Top highlight

Understanding Entropy, Cross-Entropy and Cross-Entropy Loss

Vijendra Singh
Follow

Apr 3, 2018 · 3 min read

Image source: Key Step Media

Cross Entropy loss is one of the most widely used loss function in Deep learning and this almighty loss function rides on the concept of Cross Entropy. When I started to use this loss function, it was hard for me to get the intuition behind it. After Googling a bit and munching on the concepts I got from different sources, I was able to get a satisfactory understanding and I would like to share it in this article.

In order to develop complete understanding, we need to understand concepts in the following order: Surprisal, Entropy, Cross-Entropy, Cross Entropy Loss

Surprisal:

"*Degree to which you are surprised to see the result*"

Now its easy to digest my word when I say that I will be more surprised to see an outcome with low probability in comparison to an outcome with high probability. Now, if *yi* is the probability of ith outcome then we could represent surprisal (s) as:

$$s = \log(1/y_i)$$

Surprisal

Entropy:

Since I know surprisal for individual outcomes, I would like to know

surprisal for the event. It would be intuitive to take a weighted average of surprisals. Now the question is what weight to choose? Hmmm…since I know the probability of each outcome, taking probability as weight makes sense because this is how likely each outcome is supposed to occur. This weighted average of surprisal is nothing but Entropy (e) and if there are *n* outcomes then it could be written as:

$$e = \sum_{0}^{n} y_i \log(1/y_i)$$

Entropy

Cross-Entropy:

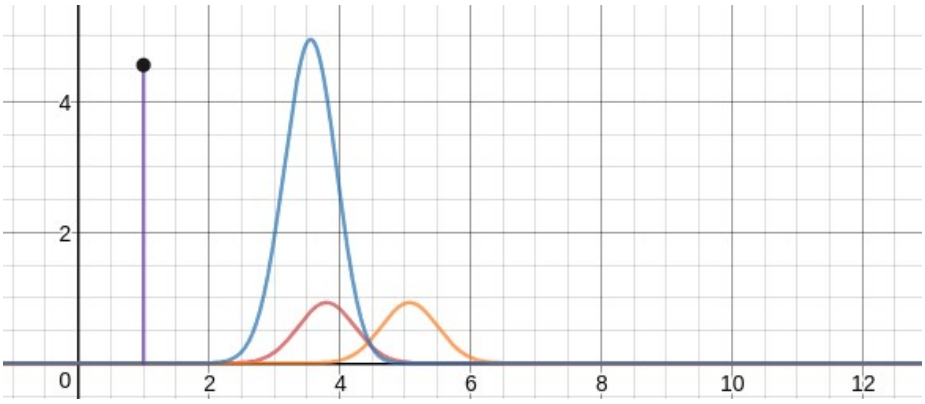
Now, what if each outcome's actual probability is *pi* but someone is estimating probability as *qi*. In this case, each event will occur with the probability of *pi* but surprisal will be given by *qi* in its formula (since that person will be surprised thinking that probability of the outcome is *qi*). Now, weighted average surprisal, in this case, is nothing but cross entropy(c) and it could be scribbled as:

$$c = \sum_{0}^{n} p_i \log(1/q_i)$$

Cross-Entropy

Cross-entropy is always larger than entropy and it will be same as entropy only when *pi=qi*. You could digest the last sentence after seeing really nice plot given by desmos.com

Cross-Entropy Loss:



cross entropy (purple line=area under the blue curve), estimated probability distribution (orange), actual probability distribution (red)

In the plot I mentioned above, you will notice that as estimated probability distribution moves away from actual/desired probability distribution, cross entropy increases and vice-versa. Hence, we could say that minimizing cross entropy will move us closer to actual/desired distribution and that is what we want. This is why we try to reduce cross entropy so that our predicted probability distribution end up being close to the actual one. Hence, we get the formula of cross-entropy loss as:

$$c = \sum_{0}^{n} p_i \log(1/q_i)$$

Cross-Entropy Loss

And in the case of binary classification problem where we have only two classes, we name it as binary cross-entropy loss and above formula becomes:

$$c = \sum_{0}^{1} p_i \log(1/q_i) = p_0 \log(1/q_0) + p_1 \log(1/q_1) = p_0 \log(1/q_0) + (1 - p_0) \log(1/(1 - q_0))$$

Binary Cross-Entropy Loss

Machine Learning | Activation Functions | Cross Entropy | Deep Learning | Understanding

WRITTEN BY

Vijendra Singh

Follow

Software Engineer | Computer Vision and DL Enthusiast

See responses (5)

More From Medium

Also tagged Deep Learning

3 Myths about Data-Science in the corporate world

M. Emmanuel in Towards Data Science

Jul 11 · 6 min read ★

Also tagged Cross Entropy

Intuitive Explanation of Cross Entropy

Lili Jiang in Towards Data Science

Jan 18 · 6 min read

213

Related reads

A Look at Gradient Descent and RMSprop Optimizers

Rohith Gandhi in Towards Data Science

Jun 19, 2018 · 6 min read

513