on training data: 65 / 95
Cost on evaluation data: 0.17898859908100612
Accuracy on evaluation data: 35 / 55

Epoch 9 training complete
Cost on training data: 0.1737507681130413
Accuracy on training data: 65 / 95
Cost on evaluation data: 0.1776028110886066
Accuracy on evaluation data: 35 / 55

Epoch 10 training complete
Cost on training data: 0.17264571173278204
Accuracy on training data: 65 / 95
Cost on evaluation data: 0.17656425232755785
Accuracy on evaluation data: 35 / 55

Epoch 11 training complete
Cost on training data: 0.17179130790365219
Accuracy on training data: 65 / 95
Cost on evaluation data: 0.17575844050919673
Accuracy on evaluation data: 35 / 55

Epoch 12 training complete
Cost on training data: 0.17111209145146633
Accuracy on training data: 65 / 95
Cost on evaluation data: 0.1751159110291929
Accuracy on evaluation data: 35 / 55

Epoch 13 training complete
Cost on training data: 0.1705598755089031
Accuracy on training data: 65 / 95
Cost on evaluation data: 0.17459216589672671
Accuracy on evaluation data: 35 / 55

Epoch 14 training complete
Cost on training data: 0.1701025186391921
Accuracy on training data: 65 / 95
Cost on evaluation data: 0.1741574243227054
Accuracy on evaluation data: 35 / 55

Training error:
[0.3344311654023177,
0.33175799332119593,
0.2888319510914468,
0.21879450487789684,
0.19532043187167233,
0.18552971480253466,
0.18041560683139668,
0.1773085579325556,
0.17523125989906965,
0.1737507681130413,
0.17264571173278204,
0.17179130790365219,
0.17111209145146633,
0.1705598755089031,
0.1701025186391921]

Training correct:
[32, 33, 62, 62, 65, 65, 65, 65, 65, 65, 65, 65, 65, 65]

Test error:
[0.3344311654023177,
0.33175799332119593,
0.2888319510914468,
0.21879450487789684,
0.19532043187167233,
0.18552971480253466,
0.18041560683139668,
0.1773085579325556,
0.17523125989906965,
0.1737507681130413,
0.17264571173278204,
0.17179130790365219,
0.17111209145146633,
0.1705598755089031,
0.1701025186391921]

Test correct:
[18, 17, 38, 38, 35, 35, 35, 35, 35, 35, 35, 35, 35, 35]

(8) Tanh + Tanh + Quadratic

```
In [13]: net2 = network2.load_network("iris-423.dat")
# Set hyper-parameter values individually after the network
net2.set_parameters(cost=network2.QuadraticCost, act_hidden=network2.Tanh, act_output=ne

train_error, train_correct, test_error, test_correct = net2.SGD(iris_train,
                                                                15, 10, 1.0,
                                                                evaluation_data=iris_te
                                                                monitor_evaluation_cost
                                                                monitor_evaluation_accur
                                                                monitor_training_cost=Tr
                                                                monitor_training_accuracy=T

print('Training error: \n %s\n' %format_res(train_error))
print('Training correct: \n %s\n' %train_correct)
print('Test error: \n %s\n' %format_res(test_error))
print('Test correct: \n %s' %test_correct)
```

Epoch 0 training complete
Cost on training data: 1.5632335301767852
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.64478175751216
Accuracy on evaluation data: 17 / 55

Epoch 1 training complete
Cost on training data: 1.7700632291152376
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.8449848877028372
Accuracy on evaluation data: 17 / 55

Epoch 2 training complete
Cost on training data: 1.5001812828559293
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.5351845062561134
Accuracy on evaluation data: 17 / 55

Epoch 3 training complete
Cost on training data: 1.5570778587243312
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.5640765578847946
Accuracy on evaluation data: 17 / 55

Epoch 4 training complete
Cost on training data: 1.4008025521176595
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.4146963617680761
Accuracy on evaluation data: 17 / 55

Epoch 5 training complete
Cost on training data: 1.7515323812687178
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.8225619187617095
Accuracy on evaluation data: 17 / 55

Epoch 6 training complete
Cost on training data: 1.6150277984560082
Accuracy on training data: 30 / 95
Cost on evaluation data: 1.562815580801297
Accuracy on evaluation data: 20 / 55

Epoch 7 training complete

Cost on training data: 1.61285966722797
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.6450334946113043
Accuracy on evaluation data: 17 / 55

Epoch 8 training complete
Cost on training data: 1.5134550378166842
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.5728424434088755
Accuracy on evaluation data: 17 / 55

Epoch 9 training complete
Cost on training data: 1.7513250558857554
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.8214007961862155
Accuracy on evaluation data: 17 / 55

Epoch 10 training complete
Cost on training data: 1.4490755085588816
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.4702886285771268
Accuracy on evaluation data: 17 / 55

Epoch 11 training complete
Cost on training data: 1.5583510614176772
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.565059433182053
Accuracy on evaluation data: 17 / 55

Epoch 12 training complete
Cost on training data: 1.4579799250799323
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.5021212564337967
Accuracy on evaluation data: 17 / 55

Epoch 13 training complete
Cost on training data: 1.5744372542832024
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.5905698581047212
Accuracy on evaluation data: 17 / 55

Epoch 14 training complete
Cost on training data: 1.4493225175624624
Accuracy on training data: 33 / 95
Cost on evaluation data: 1.4643185339476041
Accuracy on evaluation data: 17 / 55

Training error:
[1.5632335301767852,
1.7700632291152376,
1.5001812828559293,
1.5570778587243312,
1.4008025521176595,
1.7515323812687178,
1.6150277984560082,
1.61285966722797,
1.5134550378166842,
1.7513250558857554,
1.4490755085588816,
1.5583510614176772,
1.4579799250799323,
1.5744372542832024,
1.4493225175624624]

Training correct:
[33, 33, 33, 33, 33, 33, 33, 30, 33, 33, 33, 33, 33, 33, 33]

```
Test error:
[1.5632335301767852,
1.7700632291152376,
1.5001812828559293,
1.5570778587243312,
1.4008025521176595,
1.7515323812687178,
1.6150277984560082,
1.61285966722797,
1.5134550378166842,
1.7513250558857554,
1.4490755085588816,
1.5583510614176772,
1.4579799250799323,
1.5744372542832024,
1.4493225175624624]
```

```
Test correct:
[17, 17, 17, 17, 17, 17, 17, 20, 17, 17, 17, 17, 17, 17, 17]
```

(9) Sigmoid + Sigmoid + Quadractic + L2 + lambda(3.0)

```
In [14]: net2 = network2.load_network("iris-423.dat")
# Set hyper-parameter values individually after the network
net2.set_parameters(cost=network2.QuadraticCost, act_hidden=network2.Sigmoid, act_output

train_error, train_correct, test_error, test_correct = net2.SGD(iris_train,
                                                                15, 10, 1.0,
                                                                evaluation_data=iris_te
                                                                monitor_evaluation_cost
                                                                monitor_evaluation_accur
                                                                monitor_training_cost=Tr
                                                                monitor_training_accuracy=T

print('Training error: \n %s\n' %format_res(train_error))
print('Training correct: \n %s\n' %train_correct)
print('Test error: \n %s\n' %format_res(test_error))
print('Test correct: \n %s' %test_correct)
```

```
Epoch 0 training complete
Cost on training data: 0.3392795030659105
Accuracy on training data: 21 / 95
Cost on evaluation data: 0.34444833747414394
Accuracy on evaluation data: 15 / 55
```

```
Epoch 1 training complete
Cost on training data: 0.3332680387432046
Accuracy on training data: 33 / 95
Cost on evaluation data: 0.3357508215659017
Accuracy on evaluation data: 17 / 55
```

```
Epoch 2 training complete
Cost on training data: 0.33169864345910943
Accuracy on training data: 33 / 95
Cost on evaluation data: 0.33331819877313307
Accuracy on evaluation data: 17 / 55
```

```
Epoch 3 training complete
Cost on training data: 0.3280047701277544
Accuracy on training data: 33 / 95
```

Cost on evaluation data: 0.3293129769432788
Accuracy on evaluation data: 17 / 55

Epoch 4 training complete
Cost on training data: 0.3105024143021552
Accuracy on training data: 65 / 95
Cost on evaluation data: 0.311452504680719
Accuracy on evaluation data: 35 / 55

Epoch 5 training complete
Cost on training data: 0.2822226213778409
Accuracy on training data: 64 / 95
Cost on evaluation data: 0.28266644748693787
Accuracy on evaluation data: 39 / 55

Epoch 6 training complete
Cost on training data: 0.2618530851998504
Accuracy on training data: 65 / 95
Cost on evaluation data: 0.2622221163236103
Accuracy on evaluation data: 40 / 55

Epoch 7 training complete
Cost on training data: 0.24637620748622577
Accuracy on training data: 64 / 95
Cost on evaluation data: 0.24693122708939252
Accuracy on evaluation data: 40 / 55

Epoch 8 training complete
Cost on training data: 0.23397287354343002
Accuracy on training data: 65 / 95
Cost on evaluation data: 0.234842439155326
Accuracy on evaluation data: 41 / 55

Epoch 9 training complete
Cost on training data: 0.2240389321795399
Accuracy on training data: 67 / 95
Cost on evaluation data: 0.22523745295521982
Accuracy on evaluation data: 41 / 55

Epoch 10 training complete
Cost on training data: 0.21604989548513004
Accuracy on training data: 70 / 95
Cost on evaluation data: 0.2175515310927702
Accuracy on evaluation data: 44 / 55

Epoch 11 training complete
Cost on training data: 0.20957620155325693
Accuracy on training data: 73 / 95
Cost on evaluation data: 0.21134700807259035
Accuracy on evaluation data: 47 / 55

Epoch 12 training complete
Cost on training data: 0.2042808031846783
Accuracy on training data: 81 / 95
Cost on evaluation data: 0.20628846293823358
Accuracy on evaluation data: 52 / 55

Epoch 13 training complete
Cost on training data: 0.19990418595499365
Accuracy on training data: 94 / 95
Cost on evaluation data: 0.20212044282632813
Accuracy on evaluation data: 54 / 55

Epoch 14 training complete
Cost on training data: 0.19624754621498341
Accuracy on training data: 65 / 95

Cost on evaluation data: 0.19864854335473608
Accuracy on evaluation data: 35 / 55

Training error:

```
[0.3392795030659105,  
0.3332680387432046,  
0.33169864345910943,  
0.3280047701277544,  
0.3105024143021552,  
0.2822226213778409,  
0.2618530851998504,  
0.24637620748622577,  
0.23397287354343002,  
0.2240389321795399,  
0.21604989548513004,  
0.20957620155325693,  
0.2042808031846783,  
0.19990418595499365,  
0.19624754621498341]
```

Training correct:

```
[21, 33, 33, 33, 65, 64, 65, 64, 65, 67, 70, 73, 81, 94, 65]
```

Test error:

```
[0.3392795030659105,  
0.3332680387432046,  
0.33169864345910943,  
0.3280047701277544,  
0.3105024143021552,  
0.2822226213778409,  
0.2618530851998504,  
0.24637620748622577,  
0.23397287354343002,  
0.2240389321795399,  
0.21604989548513004,  
0.20957620155325693,  
0.2042808031846783,  
0.19990418595499365,  
0.19624754621498341]
```

Test correct:

```
[15, 17, 17, 17, 35, 39, 40, 40, 41, 41, 44, 47, 52, 54, 35]
```

(10) Sigmoid + Sigmoid + Quadractic + L1 + lambda(3.0)

```
In [15]: net2 = network2.load_network("iris-423.dat")  
# Set hyper-parameter values individually after the network  
net2.set_parameters(cost=network2.QuadraticCost, act_hidden=network2.Sigmoid, act_output  
  
train_error, train_correct, test_error, test_correct = net2.SGD(iris_train,  
                                                                15, 10, 1.0,  
                                                                evaluation_data=iris_te  
                                                                monitor_evaluation_cost  
                                                                monitor_evaluation_accur  
                                                                monitor_training_cost=Tr  
                                                                monitor_training_accuracy=T  
  
print('Training error: \n %s\n' %format_res(train_error))  
print('Training correct: \n %s\n' %train_correct)  
print('Test error: \n %s\n' %format_res(test_error))  
print('Test correct: \n %s' %test_correct)
```

Epoch 0 training complete
Cost on training data: 0.4123917551703461
Accuracy on training data: 4 / 95
Cost on evaluation data: 0.4244112968465798
Accuracy on evaluation data: 3 / 55

Epoch 1 training complete
Cost on training data: 0.4123917551400401
Accuracy on training data: 4 / 95
Cost on evaluation data: 0.42441129682437867
Accuracy on evaluation data: 3 / 55

Epoch 2 training complete
Cost on training data: 0.4123917551400401
Accuracy on training data: 4 / 95
Cost on evaluation data: 0.42441129682437867
Accuracy on evaluation data: 3 / 55

Epoch 3 training complete
Cost on training data: 0.4123917551400401
Accuracy on training data: 4 / 95
Cost on evaluation data: 0.42441129682437867
Accuracy on evaluation data: 3 / 55

Epoch 4 training complete
Cost on training data: 0.4123917551400401
Accuracy on training data: 4 / 95
Cost on evaluation data: 0.42441129682437867
Accuracy on evaluation data: 3 / 55

Epoch 5 training complete
Cost on training data: 0.4123917551400401
Accuracy on training data: 4 / 95
Cost on evaluation data: 0.42441129682437867
Accuracy on evaluation data: 3 / 55

Epoch 6 training complete
Cost on training data: 0.4123917551400401
Accuracy on training data: 4 / 95
Cost on evaluation data: 0.42441129682437867
Accuracy on evaluation data: 3 / 55

Epoch 7 training complete
Cost on training data: 0.4123917551400401
Accuracy on training data: 4 / 95
Cost on evaluation data: 0.42441129682437867
Accuracy on evaluation data: 3 / 55

Epoch 8 training complete
Cost on training data: 0.4123917551400401
Accuracy on training data: 4 / 95
Cost on evaluation data: 0.42441129682437867
Accuracy on evaluation data: 3 / 55

Epoch 9 training complete
Cost on training data: 0.4123917551400401
Accuracy on training data: 4 / 95
Cost on evaluation data: 0.42441129682437867
Accuracy on evaluation data: 3 / 55

Epoch 10 training complete
Cost on training data: 0.4123917551400401
Accuracy on training data: 4 / 95
Cost on evaluation data: 0.42441129682437867
Accuracy on evaluation data: 3 / 55

Epoch 11 training complete
Cost on training data: 0.4123917551400401
Accuracy on training data: 4 / 95
Cost on evaluation data: 0.42441129682437867
Accuracy on evaluation data: 3 / 55

Epoch 12 training complete
Cost on training data: 0.4123917551400401
Accuracy on training data: 4 / 95
Cost on evaluation data: 0.42441129682437867
Accuracy on evaluation data: 3 / 55

Epoch 13 training complete
Cost on training data: 0.4123917551400401
Accuracy on training data: 4 / 95
Cost on evaluation data: 0.42441129682437867
Accuracy on evaluation data: 3 / 55

Epoch 14 training complete
Cost on training data: 0.4123917551400401
Accuracy on training data: 4 / 95
Cost on evaluation data: 0.42441129682437867
Accuracy on evaluation data: 3 / 55

Training error:
[0.4123917551703461,
0.4123917551400401,
0.4123917551400401,
0.4123917551400401,
0.4123917551400401,
0.4123917551400401,
0.4123917551400401,
0.4123917551400401,
0.4123917551400401,
0.4123917551400401,
0.4123917551400401,
0.4123917551400401,
0.4123917551400401,
0.4123917551400401,
0.4123917551400401,
0.4123917551400401,
0.4123917551400401,
0.4123917551400401]

Training correct:
[4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4]

Test error:
[0.4123917551703461,
0.4123917551400401,
0.4123917551400401,
0.4123917551400401,
0.4123917551400401,
0.4123917551400401,
0.4123917551400401,
0.4123917551400401,
0.4123917551400401,
0.4123917551400401,
0.4123917551400401,
0.4123917551400401,
0.4123917551400401,
0.4123917551400401,
0.4123917551400401,
0.4123917551400401,
0.4123917551400401,
0.4123917551400401]

Test correct:
[3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3]

Experiments for dropout

(11) Sigmoid + Sigmoid + Quadractic + dropout(0.1)

```
In [16]: net2 = network2.load_network("iris4-20-7-3.dat")
# Set hyper-parameter values individually after the network
net2.set_parameters(cost=network2.QuadraticCost,
                    act_hidden=network2.Sigmoid,
                    act_output=network2.Sigmoid,
                    dropoutpercent=0.1
                    )

train_error, train_correct, test_error, test_correct = net2.SGD(iris_train,
                                                                15, 10, 1.0,
                                                                evaluation_data=iris_te
                                                                monitor_evaluation_cost
                                                                monitor_evaluation_accur
                                                                monitor_training_cost=Tr
                                                                monitor_training_accuracy=T

print('Training error: \n %s\n' %format_res(train_error))
print('Training correct: \n %s\n' %train_correct)
print('Test error: \n %s\n' %format_res(test_error))
print('Test correct: \n %s' %test_correct)
```

Epoch 0 training complete
Cost on training data: 0.3336562793195007
Accuracy on training data: 33 / 95
Cost on evaluation data: 0.33374796857762645
Accuracy on evaluation data: 17 / 55

Epoch 1 training complete
Cost on training data: 0.33322942032716346
Accuracy on training data: 33 / 95
Cost on evaluation data: 0.33374738250400454
Accuracy on evaluation data: 17 / 55

Epoch 2 training complete
Cost on training data: 0.3330924835386632
Accuracy on training data: 33 / 95
Cost on evaluation data: 0.33391464524861886
Accuracy on evaluation data: 17 / 55

Epoch 3 training complete
Cost on training data: 0.33296986261605255
Accuracy on training data: 33 / 95
Cost on evaluation data: 0.33404073632828696
Accuracy on evaluation data: 17 / 55

Epoch 4 training complete
Cost on training data: 0.332759584338502
Accuracy on training data: 33 / 95
Cost on evaluation data: 0.3339407005146947
Accuracy on evaluation data: 17 / 55

Epoch 5 training complete
Cost on training data: 0.3326345209679276
Accuracy on training data: 33 / 95
Cost on evaluation data: 0.3335124332352579
Accuracy on evaluation data: 17 / 55

Epoch 6 training complete
Cost on training data: 0.3324257813411286
Accuracy on training data: 33 / 95
Cost on evaluation data: 0.3333507991473653
Accuracy on evaluation data: 17 / 55

Epoch 7 training complete
Cost on training data: 0.33220527282392487
Accuracy on training data: 33 / 95
Cost on evaluation data: 0.3336219344575466
Accuracy on evaluation data: 17 / 55

Epoch 8 training complete
Cost on training data: 0.33183481440417845
Accuracy on training data: 33 / 95
Cost on evaluation data: 0.3327221959673018
Accuracy on evaluation data: 17 / 55

Epoch 9 training complete
Cost on training data: 0.33146471678041634
Accuracy on training data: 33 / 95
Cost on evaluation data: 0.33244192793605737
Accuracy on evaluation data: 17 / 55

Epoch 10 training complete
Cost on training data: 0.330976806186165
Accuracy on training data: 33 / 95
Cost on evaluation data: 0.33189262552278004
Accuracy on evaluation data: 17 / 55

Epoch 11 training complete
Cost on training data: 0.33013090265519074
Accuracy on training data: 33 / 95
Cost on evaluation data: 0.33053029588912813
Accuracy on evaluation data: 17 / 55

Epoch 12 training complete
Cost on training data: 0.32895857766118675
Accuracy on training data: 33 / 95
Cost on evaluation data: 0.32973958115015894
Accuracy on evaluation data: 17 / 55

Epoch 13 training complete
Cost on training data: 0.3274752083541871
Accuracy on training data: 33 / 95
Cost on evaluation data: 0.32854779764748265
Accuracy on evaluation data: 17 / 55

Epoch 14 training complete
Cost on training data: 0.3256517718358024
Accuracy on training data: 33 / 95
Cost on evaluation data: 0.32616590363174364
Accuracy on evaluation data: 17 / 55

Training error:
[0.3336562793195007,
0.33322942032716346,
0.3330924835386632,
0.33296986261605255,
0.332759584338502,
0.3326345209679276,
0.3324257813411286,
0.33220527282392487,
0.33183481440417845,
0.33146471678041634,

```
0.330976806186165,  
0.33013090265519074,  
0.32895857766118675,  
0.3274752083541871,  
0.3256517718358024]
```

Training correct:

```
[33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33, 33]
```

Test error:

```
[0.3336562793195007,  
0.33322942032716346,  
0.3330924835386632,  
0.33296986261605255,  
0.332759584338502,  
0.3326345209679276,  
0.3324257813411286,  
0.33220527282392487,  
0.33183481440417845,  
0.33146471678041634,  
0.330976806186165,  
0.33013090265519074,  
0.32895857766118675,  
0.3274752083541871,  
0.3256517718358024]
```

Test correct:

```
[17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17, 17]
```

(12) Sigmoid + Sigmoid + Quadratic + dropout(0.5)

```
In [17]: net2 = network2.load_network("iris4-20-7-3.dat")  
# Set hyper-parameter values individually after the network  
net2.set_parameters(cost=network2.QuadraticCost,  
                    act_hidden=network2.Sigmoid,  
                    act_output=network2.Sigmoid,  
                    dropoutpercent=0.5  
                    )  
  
train_error, train_correct, test_error, test_correct = net2.SGD(iris_train,  
                                                                15, 10, 1.0,  
                                                                evaluation_data=iris_te  
                                                                monitor_evaluation_cost  
                                                                monitor_evaluation_accur  
                                                                monitor_training_cost=Tr  
                                                                monitor_training_accuracy=T  
                                                                )  
  
print('Training error: \n %s\n' %format_res(train_error))  
print('Training correct: \n %s\n' %train_correct)  
print('Test error: \n %s\n' %format_res(test_error))  
print('Test correct: \n %s' %test_correct)
```

Epoch 0 training complete

Cost on training data: 0.333883101360267

Accuracy on training data: 33 / 95

Cost on evaluation data: 0.33637254416301443

Accuracy on evaluation data: 17 / 55

Epoch 1 training complete

Cost on training data: 0.3331862874368164

Accuracy on training data: 33 / 95

Cost on evaluation data: 0.33417462654444663
Accuracy on evaluation data: 17 / 55

Epoch 2 training complete
Cost on training data: 0.3329786901639483
Accuracy on training data: 33 / 95
Cost on evaluation data: 0.33489370299934934
Accuracy on evaluation data: 17 / 55

Epoch 3 training complete
Cost on training data: 0.3327175803000471
Accuracy on training data: 33 / 95
Cost on evaluation data: 0.3336991809123039
Accuracy on evaluation data: 17 / 55

Epoch 4 training complete
Cost on training data: 0.3321986444401344
Accuracy on training data: 33 / 95
Cost on evaluation data: 0.33394055738693246
Accuracy on evaluation data: 17 / 55

Epoch 5 training complete
Cost on training data: 0.33272344606066
Accuracy on training data: 30 / 95
Cost on evaluation data: 0.3316918668676193
Accuracy on evaluation data: 20 / 55

Epoch 6 training complete
Cost on training data: 0.33231172101064377
Accuracy on training data: 33 / 95
Cost on evaluation data: 0.332840474845101
Accuracy on evaluation data: 17 / 55

Epoch 7 training complete
Cost on training data: 0.3323523113659325
Accuracy on training data: 33 / 95
Cost on evaluation data: 0.3335134381459856
Accuracy on evaluation data: 17 / 55

Epoch 8 training complete
Cost on training data: 0.33195451433280393
Accuracy on training data: 33 / 95
Cost on evaluation data: 0.33386719596853587
Accuracy on evaluation data: 17 / 55

Epoch 9 training complete
Cost on training data: 0.33159634256548565
Accuracy on training data: 33 / 95
Cost on evaluation data: 0.3329990288152973
Accuracy on evaluation data: 17 / 55

Epoch 10 training complete
Cost on training data: 0.33155321130751486
Accuracy on training data: 33 / 95
Cost on evaluation data: 0.3332476708640554
Accuracy on evaluation data: 17 / 55

Epoch 11 training complete
Cost on training data: 0.33137683189435946
Accuracy on training data: 33 / 95
Cost on evaluation data: 0.3333374989591965
Accuracy on evaluation data: 17 / 55

Epoch 12 training complete
Cost on training data: 0.33159679909119627
Accuracy on training data: 32 / 95

Cost on evaluation data: 0.33291829817232405
Accuracy on evaluation data: 18 / 55

Epoch 13 training complete
Cost on training data: 0.3303755700010149
Accuracy on training data: 32 / 95
Cost on evaluation data: 0.3317501866458422
Accuracy on evaluation data: 18 / 55

Epoch 14 training complete
Cost on training data: 0.328966376933048
Accuracy on training data: 32 / 95
Cost on evaluation data: 0.3289372102120124
Accuracy on evaluation data: 18 / 55

Training error:
[0.333883101360267,
0.3331862874368164,
0.3329786901639483,
0.3327175803000471,
0.3321986444401344,
0.33272344606066,
0.33231172101064377,
0.3323523113659325,
0.33195451433280393,
0.33159634256548565,
0.33155321130751486,
0.33137683189435946,
0.33159679909119627,
0.3303755700010149,
0.328966376933048]

Training correct:
[33, 33, 33, 33, 33, 30, 33, 33, 33, 33, 33, 33, 32, 32, 32]

Test error:
[0.333883101360267,
0.3331862874368164,
0.3329786901639483,
0.3327175803000471,
0.3321986444401344,
0.33272344606066,
0.33231172101064377,
0.3323523113659325,
0.33195451433280393,
0.33159634256548565,
0.33155321130751486,
0.33137683189435946,
0.33159679909119627,
0.3303755700010149,
0.328966376933048]

Test correct:
[17, 17, 17, 17, 17, 20, 17, 17, 17, 17, 17, 17, 18, 18, 18]