COMS 4030A

Adaptive Computation and Machine Learning

EXERCISES:

(1) Given the function $f(w) = \sigma(3w-2)$ and the point $\underline{w} = 0.5$, apply the gradient descent update rule with $\eta = 0.1$.

Solution:

$$f(w) = \frac{1}{1+e^{3w-2}},$$

$$\tfrac{df}{dw} = \frac{3e^{-(3w-2)}}{(1+e^{-(3w-2)})^2}$$

$$\tfrac{df}{dw}\Big|_{0.5} = \frac{3e^{-(3w-2)}}{(1+e^{-(3w-2)})^2}\Bigg|_{0.5} = \frac{3e^{-(3(0.5)-2)}}{(1+e^{-(3(0.5)-2)})^2} = 0.705$$

Thus,

$$\underline{w} \leftarrow \underline{w} - \eta(0.705) = 0.5 - (0.1)(0.705) = 0.4295.$$

(2) Given the function $F(w_1, w_2) = e^{(2w_1-3w_2+1)}$ and the point $\underline{\boldsymbol{w}} = (\underline{w}_1, \underline{w}_2) = (0.4, 0.5)$, apply the gradient descent update rule with $\eta = 0.1$.

Solution:

$$\frac{\partial F}{\partial w_1} = 2e^{(2w_1-3w_2+1)} \qquad\qquad \frac{\partial F}{\partial w_2} = -3e^{(2w_1-3w_2+1)}$$

$$\frac{\partial F}{\partial w_1}\Bigg|_{(0.4,0.5)} = 2e^{(2w_1-3w_2+1)}\Bigg|_{(0.4,0.5)} = 2e^{(2(0.4)-3(0.5)+1)} = 2.300$$

$$\frac{\partial F}{\partial w_2}\Bigg|_{(0.4,0.5)} = -3e^{(2w_1-3w_2+1)}\Bigg|_{(0.4,0.5)} = -3e^{(2(0.4)-3(0.5)+1)} = -4.050$$

Thus,

$$\underline{w}_1 \leftarrow \underline{w}_1 - \eta(2.300) = 0.4 - (0.1)(2.300) = 0.17.$$

$$\underline{w}_2 \leftarrow \underline{w}_2 - \eta(-4.050) = 0.5 - (0.1)(-4.050) = 0.905.$$

EXERCISES:

(1) If $y = \sigma(z)$, show that $\frac{dy}{dz} = \sigma(z)(1 - \sigma(z)) = y(1-y)$.

Solution:

$$y = \sigma(z) = \frac{1}{1+e^{-z}} = (1+e^{-z})^{-1}$$

Therefore,

$$\begin{aligned}
\tfrac{dy}{dz} &= (-1)(1+e^{-z})^{-2}(e^{-z})(-1) \\
&= \frac{e^{-z}}{(1+e^{-z})^2} \\
&= \left(\frac{1}{1+e^{-z}}\right)\left(\frac{e^{-z}}{1+e^{-z}}\right) \\
&= \left(\frac{1}{1+e^{-z}}\right)\left(\frac{1+e^{-z}-1}{1+e^{-z}}\right) \\
&= \left(\frac{1}{1+e^{-z}}\right)\left(1 - \frac{1}{1+e^{-z}}\right) \\
&= \sigma(z)(1-\sigma(z)) \\
&= y(1-y).
\end{aligned}$$

(2) Consider a network with 2 input nodes, one hidden layer with 2 nodes, and 2 output nodes. The weights and the bias values are given by W_1, W_2, $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$:

$$W_1 = \begin{bmatrix} 1 & -1 \\ -3 & -2 \end{bmatrix} \quad W_2 = \begin{bmatrix} -2 & 0 \\ 0 & 3 \end{bmatrix} \quad \boldsymbol{b}_1 = (0,1) \quad \boldsymbol{b}_2 = (1,-2).$$

The activation function at each node is the sigmoid function.

Suppose we want to train the network using input $\boldsymbol{x} = (1,-2)$ with target $\boldsymbol{t} = (1,0)$.

(a) Calculate the output vector $\boldsymbol{y} = (y_1, y_2)$ for the input $\boldsymbol{x}$.

Solution: $\boldsymbol{y} = (0.269, 0.720)$

(b) Compute the sum-of-squares loss for the input $\boldsymbol{x}$.

Solution: 0.5264

(c) Do the weight updates for each edge weight and bias according to the Neural Network Training Algorithm with sum-of-squares loss and sigmoid activation functions.

Solution: The new values are (please check ... and note that rounding errors may give different answers):

$$W_1 = \begin{bmatrix} 0.99997 & -1.0008 \\ -2.9999 & -1.9985 \end{bmatrix} \quad W_2 = \begin{bmatrix} -1.986 & -0.015 \\ 0.014 & 2.986 \end{bmatrix}$$

$$\boldsymbol{b}_1 = (-0.00001, 0.999) \quad \boldsymbol{b}_2 = (1.014, -2.015).$$

(d) After the updates, feed $\boldsymbol{x}$ into the updated network to get the output.

Solution: $\boldsymbol{y} = (0.2778, 0.711)$

(e) Compute the sum-of-squares loss for the input $\boldsymbol{x}$ and compare with (b) (it should have decreased).

Solution: 0.5134

(3) Consider a neural network that uses the sum-of-squares loss function and the *relu* activation function at each layer. Do the computation in Section 2.15 to obtain the rules for computing delta values, that is, obtain the formulas (8) and (9) in the case of the *relu* activation function.

Solution: We need to calculate the values δ_n at the output nodes, where

$$\delta_n = (a_n - t_n)\left(\tfrac{dy_n}{dz_n}\Big|_{\boldsymbol{xtW}}\right)$$

so we need to calculate $\frac{dy_n}{dz_n}\big|_{\boldsymbol{xtW}}$ in the case that g_n is *relu*. Recall that

$$relu(z_n) = \begin{cases} z_n & \text{if } z_n > 0 \\ 0 & \text{if } z_n \leq 0 \end{cases}.$$

For $y_n = relu(z_n)$, therefore,

$$\tfrac{dy_n}{dz_n} = \begin{cases} 1 & \text{if } z_n > 0 \\ 0 & \text{if } z_n \leq 0 \end{cases}$$

Then,

$$\tfrac{dy_n}{dz_n}\Big|_{\boldsymbol{xtW}} = \begin{cases} 1 & \text{if } z_n\big|_{\boldsymbol{xtW}} > 0 \\ 0 & \text{if } z_n\big|_{\boldsymbol{xtW}} \leq 0 \end{cases}$$

The following observation simplifies things a bit:
$a_n = z_n\big|_{\boldsymbol{xtW}}$ if $z_n\big|_{\boldsymbol{xtW}} > 0$, and $a_n = 0$ if $z_n\big|_{\boldsymbol{xtW}} \leq 0$.
Using this observation, we can write

$$\tfrac{dy_n}{dz_n}\Big|_{\boldsymbol{xtW}} = \begin{cases} 1 & \text{if } a_n > 0 \\ 0 & \text{if } a_n = 0 \end{cases}$$

This means that the derivative $\frac{dy_n}{dz_n}\big|_{\boldsymbol{xtW}}$ can be obtained just from the a_n value.

Then, the rule for δ_n is as follows:

$$\delta_n = \begin{cases} a_n - t_n & \text{if } a_n > 0 \\ 0 & \text{if } a_n = 0 \end{cases}$$

For a node m in a hidden layer, δ_m is given by:

$$\delta_m = \left(\sum_n \delta_n \, \underline{w}_{mn}\right) \left(\frac{dy_m}{dz_m}\Big|_{\boldsymbol{xtW}}\right)$$

where the n's range over the nodes in the layer after m's layer.

Since we are using $relu$ as the activation function at the hidden layer, we have that $y_m = relu(z_m)$. Using the above derivative,

$$\frac{dy_m}{dz_m}\Big|_{\boldsymbol{xtW}} = \begin{cases} 1 & \text{if } a_m > 0 \\ 0 & \text{if } a_m = 0 \end{cases}$$

we obtain

$$\delta_m = \begin{cases} \sum_n \delta_n \, \underline{w}_{mn} & \text{if } a_m > 0 \\ 0 & \text{if } a_m = 0 \end{cases}$$

where n ranges over all nodes in the layer after m's layer.

(4) Try exercise (3) with the activation function lin.

Solution: For an output node n, if $y_n = lin(z_n) = z_n$, then $\frac{dy_n}{dz_n} = 1$. Thus,

$$\delta_n = a_n - t_n,$$

and for a node m in a hidden layer,

$$\delta_m = \sum_n \delta_n \, \underline{w}_{mn}$$

where n ranges over all nodes in the layer after m's layer.

(5) Try exercise (3) with the activation function $tanh$.

You need to show that if $y = tanh(z)$, then $\frac{dy}{dz} = 1 - tanh^2(z) = 1 - y^2$.

Solution:

$$y = tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

Therefore, by the quotient rule for differentiation,

$$\begin{aligned}\tfrac{dy}{dz} &= \frac{(e^z + e^{-z})(e^z - (-e^{-z})) - (e^z - e^{-z})(e^z + (-e^{-z}))}{(e^z + e^{-z})^2} \\ &= \frac{(e^z + e^{-z})^2 - (e^z - e^{-z})^2}{(e^z + e^{-z})^2} \\ &= 1 - \frac{(e^z - e^{-z})^2}{(e^z + e^{-z})^2} \\ &= 1 - \left(\frac{e^z - e^{-z}}{e^z + e^{-z}}\right)^2 \\ &= 1 - (tanh(z))^2 \\ &= 1 - y^2.\end{aligned}$$

For an output node n, if $y_n = tanh(z_n)$, then

$$\tfrac{dy_n}{dz_n}\Big|_{\boldsymbol{xtW}} = (1 - y_n^2)\Big|_{\boldsymbol{xtW}} = 1 - a_n^2.$$

Thus,

$$\delta_n = (a_n - t_n)(1 - a_n^2),$$

and for a node m in a hidden layer,

$$\delta_m = \left(\textstyle\sum_n \delta_n \, \underline{w}_{mn}\right)(1 - a_m^2)$$

where n ranges over all nodes in the layer after m's layer.