Matthew Ragoza
Dr. Adriana Kovashka
CS 2078 - Intro to Deep Learning
February 7, 2024

Homework 1

Part b.

Forward pass

$$
\begin{aligned}
a_j &= \sum_i w_{ji}^{(1)} x_i \\
a_1 = a_2 &= 0.05 * 1 + 0.05 * 1 + 0.05 * 1 = 0.15 \\
z_j &= \sigma(a_j) = e^{a_j}/(1 + e^{a_j}) \\
z_1 = z_2 &= e^{0.15}/(1 + e^{0.15}) = 1.16/2.16 = 0.54 \\
y_k^p &= \sum_j w_{kj}^{(2)} z_j \\
y_k^p &= 0.05 * 1 + 0.05 * 0.54 + 0.05 * 0.54 = 0.104 \\
L &= \frac{1}{2} \sum_k (y_k^p - y_k^t)^2 \\
L &= \frac{1}{2}[(0.104 - 0)^2 + (0.104 - 0)^2] = 0.0108
\end{aligned}
$$

Backward pass - Local derivatives

$$
\begin{aligned}
\frac{\partial L}{\partial y_k^p} &= (y_k^p - y_k^t) = [0.104, 0.104] \\
\frac{\partial y_k^p}{\partial w_{kj}^{(2)}} &= z_j = [1, 0.54, 0.54] \\
\frac{\partial y_k^p}{\partial z_j} &= w_{kj}^{(2)} = [0.05, 0.05, 0.05; 0.05, 0.05, 0.05] \\
\frac{\partial z_j}{\partial a_j} &= z_j(1 - z_j) = [0.248, 0.248] \\
\frac{\partial a_j}{\partial w_{ji}^{(1)}} &= x_i = [1, 1, 1] \\
\frac{\partial a_j}{\partial x_i} &= w_{ji}^{(1)} = [0.05, 0.05, 0.05; 0.05, 0.05, 0.05]
\end{aligned}
$$

Backward pass - Output layer

$$\frac{\partial L}{\partial w_{kj}^{(2)}} = \frac{\partial L}{\partial y_k^p}\frac{\partial y_k^p}{\partial w_{kj}^{(2)}}$$
$$\frac{\partial L}{\partial w_{kj}^{(2)}} = (y_k^p - y_k^t)z_j$$
$$\frac{\partial L}{\partial w_{k0}^{(2)}} = 0.104 * 1 = 0.104$$
$$\frac{\partial L}{\partial w_{k1}^{(2)}} = 0.104 * 0.54 = 0.056$$
$$\frac{\partial L}{\partial w_{k2}^{(2)}} = 0.104 * 0.54 = 0.056$$

$$\frac{\partial L}{\partial z_j} = \frac{\partial L}{\partial y_k^p}\frac{\partial y_k^p}{\partial z_j}$$
$$\frac{\partial L}{\partial z_j} = \sum_k (y_k^p - y_k^t)w_{kj}^{(2)}$$
$$\frac{\partial L}{\partial z_j} = 0.104 * 0.05 + 0.104 * 0.05 = 0.0104$$

Backward pass - Hidden layer

$$\frac{\partial L}{\partial w_{ji}^{(1)}} = \frac{\partial L}{\partial z_j}\frac{\partial z_j}{\partial a_j}\frac{\partial a_j}{\partial w_{ji}^{(1)}}$$
$$\frac{\partial L}{\partial w_{ji}^{(1)}} = \frac{\partial L}{\partial z_j}\frac{\partial z_j}{\partial a_j}\frac{\partial a_j}{\partial w_{ji}^{(1)}}$$
$$\frac{\partial L}{\partial w_{ji}^{(1)}} = 0.0104 * 0.248 * 1 = 0.00258$$

Weight update

$$
\begin{aligned}
w_{kj}^2 &\leftarrow w_{kj}^2 - \eta \frac{\partial L}{\partial w_{kj}^{(2)}} \\
w_{k0}^2 &\leftarrow 0.05 - 0.3 * 0.104 = 0.05 - 0.0301 = 0.0199 \\
w_{k1}^2 &\leftarrow 0.05 - 0.3 * 0.056 = 0.05 - 0.0168 = 0.0332 \\
w_{k2}^2 &\leftarrow 0.05 - 0.3 * 0.056 = 0.05 - 0.0168 = 0.0332
\end{aligned}
$$

$$
\begin{aligned}
w_{ji}^1 &\leftarrow w_{ji}^1 - \eta \frac{\partial L}{\partial w_{ji}^{(1)}} \\
w_{ji}^1 &\leftarrow 0.05 - 0.3 * 0.00258 = 0.05 - 0.000774 = 0.0492
\end{aligned}
$$

Updated loss

$$
\begin{aligned}
a_j &= \sum_i w_{ji}^{(1)} x_i \\
a_1 &= a_2 = 0.0492 * 1 + 0.0492 * 1 + 0.0492 * 1 = 0.1476 \\
z_j &= \sigma(a_j) = e^{a_j}/(1 + e^{a_j}) \\
z_1 &= z_2 = e^{0.1476}/(1 + e^{0.1476}) = 1.159/2.159 = 0.535 \\
y_k^p &= \sum_j w_{kj}^{(2)} z_j \\
y_k^p &= 0.0199 * 1 + 0.0332 * 0.535 + 0.0332 * 0.535 = 0.0554 \\
L &= \frac{1}{2} \sum_k (y_k^p - y_k^t)^2 \\
L &= (0.0554 - 0)^2 + (0.0554 - 0)^2 = 0.00307
\end{aligned}
$$

The loss decreased, as expected.

Part d.

After implementing the neural network training script, I evaluated it by training on the wine quality dataset for 1000 iterations and 30 hidden units using a learning rate of 0.1, 0.01, or 0.001. The training plots, correlation plots, and RMSE values for the train and test sets are shown below.

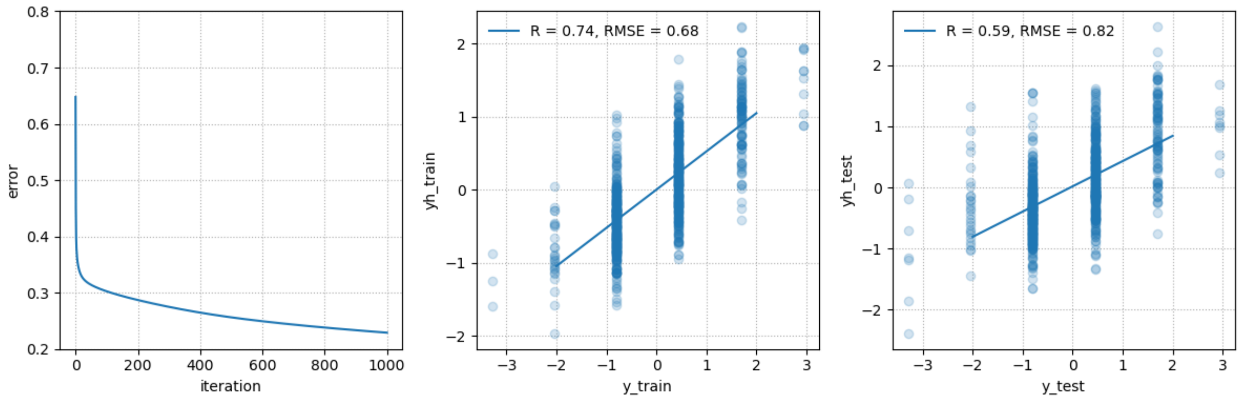

Figure 1. Learning rate = 0.1

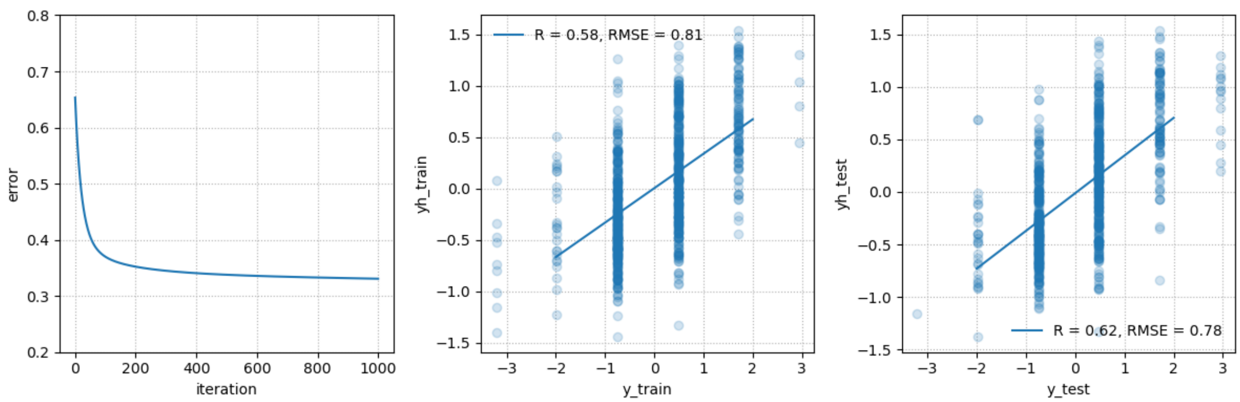

Figure 2. Learning rate = 0.01

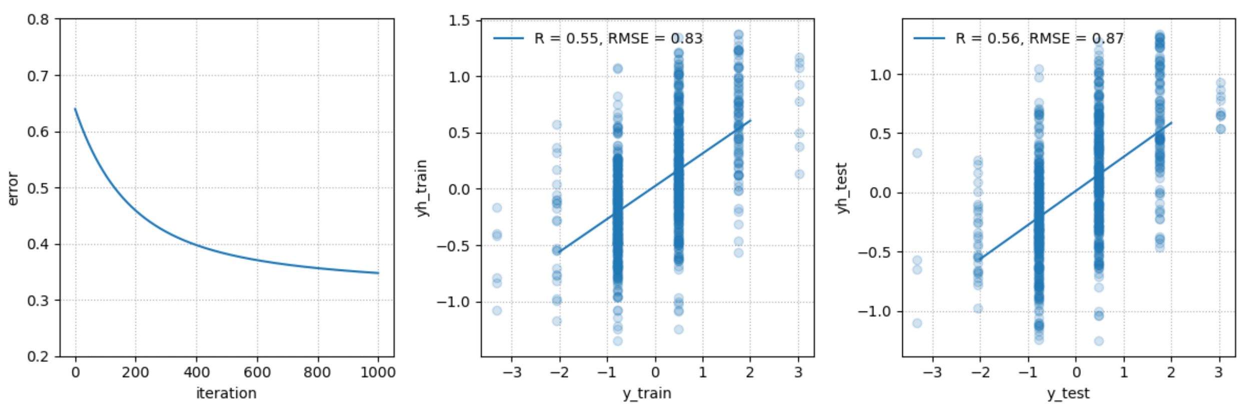

Figure 3. Learning rate = 0.001