CO553 - Introduction to Machine Learning: Neural networks

Prepared by Marek Rei

Autumn 2020/2021

1 Questions

Here is a set of various questions to improve your understanding of neural networks.

1. Remember that coursework 2 part 1 covers much of the neural network material – feedforward networks, activation functions, loss functions, backpropagation, gradient descent and data normalisation. Even if your teammates implemented some of it, make sure you go over all of it and understand how it works.
2. Given a network with the following output activations, write 1) what type of task are we likely solving with this network, 2) what would be the most common loss function for this activation and task combination, and 3) one example of this type of task.
   (a) One neuron with sigmoid
   (b) Multiple neurons with softmax
   (c) One neuron with linear output
   (d) Multiple neurons with sigmoid
3. Write out the formula for linear regression with two input features.
4. Given the linear regression from the previous question, derive the gradient descent updates for each of the parameters.
5. You have a multi-layer neural network for regression. Input layer with 2 neurons; hidden layer with 2 neurons and sigmoid activation; output layer with 1 neuron and linear activation.

   The weights between the input layer and hidden layer look like this:

$$W_1 = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix}$$

$$b_1 = \begin{bmatrix} 0.2 & -0.1 \end{bmatrix}$$

   The weights between the hidden layer and the output layer look like this:

$$W_2 = \begin{bmatrix} 0.5 \\ 0.6 \end{bmatrix}$$

$$b_2 = \begin{bmatrix} 0.1 \end{bmatrix}$$

   Given the following input:

$$X = \begin{bmatrix} 0.2 & -0.4 \end{bmatrix}$$

   calculate the output of the network.

6. Calculate the mean squared error (MSE) for the network in the previous question, given the target value $y = 1.5$.

7. Calculate the gradients for the weight matrix W_1 and bias vector b_1 of that network.

8. Calculate the updated values for W_1 and b_1 after doing 1 step of stochastic gradient descent with learning rate 0.1.