ACM41020 Assignment 4

Neural Networks

DUE: 10AM FRIDAY 22ND NOVEMBER 2024

1. **Autoencoder**
   We have seen how to construct a simple neural network with one hidden layer that can successfully recognise handwritten digits. This network can be considered as an example of an *encoder*, which takes some high-dimensional input (in this case an image represented by a vector with 784 components) and encodes it into a lower-dimensional representation (a vector of 10 numbers which may be interpreted in terms of probabilities).

   a) Using the code from the "Training a Neural Network" notebook, construct and train an *encoder*, a network that takes in a vector of 784 numbers and produces a vector of 10 numbers. Your network should have no hidden layers.
   b) Construct and train a corresponding *decoder*, a network that does the reverse of this process. The decoder should take a vector of 10 numbers and reconstruct a vector with 784 numbers representing an image of a handwritten digit.
   c) Give an interpretation of the columns of the weight matrix in the decoder.
   d) Use your decoder to produce a set of 10 images, one for each "perfect" example of a digit. Here, perfect means the input vector only has one non-zero entry and it is equal to 1.
   e) Combine the two networks to produce an *autoencoder*, which takes an input vector of 784 components and produces an output vector with 784 components.
   f) Test your autoencoder with a sample of 10 digits from the validation set. Include 5 cases where the encoder correctly predicts the expected output and 5 cases where it does not.
   g) Produce another autoencoder as before, completing steps (a) to (f) above, but include a hidden layer with 40 neurons in both the encoder and decoder.

2. **Training a neural network with TensorFlow and Keras**
   You wish to produce neural networks that approximate the functions

$$f(x) = \sin x, \qquad g(x) = (1 + \sin x)/2$$

   a) Create training data by evaluating $f(x)$ and $g(x)$ at 10,000 equally-spaced points in the range $x \in [0,2\pi]$.
   b) Using TensorFlow and Keras, construct neural networks for the two functions with the following properties:
      - An input layer with a single neuron.
      - A hidden layer with 3 neurons and a logistic sigmoid activation function.
      - An output layer with a single neuron and a logistic sigmoid activation function.
   c) Train your networks using the training data you produced. Use the following parameters for training:
      - A mean squared error cost function.
      - Batch size 20.
      - 200 epochs.
   d) Produce a plot comparing your models against the true functions $f(x)$ and $g(x)$ over the range $x \in [0,2\pi]$. Comment on if one model performs better than the other, and why.
   e) Plot your model for $g(x)$ over the range $x \in [0,4\pi]$. Comment on and give an explanation for the performance of your model in this case. Your comment should include two possible explanations based on (i) the training data used and (ii) the expressivity of a network with three hidden neurons.