Softmax and categorical cross entropy loss

Pawel Wocjan

March 18, 2019

Abstract

We define softmax and categorical cross entropy for multiclass classification

1 Classification

Problem type	Last-layer activation	Loss function
Binary classification	`sigmoid`	`binary_crossentropy`
Multiclass, single-label classification	`softmax`	`categorical_crossentropy`
Multiclass, multi-label classification	`sigmoid`	`binary_crossentropy`

2 Softmax activation function

Let $z_1, \dots, z_m$ be the weighted inputs of the m neurons of the last layer. Each of the neurons corresponds to one of the m classes of the multiclass classification problem at hand. We consider the single-label situation. We convert the weighted input vector $\boldsymbol{z} = (z_1, \dots, z_m)^T \in \mathbb{R}^m$ into a probability vector $\boldsymbol{a} = (a_1, \dots, a_m)^T \in \mathbb{R}^n$ by applying the so-called softmax activation function.

For $k \in [m]$, the activation a_k of the kth neuron is defined as follows:

$$a_k = \frac{e^{z_k}}{\sum_{j=1}^n e^{z_j}}. \tag{1}$$

It is straightforward to verify that this yields a probability distribution. The values a_k are all positive because the range of the exponential function is $(0, \infty)$. They sum up to 1 because of the normalization in the denominator.

Using the product and chain rule, we can show that

$$\frac{\partial a_k}{\partial z_j} = a_k \cdot (\delta_{jk} - a_j), \tag{2}$$

where δ_{jk} is the so called Kronecker delta, which is equal to 1 if $j = k$ and 0 if $j \neq k$.

3 Cross entropy

Let $\boldsymbol{p} = (p_1, \ldots, p_m)$ and $\boldsymbol{q} = (q_1, \ldots, q_m)$ be two probability distributions. In information theory, the cross entropy is defined by

$$H(\boldsymbol{p}, \boldsymbol{q}) = -\sum_{k=1}^{m} p_k \log q_k. \tag{3}$$

See `https://en.wikipedia.org/wiki/Cross_entropy` for a quick overview.

4 Categorical cross entropy loss function

Let $y \in [m]$ be a label. Using the so-called one-hot or categorical encoding, we construct a corresponding vector $\boldsymbol{y} = (y_1, \ldots, y_m)^T \in \mathbb{R}^n$ such that $m-1$ of its entries are equal to 0 and exactly one entry is equal to 1. The position of the entry 1 is given by the label y. For instance, for $m = 3$ classes, label 1 corresponds to $(1,0,0)$, the label 2 to $(0,1,0)$, and the label 3 to $(0,0,1)$.

Assume that the feature vector $\boldsymbol{x}$ produces the activation vector $\boldsymbol{a} = (a_1, \ldots, a_m)$ in the last layer. Assume that the correct label is y. Then the categorical cross entropy loss $\mathcal{L}$ is defined by

$$\mathcal{L} = -\sum_{k=1}^{m} y_k \log a_k, \tag{4}$$

where $\boldsymbol{y} = (y_1, \ldots, y_m)$ is the categorical encoding of y. Observe that $\mathcal{L} = H(\boldsymbol{y}, \boldsymbol{a})$.

The partial derivatives of $\mathcal{L}$ with respect to a_k is

$$\frac{\partial \mathcal{L}}{\partial a_k} = -\frac{y_k}{a_k}. \tag{5}$$

The partial derivatives of $\mathcal{L}$ with respect to z_j is

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial z_j} &= \sum_{k=1}^{m} \frac{\partial \mathcal{L}}{\partial a_k} \cdot \frac{\partial a_k}{\partial z_j} && (6)\\
&= \sum_{k=1}^{m} -\frac{y_k}{a_k} \cdot a_k \cdot (\delta_{kj} - a_j) && (7)\\
&= \sum_{k=1}^{m} y_k \cdot (a_j - \delta_{kj}) && (8)\\
&= \left(\sum_{k=1}^{m} y_k\right) \cdot a_j - \sum_{k=1}^{m} y_k \cdot \delta_{kj} && (9)\\
&= a_j - y_j && (10)
\end{aligned}$$

since $\sum_{k=1}^{m} y_k = 1$. Recall that the y_k are the entries of the one-hot-encoding.

In matrix notation, this is given by

$$\nabla_{\boldsymbol{z}} \mathcal{L} = \boldsymbol{a} - \boldsymbol{y} \tag{11}$$

5 Simple neural network with softmax activation and categorical cross entropy

Let us consider a network that takes feature vectors of the form $\boldsymbol{x} = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$ as input and has m output neurons with softmax activation. Then for $j \in [m]$, the weighted input of the jth neuron is given by

$$z_j = \sum_{i=1}^{n} w_{ji} x_i + b_j. \tag{12}$$

The weights and the bias of jth neuron are $(w_{j1}, \ldots, w_{jn})$ and b_j, respectively. The weighted inputs z_j are transformed into probabilities by the softmax activation as discussed above.

The loss function changes as follows with respect to the weights and biases:

$$\frac{\partial \mathcal{L}}{\partial w_{ji}} = \frac{\partial \mathcal{L}}{\partial z_j} \cdot x_i \tag{13}$$

$$\frac{\partial \mathcal{L}}{\partial b_j} = \frac{\partial \mathcal{L}}{\partial z_j} \tag{14}$$