Caige Middaugh
10/10/21
Intro to Machine Learning

Assignment 2

1 Theoretical Part

1.1 Gradient Descent Derivation

We have the equation $o = w0 + w1(x1+x1^2) +\ldots+ wn(xn+xn^2)$ for the predicted value. The error function we use is $E_i = \frac{1}{2} \sum_{d \in D}(t_{id} - o_{id})^2$ here is is the training example instance number, and i is the specific weight number we will be deriving with respect to such as w1 or wn. The training rule for gradient descent is $\Delta wij = -\eta*(\partial Ei/\partial wij)$ which the $(\partial Ei/\partial wij)$ represents the gradient of the weight vector.

$$(\partial Ei/\partial wij) = \partial(1/2\Sigma d\in D(tid - oid)^2)/\partial wij$$

We will take the derivative of the inside.

$$= 1/2\Sigma d\in D2(tid - oid)*\partial(tid - oid)/\partial wi$$

$$= \Sigma d\in D(tid - oid)*\partial(-oid)/\partial wi$$

Here the tid in the derivative goes to 0 since there is no weights at all in the target.

$$= \Sigma d\in D(tid - oid)*\partial(-f(x)*\Sigma k=1 \text{ to } n(wik*xkd))/\partial wi$$

The f(x) is the activation function such as sigmoid, or sign activation functions, etc.

For simplicity I will summarize and say **sum_id = Σn, k=1(wik*xkd)**

$$= \Sigma d\in D(tid - oid)*\partial(-f(x)*sum\_id)/\partial wi$$

$\partial f(g(x))/\partial x = \partial f/\partial g(x)\times\partial g(x)/\partial x$ Hence we apply this to

$$\partial(f(x)*(sum\_id))/\partial(wij) = \partial(f(sumid))/\partial(sum\_id) * \partial(sum\_id)/\partial(wi)$$

$$\partial(sum\_id)/\partial(wi) = \partial(\Sigma n,\ k=1\ w\_ik*x\_kd)/\partial(w\_ij)$$

$$= \partial(w\_i1(x\_1d+x\_1d^2) +\ldots+ w\_in(x\_nd+x\_nd^2))/\partial(w\_ij)$$

$$= \partial(w\_i1*(x\_1d+x\_1d^2))/\partial(w\_ij) +\ldots+ \partial(w\_ij*(x\_jd+x\_jd^2))/\partial(w\_ij) +\ldots+ \partial(w\_in*(x\_nd+x\_nd^2))/\partial w\_ij$$

$$= 0 +\ldots+ (x\_jd+x\_jd^2) +\ldots+ 0 = (x\_jd+x\_jd^2)$$

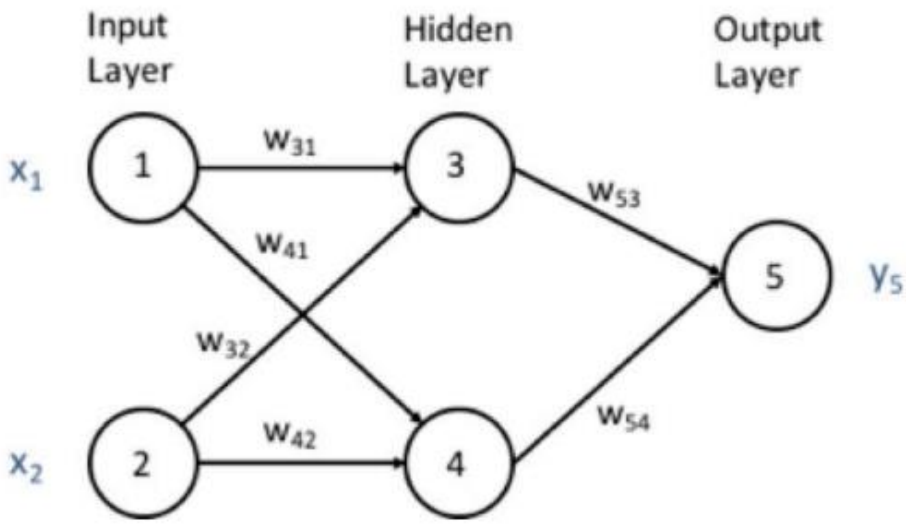
Now let $h = \partial(f(x))/\partial(wij)$, thus we get

$$(\partial Ei/\partial wij) = -\Sigma d\in D(tid - oid)*h*(x\_jd+x\_jd^2)$$

I am leaving the activation function as a general answer since it was not provided in the question. With this, we can now define our training rule for the prediction function.

$$\Delta wij = -\ \eta*(-\Sigma d\in D(tid - oid)*h*(x\_jd+x\_jd^2))$$

1.2 Comparing Activation Function



a.)

This answer assumes that the equation for o for 1.1 is not applied. If that is the case refer to part a part 2.

Hidden layer Computation:

net3=W31*x1 + W32*x2   net4=W41*x1+W42*x2

x3 = Node3 = h(net3)   x4=node4= h(net4)

Output Layer Computation:

net5 = W53*x3 + W54*x4   y5 = h(net5) = h(W53*x3 + W54*x4)

= h(W53*h(W31*x1 + W32*x2) + W54*h(W41*x1+W42*x2)

a part 2.)

net3=W31*(x1+x1^2) + W32*(x2+x2^2)   net4=W41*(x1+x1^2)+W42*(x2+x2^2)

x3 = Node3 = h(net3)   x4=node4= h(net4)

Output Layer Computation:

net5 = W53*(x3+x3^2) + W54*(x4+x4^2)

y5 = h(net5) = h(W53*(x3+x3^2) + W54*(x4+x4^2))

= h(W53*h((W31*(x1+x1^2)+W32*(x2+x2^2))+(W31*(x1+x1^2) + W32*(x2+x2^2))^2 + W54*h((W41*(x1+x1^2)+W42*(x2+x2^2)) + (W41*(x1+x1^2)+W42*(x2+x2^2))^2) b.)

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

$$\begin{pmatrix} net3 \\ net4 \end{pmatrix} = W^{(1)} \cdot X = net_{34} \qquad x_{34} = \begin{pmatrix} x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} h(net3) \\ h(net4) \end{pmatrix}$$

$$net_5 = (net_5) = W^{(2)} \cdot x_{34} \quad y_5 = (y_5) = (h(net_5))$$

$$y_5 = (h(w^{(2)} \cdot x_{34})) = (h(w^{(2)} \cdot \begin{pmatrix} h(net_3) \\ h(net_4) \end{pmatrix}))$$

$$= \left(h\left(w^{(2)} \cdot \begin{pmatrix} h(w_{31} \cdot x_1 + w_{32} \cdot x_2) \\ h(w_{41} \cdot x_1 + w_{42} \cdot x_2) \end{pmatrix}\right)\right)$$

$$= \left(h\begin{pmatrix} w_{53} \cdot h(w_{31} \cdot x_1 + w_{32} \cdot x_2) \\ w_{54} \cdot h(w_{41} \cdot x_1 + w_{42} \cdot x_2) \end{pmatrix}\right)$$

c.)

sigmoid

$\sigma(t) = h_s(t) = \frac{1}{1+e^{-x}}$

Tanh

$\tanh(x) = h_t(t) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

we can say $1-\sigma(t) = \sigma(-t)$ since there is symmetry in the sigmoid function and has constraint range [0,1]

$$1 - \frac{1}{1+e^{-x}} = \frac{1}{1+e^{x}} = 1-\sigma(x) \text{ let this be } ①$$

I must now show tanh(t), and I will start with adding and subtracting e^{-x}.

$$h_t(t) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^x - e^{-x} + e^{-x} - e^{-x}}{e^x + e^{-x}} = \frac{e^x + e^{-x} - 2e^{-x}}{e^x + e^{-x}}$$

$$= \cancel{\frac{e^x + e^{-x}}{e^x + e^{-x}}} - \frac{2e^{-x}}{e^x + e^{-x}} = 1 - \frac{2e^{-x}}{e^x + e^{-x}} = 1 - \frac{2}{e^x(e^x + e^{-x})}$$

$$= 1 - \frac{2}{e^x \cdot e^x + e^x \cdot e^{-x}} = 1 - \frac{2}{e^{2x}+1} = 1 - 2\sigma(-2x) - \text{From } ①$$

$$= 1 - 2(1 - \sigma(2t)) \rightarrow \text{From logic to get } ①$$

$$= 1 - 2 + 2\sigma(2x = 2\sigma(2t) - 1 = 2h_s(2x) - 1$$

Hence $h_t(t) = 2h_s(2x) - 1$, note tanh(t) is just sigmoid but different by a constant rate of 2 and subtracted by constant value 1. Because the any difference is constants then they can generate same output functions.