Understanding Cross-Entropy Loss, Binary Cross-Entropy Loss, Softmax Loss

Multi-Class Classification

One-of-many classification. Each sample can belong to ONE of C classes. The neural network (NN) will have C output neurons that can be gathered in a vector $\mathbf{s}$ (Scores). The target (ground truth) vector t will be a one-hot vector with a positive class and $C-1$ negative classes. This task is treated as a single classification problem of samples in one of C classes.

Output Activation Functions

These functions are transformations we apply to vectors coming out from NNs($\mathbf{s}$) before the loss computation.

Sigmoid

$$f(s_i) = \frac{1}{1+e^{s_i}}$$

It squashes a vector in the range $(0,1)$. It is applied independently to each element of $\mathbf{s}$. It's also called logistic function.

Softmax

Softmax it's a function, not a loss. It squashes a vector in the range $(0,1)$ and all the resulting elements add up to 1. It is applied to the output scores $\mathbf{s}$. As elements represent a class, they can be interpreted as class probabilities. The Softmax function cannot be applied independently to each s_i, since it depends on all elements of $\mathbf{s}$. For a given class s_i, the Softmax function can be computed as:

$$f(s_i) = \frac{e^{s_i}}{\sum_{j=1}^{C} e^{s_j}},$$

where s_j are the scores inferred by the net for each class in C. Note that the Softmax activations for a class s_i depends on all the scores in $\mathbf{s}$.

Losses

Cross-Entropy loss

The Cross-Entropy Loss is actually the only loss we are discussing here. The other losses names written in the title are other names or variations of it. The CE Loss is defined as:

$$CE = -\sum_{i=1}^{C} t_i \log(s_i),$$

where t_i and s_i are the groundtruth and the CNN score for each class i in C. As usually an activation function (Sigmoid / Softmax) is applied to the scores before the CE Loss computation, we write $f(s_i)$ to refer to the activations.

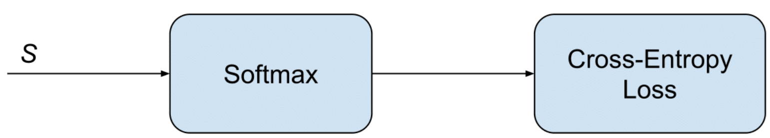
In a binary classification problem, where $C' = 2$, the Cross Entropy Loss can be defined also as

$$CE = -\sum_{i=1}^{C'=2} t_i \log(s_i) = -t_1 \log(s_1) - (1 - t_1) \log(1 - s_1),$$

where it's assumed that there are two classes: C_1 and C_2. $t_1 \in \{0, 1\}$ and s_1 are the groundtruth and the score for C_1, and $t_2 = 1 - t_1$ and $s_2 = 1 - s_1$ are the groundtruth and the score for C_2.

Categorical Cross-Entropy loss

Also called **Softmax Loss**. It is a **Softmax activation** plus a **Cross-Entropy loss**. If we use this loss, we will train a NN to output a probability over the C classes for each image. It is used for multi-class classification.



$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}} \qquad CE = -\sum_i^C t_i log(f(s)_i)$$

In the specific (and usual) case of Multi-Class classification the labels are one-hot, so only the positive class C_p keeps its term in the loss. There is only one element of the Target vector t which is not zero $t_i = t_p$. So discarding the elements of the summation which are zero due to target labels, we can write:

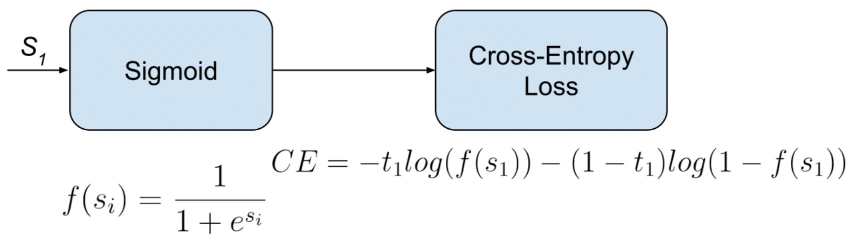
$$CE = -\log\left(\frac{e^{s_p}}{\sum_j^C e^{s_j}}\right) \tag{1}$$

where s_p is the NN score for the positive class.

Binary Cross-Entropy Loss

Also called **Sigmoid Cross-Entropy loss**. It is a **Sigmoid activation** plus a **Cross-Entropy loss**. Unlike **Softmax loss** it is independent for each vector component (class), meaning that the loss computed for every vector component is not affected by other component values. It's called **Binary Cross-Entropy Loss** because it sets up a binary classification problem between $C' = 2$ classes for every class in C, as explained above. So when using this Loss, the formulation of Cross Entroypy Loss for binary problems is often used:

$$CE = -\sum_{i=1}^{C'=2} t_i \log(f(s_i)) = -t_1 \log(f(s_1)) - (1 - t_1) \log(1 - f(s_1)) \tag{2}$$



This would be the pipeline for each one of the C classes. We set C independent binary classification problems ($C' = 2$). Then we sum up the loss over the different binary problems. s_1 and t_1 are the score and the gorundtruth label for the class C_1, which is also the class C_i in C. $s_2 = 1 - s_1$ and $t_2 = 1 - t_1$ are the score and the groundtruth label of the class C_2, which is not a "class" in our original problem with C classes, but a class we create to set up the binary problem with $C = C_i$. We can understand it as a background class.

The loss can be expressed as:

$$CE = \begin{cases} -\log(f(s_1)) & if \quad t_1 = 1 \\ -\log(1 - f(s_1)) & if \quad t_1 = 0 \end{cases}$$

where $t_1 = 1$ means that the class $C = C_i$ is positive for this sample.

Name: **Student ID**:

Quiz

1. Please explain for the binary classification problem, how can the categorical Cross-Entropy loss (1)

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

reduce to (2)

$$CE = -t_1 \log(f(s_1)) - (1 - t_1) \log(1 - f(s_1)),$$

where $f(s)$ is the sigmoid function.

2. The score of the linear classifier for the i-th sample has the form of $\mathbf{s} = \mathbf{W}^T\mathbf{x}_i$ with $\mathbf{w}_j$ being the j column of $\mathbf{W}$. Let y_i be the label of the i-th sample. The SVM loss function can expressed as

$$L_i = \sum_{j \neq y_i} \max(0, \mathbf{w}_j^T\mathbf{x}_i - \mathbf{w}_{y_i}^T\mathbf{x}_i + 1).$$

Please show that the gradients of L_i w.r.t. $\mathbf{w}_j$ and $\mathbf{w}_{y_i}$ are

$$\nabla_{\mathbf{w}_j} L_i = 1_{(\mathbf{w}_j^T\mathbf{x}_i - \mathbf{w}_{y_i}^T\mathbf{x}_i + 1 > 0)}\mathbf{x}_i, \tag{3}$$

$$\nabla_{\mathbf{w}_{y_i}} L_i = -\left(\sum_{j \neq y_i} 1_{(\mathbf{w}_j^T\mathbf{x}_i - \mathbf{w}_{y_i}^T\mathbf{x}_i + 1 > 0)}\right)\mathbf{x}_i, \tag{4}$$

where 1_A is the indicator function; $1_A = 1$ if A is true, otherwise $1_A = 0$.

(**Hints**: You can let $\mathbf{w}_j$ and $\mathbf{x}_i$ as scalars to get the same result by using the chain rule. Then, generalize the results to the vector case.