Single Node Math Implementation with Coding

Notebook:	Deep Learning		
Created:	10/4/2020 4:44 PM	**Updated:**	10/4/2020 5:34 PM
Author:	Aninda Ghosh		

Neural Network from Scratch

(General AI) Deep learning.

Back propagation [performing 1st order differentiation]

finding the gradient

Input nodes $\xrightarrow{w_i}$ Dense node → Output → Classification / Regression

Activation (Relu)

non-linear

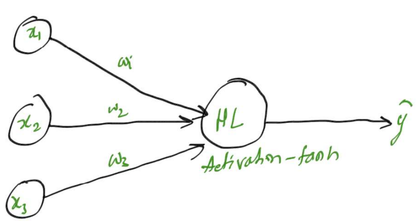
Can use (tanh)

Differentiation of what?

the loss function (1) Mean square Error

(2) Absolute difference.

1 node NN (Implementation from Scratch)



Forward propagation Block

$$\hat{y} = f(x_1W_1 + x_2W_2 + x_3W_3 + b_0) \quad \text{— (i)}$$

f = tanh

$$\boxed{\hat{y} = \tanh\left(\sum_{i=0} x_i w_i\right)}$$

Training Phase

let's findout the error.

y = actual output $\quad \hat{y}$ = Calculated output

$$e = (y - \hat{y}) \quad \text{— (ii)}$$

from (i) & (ii)

$$e = \left[y - \tanh(xW^T)\right]^2 \quad \text{— (iii)}$$

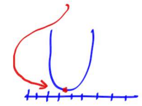
e is also called (1) Cost function (2) Error function

$\boxed{\text{TARGET}}$ → MINIMUM e

using 1st order derivative of e wrt w will be calculated.

... is the inflection points of the

i.e. $\frac{\partial e}{\partial W} = 0$ This gives ... curve]



We need the weights value which will make the above equation almost close to 0

Since we have 3 independent inputs we can calculate the individual derivatives

i.e $\rightarrow$ $\frac{\partial e}{\partial w_1} \simeq 0$ $\quad \frac{\partial e}{\partial w_3} \simeq 0$

$\frac{\partial e}{\partial w_2} \simeq 0$

$$\frac{\partial e}{\partial w_1} = \frac{\partial}{\partial w_1}\left[y - \tanh(x_1 w_1 + x_2 w_2 + x_3 w_3 + 0)\right]^2$$

[Considering bias = 0] lets assume

$$= \frac{\partial}{\partial w_1}\left[y - \tanh(xW^T)\right]^2$$

$$= \frac{\partial}{\partial w_1}\left[y^2 - 2y\tanh(xW^T) + \tanh(xW^T)^2\right]$$

$$= \frac{\partial}{\partial w_1}(y^2) - \frac{\partial}{\partial w_1}\left(2y\tanh(x_1 w_1 + k)\right) + \frac{\partial}{\partial w_1}\tanh(x_1 w_1 + k)^2$$

$\frac{\partial}{\partial x}(\tanh x) = 1 - \tanh(x)^2$

$$= 0 - 2y\left(1 - \tanh^2(x_1 w_1 + k)\right)x_1 + 2\tanh(x_1 w_1 + k)\left(1 - \tanh^2(x_1 w_1 + k)\right)x_1$$

$$= -2yx_1 + 2yx_1\tanh^2(x_1 w_1 + k) + 2x_1\tanh(x_1 w_1 + k) - 2x_1\tanh^3(x_1 w_1 + k)$$

= from eq (i) [For computation]

$$= -2yx_1 + 2yx_1(\hat{y})^2 + 2x_1\hat{y} - 2x_1(\hat{y})^3$$

$$= 2x_1(\hat{y} - y) + 2x_1 y(\hat{y})^2 - 2x_1(\hat{y})^3$$

$$= 2x_1\left(-y + \hat{y} + y(\hat{y})^2 - (\hat{y})^3\right) \quad \text{— (iv)}$$

Similarly

$$\frac{\partial e}{\partial w_2} = 2x_2\left(-y + \hat{y} + y(\hat{y})^2 - (\hat{y})^3\right) \quad \text{— (v)}$$

$$\frac{\partial e}{\partial w_3} = 2x_3\left(-y + \hat{y} + y(\hat{y})^2 - (\hat{y})^3\right) \quad \text{— (vi)}$$

We need to Reduce the error on a global level

These are my individual adjustments.

$$\boxed{\Delta W \propto \frac{\partial E}{\partial W}}$$

i.e. $W_{1(new)} = W_{1(old)} - \frac{\partial e}{\partial w_{1(old)}}$

$W_{2(new)} = W_{2(old)} - \frac{\partial e}{\partial w_{1(old)}}$

$W_{3(new)} = W_{3(old)} - \frac{\partial e}{\partial w_{3(old)}}$