Data Science Toolbox Portfolio Questions

07 Perceptrons and Neural Networks

Daniel Lawson — University of Bristol

Block 7

Portfolio 07

Choose **one question** and write up to **one page** about it. You are free to conduct further experiments to add weight to your results, and any additional material you generate can be submitted as an appendix. See The Assessment Page for advice.

These questions may make reference to the content from the current block.

Question R07.1: Investigate the Universal Approximation Theorem for MLPs, which says that an MLP with a single hidden layer of arbitrary width can approximate any continuous function.

Take `model_1`, the first MLP from lab code and modify its architecture. First train it with 1 hidden layer with 64 neurons, compare this with 2 hidden layers with 32 each etc. for a variety of different configuration of hidden layers (make sure you have activation functions between, you might want to use a loop inside your class to make this easier to define).

For a fixed number of epochs, record the training speed, model performance, minimum test loss etc. Produce plots (as an appendix) and explain your observations in the portfolio. You might want to watch this video for some inspiration.

Question R07.2: Improving CNN Performance. Augment the code for `model_2` (the CNN from the workshop) to improve performance by exploring hyper-parameters. Explore two of the following parameters (or others) to see what works best for performance. For each, write a short summary explaining what it does and document how it affects performance:

- Adjusting learning rate
- Use a different optimizer like Adam or AdamW
- Adjusting batch size
- Implement random transforms on your dataloaders (think about which transforms are appropriate for train/test loader)
- Dropout layers

Question R07.3: Read the paper "**Transformers need glasses!** Information over-squashing in language tasks" You could also listen to the podcast with the authors as well.

You're going to try and replicate some of the experiments and give an intuitive explanation of the results. To replicate the results, you should make a new google collab notebook and use some of the free transformer models. Within the collab notebook, remember to change runtime to GPU. You should use models from the transformers library for experiments (see example code at the bottom of the collab notebook). These are free but significantly worse performing than the models used in the paper so you should expect your results to reflect that!