Deep Learning – Autumn 2019

Assignment 1 – Building Neural Network Architectures

PART A – Vanilla Neural Network (from scratch)

The company situation:

You recently joined a very large corporate as a data scientist. This large corporate has been very risk averse previously regarding data science work, however the junior data scientist who has been working there for around a year now (the only data scientist prior to your arrival) has been making good headway with analytical work and some 'classic' machine learning models. Management has been impressed with her work. However, management has been very sceptical about her push to move into using 'A.I' to solve some other problems around the company. Frankly, they don't trust deep learning, neural networks and don't understand why it is needed when they have been making progress with classic models.

Just as you are about to join, the previous data scientist has gone on holiday and left you their initial deep learning project to complete for an upcoming management presentation in a few weeks. Your task is to pick up their work, finish it as much as you can and prepare the notebooks and explanatory notes for you to compile a report and presentation together when the junior data scientist returns. You think you may be even able to add some extensions to prove yourself in this new role.

This task is largely technical in nature. You do not need to compile the report and presentation for this task.

The business brief:

Your absent colleague has convinced management to allow a proof-of-concept neural network project to be undertaken on the only area of the company they are comfortable with this experimentation; the carpark. Specifically, they currently employ several people whose full-time jobs is to press the buttons to open the security gate and let cars into the carpark.

Your colleague has been allowed to set up a motion-activated camera that takes a picture of anything approaching the car park and hopes to build a model that will take this picture, determine if it is an automobile and therefore automatically decide whether to lift the gate or not.

Unfortunately, due to the location of the company headquarters there appears to have been a lot of other things triggering the camera to take photos which you definitely don't want to let into the secure facility.

The technical brief:

You must build a notebook that undertakes feed-forward and back-propagation for a vanilla neural network. Some parts of this have been completed for you and others need to be completed.

It would greatly aid your colleague if you could do a quick benchmark of performance for 'classic' ML models on this dataset. Run a quick logistic regression or tree-based model and note the performance and efficiency of this method.

Your functions and work should map to a general methodology of:

1. Decide & encode number of layers, size of layers
2. Initialise weights & biases
3. (For each training cycle):
    a. Feed forward (linear, activation)
    b. Compute cost
    c. Back prop
    d. Update parameters
4. Finally make predictions on your test set.

The notes from your colleague indicate your initial model should be:

- 1 input layer, 3 hidden layers and 1 output layer
- Hidden layer sizes of 10, 25, 10
- Relu activation function for all layers except the last (sigmoid)

The technology:

Due to the scepticism of management, you **will not** be able to use any deep learning frameworks (keras, pytorch, tensorflow etc) for this task. You will have access to numpy, pandas and various other python data analytics packages for your work. However, your neural network must be *handcrafted*.

The existing work:

You have been provided a jupyter notebook that contains the work and notes completed so far by your colleague. The basic outline is included of what they think will be a good approach as well as some useful functions. It would be best to firstly complete their work before moving on to extension components or altering the main brief. After all, you don't want to ruffle any feathers in your first days at the new job!

One **important** thing to note is the current network is very unstable due to the weight initialisation. You are strongly encouraged to substitute the (*0.01) in the weight initialisation for Xavier initialisation (Source, Glorot & Bengio et al, 2010). You will not be able to do a very deep network (perhaps not even the current suggestion….) without

adjusting this.

Submission requirements:

They won't have much time upon their return to collate everything into a nice report, so good formatting (markdown TOC levels), detailed comments and explanations are a must. Though of course, she won't want to read thousands of lines of code and text; so be kind to your colleague with your compiled work.

It will be important to ensure your work (whether within the notebook, at the end of the notebook in a separate section, or a standalone 1-2 pager) includes an outline of your methodology, decisions and results if any of the extension components are attempted.

Extension Tasks:

Assessment note: To gain marks for the extension component, you only need to make a good attempt at one extension task. Feel free to propose and outline other extension to your colleague's work, though any of these listed below should be sufficient.

- Try different numbers of hidden layers, hidden layer sizes
- Try different activation functions (remember you will have to look up how to differentiate them!)
    - Leaky Relu? Elu? Maxout?
- Try a different weight initialisation. If you want to try a much deeper network than the initial brief then this will likely be needed.
    - He initialisation (Source He at al., 2015)

The weight initialisation is an important element and is advisable to try first, as part of the original tasks rather than extension tasks.

Do not try to recreate a CNN by hand (pooling layers etc). The focus for this task is on a vanilla neural network.