Machine Learning Club 2.08.17

Classifying MNIST Data (Part 1) Neural Networks Assignment

Your end task is to create a neural network that classifying the MNIST database as best as possible. You will not get that far this week. Your task this week is to set up the data environment and write the feedforward method.

Parameters

- Have numpy installed (pip3 install numpy)
- Do not use a library (Scikit-Learn, PyBrain, TensorFlow, etc.)
- Write in python3
- Submit your code by Sunday midnight to `tjmachinelearning.com/submit`

MNIST Database

The MNIST database is large database of handwritten digits. The data consists of 70,000 28x28 grayscale images of digits.

Figure 1. *Digits from the MNIST database.*

The first part of the database consists of 50,000 images for training your neural network. The second part of the database consists of 10,000 images for validation of hyperparameters. The third part contains 10,000 images for testing. Your rankings in this competition will be based on your score for the testing images.

Loading Data

Download and unzip the data and code from

`tjmachinelearning.com/resources/mnist.zip`

Open command prompt and navigate to the unzipped mnist folder. Run python and type:

```
>>> import mnist_loader
>>> training_data, validation_data, test_data = \
... mnist_loader.load_data_wrapper()
```

Ensure this does not return any errors. If it does, troubleshoot and then contact us at `tjmachinelearning@gmail.com`

If there are no errors, proceed.

Writing the Neural Network

Each sigmoid neuron of the network has a weight and a bias, which will change as the network learns. The weights and biases are stored as lists of numpy matrices.

Example: `net.weights[1]` is the Numpy matrix storing the weights connecting the second and third layers of neurons.

Why do we store the weights this way? It makes vector activations extremely simple.

Each layer follows the equation: $a' = \sigma(wa+b)$.

In other words, to go from layer n to layer n+1, multiply the vector a by the weight vector w, and add the bias. Then apply the sigmoid function (σ) to every element in this new vector matrix.

Initializing the Neural Network

We wish to set the biases randomly, sampled from a Gaussian distribution.

The line:

```
self.biases = [np.random.randn(y, 1) for y in sizes[1:]]
```

uses np.random.randn, which does the Gaussian sampling for us.

We repeat this for the weights.

An Explanation of the zip() function

```
a = []*20
b = []*20
c = zip(a, b)
```

Here, zip() creates a list of length 20 with each element as a tuple.

Feedforward Method

Write the method in `Network.py` that returns the feedforward network's output for a given input. This method should be less than 10 lines, but it is important you understand each one. You may write a helper method if you wish.

Testing and Submission

There is no testing or ranking this week. This assignment (Part 1) is purely completion. When you finish, name your code `nn1_lastname_firstname.py` and turn it in.

Next week you will write the code for backpropagation and stoichiometric descent.