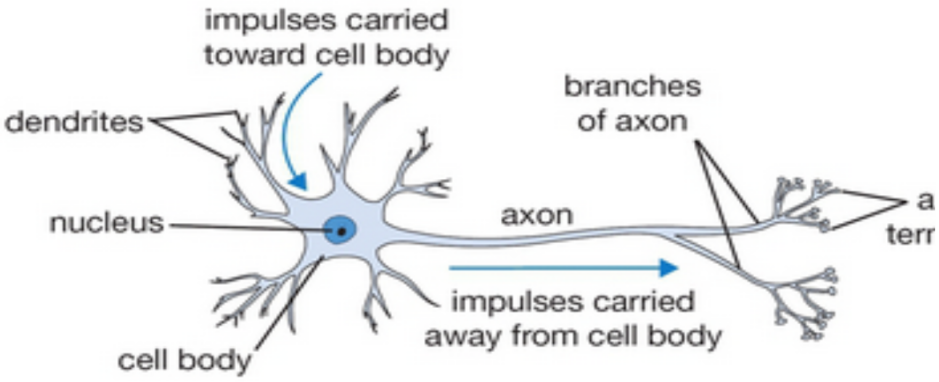
Université de Bordeaux

Deep Learning

Deep Neural Networks (lab2)



Abderrahim Serghini

5 octobre 2018

One Neuron : Following the course, we first used the architecture of one neuron :

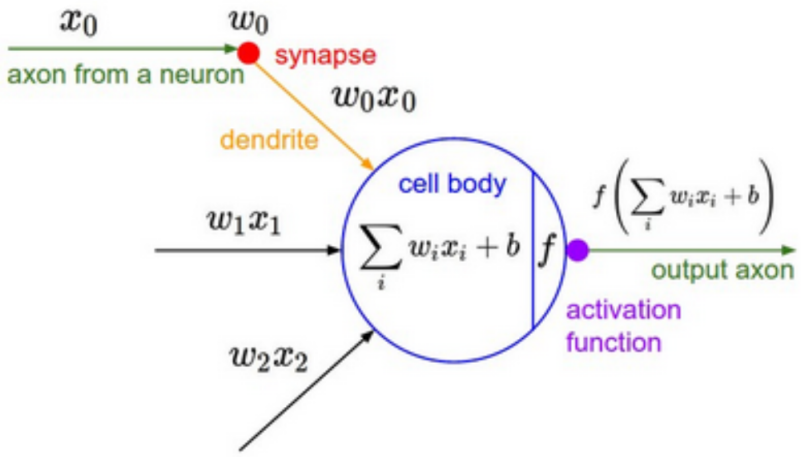


FIGURE 1 – A simple architecture of a neural network

Where we take the pixels of the image as the input, the weights are taken as a random vector while we f is the sigmoid functions :

$$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}}$$

And Y as the output as $\sigma(w^T x + b)$

Forward Propagation : The mathematical expression for the loss function using the cross entropy is :

$$L(x) = -\sum Y log(\hat{Y}) + \sum (1 - Y) log(1 - \hat{Y})$$

Backward Propagation : The derivative of L in accordance to its component W (using the chain rule, and m being the number of examples)

$$\frac{dL}{dW} = \frac{1}{m}(\hat{Y} - Y).X^T$$

We also need the derivative of b in which we get using the same rule :

$$db = \frac{1}{m}\sum(\hat{Y} - Y)$$

Results : Testing the model for 500 epochs we get the following result :

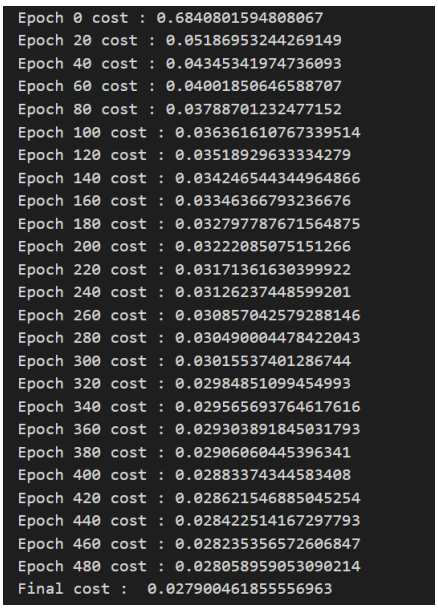


FIGURE 2 – Test using the one Neuron model, the cost being calculated each 20th iteration

A Hidden layer : With the same activation function and the loss function, we can add another layer and calculate the derivative once again with the same method. We get then the following results :

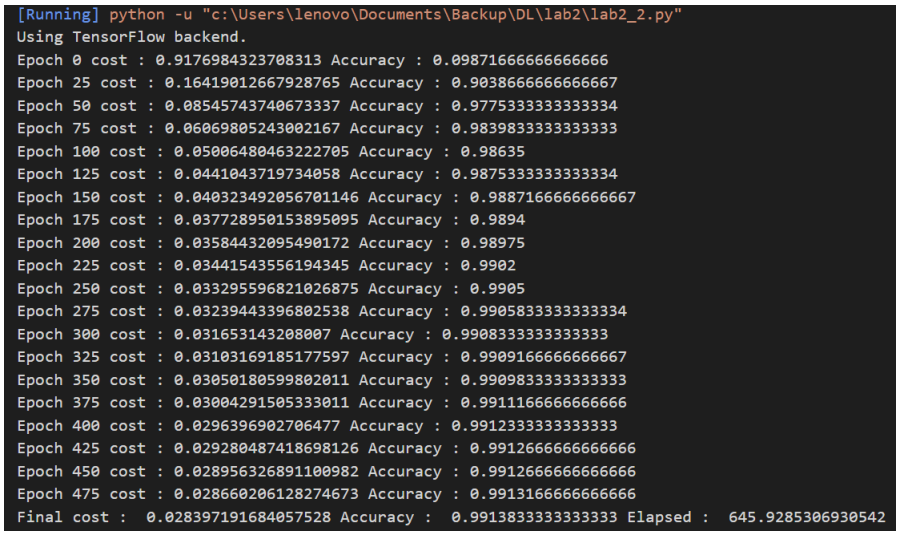


FIGURE 3 – Test adding a Hidden layer of 64 elements achieving a 99,13 % accuracy

The results do not change drastically when adding more elements to the hidden layer :

```
[Running] python -u "c:\Users\lenovo\Documents\Backup\DL\lab2\lab2_2.py"
Using TensorFlow backend.
Epoch 0 cost : 0.6662941976132445 Accuracy : 0.9012833333333333
Epoch 25 cost : 0.10248775920054211 Accuracy : 0.9697333333333333
Epoch 50 cost : 0.06325975019280071 Accuracy : 0.9827
Epoch 75 cost : 0.05044400373242353 Accuracy : 0.9859333333333333
Epoch 100 cost : 0.043872647135917264 Accuracy : 0.9873666666666666
Epoch 125 cost : 0.03990044870013563 Accuracy : 0.98855
Epoch 150 cost : 0.037255029502269964 Accuracy : 0.9893
Epoch 175 cost : 0.035371103848745 Accuracy : 0.9898666666666667
Epoch 200 cost : 0.03396259620316097 Accuracy : 0.9901666666666666
Epoch 225 cost : 0.03287034100654196 Accuracy : 0.9904833333333334
Epoch 250 cost : 0.03199872367742637 Accuracy : 0.99075
Epoch 275 cost : 0.031286594641276484 Accuracy : 0.99085
Epoch 300 cost : 0.030692831305238852 Accuracy : 0.9908666666666667
Epoch 325 cost : 0.03018865875365686 Accuracy : 0.9910833333333333
Epoch 350 cost : 0.029753304050867943 Accuracy : 0.9911166666666666
Epoch 375 cost : 0.02937140001688888 Accuracy : 0.9911
Epoch 400 cost : 0.02903136171671551 Accuracy : 0.9911833333333333
Epoch 425 cost : 0.028724332070374902 Accuracy : 0.9912333333333333
Epoch 450 cost : 0.028443474157282843 Accuracy : 0.9913333333333333
Epoch 475 cost : 0.02818348063370941 Accuracy : 0.99145
Final cost :  0.027949673620899257 Accuracy :  0.9915166666666667 Elapsed :  879.5334870815277
```

FIGURE 4 – Test adding a Hidden layer of 128 elements achieving a 99,15 % accuracy

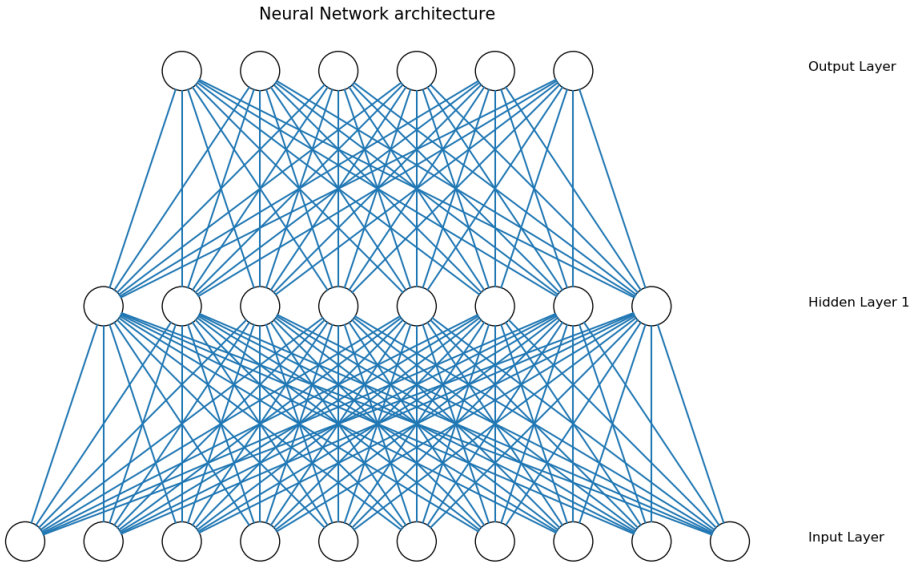


FIGURE 5 – Architecture of the neural network

Multi neural network Above is the architecture of our neural network, composing of an input of 728 elements, A hidden layer of 64 elements and an output of 10 elements (representing the hot encoding of the numbers ranging from 0 to 9)

After running the test we get the following result :

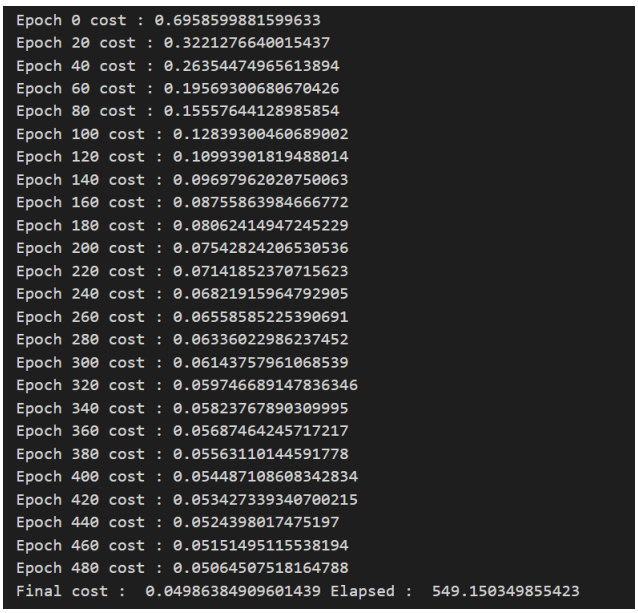


FIGURE 6 – test of multi neural network