Introduction to Machine Learning
Problem Set: Neural Networks

Summer 2021

1. **This is not a programming question - compute the answers by hand, with a calculator, not by writing code.**

   Consider a neural network for classification with two features at the input, two hidden-layer nodes in a single hidden layer, and one output node. There is a sigmoid activation function at the hidden layer nodes *and* at the output layer node. The weights are indicated on the graph below. (The bias input is always +1.)

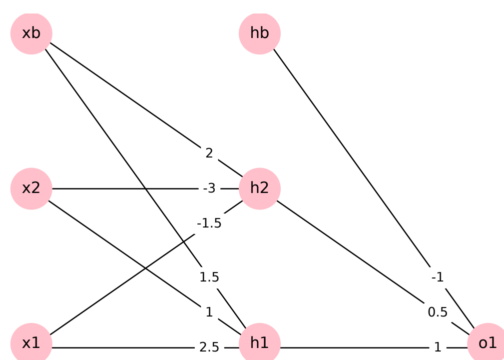

Figure 1: Network for neural network training example.

   (a) For the input $x_1 = 0, x_2 = 1$, do a forward pass on the network and compute the output.
   Solution:
   - Output of h_1: $\sigma(2.5 \times 0 + 1 \times 1 + 1.5 \times 1) = \sigma(2.5) = 0.924$
   - Output of h_2: $\sigma(-1.5 \times 0 - 3 \times 1 + 2 \times 1) = \sigma(-1) = 0.269$
   - Output of o_1: $\sigma(1 \times 0.924 + 0.5 \times 0.269 - 1 \times 1) = \sigma(0.058) = 0.515$

   (b) Suppose the true value is $y = 1$ when the input is $x_1 = 0, x_2 = 1$. Compute the gradients of the loss function with respect to the weights, using the backpropagation algorithm and the squared loss function:

$$L(W) = \frac{1}{2}\sum_{N}(y - u_O)^2$$

   Show your work.

Note that because of the sigmoid activation function at the output node, $u_O = \sigma(z_O)$, you will compute the backpropagation error at the output with

$$\delta_O = \frac{\partial L}{\partial z_O} = \frac{\partial L}{\partial u_O}\frac{\partial u_O}{\partial z_O}$$

(Note: we would usually use the binary cross entropy loss function for a classification problem, but the math is easier to work with for the squared error loss! So in this case, we'll use squared error.)

Solution:

- $\delta_{O_1} = -(1 - 0.515) \times \sigma'(0.058) = -0.121$
- $\delta_{h_1} = 1 \times (-0.121) \times \sigma'(2.5) = -0.0085$
- $\delta_{h_2} = 0.5 \times (-0.121) \times \sigma'(-1) = -0.0119$
- $\frac{\partial L}{\partial W_O} = -0.121 \times [0.924, 0.269, 1] = [-0.112, -0.0326, -0.121]$ from h_1, h_2, h_b to o_1
- $\frac{\partial L}{\partial W_{h_1}} = -0.0085 \times [0, 1, 1] = [-0, -0.0085, -0.0085]$ from x_1, x_2, x_b to h_1
- $\frac{\partial L}{\partial W_{h_2}} = -0.0119 \times [0, 1, 1] = [-0, -0.0119, -0.0119]$ from x_1, x_2, x_b to h_2

(c) Compute the updated weights for both the hidden layer and the output layer by performing one step of gradient descent. Use a learning rate of 0.1.

Solution:

- New W_{O_1}: $[1, 0.5, -1] - 0.1[-0.112, -0.0326, -0.121]^T = [1.011, 0.503, -0.988]$
- New W_{h_1}: $[2.5, 1, 1.5] - 0.1[0, -0.0085, -0.0085]^T = [2.5, 1.00085, 1.50085]$
- New W_{h_2}: $[-1.5, -3, 2] - 0.1[0, -0.0119, -0.0119]^T = [-1.5, -2.999, 2.001]$

2. Neural networks for musical instrument classification

   Please refer to the Colab notebook on the course site.

   Solution: Please refer to the solution notebook.