↑ for ~~my tensorflow~~ submission

~~as well I had~~ sub

The model Created By ~~Ao~~ no-framework

* (Only one hidden-layer.)

Explanation:→ ***(For my tensorflow Submission as well I ~~co~~ Commented By code Propery and Submitted one more doc)

1) Imported numpy, Panda

(2) csv file through ~~internet~~ Panda.

(3) ~~Sa~~ Stored in an array (name-data)

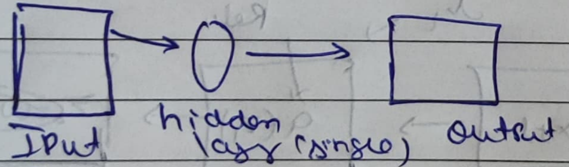
(4) Shuffled data and split ~~for~~ it for training.

Functions for neural network.

(i) init_Params:→ it will intialize the parameter weight and bias, ~~it~~ I took random value in initialy weight and bias.

(ii) defined softmax (for final output Prediction)

· ~~Re~~ relu (activation for each layer though i used only one hidden layer)



(iii) For forward Propagation:→

as we know the maths

we Create $Z_1 = W \cdot X + B$ ... for~~ward~~

weight Bias

Z_1 is linear function

we pass it through Activation function Which also make it non-linear and changes it range.

so $A(Z_1)$, here A is relu → for hidden layer

so, $Z_2 = W_2 A(Z_1) + b_2$ ... for ~~output~~ last layer

for final output

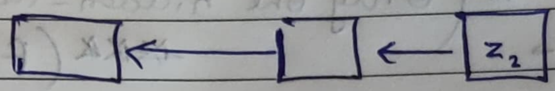
we use $A_2 = \text{Softmax}(A_2)$

give us value between

Completed all video

Refrence :→ Cousera Andrew N.S. deep learning Course

(IV) Backword Propasation :→



Derivation of Cost function 1

$dZ_2 = A_2 - Y$

$dw_2 = (dZ_2 * A1.T)/m$

$db_2 = \frac{sum(dZ_2)}{m}$

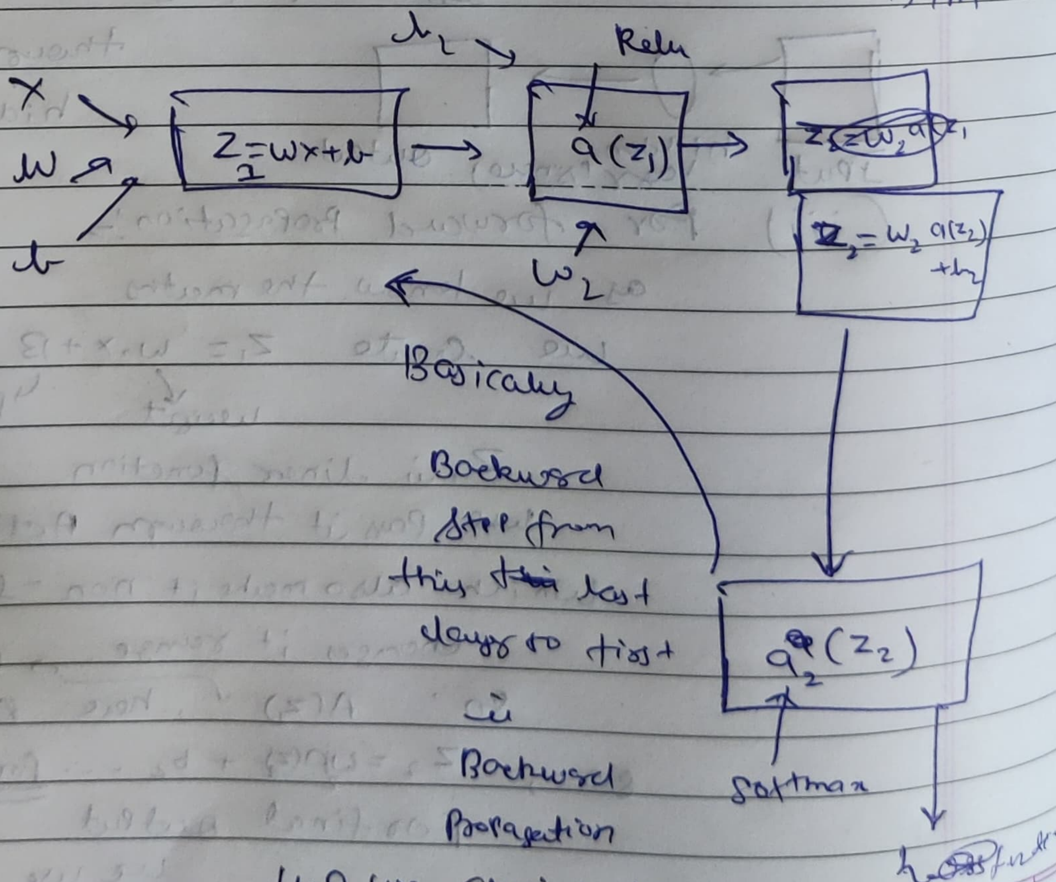
$dz_1 = (w_2.T * dZ_2) * g'1(z$

Element wise Product

$dw_1 = (dz_1 * \overset{X}{\cancel{A}}.T)/m$

$db = sum(dz_1)/m$

Wrote this function



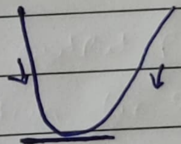
Basically Backward step from this ~~this~~ last layers to first is Backward Propagation We use Chainrule to derive

(V) Updating Peramets By deciding α (learning rate)

$$w_2 = w_2 - \alpha dw_2$$

Similiar $\left\{ \vdots \right.$
for
w_1, w_2, b_1, b_2

(VI) gradient descent :→ it is main function which iterate no. of times and written the final W_1, W_2, b_1, b_2 By training the Paramter (or say remembering the given data)



(VII) cost function are used to make Prediction using the $w_1 W_2, b_1, b_1$ we we got after gradient descent.