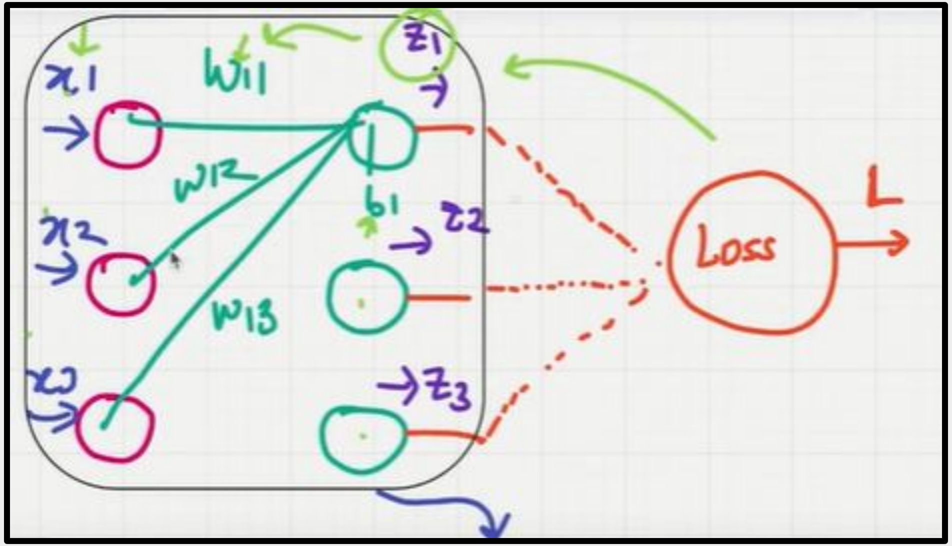
BACKWARD PROPAGATION IN NEURAL NETWORKS FROM SCRATCH

EXPLANATION OF DENSE CLASS WITH BACKPROPAGATION

DERIVATIVES OF BAIS AND WEIGHTS NEURAL NETWORKS



DERIVATIVES OF BAIS AND WEIGHTS NEURAL NETWORKS

- z1=w11(x1)+
  w12(x2)+
  w13(x3)+b

- Partial derivative of loss Wrt x1: It effects z1,z2, z3. In z1,z2,z3 it appears as x1

$$\frac{\partial L}{\partial w_{11}} = \frac{\partial L}{\partial z_1} * \frac{\partial z_1}{\partial w_{11}} = \boxed{\frac{\partial L}{\partial z_1}} * x_1$$

$$\frac{\partial L}{\partial b_1} = \frac{\partial L}{\partial z_1} * \frac{\partial z_1}{\partial b_1} = \frac{\partial L}{\partial z_1}$$

$$\frac{\partial L}{\partial x_1} = \frac{\partial L}{\partial z_1} * \frac{\partial z_1}{\partial x_1} + \frac{\partial L}{\partial z_2} * \frac{\partial z_2}{\partial x_1} + \frac{\partial L}{\partial z_3} * \frac{\partial z_3}{\partial x_1}$$

$$= \frac{\partial L}{\partial z_1} * w_{11} + \frac{\partial L}{\partial z_2} * w_{21} + \frac{\partial L}{\partial z_3} * w_{31}$$

DERIVATIVES OF BAIS AND WEIGHTS NEURAL NETWORKS - Matrices

In matrix form

$$W: \begin{bmatrix} w_{11} & w_{21} & w_{31} \\ w_{12} & w_{22} & w_{32} \\ w_{13} & w_{23} & w_{33} \end{bmatrix} \quad X: \text{Inputs}: \begin{bmatrix} x_1 & x_2 & x_3 \\ x_1 & x_2 & x_3 \\ x_1 & x_2 & x_3 \end{bmatrix};$$

$$dL\text{-}dZ: \begin{bmatrix} \partial L/\partial z_1 & \partial L/\partial z_2 & \partial L/\partial z_3 \\ \partial L/\partial z_1 & \partial L/\partial z_2 & \partial L/\partial z_3 \\ \partial L/\partial z_1 & \partial L/\partial z_2 & \partial L/\partial z_3 \end{bmatrix}$$

DERIVATIVES OF BAIS AND WEIGHTS NEURAL NETWORKS - Conclusion

- **Partial Derivative of loss wrt to weight:**

  Transpose (Inputs_matrix) , DL/DZ (partial der wrt z) => np.dot
- **Partial Derivative of loss wrt to bais:**

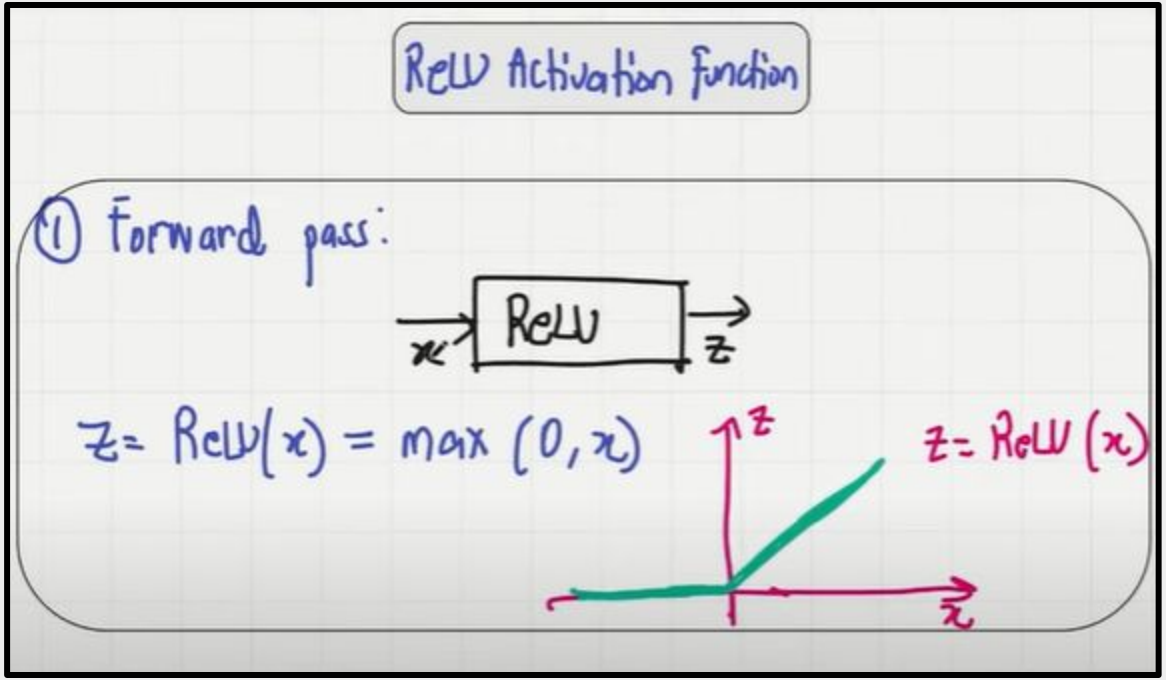
  Sum of rows of dl_dz => np.sum gives(1,3) dimensional result

  Axis = 0 gives every column sum
- **Partial Derivative of loss wrt input of next Layer:**

  *DL/DZ (partial der wrt z) * Transpose (Weights_matrix)*

EXPLANATION OF RELU WITH BACKPROPAGATION

EXPLANATION OF RELU WITH BACKPROPAGATION



EXPLANATION OF RELU WITH BACKPROPAGATION

(2) Backward pass:

$$x \rightarrow \boxed{\text{ReLU}} \rightarrow z$$

Given: $\frac{\partial L}{\partial z}$. To find: $\frac{\partial L}{\partial x}$

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial z} * \frac{\partial z}{\partial x} \rightarrow \begin{pmatrix} 1 \text{ if } x > 0 \\ 0 \text{ if } x \leq 0 \end{pmatrix}$$

Given: $\boxed{\frac{\partial L}{\partial z}}$ To find: $\frac{\partial L}{\partial x}$

$$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial z} * \frac{\partial z}{\partial x} \rightarrow \begin{pmatrix} 1 \text{ if } x > 0 \\ 0 \text{ if } x \leq 0 \end{pmatrix}$$

In practice: set answer = $\frac{\partial L}{\partial z}$

Find all index where $x \leq 0$.

Set all values corresponding to those index

EXPLANATION OF SOFTMAX AND LOSS WITH BACKPROPAGATION

To compute the gradient of loss wrt to softmax inputs,
We have a formula

$$[dL/dx1 , dL/dx2, dL/dx3] = \text{Actual} - \text{Prediction}$$

We divide by number of samples to as to prevent gradient explosion

FORMULA OF SGD OPTIMIZER

$$w = w - \eta \cdot \nabla L(w)$$