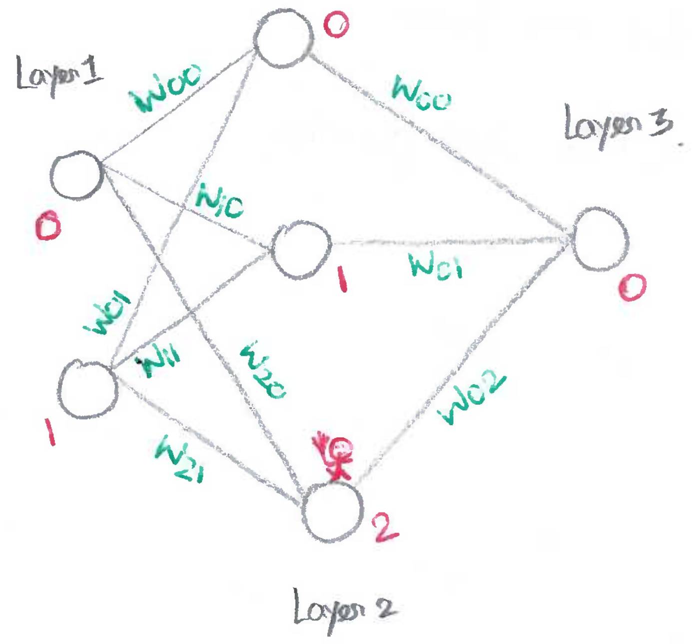
[2,3,1]

BACK PROPAGATION



Weight_Matrix (L1-L2) = $\begin{bmatrix} 3 \times 2 \end{bmatrix}$

Weight_Matrix (L2-L3) = $[\ 1 \times 3\ ]$

bias_L2 = $[\ 1 \times 3\ ]$

bias_L3 = $[\ 1 \times 1\ ]$

W_{jk}^{l} → weight for the connection from the k^{th} neuron in the $(l-1)^{th}$ layer to the j^{th} neuron in the l^{th} layer.

b_j^l → Bias for the j^{th} neuron in the l^{th} layer

a_j^l → activation of the j^{th} neuron in the l^{th} layer

δ_j^l → Delta is the error in the j^{th} neuron in the l^{th} layer (Measure of error in the NEURON)

BACK PROPAGATION FOUR FONDAMENTAL EQUATIONS

* Compute the ERROR for each Layer

* Compute gradient of the Cost function

1) Equation for the error in the output Layer

$$\delta_j^L = \frac{\partial C}{\partial a_j^L} \sigma'(z_j^L)$$

$\sigma'(z_j^L) \rightarrow$ FEED FORWARD ALGORITHM

$\frac{\partial C}{\partial a_j^L} \rightarrow$ Differentiate COST FONCTION

$$\boxed{C = \frac{1}{2} \sum_j (y_j - a_j^L)^2}$$

$$\delta^L = (a^L - y) \odot \sigma'(z^L) \quad \text{—— } (1)$$

$$\text{delta} = \underbrace{\text{self.cost\_derivative}(\text{activations}[-L], y)}_{\text{RATE OF CHANGE OF "C" WITH RESPECT TO THE OUTPUT ACTIVATIONS}} * \text{Sigmoid\_Prime}(\text{ZS}[-L])$$

2) Equation for the error δ^L in terms of error in Next Layer δ^{L+1}

$$\delta^L = ((w^{L+1})^T \delta^{L+1}) \odot \sigma'(z^L) \quad \text{—— } (2)$$

```
SP = Sigmoid_Prime (Z)
delta = np.dot (self.weight [-L+1].transpose() , delta ) * SP
```

Suppose we know the error δ^{L+1} at the $(L+1)^{th}$ Layer, when we apply the transpose weight matrix, $(w^{L+1})^T$, we can think Intiudively of this as moving the ERROR BACKWARD through the network, gives us some sort of measure of the ERROR at the output of the l^{th} Layer.

3) Equation for rate of change of Cost with respect to any BIAS in the network

$$\frac{\partial c}{\partial b_j^L} = \delta_j^L$$

δ_j^L is exactly equal to the rate of change $\frac{\partial c}{\partial b_j^L}$, this is already computed in Equation (1) so

$$\frac{\partial c}{\partial b} = \delta \quad \text{———} \quad (3)$$

nabla_b [-L] = delta

Suppose we know the error δ^{L+1} at the $(L+1)^{th}$ Layer, when we apply the transpose weight matrix, $(w^{L+1})^T$, we can think Intiutively of this as moving the ERROR BACKWARD through the network, gives us some sort of measure of the ERROR at the output of the l^{th} Layer.

3) Equation for rate of change of cost with respect to any BIAS in the network

$$\frac{\partial c}{\partial b_j^L} = \delta_j^L$$

δ_j^L is exactly equal to the rate of change $\frac{\partial c}{\partial b_j^L}$, this is already computed in Equation 1 so

$$\frac{\partial c}{\partial b} = \delta \quad \text{———} \quad (3)$$

nabla_b [-L] = delta

4) Equation for the rate of change of the cost with respect to any weight in the network

$$\frac{\partial c}{\partial w_{jk}^{L}} = a_k^{L-1}\,\delta_j^{L} \quad \text{—} \ (4)$$

this equation can be written as

$$\frac{\partial c}{\partial w} = a_{in}\,\delta_{out}$$

$a_{in} \rightarrow$ activation of the neuron Input to the weight "w" and "δ_{out}" is the error of the neuron output from w

nabla_w[-L] = np.dot(delta, activations[-2].transpose())

TEST STATISTICS

TOTAL TEST IMAGES = 200

POSITIVE PREDICTION = 150

NEGATIVE PREDICTION = 50

0 – POSITIVE

1 – NEGATIVE

	0	1
Non crater 0	TP 25	FN 20
crater 1	FP 30	TN 125

CRATER LABELLED INCORRECTLY AS NON CRATER

TP = 25/200. Refers to the Positive tuple that were correctly labelled by NN

TN = 125/200. Refers to the Negative tuple that were correctly labelled by NN

FP = 30/200. These are the Negative tuple that we In-correctly labelled as Positive

FN = 20/200. These are the Positive tuple that were mis labeled as Negative