

Neural Network Training – Sanity checks

Check initialization

Make sure the observed loss is what you expect it to be for a random prediction.

E.g. for MNIST (10 classes) we expect a probability of 0.1 for each class. With Softmax objective function the expected loss is $-\log(0.1) = 2.303$

```
32/60000 [.....] - ETA: 10:17 - loss: 2.3875 - acc: 0.0938
```

Regularization check

Overfit on a small dataset

Neural Network Training – Sanity checks

Check initialization

Regularization check

Increasing the regularization strength should increase the training loss. Such as L2 parameter constrain.

```
32/60000 [.....] - ETA: 14:24 - loss: 8.7382 - acc: 0.1250
```

Overfit on a small dataset

Neural Network Training – Sanity checks

Check initialization
Regularization check

Overfit on a small dataset

Train your model on a few samples (e.g. 50), you should reach a training loss of zero. Do not use regularization methods during this test.

```
model.fit(x_train[0:50], y_train[0:50], epochs=1000, shuffle=True)
```

```
Epoch 1000/1000
```

```
32/50 [=====>.....] - ETA: 0s - loss: 1.5039e-05 - acc: 1.0000
```

```
50/50 [=====] - 0s 350us/step - loss: 1.6422e-05 - acc: 1.0000
```