Machine Learning

Assignment 7.2

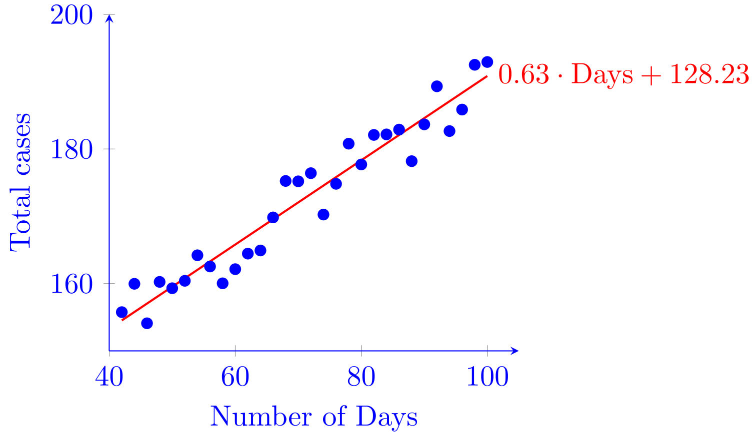
Submitted By: Ranji Raj

December 9, 2020

Preamble

- **Activation function** activates the unit that is required for the desired output.
- **Loss function** aids in recording the model performance, i.e. how good the model is able to generalize.
- Choosing appropriate activation and loss function is directly proportional on the kind of output desired to predict.

Case1: Desired output is a numerical value prediction-Linear activation function



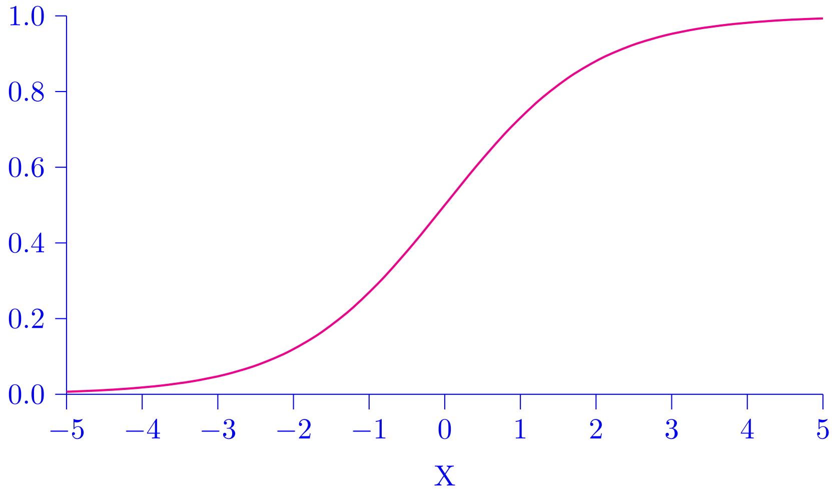
- Predicting the emerging total pandemic cases based on a country profile provided with various independent variables as announced by the European Centre for Disease Prevention and Control (ECDC),. In this case the output layer of a neural network would comprise of only one unit which gives the prediction as a numerical quantity.

- For calculating the accuracy scores the predicted values are compared to the true/actual values as the **loss function**,

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

- A **Linear activation** function is used under such requirement which gives the output in a numeric format.

Case2: Desired output is a binary classification-Sigmoid activation function



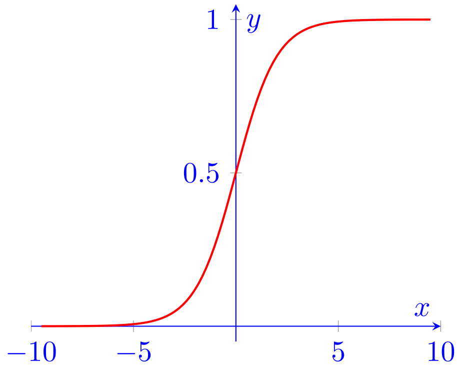
- From the NASA Technical Reports Server (NTRS), motivated by the LANDSAT problem of estimating the probability of crop types based on multi-channel satellite imagery data, into *healthy* and *corrupt* crops comes under binary classification problem.
- For such kind of problems, the output layer consists of only one unit that is responsible to result in a value that is between 0 and 1.
- For the accuracy of the prediction, it is compared against with the true labels.
- A **Sigmoid activation** function is employed for such problems which squashes the output to either 0 or 1.

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

- The possible **loss function** used under such scenarios is **Binary Cross Entropy**- the difference between the two probability distributions is given by BCE. (p, 1-p) is the model distribution predicted by the model, to compare it with true distribution BCE is used.

$$BCE = -\frac{1}{N}\sum_{i=1}^{n} y_i log(p_i) + (1 - y_i) log(1 - p_i)$$

Case3: Desired output is a single class from multiple classes-Softmax activation function



- From the different kinds of biological signals (EEG, ECG, EMG) from a patients body suffering from various co-morbidities the task of differentiating between a healthy individual and patients with larynx diseases.
- Such complicated multi-class problem where the output unit will revert a value between 0 and 1, and the output is the probability distribution that results in 1 are all added.
- Each output is checked with its respective true value to get the accuracy. Further, they are one-hot-encoded meaning, it will be 1 for the correct class otherwise would be zero.
- A **Softmax activation** function is used under such circumstances which outputs between 0 and 1 as probability scores which gives the result as 1 when added.

$$\sigma(x)_j = \frac{e^{x_j}}{\sum_{i=1}^{k} e^{x_k}}$$

- For example, three class labels will be integer encoded as 0, 1, and 2. Then encoded to vectors as follows:

$$Class0 : [1, 0, 0], Class1 : [0, 1, 0], Class2 : [0, 0, 1]$$

- The ideal, loss function for such problems is to model using **Cross Entropy-** computes the difference between multi-nomial probability distributions.
  For $M > 2$,

$$CE = -\sum_{c=1}^{M} y_i log(p_i)$$

Problem	**Output Type**	**Activation Function**	**Loss Function**
Regression	Continuous	Linear	Sum-Of-Squared-Error
Classification	Binary	Sigmoid	Binary Cross Entropy
Classification	Multi-class	Softmax	Cross Entropy

Table 1: Summary: Activation and Loss functions