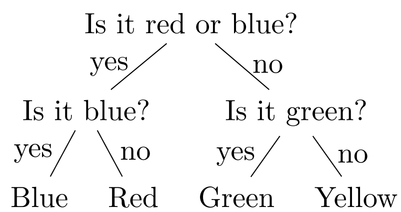
XEntropy explained

J Hope

July 24, 2017

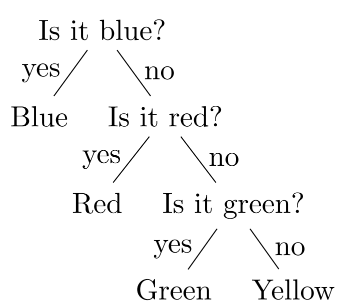
Entropy and Cross Entropy is used extensively in machine learning to evaluate the effectiveness of a given strategy under different situations. For example, cross entropy is used to measure the error as a cost function of a softmax layer in neural networks for the purposes of back propagation.

Entropy can be explained in the context of a simple game. The game works as follows. A player draws a ball from a bag of balls which contains: a blue ball, a red ball, a green ball, and an orange ball. The objective of the game is to guess the correct colour drawn from the bag, with as fewer guesses or 'questions' as possible. One strategy, is described by the tree below.



Under this strategy, each ball has a $1/4$ of probability of getting chosen ($1/2 \times 1/2$), and takes 2 questions to guess the colour of the ball drawn. So the expected number of questions to guess the ball is 2.

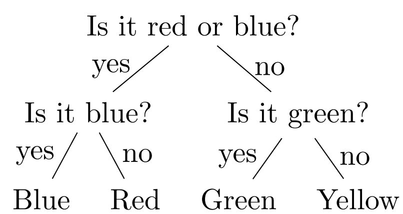
Let's consider a new game. Ball are now drawn from a different bag: $1/2$ of them are blue, $1/4$ are red, $1/8$ are green, and $1/8$ are orange. The strategy that yields the fewest questions - the optimal strategy - is described by the tree below.



$1/2$ of the time it is blue, and it takes 1 question to guess the colour of the ball drawn. $1/4$ of the time it is red, and it takes 2 questions to guess the colour of the ball drawn. The expected (mean) number of questions to guess a ball is calculated as $1/2 \times 1$ question (blue) + $1/4 \times 2$ question (red) + $1/8 \times 3$ questions (green) + $1/8 \times 3$ questions (orange) = 1.75.

Let's consider another scenario; a bag with all blue balls. It takes 0 questions to guess the colour of my ball, or $\log_2(1)$ question. It takes 0 questions if you know that it is a bag of exclusively blues. In fact, a ball with p probability takes $\log_2\left(\frac{1}{p}\right)$ to guess it correctly. For example, when p = $1/4$, $\log_2(1) = 2\log_2(4) = 2$ questions. So in total, the expected number of questions for this game is $\sum_i p_i \log_2\left(\frac{1}{p_i}\right)$. And that is the expression for entropy. Intuitively, it is the expected number of questions to guess the colour under the optimal strategy for this game. The more uncertain the setup is, the higher the entropy.

Cross entropy (often written as XEntropy) is a related idea. Consider again the bag where $1/2$ of the balls are blue, $1/4$ are red, $1/8$ are green, and $1/8$ are orange, under the first strategy (shown again below):



Under this strategy, $1/8$ of the time, the ball is orange, and it takes 2 questions to get it right. $1/2$ of the time, it?s blue but it still takes 2 questions to guess the correct colour. On average, it takes $1/8 \times 2 + 1/4 \times 2 + 1/2 \times 2 = 2$ questions to guess the correct colour. So, 2 is the cross entropy for using this strategy in this scenario.

Cross entropy for a given strategy is simply the expected number of questions to guess the colour under that strategy. For a given setup, the better the strategy is, the lower the cross entropy is. The lowest cross entropy is that of the optimal strategy, which is just the entropy defined above. This is why in machine learning classification problems, we want to minimize the cross entropy.

More formally, cross entropy is $\sum_i p_i \log_2\left(\frac{1}{\hat{p}_i}\right)$, where p_i is the true probability (for example, $1/8$ for orange and green, $1/4$ for red and $1/2$ for blue) and $\hat{p_i}$ is the wrongly assumed or predicted probability (for example, using the first strategy, we are assuming or predicting p = $1/4$ for all colours). It may be easy to confuse whether it?s p_i or $\hat{p_i}$ that is in the log. The logarithm is used to calculate the number of questions under the strategy you are applying; in other words, what is under the log is your predicted probability, $\hat{p_i}$.

References

[1] Intuitive explanations of technical concepts
https://www.quora.com/Whats-an-intuitive-way-to-think-of-cross-entropy