Neural Networks and Deep Learning

Complete Course Review

This is my first step in Deep Learning after a brief introduction of Neural Networks in Andrew Ng's Machine Learning Course by Stanford University.

The course begins with the phrase "AI is the new Electricity", similar to Electricity which transformed all the industries 100 years back and now AI would bring a similar transformation and Deep Learning is driving these developments.

So, this pretty much answers why I am doing this specialization- I want to be a part of these change rather than seeing the change.

We start the course by giving a very simple definition for Deep Learning as follows:
"Training neural networks or sometimes very large neural networks".

In the 1st week, Andrew Ng starts off with his classic housing prediction example to show how features (parameters) are selected in a Neural Network (NN), evaluated the reason why Deep Learning is taking off so well.

I also understood that there is no hard rule to know the values of hyper-parameters (learning rate, # iterations, # Layers, etc), we need to test them with different set of values and compare the results. The general rule of thumb is "Applied Deep Leaning is a very empirical process". We generally follow:

Idea -> Code -> Experiment

Set Values -> Implement -> Analyse Result

In the process, I understood the importance of Vectorising rather than using loops as using loops significantly increases the time complexity as the data size increases and # iterations increases

In the 2nd week, we have built a cat recognizer initially with logistic regression with a success rate of 70%. Later, we added few hidden layers (~1), to see what was the result and then later increased the depth (more number of hidden layers) thereby getting a first-hand experience of how deep learning works.

In week 2, we take the 1st formal step into neural network by understanding how logistic regression works, we see it as a neural network with '0' hidden layers to build a cat classifier with the help of sigmoid activation function to determine whether a given image is cat or non-cat, we got a training accuracy of 99% and Test accuracy of 70% indicating that we are over-fitting the data.

In Week 3, we continue from where we left in week 2 by defining logistic regression as a neural network with '0' hidden units to shallow neural network with '1' hidden layer.

We understood, the intuition behind the back-propagation algorithm, the forward propagation computes the final output and computes the cost(loss) and in the back propagation we try to optimize the parameters, in such a way to reduce the cost very similar to how we try to do optimize something to reduce error and make it more efficient, we try to correct it next time when we perform.

We have seen the importance of random-initialisation instead of initialising them to zeros because if you initialise all of them to zeros all your hidden units of a given layer would compute the same value, which is redundant.

After our 1st 3 weeks spent in building a solid foundation of how a Neural Networks, we took a deep dive to see the functioning of Deep Neural Networks, understood the complete flow of Forward-Propagation and Backward-Propagation.

I think Andrew did a great job especially in giving intuitions e.g. how he used computing graphs to explain the concept of derivatives making it easy to understand even for a non-programmer.

One drawback I felt with the Stanford Machine Learning course was we were asked to program in Octave/Matlab, but I think that was heard by the course instructors and we are using Python 3 in this course.

The programming assignments were really helpful as they helped me to the feel of the algorithm better, the way we built logistic regression as a simple neural network with no hidden layer and by the end of the course with a deep neural network.

We have seen different activation functions namely: sigmoid, tanh and Relu (Rectified Linear Unit) and also understood the pros and cons of each one of them. The why part really helped me to get a better insight while implementing these.

I have learnt a very important debugging tool: matrix dimensions -> this reduced 90% of all my errors, checking whether my matrix dimensions are right, if yes go to next step else fix that.

The analogy from circuit theory helps you to understand why deep layers are needed, more informally, there are functions you can compute with a “small” L-layer deep neural network (NN) that shallow NN require exponentially more hidden units to compute. Intuitions from what different layers of a deep network does in Image recognition, Speech recognition helped me get a better understanding of how deep learning works.

In Machine Learning/Deep Learning, lots of complexity comes from data and not from code i.e. the reason you get magical results even with few lines of code.

At the end of this course, I can confidently say how a simple deep network, understood the power of a simple yet extremely efficient back-propagation algorithm. I feel I have the foundation set to dive deep into deep learning. I am excited for the 2nd course….