Deep learning

For the following tasks, you can use any Machine Learning library (such as PyTorch or JAX). A skeleton notebook in PyTorch is provided, but you do not need to use it.

1. Using automatic differentiation (AD), calculate the derivative of the following function: $f(x) = \sigma^2(x)$, where $\sigma(x)$ is the logistic sigmoid function.

   a. Apply AD on the interval $[-3, 3]$ and plot your result.
   b. Check the result by comparing against the derivative calculated using the analytically derived formula.

2. Find a minimum point of the Himmelblau function: $f(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2$.

   a. Plot the 2D or 3D contour of the function on the $[-6, 6]^2$ interval.
   b. Using automatic differentiation (but without using an optimiser), write a loop that minimises the function.
   c. Try different built-in optimisers (at least 3) to minimise the same function. Tune their hyperparameters (especially the learning rate) by trying out a few sensible values.
   d. Save the trajectories, and plot each trajectory on top of the contour plot, and compare the behaviour of the different optimisers.

3. Classify the MNIST dataset using a deep learning model. MNIST is a classification dataset consisting of 28×28 grayscale images of handwritten digits. There are 10 classes in the dataset (corresponding to each digit).

   a. Download and load into memory the training and test datasets.
   b. Create a suitable convolutional neural network model, and specify an appropriate loss function (categorical cross entropy) and an optimiser (it may turn out that for this task, the optimisers perform differently than in Task 2).
   c. Train the model, using minibatches of suitable size (e.g., depending on your hardware), evaluate on the test set. Plot some of the misclassified samples from the test set.
   d. Create a learning curve that plots the loss measured on the training and the test dataset for each epoch (one pass through the whole training dataset).

4. The task is to reproduce one of the original double descent experiments from [1], Appendix C3. This is also an MNIST classification task, so modify and use the code from Task 3.

   a. Create a new model type that is a fully connected neural network with one hidden layer. This layer should have a configurable width.
   b. From layer sizes 3 to 800, train the model on 4000 randomly selected training samples, then evaluate on the test dataset.
   c. Plot the training and test accuracies (or zero-one errors) versus the layer sizes. Observe any similarities, differences with those of the original paper.

[1] M. Belkin, D. Hsu, S. Ma, and S. Mandal, "Reconciling modern machine learning and the bias-variance trade-off," *CoRR*, vol. abs/1812.11118, 2018, Available: http://arxiv.org/abs/1812.11118