CME 216, ME 343 - Spring 2020

Eric Darve, ICME

Stanford University

The loss function for classification problems is usually defined using the cross-entropy.

Let us review the definition of cross-entropy and its interpretation.

This will require some probabilities.

Take a distribution p_i, which is assumed to be the true distribution.

Take an approximation q_i. Then the cross-entropy $H(p,q)$ is

$$H(p,q) = -\sum_i p_i \log q_i$$

What is the interpretation of cross-entropy?

The name entropy comes from the definition of the entropy of p

$$H(p) = -\sum_i p_i \log p_i$$

To understand these concepts, we need to run the following thought experiment.

Assume that we generate random samples i_k drawn from the probability p.

If we number of samples N is really large, the probability of observing the sequence i_k is given by

$$\prod_i (p_i)^{Np_i}$$

Explanation

The probability of seeing a sample $i = i_k$ is p_i and the number of times the sample i is going to appear is Np_i.

So, the associated probability is

$$(p_i)^{Np_i}$$

The product of these probabilities is the probability of seeing the entire sequence.

The entropy is then equal to the negative log of

$$P_H(\{i_k\}) = \prod_i (p_i)^{Np_i}$$

$$H(p) = -\frac{1}{N} \log P_H(\{i_k\}) = -\sum_i p_i \log p_i$$

If we have a system where only one state is possible (very low entropy), then

$$-\sum_i p_i \log p_i = -\log 1 = 0$$

If we have a system where all states have equal probability, the entropy is high:

$$-\sum_i p_i \log p_i = -\log n^{-1} = \log n$$

where n is the total number of states in the system.

What is now the cross-entropy?

We can repeat the same thought experiment with a slightly different setup.

Assume we generate the sequence i_k using probability p_i.

But p_i is unknown, and we only have some approximation q_i.

Then our approximation of the probability of seeing the sequence $\{i_k\}$ is

$$P_H^q(\{i_k\}) = \prod_i (q_i)^{Np_i}$$

The log of $P_H^q(\{i_k\})$ is the cross-entropy:

$$-\frac{1}{N}\log P_H^q(\{i_k\}) = -\sum_i p_i \log q_i$$

If our guess of p_i is correct, we have $q_i = p_i$ and the cross-entropy will be small.

Our estimated probability $P_H^q(\{i_k\})$ is large.

Note that the cross-entropy $H(p, q)$ is always greater than $H(p)$.

If $q_i = p_i$, we get

$$H(p, q) = -\sum_i p_i \log q_i = -\sum_i p_i \log p_i = H(p)$$

The cross-entropy is minimal.

If our guess is wildly off, then the probability we estimate for the sequence $\{i_k\}$ will be very low.

In that case, $H(p, q)$ will be very large.

Let's take a simple example. Let's consider a dice that has written 6 on all its faces.

In that scenario, the only sequence we can generate is

$$(6, 6, 6, \cdots)$$

If we believe that the dice is a normal one, we will assign a small probability to the sequence we are seeing.

We get:

$$H(p, q) = -\sum_i p_i \log q_i = -\log 1/6 = \log 6$$

If instead, we know that only 6 can show up, we will use

$q_i = 0$, when $i \neq 6$

$q_i = 1$, when $i = 6$.

This gives us

$$H(p, q) = -\sum_i p_i \log q_i = \log 1 = 0$$

The cross-entropy is much lower.

For our deep learning problem, the cross-entropy can be used as the loss function.

Let's apply this to the classification problem.

Using softmax, we get some output probabilities $\hat{y}_i$.

We use the notation y_i and $\hat{y}_i$ because this is the convention for the output variable although it represents a probability in this case.

The true probability in this case is often the one-hot vector. That is, the vector

$y_i = 0$, if $i \neq t$

$y_i = 1$, if $i = t$

where i is a label and t is the true label associated with the input x.

If our DNN guesses $\hat{y}_i$, the cross-entropy is

$$-\sum_i y_i \log \hat{y}_i = -\log \hat{y}_t$$

If $\hat{y}_t = 1$, the DNN has correctly guessed the label and its certainty is maximum.

If $\hat{y}_t \approx 0$, the loss function (cross-entropy) is very large.

This is what we should expect.