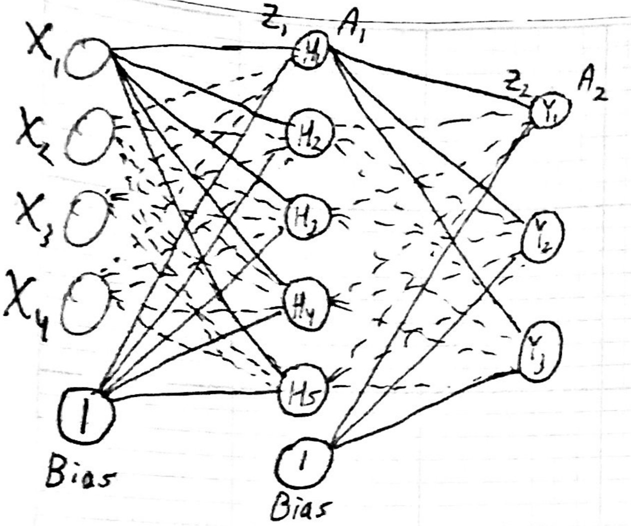


Activation Functions

Sigmoid: $\sigma(x_i) = \dfrac{1}{1+e^{-x_i}}$

Softmax: $\sigma(x)_i = \dfrac{e^{x_i}}{\sum_j^C e^{x_j}}$

Derivative of Sigmoid: $\sigma'(x) = (\sigma(x))(1-\sigma(x))$

$W^{Layer}_{From,To}$

The Problem we are aiming to solve is 'Multi-Class Classification'

Every Neural Network Requires a Cost Function. Cross Entropy is a common loss function for Classification

Categorial Cross Entropy Loss

$$Loss_{CE} = -\sum_i^C t_i \log(f(a)_i) \qquad f(a)_i^{Softmax} = \frac{e^{a_i}}{\sum_j^C e^{a_j}} = A(z)$$

in One-hot classification:

$$\boxed{CE = -\log\left(\frac{e^{s_p}}{\sum_j^C e^{a_j}}\right)}$$

s_p = positive Class
(only the positive Class C_p keeps its term in the loss)
S_n = negative Class

Four inputs
Three Output
(1 per each Class)

$$\frac{\partial CE}{\partial S_p} = \left(\frac{e^{a_p}}{\sum_j^C e^{s_j}} - 1\right)$$

$$\frac{\partial CE}{\partial S_n} = \left(\frac{e^{S_n}}{\sum_j^C e^{s_j}}\right)$$

$$\boxed{\frac{\partial CE}{\partial S} = \hat{Y}_1 - Y_1} = \frac{\partial CE}{\partial Z^2}$$

Layer 2

Summary $\quad \delta^2 = \hat{Y}_1 - Y_1$

$$\frac{\partial CE}{\partial W_2} = A_1^T \delta^2 \qquad \frac{\partial CE}{\partial B_2} = \sum \delta^2$$

$$\boxed{\begin{aligned} W_2 &= W_2 - LR \cdot (A_1^T \delta^2) \\ B_2 &= B_2 - LR \cdot colSums(\delta^2) \end{aligned}}$$

$$\frac{\partial CE}{\partial W_1} = X^T \delta^1 \qquad \frac{\partial CE}{\partial B_1} = \sum \delta^1$$

$$\delta^1 = \delta^2 W_2^T * \sigma'(z_1)$$

Element wise

$$\boxed{\begin{aligned} W_1 &= W_1 - LR \cdot (X^T \delta^1) \\ B_1 &= B_1 - LR \cdot colSums(\delta^1) \end{aligned}}$$

Below Follows the math for one row of input being entered to the NN

Feed-Forward (simple example) $W^L_{From,To}$

$$\underset{1\times 4}{\overset{X}{[X_1\ X_2\ X_3\ X_4]}}\ \overset{W_1}{\begin{bmatrix} W_{11} & W_{12} & W_{13} & W_{14} & W_{15} \\ W_{21} & W_{22} & W_{23} & W_{24} & W_{25} \\ W_{31} & W_{32} & W_{33} & W_{34} & W_{35} \\ W_{41} & W_{42} & W_{43} & W_{44} & W_{45} \end{bmatrix}} + \underset{1\times 5}{\overset{B_1}{[B_1\ B_2\ B_3\ B_4\ B_5]}}\ \underset{\text{bias}}{[1]} = \overset{Z_1}{\begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \\ Z_5 \end{bmatrix}^T}$$

Bias weights

$Z_1 = 1\times 5$ (nSamples × nodes)

Hidden Layer

$4\times 5 = \text{len}(x) \times \text{nodes}$ (nInputs × nOutputs) Columns input x nodes

$A_1 = 1\times 5$ matrix (nSamples × nNodes)

$$\boxed{Z_1 = XW_1 + B_1} \xrightarrow{\text{activation}} \boxed{A_1 = \sigma(Z_1)}$$

$$\underset{1\times 5}{\overset{A_1}{[A_1\ A_2\ A_3\ A_4\ A_5]}}\ \overset{W_2}{\begin{bmatrix} W_{11} & W_{12} & W_{13} \\ W_{21} & W_{22} & W_{23} \\ W_{31} & W_{32} & W_{33} \\ W_{41} & W_{42} & W_{43} \\ W_{51} & W_{52} & W_{53} \end{bmatrix}} + \underset{1\times 3}{\overset{B_2}{[B_1\ B_2\ B_3]}}\ [1] = \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \end{bmatrix}^T$$

$Z_2 = 1\times 3$ (Samples × levels of factor / # of classes / output)

5×3 matrix (nInputs × nOutputs)

$$\boxed{Z_2 = A_1 W_2 + B_2} \xrightarrow{\text{activate}} \boxed{A_2 = \sigma(Z_2)}$$

Below Follows the Math for back propagation to derive the gradient of the error with respect to the Networks Weights:

$$W_{New} = W_{old} - \underset{\substack{\uparrow \\ \text{Learning} \\ \text{Rate}}}{LR} \cdot \underset{\substack{\uparrow \\ \text{Gradient} \\ \text{of Error}}}{\nabla E}$$

$$\delta^2 = (\hat{Y} - Y)$$

$$\frac{\partial CE}{\partial W_2} = \underbrace{\frac{\partial CE}{\partial A_2} \cdot \frac{\partial A_2}{\partial Z_2}}_{\delta^2} \cdot \frac{\partial Z_2}{\partial W_2} \Rightarrow \frac{\partial CE}{\partial W_2} = A_1^T \delta^2$$

Update With:

$$\boxed{W_2 = W_2 - LR \cdot (A_1^T \delta^2)}$$

$$\frac{\partial CE}{\partial B_2} = \frac{\partial CE}{\partial A_2} \cdot \frac{\partial A_2}{\partial Z_2} \cdot \frac{\partial Z_2}{\partial B_2} \Rightarrow \frac{\partial CE}{\partial B_2} = \sum \delta^2$$

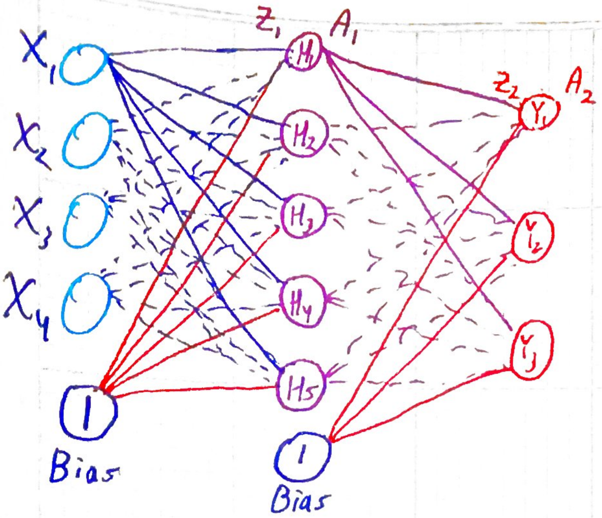
$$\boxed{B_2 = B_2 - LR \cdot \text{colSums}(\delta^2)}$$

$$\frac{\partial CE}{\partial W_1} = \underbrace{\underbrace{\frac{\partial CE}{\partial A_2} \cdot \frac{\partial A_2}{\partial Z_2}}_{\delta^2} \cdot \frac{\partial Z_2}{\partial A_1} \cdot \frac{\partial A_1}{\partial Z_1}}_{\delta^1} \cdot \frac{\partial Z_1}{\partial W_1} \Rightarrow \frac{\partial CE}{\partial W_1} = X^T \delta^1$$

$$\boxed{W_1 = W_1 - LR \cdot (X^T \delta^1)}$$

$$\boxed{\delta^1 = \delta^2 W_2^T \underset{\substack{\text{element wise mult} \\ \downarrow}}{*} \sigma'(Z_1)}$$

$$\frac{\partial CE}{\partial B_1} = \frac{\partial CE}{\partial A_2} \cdot \frac{\partial A_2}{\partial Z_2} \cdot \frac{\partial Z_2}{\partial A_1} \cdot \frac{\partial A_1}{\partial Z_1} \cdot \frac{\partial Z_1}{\partial B_1} \Rightarrow \frac{\partial CE}{\partial B_1} = \sum \delta^1 \quad B_1 = B_1 - LR \cdot \text{colSums}(\delta^1)$$



Activation Functions

Sigmoid: $\sigma(x_i) = \frac{1}{1+e^{-x_i}}$

Softmax: $\sigma(x)_i = \frac{e^{x_i}}{\sum_j^C e^{x_j}}$

Derivative of Sigmoid: $\sigma'(x) = (\sigma(x))(1-\sigma(x))$

$W^{Layer}_{From,To}$

The Problem we are aiming to solve is 'Multi-Class Classification'

Every Neural Network Requires a Cost Function. Cross Entropy is a common loss function for Classification

Categorial Cross Entropy Loss

$$Loss_{CE} = -\sum_i^C t_i \log(f(a)_i) \qquad f(a)_i = \frac{e^{a_i}}{\sum_j^C e^{a_j}} = A(z)$$

Softmax

in one-hot classification:

$$\boxed{CE = -\log\left(\frac{e^{s_p}}{\sum_j^C e^{a_j}}\right)}$$

s_p = positive Class

(only the positive Class C_p keeps its terms in the loss)

s_n = negative Class

Four inputs

Three Output (1 per each Class)

$$\frac{dCE}{ds_p} = \left(\frac{e^{a_p}}{\sum_j^C e^{s_j}} - 1\right)$$

$$\frac{dCE}{ds_n} = \left(\frac{e^{s_n}}{\sum_j^C e^{s_j}}\right)$$

$$\boxed{\frac{dCE}{ds} = \hat{Y}_1 - Y_1} = \frac{dCE}{dZ^2}$$

Layer 2

Summary $\delta^2 = \hat{Y}_1 - Y_1$

$\frac{dCE}{dW_2} = A_1^T \delta^2 \qquad \frac{dCE}{dB_2} = \sum \delta^2$

$W_2 = W_2 - LR \cdot (A_1^T \delta^2)$
$B_2 = B_2 - LR \cdot colSums(\delta^2)$

$\frac{dCE}{dW_1} = X^T \delta^1 \qquad \frac{dCE}{dB_1} = \sum \delta^1$

$\delta^1 = \delta^2 W_2^T * \sigma'(z_1)$

Element wise

$W_1 = W_1 - LR \cdot (X^T \delta^1)$
$B_1 = B_1 - LR \cdot colSums(\delta^1)$

Below Follows the math for one row of input being entered to the NN

Feed-Forward (simple example) $W^{L}_{From,To}$

$$\underset{1\times 4}{\overset{X}{\begin{bmatrix} X_1 & X_2 & X_3 & X_4 \end{bmatrix}}} \overset{W_1}{\begin{bmatrix} W_{11} & W_{12} & W_{13} & W_{14} & W_{15} \\ W_{21} & W_{22} & W_{23} & W_{24} & W_{25} \\ W_{31} & W_{32} & W_{33} & W_{34} & W_{35} \\ W_{41} & W_{42} & W_{43} & W_{44} & W_{45} \end{bmatrix}} + \underset{1\times 5}{\overset{B_1}{\begin{bmatrix} B_1 & B_2 & B_3 & B_4 & B_5 \end{bmatrix}}} \underset{\text{bias}}{\begin{bmatrix} 1 \end{bmatrix}} = \overset{Z_1}{\begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \\ Z_5 \end{bmatrix}^T}$$

Bias Weights

nSamples nNodes

$Z_1 = 1 \times 5$

Hidden Layer

$4 \times 5 = len(x) \times nodes$

nInputs nOutputs

Column input x nodes

nSamples nNodes

$A_1 = 1 \times 5$ matrix

$$\boxed{Z_1 = X W_1 + B_1} \xrightarrow{\text{Activation}} \boxed{A_1 = \sigma(Z_1)}$$

$$\underset{1\times 5}{\overset{A_1}{\begin{bmatrix} A_1 & A_2 & A_3 & A_4 & A_5 \end{bmatrix}}} \overset{W_2}{\begin{bmatrix} W_{11} & W_{12} & W_{13} \\ W_{21} & W_{22} & W_{23} \\ W_{31} & W_{32} & W_{33} \\ W_{41} & W_{42} & W_{43} \\ W_{51} & W_{52} & W_{53} \end{bmatrix}} + \underset{1\times 3}{\overset{B_2}{\begin{bmatrix} B_1 & B_2 & B_3 \end{bmatrix}}} \begin{bmatrix} 1 \end{bmatrix} = \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \end{bmatrix}^T$$

Samples levels of factor/ # of Classes/ output

$Z_2 = 1 \times 3$

5×3 matrix

nInputs nOutputs

$$\boxed{Z_2 = A_1 W_2 + B_2} \xrightarrow{\text{activate}} \boxed{A_2 = \sigma(Z_2)}$$

Below Follows the Math for back propagation to derive the gradient of the error with respect to the Networks Weights:

$$W_{New} = W_{Old} - \underset{\substack{\uparrow \\ \text{Learning Rate}}}{LR} \cdot \underset{\substack{\uparrow \\ \text{Gradient of Error}}}{\nabla E}$$

$$\delta^2 = (\hat{Y} - Y)$$

$$\frac{\partial CE}{\partial W_2} = \underbrace{\frac{\partial CE}{\partial A_2} \cdot \frac{\partial A_2}{\partial Z_2}}_{\delta^2} \cdot \frac{\partial Z_2}{\partial W_2} \Rightarrow \frac{\partial CE}{\partial W_2} = A_1^T \delta^2$$

Update With:

$$\boxed{W_2 = W_2 - LR \cdot (A_1^T \delta^2)}$$

$$\frac{\partial CE}{\partial B_2} = \frac{\partial CE}{\partial A_2} \cdot \frac{\partial A_2}{\partial Z_2} \cdot \frac{\partial Z_2}{\partial B_2} \Rightarrow \frac{\partial CE}{\partial B_2} = \Sigma \delta^2$$

$$\boxed{B_2 = B_2 - LR \cdot \text{colSums}(\delta^2)}$$

$$\frac{\partial CE}{\partial W_1} = \underbrace{\underbrace{\frac{\partial CE}{\partial A_2} \cdot \frac{\partial A_2}{\partial Z_2}}_{\delta^2} \cdot \frac{\partial Z_2}{\partial A_1} \cdot \frac{\partial A_1}{\partial Z_1}}_{\delta^1} \cdot \frac{\partial Z_1}{\partial W_1} \Rightarrow \frac{\partial CE}{\partial W_1} = X^T \delta^1$$

$$\boxed{W_1 = W_1 - LR \cdot (X^T \delta^1)}$$

$$\boxed{\delta^1 = \delta^2 W_2^T \overset{\substack{\text{element wise mult} \\ \downarrow}}{*} \sigma'(Z_1)}$$

$$\frac{\partial CE}{\partial B_1} = \frac{\partial CE}{\partial A_2} \cdot \frac{\partial A_2}{\partial Z_2} \cdot \frac{\partial Z_2}{\partial A_1} \cdot \frac{\partial A_1}{\partial Z_1} \cdot \frac{\partial Z_1}{\partial B_1} \Rightarrow \frac{\partial CE}{\partial B_1} = \Sigma \delta^1 \quad B_1 = B_1 - LR \cdot \text{colSums}(\delta^1$$