Spiral Branch Determiner Project

- Activation Function:
  - ↳ Hidden Layers: Rectified Linear (ReLU) Activation Function:

$$\boxed{f(x) = \max(0,x)}$$

Derivative:

$$\boxed{f'(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases}}$$

  - ↳ Output Layer: Softmax activation function:

$$S_i = \frac{e^{z_i}}{\sum_{j=0}^{n} e^{z_j}}$$

- To prevent overflow we will actually use the following:

$$\boxed{S_i = \frac{e^{(z_i - \max(z))}}{\sum_{j=0}^{n} e^{(z_j - \max(z))}}}$$

Derivative:

$$\boxed{\frac{\partial S_i}{\partial z_k} = \begin{cases} S_i - S_i^2, & \text{if } k=i \\ -S_i \cdot S_k, & \text{otherwise} \end{cases}}$$

$$S_i(\delta_{ik} - S_k)$$

$$\Rightarrow S_k(\delta_{ik} - S_i)$$

$$\begin{cases} \delta = 1 & \text{if } k=i \\ \delta = 0 & \text{otherwise} \end{cases}$$

- Input:
  - ↳ P points (x,y) in the cartesian plane.
  - ↳ Thus, we have an input layer with 2 neurons

- Output/ Desired output:
  - ↳ Desired output: array y where $y[i] = c$ means that the ith point belongs to class c.
  - ↳ we can have as many output classes we want, as well as we can have as many samples (points) we want.
  - ↳ The predicted output is given by the class with higher probability outputed in the last layer (after applying the softmax activation function

- Loss/ cost Function:
  - The categorical cross entropy loss Function:

$$\boxed{C = -\sum_j y_j \log \hat{y}_j} \Rightarrow C = -\log \hat{y}_k$$

, where k is the true class.

Obs.: to avoid overflow, $\hat{y}$ will be clipped:

$$\hat{y} = \begin{cases} \hat{y}, \text{ if } 1e{-7} \leq \hat{y} \leq 1 - 1e{-7} \\ 1 - 1e^{-7}, \text{ if } \hat{y} > 1 - 1e^{-7} \\ 1e^{-7}, \text{ if } \hat{y} < 1e^{-7} \end{cases}$$

→ avoiding average Loss shifting.

Derivative :

$$\boxed{\frac{\partial C}{\partial \hat{y}_k} = \frac{-y_k}{\hat{y}_k}}$$

- Gradient Descent :

1) Formulas to calculate gradient with respect to the last layer:

$$\frac{\partial C}{\partial x_i^L} = \frac{\partial C}{\partial s_k}\left(\sum_{j=0}^{m_L} \frac{\partial s_k}{\partial r_j}\cdot\frac{\partial r_j}{\partial z_j}\cdot\frac{\partial z_j}{\partial x_i}\right)$$

$$= \frac{-1}{\cancel{s_k}}\left(\sum_{j=0}^{m_L} \cancel{s_k}\underbrace{(\delta(k,j)-s_j)\cdot(z_j>0\,?\,1:0)}_{\text{softmax Relu Derivatives}}\cdot w_{ij}\right)$$

$$\frac{\partial C}{\partial w_{ip}^L} = \frac{\partial C}{\partial s_k}\left(\sum_{j=0}^{m_L} \frac{\partial s_k}{\partial r_j}\cdot\frac{\partial r_j}{\partial z_j}\cdot\overbrace{\frac{\partial z_j}{\partial w_{ip}^L}}^{0 \text{ for each } j\neq p}\right)$$

$$= \frac{-1}{\cancel{s_k}}\,\cancel{s_k}(\delta(k,p)-s_p)(z_p>0\,?\,1:0)\cdot x_i$$

$$= (s_p-\delta(k,p))(z_p>0\,?\,1:0)\,x_i$$

$$\frac{\partial C}{\partial b_p^L} = \frac{\partial C}{\partial s_k}\left(\sum_{j=0}^{m_L} \frac{\partial s_k}{\partial r_j}\cdot\frac{\partial r_j}{\partial z_j}\cdot\overbrace{\frac{\partial z_j}{\partial b_p}}^{0 \text{ for each } j\neq p}\right)$$

$$\vdots$$

$$= (s_p-\delta(k,p))(z_p>0\,?\,1:0)$$

2) Formulas to calculate gradient with respect to a layer having the gradiant of the next layer:

$$\frac{\partial C}{\partial x_i^l} = \sum_{j=0}^{m_l} \underbrace{\frac{\partial C}{\partial a_j^l}}_{\frac{\partial C}{\partial x_j^{l+1}}} \cdot \frac{\partial a_j^l}{\partial x_i^l} = \sum_{j=0}^{m_l} \frac{\partial C}{\partial x_j^{l+1}} \cdot \frac{\partial x_j^{l+1}}{\partial z_j^l} \cdot \frac{\partial z_j^l}{\partial x_i^l} \Rightarrow$$

$$\Rightarrow \boxed{\frac{\partial C}{\partial x_i^l} = \sum_{j=0}^{m_l} \frac{\partial C}{\partial x_j^{l+1}} \cdot (z_j^l > 0\,?\,1:0) \cdot w_{ij}^l}$$

$$\frac{\partial C}{\partial w_{ip}^l} = \sum_{j=0}^{m_l} \frac{\partial C}{\partial x_j^{l+1}} \cdot \frac{\partial x_j^{l+1}}{\partial z_j^l} \cdot \underbrace{\frac{\partial z_j^l}{\partial w_{ip}^l}}_{0 \text{ for each } j \neq p,\ x_i \text{ otherwise}} \Rightarrow$$

$$\Rightarrow \boxed{\frac{\partial C}{\partial w_{ip}^l} = \frac{\partial C}{\partial x_p^{l+1}} \cdot (z_p^l > 0\,?\,1:0) \cdot x_i^l}$$

$$\frac{\partial C}{\partial b_p^l} = \sum_{j=0}^{m_l} \frac{\partial C}{\partial x_j^{l+1}} \cdot \frac{\partial x_j^{l+1}}{\partial z_j^l} \cdot \underbrace{\frac{\partial z_j^l}{\partial b_p}}_{0 \text{ for each } j \neq p,\ 1 \text{ otherwise}} \Rightarrow$$

$$\boxed{\frac{\partial C}{\partial b_j^l} = \frac{\partial C}{\partial x_j^{l+1}} \cdot (z_j^l > 0\,?\,1:0)}$$