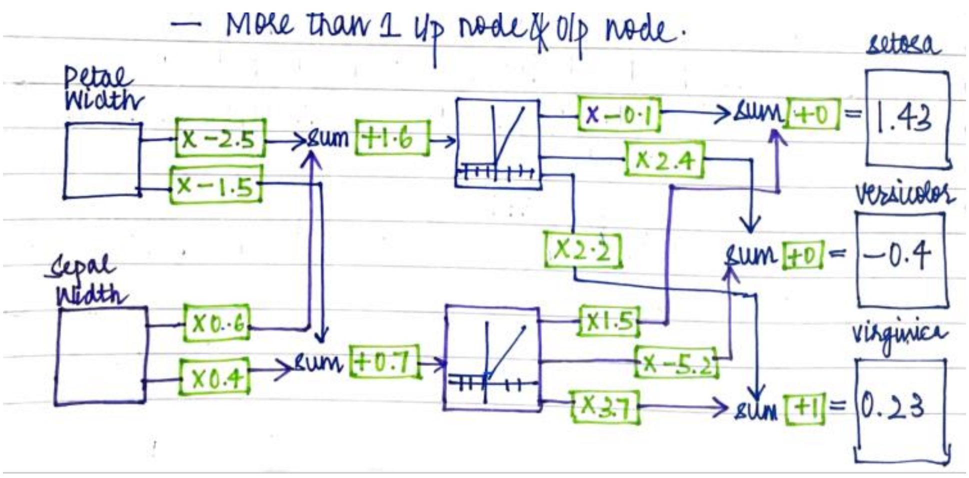
Neural Networks - Small Pointers

14 May 2023 21:19

- As already observed NN raw outputs are not always in between 0 to 1.

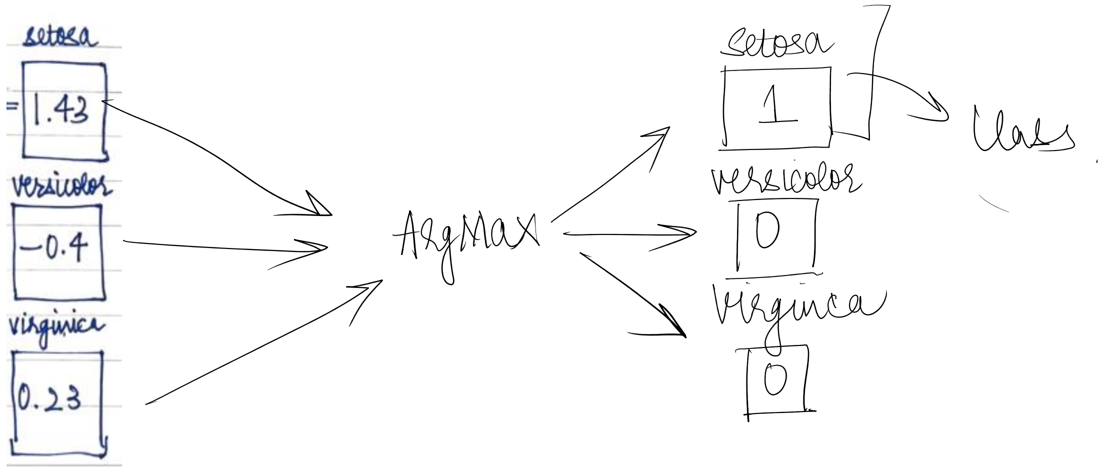


- It can be >1 or <0
- Broad Range of output and that makes it extremely hard to make predictions.
- Raw Output

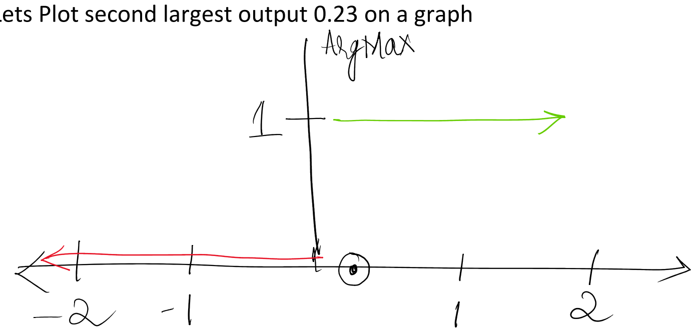
- Sent to ArgMax Layer/ SoftMax Layer before the final decision is made.

ARGMAX

- Sets the largest value to 1 and all the other values to 0 automatically



- PROBLEM W ARGMAX
  - Can't be used to optimize weights and biases in NN
  - Reason: Output values from Argmax are always going to be 0 and 1
  - Lets Plot second largest output 0.23 on a graph



- PROBLEM W ARGMAX
  - Since this is the second largest value, ArgMax will output 1 for any other value that is greater than 0.23 and will output 0 for all values <0.23.
  - Slopes of 2 line = 0
    - Their derivatives = 0
  - If we want to find the optimal value of any weights and biases in NN we will be plugging 0 **in Chain** Rule for the derivative ArgMax

$$\frac{d\ \text{Loss Function}}{d\ \text{Some Parameter}} = \frac{d\ \text{LossFunction}}{d\ \text{ArgMax}} \times \underbrace{\frac{d\ \text{ArgMax}}{d\ \text{RAW o/p}}}_{0} \cdots$$

$$= 0$$

- PROBLEM W ARGMAX
  - If we plug 0 in Gradient Descent we won't get any optimal values for the parameter
- ARGMAX IS NOT A SUITABLE FUNCTION FOR BACKPROPOGATION.

SOFTMAX

- How SoftMax is used in common NN: COMMON PRACTICE:
  - ArgMax is used for Output
  - SoftMax for optimizing weights and biases in backpropogation

$$\text{SoftMax}_{\text{setosa}}(\text{Output value}) = \frac{e^{\text{setosa}}}{e^{\text{setosa}} + e^{\text{versicolor}} + e^{\text{virginica}}}$$

$$= 0.69$$

$$\text{SoftMax}_{\text{versicolor}} = 0.10$$

$$\text{SoftMax}_{\text{virginica}} = 0.21$$

- SoftMax preserves the order of the output values
- All outputs are between 0 and 1
- Sum total of outputs for all class = 1
  - This means that SoftMax approximately gives us predictive probabilities for the classes
    - This is not to be trusted!
    - BECAUSE, the output is dependent on weights and biases, and that is dependent on selected initial values.

General Form of SoftMax:

$$\text{SoftMax}_i(\text{o/p}) = \frac{e^{\text{o/p value } i}}{\sum_{j=1}^{K} e^{\text{o/p value } j}}$$

- SoftMax has a derivative that can be used for BACKPROPOGATION

$$\boxed{\frac{d\ \text{"}p_{\text{setosa}}\text{"}}{d\ \text{RAW}_{\text{setosa}}} = p_{\text{setosa}} \times (1 - p_{\text{setosa}})}$$

$$= 0.69 \times 0.31$$
$$= 0.21$$

$$\frac{d\,P_{setosa}}{d\,Raw_{versicolor}} = -\,P_{setosa} \times P_{versicolor} = -0.07$$

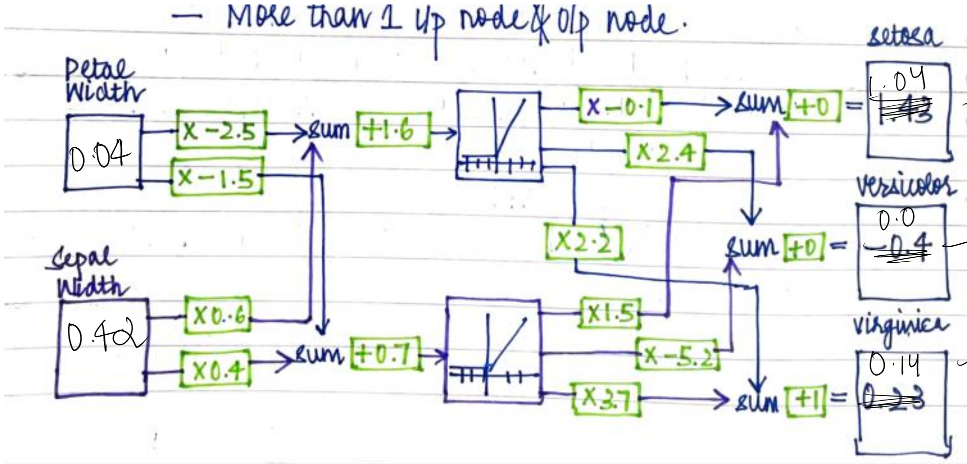
$$\frac{d\,P_{setosa}}{d\,Raw_{virginica}} = -\,P_{setosa} \times P_{virginica} = -0.15$$

CROSS ENTROPY

Petal	sepal	Species	P	cross entropy
0.04	0.42	Setosa	0.57	0.56
1	0.54	versicolor	0.58	0.54
0.50	0.37	virginica	0.52	0.65

Total cross entropy = 1.75

— More than 1 i/p node & o/p node.



0.57

SoftMax 0.20

0.23

$$\text{Cross Entropy}_{setosa} = -\log(0.57) = 0.56.$$

→ the cross entropy is the −log (base e) of SoftMax O/P value for setosa.

<u>General Term</u> (Equation)

$$-\sum_{c=1}^{M} \text{Observed}_c \times \log(\text{predicted}_c)$$

N×1 only need a simplified version

M = no of O/P classes (3)

M = no of o/p classes (3)

$$= \boxed{-\text{Observed}_{setosa}} \times \log(P_{setosa}) - \underbrace{\text{Obs}_{versi}}_{0} \times \log(P_{ve}) - \underbrace{\text{Observed}_{virginica}}_{0} \times \log(P_{virgin})$$

1

versicolor

$$(\text{Residual})^2 = (\text{obs} - \text{pred})^2$$

$$SSR = \sum_{i=1}^{n} (\text{obs}_i - \text{pred}_i)^2$$