

INF3490 Biologically Inspired Computing

Stephen Simei Kimogol, Username: stephesk

October 30, 2016

1 Running the program

I've used Python 2.7.12 in this assignment. The assignment has three files: `main.py`, `movements.py` and `mlp.py`. The file **`movements.py`** reads data file and split it into training, validation and test sets. **`mlp.py`** contains the necessary methods for training the network. The file **`main.py`** creates the mlp and runs the training of the network on the given train and validation data sets.

To run the program use: `python main.py`

2 The program

Parameters

beta = Sigmoid activation function

Sigmoid activation function = $1 / (1 + np.exp(-self.beta * input))$

Derivative of sigmoid activation function = $outputfromforward * (1 - outputfromforward)$

eta = the learning rate

bias = The bias input

2.1 Stopping approach

In *earlystopping* function, the goal is to train the network well such that it can generalise well. The training data sets are used for training while the validation data sets are used to detect overfitting. This is achieved by calling *train* function which does the back propagation algorithm and this is done until no further error changes is detected.

2.2 Train

In training, *forward* function is used to generate the activation on input and output nodes. Depending on the number of iterations, compute the activation using the *sigmoid* function. Likewise, the error is calculated using the derivative of *sigmoid* function and sent backward through the network. The weights of the hidden and output layers are updated based on the computed deltas.

2.3 Result

Few number of nodes e.g 2 are not sufficient to train the mlp. The results improve with increased number of nodes to 6 and 8, and the performance reduces again at 12 nodes. The network seems to classify well with 8 nodes.

From the matrix generated after several runs, motion class 7 is often mistaken for class 6

The images below show a sample of the confusion matrix and the percentages. However, this changes with every time the program is run

Number of hidden nodes = 2

Confusion matrix

```
[[ 0.  0.  2.  5.  0.  0.  0.  1.]
 [ 0.  3.  0.  1.  7.  0.  1.  2.]
 [ 0.  0. 13.  0.  0.  0.  0.  1.]
 [ 0.  1.  0.  5.  2.  0.  0.  7.]
 [ 0.  0.  0.  0. 10.  0.  0.  0.]
 [ 0.  0.  1.  0.  1. 14.  0.  0.]
 [ 0.  0.  2.  0.  0.  7.  6.  0.]
 [ 0.  0.  0.  0.  0.  0.  0. 19.]]
```

```
Class Prediction Percentages 1 : 0%
Class Prediction Percentages 2 : 75.0%
Class Prediction Percentages 3 : 72.2222222222%
Class Prediction Percentages 4 : 45.4545454545%
Class Prediction Percentages 5 : 50.0%
Class Prediction Percentages 6 : 66.6666666667%
Class Prediction Percentages 7 : 85.7142857143%
Class Prediction Percentages 8 : 63.3333333333%
The average percentage is 57.2988816739%
```

Number of hidden nodes = 6

Confusion matrix

```
[[ 7.  0.  0.  0.  0.  0.  0.  1.]
 [ 0. 14.  0.  0.  0.  0.  0.  0.]
 [ 1.  0. 13.  0.  0.  0.  0.  0.]
 [ 0.  0.  0. 14.  1.  0.  0.  0.]
 [ 0.  0.  0.  0. 10.  0.  0.  0.]
 [ 0.  0.  0.  0.  1. 15.  0.  0.]
 [ 0.  0.  0.  0.  0.  1. 14.  0.]
 [ 0.  0.  0.  2.  0.  0.  0. 17.]]
```

```
Class Prediction Percentages 1 : 87.5%
Class Prediction Percentages 2 : 100.0%
Class Prediction Percentages 3 : 100.0%
Class Prediction Percentages 4 : 87.5%
Class Prediction Percentages 5 : 83.3333333333%
Class Prediction Percentages 6 : 93.75%
Class Prediction Percentages 7 : 100.0%
Class Prediction Percentages 8 : 94.4444444444%
The average percentage is 93.3159722222%
```

Number of hidden nodes = 8

Confusion matrix

```
[[ 7.  0.  0.  0.  0.  0.  0.  1.]
 [ 0. 14.  0.  0.  0.  0.  0.  0.]
 [ 0.  0. 13.  1.  0.  0.  0.  0.]
 [ 0.  0.  0. 15.  0.  0.  0.  0.]
 [ 0.  0.  0.  0. 10.  0.  0.  0.]
 [ 0.  0.  0.  0.  1. 15.  0.  0.]
 [ 0.  0.  0.  0.  0.  1. 14.  0.]
 [ 0.  0.  0.  1.  0.  0.  0. 18.]]
```

Class Prediction Percentages 1 : 100.0%
Class Prediction Percentages 2 : 100.0%
Class Prediction Percentages 3 : 100.0%
Class Prediction Percentages 4 : 88.2352941176%
Class Prediction Percentages 5 : 90.9090909091%
Class Prediction Percentages 6 : 93.75%
Class Prediction Percentages 7 : 100.0%
Class Prediction Percentages 8 : 94.7368421053%
The average percentage is 95.9539033915%

Number of hidden nodes = 12

Confusion matrix

```
[[ 0.  2.  2.  0.  3.  0.  0.  1.]
 [ 0. 14.  0.  0.  0.  0.  0.  0.]
 [ 0.  0. 14.  0.  0.  0.  0.  0.]
 [ 0.  0.  1. 13.  1.  0.  0.  0.]
 [ 0.  0.  0.  0. 10.  0.  0.  0.]
 [ 0.  0.  0.  0.  1. 15.  0.  0.]
 [ 0.  0.  0.  0.  0.  1. 14.  0.]
 [ 0.  0.  0.  1.  0.  0.  0. 18.]]
```

Class Prediction Percentages 1 : 0%
Class Prediction Percentages 2 : 87.5%
Class Prediction Percentages 3 : 82.3529411765%
Class Prediction Percentages 4 : 92.8571428571%
Class Prediction Percentages 5 : 66.6666666667%
Class Prediction Percentages 6 : 93.75%
Class Prediction Percentages 7 : 100.0%
Class Prediction Percentages 8 : 94.7368421053%
The average percentage is 77.2329491007%