**DL LAB WiSe 18 – Assignment 01**

Implementation of a simple feedforward neural network was relatively easy than figuring out a reasonable network that achieves good performance. The key takeaway from this exercise for me is how important the concept of hyperparameter optimisation is.

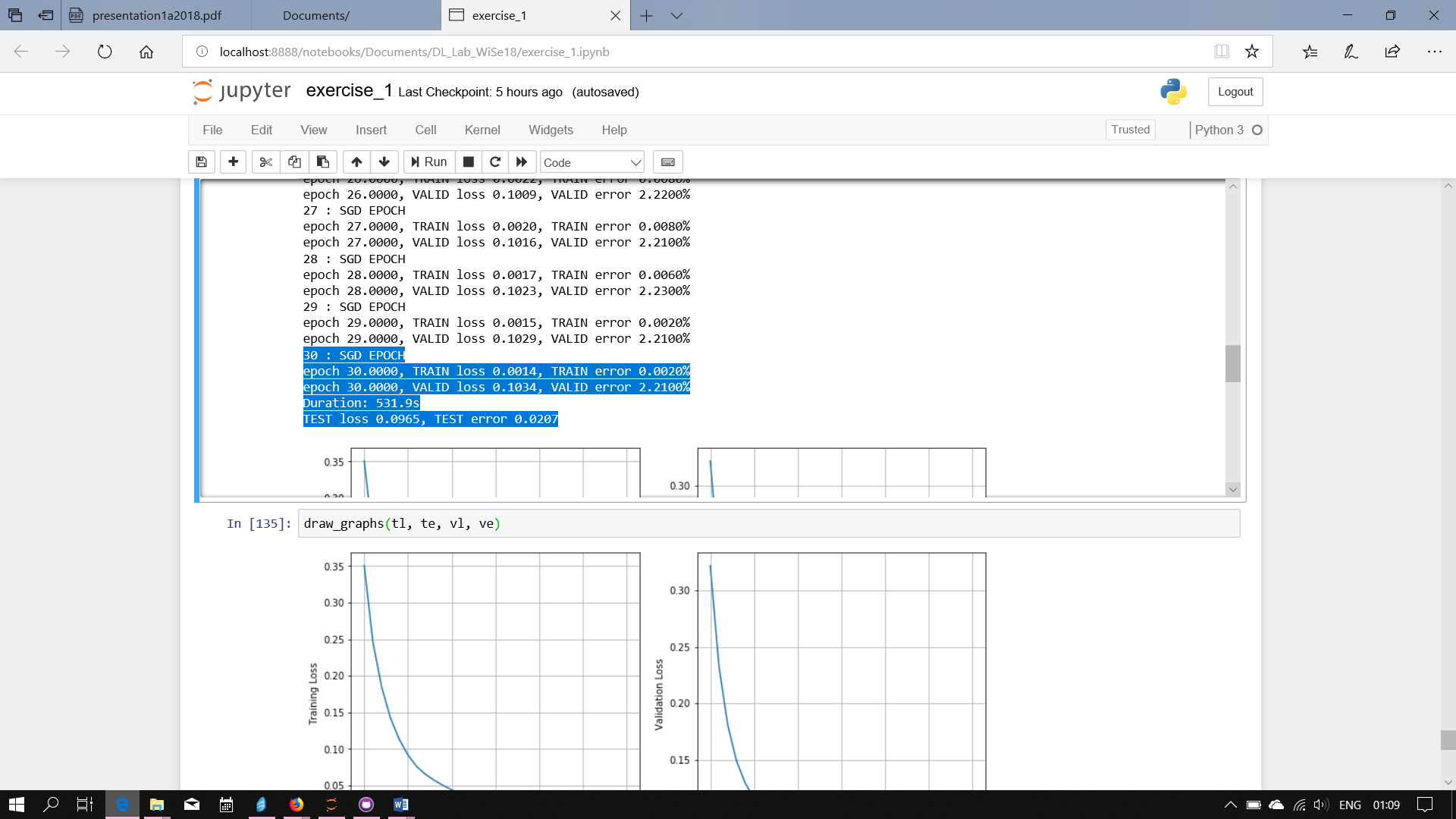
After extensive application of the “Graduate Student Descent” / Random Search Algorithm, this is the best performance that I have observed.

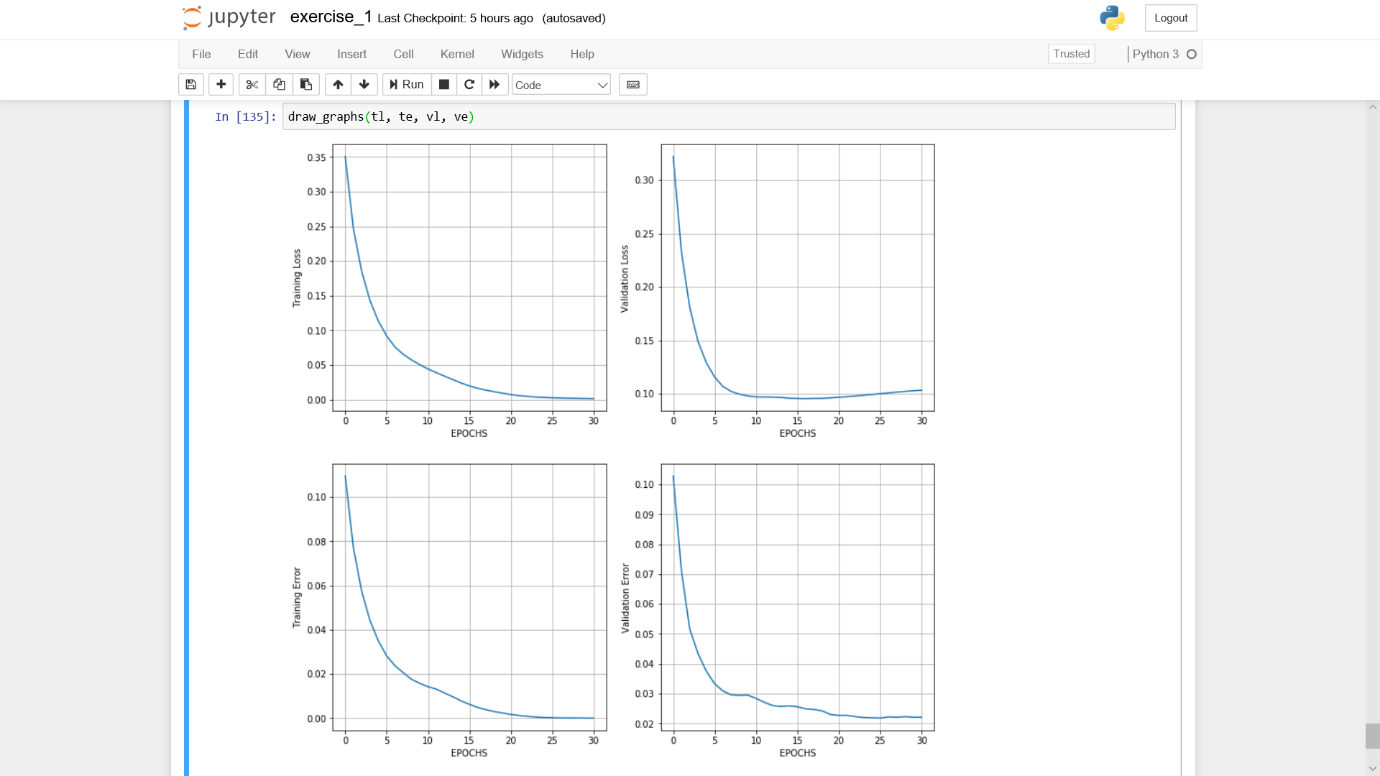
|  |  |
| --- | --- |
| Number of Hidden Layers (with nonlinearity) | 2 |
| Layer 1 | Tanh |
| Units | 200 |
| Layer 2 | Relu |
| Units | 500 |
| Learning Rate | 0.1225 |
| Max Epochs | 30 |
| Batch Size SGD | 64 |

30 : SGD EPOCH

epoch 30.0000, TRAIN loss 0.0014, TRAIN error 0.0020%   
epoch 30.0000, VALID loss 0.1034, VALID error 2.2100%   
Duration: 531.9s

TEST loss 0.0965, TEST error 0.0207 (2.07%)





I’ve observed that heavier/larger networks (with hidden units > 150) take longer to train in comparison to smaller networks, but both can sometimes achieve similar performance (about 2.x% error for me), although tuning the hyperparameters manually in both cases is tricky, as the network also tends to overfit when run for long epochs (sometimes even at epochs > 50), or fails to converge without a proper learning rate.