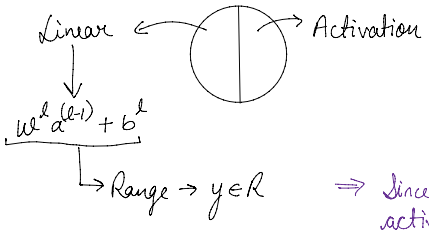
ANN Part 4 - Output and Loss Function MNIST Case Study

14 August 2023 01:24 PM

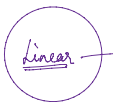
$ Output Functions

Regression ($y \in R$)

* In the functions covered so far, none of them will be able to work for regression.



⇒ Since we are getting the range for Regression from the linear Part itself, we remove the activation function in last neuron



$w^l a^{(l-1)} + b^l = \hat{y}$ { The value of $\hat{y}$ can be controlled using w & b }

∴ The output function in case of Regression : Linear Activation { In layman terms, it can also be called as absence of Activation f^n }

£ Ranges of various functions

- → Threshold function : $\{0, 1\}$
- → Sigmoid function : $[0, 1]$
- → tanh function : $[-1, 1]$
- → Softmax function : $[0, 1]$
- → ReLU : $[0, +\infty)$

$ Loss Function

aka Error function

1.) Regression :

→ $MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$

2.) Classification

→ Cross Entropy Loss :

$$L(\theta) = -\sum_{j=1}^{K} y_j \log(\hat{y}_j)$$

K → Number of classes

$L(\theta)$ → All parameters; y_j → True Label; $\hat{y}_j$ → Predicted Label

→ Classification Task (K = 3)

Image (x)

	Cat	Dog	Duck
y : Dog =	0 (y_1)	1 (y_2)	0 (y_3)
$\hat{y}_i$ =	0.3 ($\hat{y}_1$)	0.4 ($\hat{y}_2$)	0.3 ($\hat{y}_3$)

→ By using Softmax → $\hat{y} = 1$ ⇒ Dog

$$L(\theta) = -\sum_{j=1}^{3} y_j \log(\hat{y}_j) = -\{[0 \times \log 0.3] + [1 \times \log 0.4] + [0 \times \log 0.3]\}$$

$$L(\theta) = -\log 0.4$$

This $L(\theta)$ is our final loss and we need to minimize this loss. i.e. minimize $-\log(0.4)$

According to optimization theory → $\max f(x) \approx \min(-f(x))$

⇒ minimize $-\log(0.4) \approx$ maximize $\log(0.4)$

→ we are basically trying to maximize the log probability of True class.

Now, from optimization theory, w.k.t $\max f(x) \approx \max g(f(x))$ iff. $g()$ is a monotonic function.

$\Rightarrow$ maximize $\log(0.4) \approx$ maximizing 0.4

↳ We are trying to maximize probability of True class.

MNIST Case Study