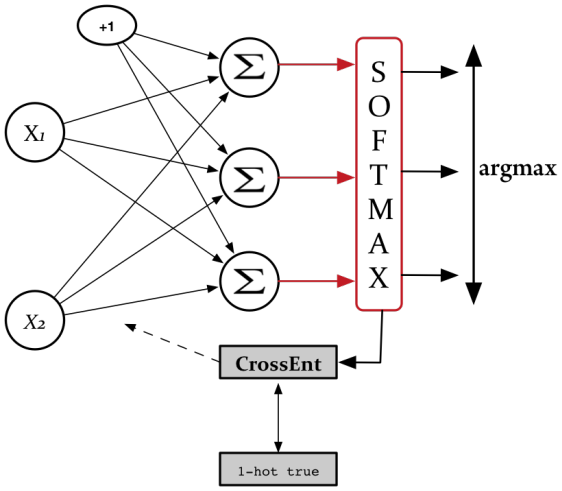
TD2

Cross Entropy Error *Vs* MSE for NN

1 Cross-entropy loss and Softmax classifiers



The Softmax classifier is a generalization of the binary form of Logistic Regression. The softmax function takes an N-dimensional vector of arbitrary real values and produces another N-dimensional vector with real values in the range (0, 1) that add up to 1.0

$$softmax(z^i) = \frac{e^{z^i}}{\sum_i e^{z^i}}$$

To illustrate the concept of softmax, let us walk through a concrete example. Let's assume we have a training set consisting of 4 samples from 3 different classes (0, 1, and 2).

$x_0 \rightarrow$ class 0
$x_1 \rightarrow$ class 1
$x_2 \rightarrow$ class 2
$x_3 \rightarrow$ class 2

Question 1 : Encode the class labels into a format that we can more easily work with by applying one-hot encoding

Question 2 : Given the following dataset of 4 samples (2 features for each sample) :

$$X = \begin{pmatrix} x_0; & x_1; & x_2; & x_3 \end{pmatrix} = \left(\begin{pmatrix} x_{00} \\ x_{01} \end{pmatrix} \begin{pmatrix} x_{10} \\ x_{11} \end{pmatrix} \begin{pmatrix} x_{20} \\ x_{21} \end{pmatrix} \begin{pmatrix} x_{30} \\ x_{31} \end{pmatrix} \right)$$

$$X^T = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} x_{00} & x_{01} \\ x_{10} & x_{11} \\ x_{20} & x_{21} \\ x_{30} & x_{31} \end{pmatrix} = \begin{pmatrix} 0.1 & 0.5 \\ 1.1 & 2.3 \\ -1.1 & -2.3 \\ -1.5 & -2.5 \end{pmatrix}$$

Data Shape (X) : (number of Samples ; number of features)
Weights Shape (W) : (size of previous layer ; size of next layer)

Using the following initial weights W and bias :

$$W = \begin{pmatrix} w_{11}^{(1)} & w_{21}^{(1)} & w_{31}^{(1)} \\ w_{12}^{(1)} & w_{22}^{(1)} & w_{32}^{(1)} \end{pmatrix} = \begin{pmatrix} 0.1 & 0.2 & 0.3 \\ 0.1 & 0.2 & 0.3 \end{pmatrix}$$

$$b^{(1)} = \begin{pmatrix} b_1^{(1)} \\ b_2^{(1)} \\ b_3^{(1)} \end{pmatrix} = \begin{pmatrix} 0.01 \\ 0.1 \\ 0.1 \end{pmatrix}$$

Compute the net output ($Z = X^T W + b$)

Question 3 : Compute the *softmax* activation that we discussed earlier.

$$softmax(z^i) = \frac{e^{z^i}}{\sum_i e^{z^i}}$$

Question 4 : What is the predicted class for each example ?

Question 5 : Since the correct class labels are [0, 1, 2, 2].

Calculate the network accuracy.

Question 6 : How can we improve the network accuracy ?

Question 7 : Which Error function is more appropriate to this network architecture ?

In order to improve the accuracy we apply GD using CE error function.

2 Cross-entropy vs MSE

Suppose you have just three training data items. The neural network uses softmax activation for the output neurons so that there are three output values that can be interpreted as probabilities.

Suppose a first neural network's computed outputs, and the target (aka desired) values are as follows:

computed	target	correct ?
0.3 0.3 0.4	0 0 1	yes
0.3 0.4 0.3	0 1 0	yes
0.1 0.2 0.7	1 0 0	no

Question 1 : Calculate the network accuracy.

Suppose a second neural network's computed outputs, and the target (aka desired) values are as follows:

computed	target	correct ?
0.1 0.2 0.7	0 0 1	yes
0.1 0.7 0.2	0 1 0	yes
0.3 0.4 0.3	1 0 0	no

Question 2 : Calculate the network accuracy.

How we can compare better the 2 networks ?

Let's compute the average CE error and the MSE error for both networks.

$$H(T_i, O_i) = -\frac{1}{N}\sum_{i}^{N} T_i \; log(O_i)$$

$$MSE = \frac{1}{N}\sum_{i}^{N} (O_i - T_i)^2$$

$\Rightarrow$ **Which of the two functions is better for network training ?**