



First Course

Liangjie Cao

VISION@OUC

July 15, 2018

1 Computation Graph

Professor Wu told us to use graph to express the logistic process. One computation graph includes the forward process and the backward process. And he show a complete process to us in figure 1. Let us see some details. The input a , b and c are 5, 3 and 2 at first. When we plus 0.001 to a . Then v will change to 11.001 and J will change to 33.003. Just a little bit change can change the output. And from the backward process, it's a gradient descent process. As a Chinese student, the derivatives are familiar to us.

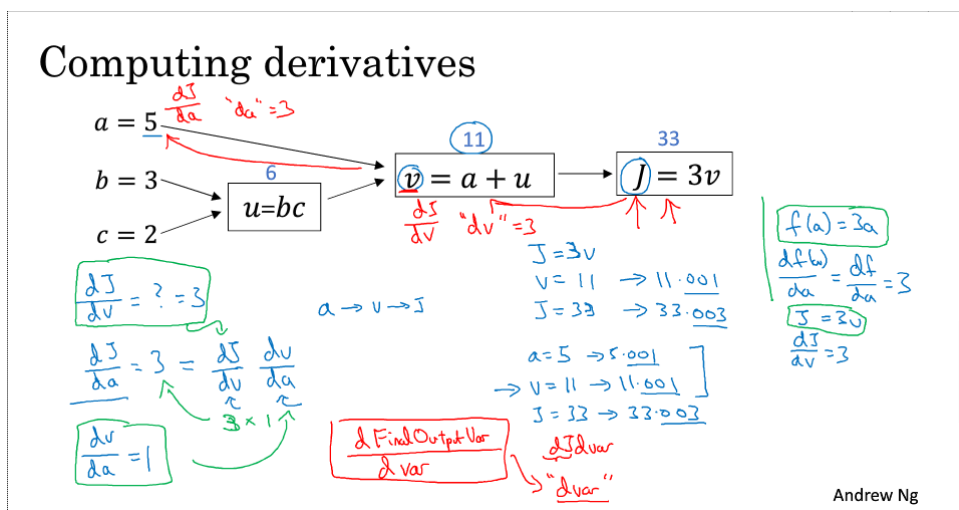


Figure 1: The computation graph

2 Vectorization

When training m of the data, we can use for loop to make it. Professor Wu tells us to use the vectorization process save time. The typical vectorization process in python code is like sigmoid function $z = np.dot(w.T, x) + b$. Where np is an import of numpy, w and x are matrixs and np.dot is a process to compute two matrix. Then b is broadcasting to a vector. This process can save 30% time to the for loop process.

3 Neural Network

Then come to the third part, professor Wu says we can form a neural network by stacking together a lot of little sigmoid units. As we can see from figure 3, it's a two layers of neural network. The input layer cannot be count into the layers. Then there're some new symbols to recognize. We can see the $z_1^{[1]}$ in figure 3, $[1]$ means this is in the second layer and 1 means this is the first unit. This can make m samples compute at the same time. Activation fuctions is like the sigmoid fuctions and other typical functions. ReLu, tanh and leaky ReLu is some different activate function. All of these functions are non-linear fuctions. It means that if we use linear function, whatever layers do we have, we can only get the linear output at last.

The initial process may cause some probelm. If we make the weights too large, when we compute the functions like sigmoid function. W will cause too large z, what's more, the gradient descent process make a little sense in the process.

Then it come the deep neural network(in figure 3), the main probelm is to make sure the demision of the compute matrix is correct. Professor Wu gives us his own method to check the demensions. When the compute unit is like sigmoid function, $z^{(1)} = W^{(1)}x + b^{(1)}$. W's row is equal to the backward units' row,

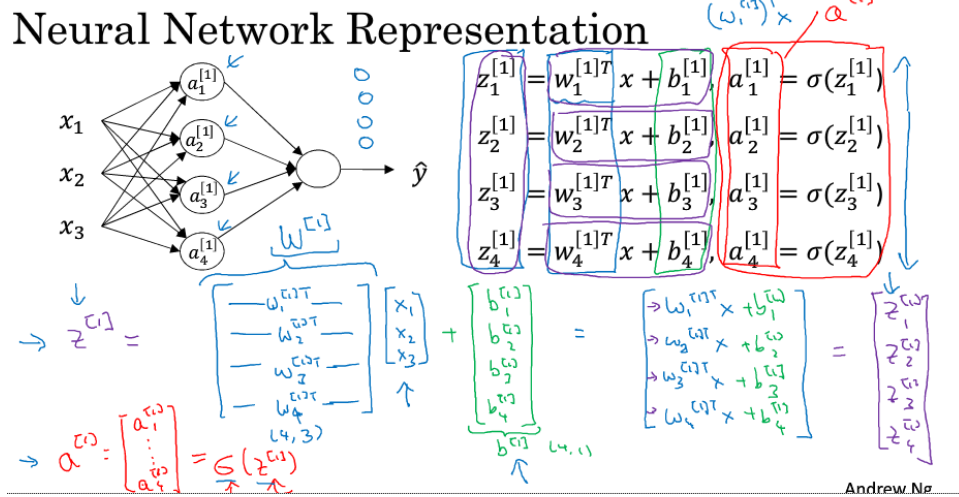


Figure 2: Shallow neural network

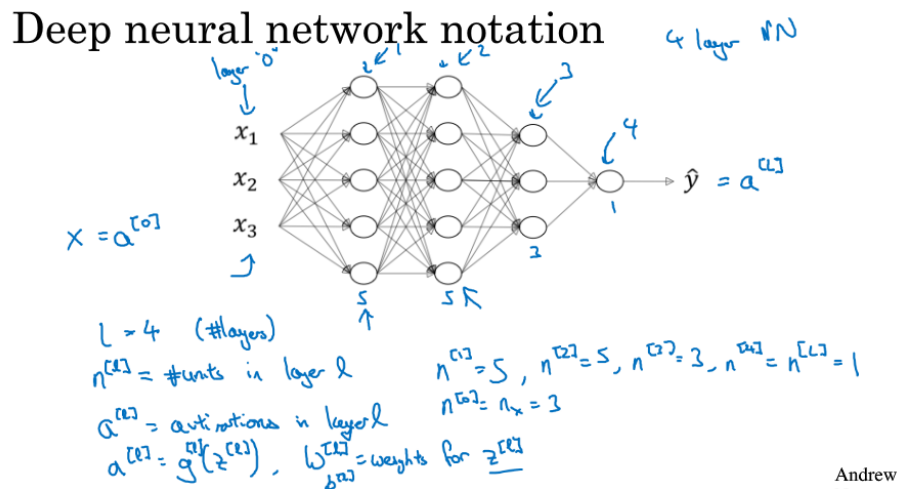


Figure 3: Deep neural network

and b 's row is equal to W 's. The column of b is $n^{(0)}$. And $n^{(0)}$ means the amount of the nodes in the input layer.

4 Hyperparameters

The parameters which can control the size of the other parameters called the hyperparameters. Learning rates is a typical hyperparameter. This may depend on ones' own experience to choose a proper hyperparameters. Maybe we can chose 0.01 at first to change the learning rate step by step when we finally get an ideal output.

5 Conclusion

I know that this week my teammates get some works to do. My creed is learning better and doing better. I really know that the practice is important. To be honest, our foundation is really poor, we are trying to change the status these days. I try to help them deal with the work as well as I can do, and hope if one day I can do it best.

This week, our senior teammates give us a report proceeding. I find maybe GAN is a really good direction for me. Actually I really like these creation works since my childhood. I will firstly realize the meaning of it and catch the main idea. I hope I can come up my own idea one day.