CS4375 Homework 2 Report

Anthony Tran axt220037
Kacie Yee

1.1 Gradient Descent

1. Find the error (simple squared error)

$$E(w) = \frac{1}{2} \sum_j (t_j - o_j)^2$$

$$w_{new} = w_{old} + \Delta w$$

$$\Delta w = -\eta \frac{\partial E}{\partial w}$$

Now, derive the partial derivative.

$$\frac{\partial E}{\partial w} = \frac{\partial}{\partial w_i} \frac{1}{n} \sum_j (t_j - o_j)^2$$

$$= \frac{1}{n} \sum_j \frac{\partial}{\partial w_i} (t_j - o_j)^2$$

$$= \frac{1}{n} \sum_j 2 (t_j - o_j) \frac{\partial}{\partial w_i} (t_j - o_j)$$

$$= \sum_j (t_j - o_j) \frac{\partial}{\partial w_i} (t_j - \vec{w} \cdot \vec{x}_{ij})$$

$$\frac{\partial E}{\partial w_i} = \sum_j (t_j - o_j)(-x_{ja})$$

so, $$\Delta w_i = \eta \sum_j (t_j - o_j) x_{ij}$$

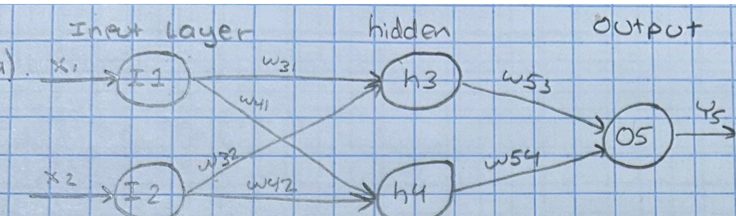
↑

our final weight update

Using this equation, we can compute the Δw_i for each weight, and use that to update each weight.

1.2 Comparing Activation Function

1.2) a)



$I1 = x_1$

$I2 = x_2$

$h3 = h(w_{31} \cdot x_1 + w_{32} \cdot x_2)$

$h4 = h(w_{41} \cdot x_1 + w_{42} \cdot x_2)$

$$\boxed{y_5 = h(w_{53} \cdot h(w_{31} \cdot x_1 + w_{32} \cdot x_2) + w_{54} \cdot h(w_{41} \cdot x_1 + w_{42} \cdot x_2))}$$

or $y_5 = h(w_{53} \cdot h_3 + w_{54} \cdot h_4)$

b) $X = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$, $W^{(1)} = \begin{pmatrix} w_{31} & w_{32} \\ w_{41} & w_{42} \end{pmatrix}$, $W^{(2)} = (w_{53} \ \ w_{54})$

hidden layer $= \begin{bmatrix} w_{31} & w_{32} \\ w_{41} & w_{42} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = W^{(1)} X = h(W^{(1)} X)$

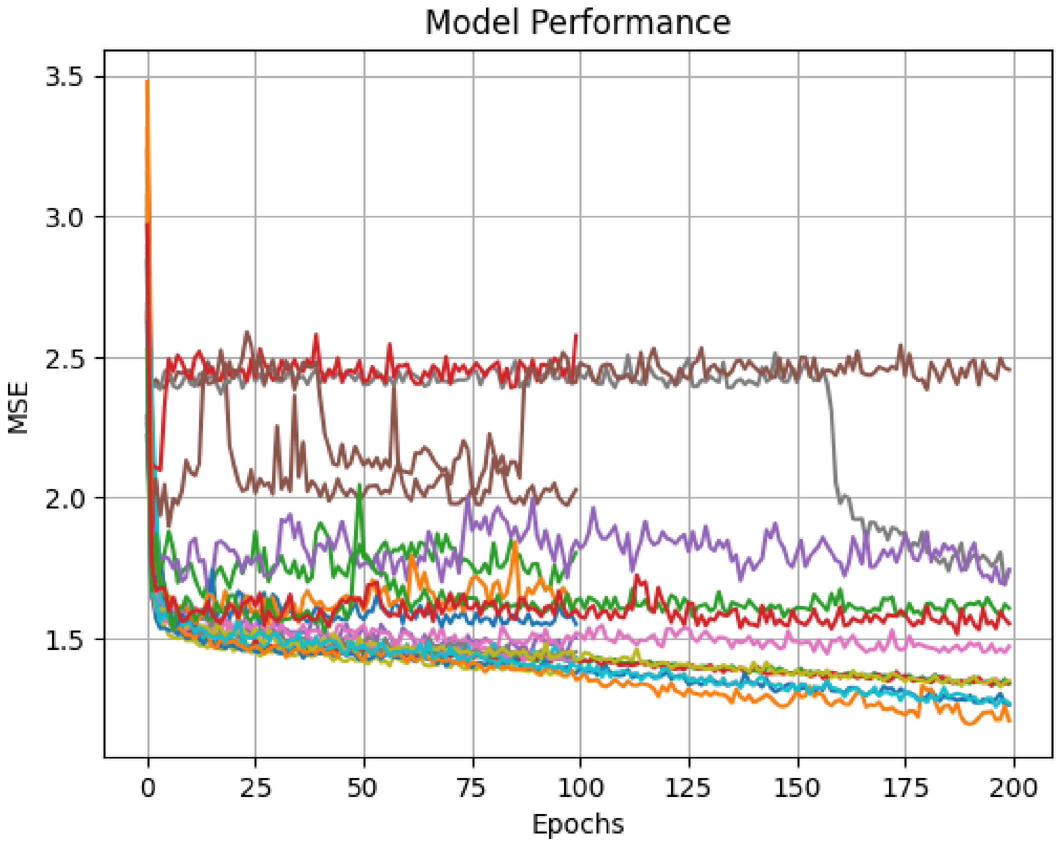
$y^5 = [w_{53} \ \ w_{54}] \cdot h(W^{(1)} X) = W^{(2)} \cdot h(W^{(1)} X) = \boxed{h(W^{(2)} \cdot h(W^{(1)} X))}$

c) Tanh function is a stretched version of the sigmoid by 2x and translated downwards by -1.

So $\rightarrow$ $h_t(x) = 2h_s(2x) - 1$

Since Tanh can be written in terms of sigmoid, the output function can be the same if you modify the weights and constants to match the linear transformation shown above.

Report:



Model Performance
3.5
3.0
2.5
2.0
1.5
MSE
0
25
50
75
100
125
150
175
200
Epochs

	activation	learning_rate	epochs	layers	test_mae	train_mae	test_loss	train_loss
0	sigmoid	0.01	100	2	1.477041	1.394334	4.416725	3.967221
1	sigmoid	0.01	100	3	1.505339	1.388505	4.443105	3.916803
2	sigmoid	0.01	200	2	1.639835	1.408230	5.026944	3.584769
3	sigmoid	0.01	200	3	1.545813	1.312767	4.785028	3.389249
4	sigmoid	0.10	100	2	1.648743	1.498624	4.904697	4.127173
5	sigmoid	0.10	100	3	1.988100	1.977970	7.295230	7.335787
6	sigmoid	0.10	200	2	1.819599	1.640983	5.943343	4.653182
7	sigmoid	0.10	200	3	1.733765	1.676414	5.491228	5.316587
8	tanh	0.01	100	2	1.523842	1.324666	4.641914	3.540610
9	tanh	0.01	100	3	1.559050	1.429311	4.574947	3.864309
10	tanh	0.01	200	2	1.581151	1.222847	4.962899	2.997593
11	tanh	0.01	200	3	1.598864	1.173827	5.345596	2.711775
12	tanh	0.10	100	2	1.757748	1.755021	5.888821	5.867476
13	tanh	0.10	100	3	2.470859	2.422840	11.839838	11.722538
14	tanh	0.10	200	2	1.845451	1.762927	6.006831	5.793617
15	tanh	0.10	200	3	2.895843	2.838145	12.776013	12.220264
16	relu	0.01	100	2	1.525081	1.371497	4.642535	3.746282
17	relu	0.01	100	3	1.593909	1.439553	4.717119	3.894003
18	relu	0.01	200	2	1.599572	1.305960	6.204692	3.329473
19	relu	0.01	200	3	1.625210	1.291479	5.106838	3.210882
20	relu	0.10	100	2	1.600400	1.538947	4.685734	4.326734
21	relu	0.10	100	3	1.597669	1.548882	5.176435	4.902181
22	relu	0.10	200	2	1.621776	1.550816	4.903563	4.588874
23	relu	0.10	200	3	1.658133	1.557553	5.010948	4.538371

0

In our model, Sigmoid and ReLu performed the best due to their low test MAE and low test loss. This could be because since calculating the age of abalone has a linear relationship with the features, sigmoid was able to show its primary strength. For ReLu, it was also effective since adding a bit of nonlinearity did lower the test loss by a percent.