"Deep learning - Information theory & Maximum likelihood."

Jan 5, 2017

Information theory

Information theory quantifies the amount of information present. In information theory, the amount of information is characterized as:

- Predictability:
  - Guaranteed events have zero information.
  - Likely events have little information. (Biased dice have little information.)
  - Random events process more information. (Random dice have more information.)
- Independent events add information. Rolling a dice twice with heads have twice the information of rolling the dice once with a head.

> *In information theory, chaos processes more information.*

Information of an event is defined as:

$$I(x) = -\log(P(x))$$

Entropy

In information theory, entropy measures the amount of information.

We define entropy as:

$$H(x) = E_{x\sim P}[I(x)]$$

So

$$H(x) = -\mathbb{E}_{x\sim P}[logP(x)]$$
$$H(x) = -\sum_{x} P(x)\log P(x)$$

If $\log$ has a base of 2, it measure the number of bits to encode the information. In information theory, information and random-ness are positively correlated. High entropy equals high randomness and requires more bits to encode it.

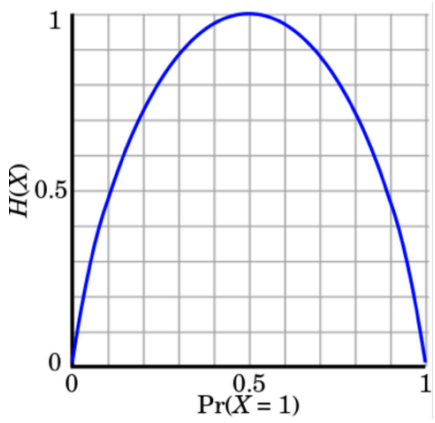
Example

Let's comput the entropy of a coin. For a fair coin:

$$H(X) = -p(head)\cdot \log_2(p(head)) - p(tail)\cdot log_2(p(tail)) = -\log_2 \frac{1}{2} = 1$$

Therefore we can use 1 bit to represent head (0 = head) and 1 bit to represent tail (1 = tail).

The entropy of a coin peaks when $p(head) = p(tail) = 0.5$.



(Source Wikipedia)

For a fair die, $H(X) = \log_2 6 \approx 2.59$. A fair die has more entropy than a fair coin because it is less predictable.

Cross entropy

If entropy measures the minimum number of bits to encode information, cross entropy measures the minimum of bits to encode x with distribution P using the wrong optimized encoding scheme from Q.

Cross entropy is defined as:

$$H(P, Q) = -\sum_{x} P(x) \log Q(x)$$

> *In deep learning, P is the distribution of the true labels, and Q is the probability distribution of the predictions from the deep network.*

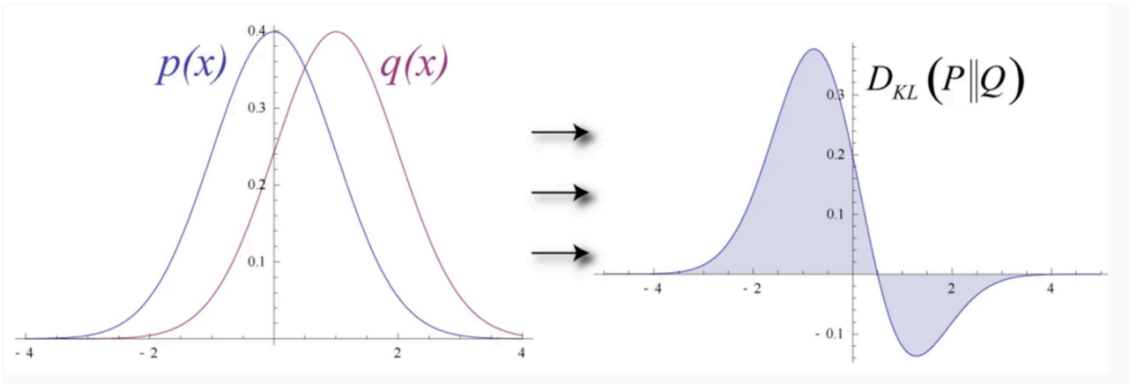
KL Divergence

In deep learning, we want a model predicting data distribution Q resemble the distribution P from the data. Such difference between 2 probability distributions can be measured by KL Divergence which is defined as:

$$D_{KL}(P||Q) = \mathbb{E}_x \log \frac{P(x)}{Q(x)}$$

So,

$$\begin{aligned} D_{KL}(P||Q) &= \sum_{x=1}^{N} P(x) \log \frac{P(x)}{Q(x)} \\ &= \sum_{x=1}^{N} P(x)[\log P(x) - \log Q(x)] \end{aligned}$$



(Source Wikipedia.)

KL-divergence is not commutative: $D_{KL}(P||Q) \neq D_{KL}(Q||P)$.

Recall:

$$
\begin{aligned}
H(P) &= -\sum P \log P, \\
H(P, Q) &= -\sum P \log Q, \quad \text{and} \\
D_{KL}(P||Q) &= \sum P \log \frac{P}{Q}.
\end{aligned}
$$

We can rewrite the cross entropy equation with KL divergence:

$$
\begin{aligned}
H(P, Q) &= -\sum P \log Q \\
&= -\sum P \log P + \sum P \log P - \sum P \log Q \\
&= H(P) + \sum P \log \frac{P}{Q} \\
H(P, Q) &= H(P) + D_{KL}(P||Q)
\end{aligned}
$$

So cross entropy is the sum of entropy and KL-divergence. Cross entropy $H(P, Q)$ is larger than $H(P)$ since we require extra amount of information (bits) to encode data with less optimized scheme from Q if $P \neq Q$. Hence, KL-divergence is always positive for $P \neq Q$ or zero otherwise.

$H(P)$ only depends on P: the probability distribution of the data. Since it is un-related with the model θ we build, we can treat $H(P)$ as a constant. Therefore, **minimizing the cross**

entropy is equivalent to minimize the KL-divergence.

$$
\begin{aligned}
H(P, Q_\theta) &= H(P) + D_{KL}(P||Q_\theta) \\
\nabla_\theta H(P, Q_\theta) &= \nabla_\theta(H(P) + D_{KL}(P||Q_\theta)) \\
&= \nabla_\theta D_{KL}(P||Q_\theta)
\end{aligned}
$$

Maximum Likelihood Estimation

We want to build a model with $\hat{\theta}$ that maximizes the probability of the observed data (a model that fits the data the best: **Maximum Likelihood Estimation MLE**):

$$
\hat{\theta} = \arg\max_\theta \prod_{i=1}^{N} p(x_i|\theta)
$$

However, multiplications overflow or underflow easily. Since $\log(x)$ is monotonic, optimize $log(f(x))$ is the same as optimize $f(x)$. We add the negative sign because the log of a probability invert the direction of $p(x)$. So instead of the MLE, we take the $\log$ and minimize the **negative log likelihood (NLL)**.

$$
\hat{\theta} = \arg\min_\theta - \sum_{i=1}^{N} \log p(x_i|\theta)
$$

NLL and minimizing cross entropy is equivalent:

$$
\begin{aligned}
\hat{\theta} &= \arg\min_\theta - \sum_{i=1}^{N} \log q(x_i|\theta) \\
&= \arg\min_\theta - \sum_{x \in X} p(x) \log q(x|\theta) \\
&= \arg\min_\theta H(p, q)
\end{aligned}
$$

Putting it together

We want to build a model that fits our data the best. We start with the maximum likelihood

estimation (MLE) which later change to negative log likelihood to avoid overflow or underflow. Mathematically, the negative log likelihood and the cross entropy have the same equation. KL divergence provides another perspective in optimizing a model. However, even they uses different formula, they both end up with the same solution.

> *Cross entropy is one common objective function in deep learning.*

Mean square error (MSE)

In a regression problem, $y = f(x; w)$. In real life, we are dealing with un-certainty and in-complete information. So we may model the problem as:

$$\hat{y} = f(x; \theta)$$
$$y \sim \mathcal{N}(y; \mu = \hat{y}, \sigma^2)$$
$$p(y|x; \theta) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(y - \hat{y})^2}{2\sigma^2}\right)$$

with σ pre-defined by users:

The log likelihood becomes optimizing the mean square error:

$$
\begin{aligned}
J &= \sum_{i=1}^{m} \log p(y|x;\theta) \\
&= \sum_{i=1}^{m} \log \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(y^{(i)} - \hat{y^{(i)}})^2}{2\sigma^2}\right) \\
&= \sum_{i=1}^{m} -\log(\sigma\sqrt{2\pi}) - \log \exp\left(\frac{(y^{(i)} - \hat{y^{(i)}})^2}{2\sigma^2}\right) \\
&= \sum_{i=1}^{m} -\log(\sigma) - \frac{1}{2}\log(2\pi) - \frac{(y^{(i)} - \hat{y^{(i)}})^2}{2\sigma^2} \\
&= -m\log(\sigma) - \frac{m}{2}\log(2\pi) - \sum_{i=1}^{m} \frac{(y^{(i)} - \hat{y^{(i)}})^2}{2\sigma^2} \\
&= -m\log(\sigma) - \frac{m}{2}\log(2\pi) - \sum_{i=1}^{m} \frac{\|y^{(i)} - \hat{y^{(i)}}\|^2}{2\sigma^2} \\
\nabla_\theta J &= -\nabla_\theta \sum_{i=1}^{m} \frac{\|y^{(i)} - \hat{y^{(i)}}\|^2}{2\sigma^2}
\end{aligned}
$$

> *Many cost functions used in deep learning, including the MSE, can be derived from the MLE.*

Maximum A Posteriori (MAP)

MLE maximizes $p(y|x;\theta)$.

$$\theta^* = \arg\max_{\theta}(P(y|x;\theta)) = \arg\max_{w} \prod_{i=1}^{n} P(y|x;\theta)$$

Alternative, we can find the most likely θ given y:

$$\theta^*_{MAP} = \arg\max_{\theta} p(\theta|y) = \arg\max_{\theta} \log p(y|\theta) + \log p(\theta)$$

Apply Bayes' theorem:

$$\begin{aligned}\theta_{MAP} &= \arg\max_{\theta} p(\theta|y) \\ &= \arg\max_{\theta} \log p(y|\theta) + \log p(\theta) - \log p(y) \\ &= \arg\max_{\theta} \log p(y|\theta) + \log p(\theta)\end{aligned}$$

To demonstrate the idea, we use a Gaussian distribution of $\mu = 0, \sigma^2 = \frac{1}{\lambda}$ as the our prior:

$$p(\theta) = \frac{1}{\sqrt{2\pi\frac{1}{\lambda}}} e^{-\frac{(\theta-0)^2}{2\frac{1}{\lambda}}}$$

$$\begin{aligned}\log p(\theta) &= -\log\sqrt{2\pi\frac{1}{\lambda}} + \log e^{-\frac{\lambda}{2}\theta^2} \\ &= C' - \frac{\lambda}{2}\theta^2\end{aligned}$$

$$-\sum_{j=1}^{N} \log p(\theta) = C + \frac{\lambda}{2}\|\theta\|^2 \qquad \text{L-2 regularization}$$

Assume the likelihood is also gaussian distributed:

$$p(y^{(i)}|\theta) \propto e^{-\frac{(y^{(i)}-y^{(i)})^2}{2\sigma^2}}$$

$$-\sum_{i=1}^{N} \log p(y^{(i)}|\theta) \propto \frac{1}{2\sigma^2}\|y^{(i)} - y^{(i)}\|^2$$

So for a Gaussian distribution prior and likelihood, the cost function is

$$
\begin{aligned}
J(\theta) &= \sum_{i=1}^{N} -\log p(y^{(i)}|\theta) - \log p(\theta) \\
&= -\sum_{i=1}^{N} \log p(y^{(i)}|\theta) - \sum_{j=1}^{d} \log p(\theta) \\
&= \frac{1}{2\sigma^2}\|y^{(i)} - y^{(i)}\|^2 + \frac{\lambda}{2}\|\theta\|^2 + constant
\end{aligned}
$$

which is the same as the MSE with L2-regularization.

If the likeliness is computed from a logistic function, the corresponding cost function is:

$$
\begin{aligned}
p(y_i|x_i, w) &= \frac{1}{1 + e^{-y_i w^T x_i}} \\
J(w) &= -\sum_{i=1}^{N} \log p(y_i|x_i, w) - \sum_{j=1}^{d} \log p(w_j) - C \\
&= \sum_{i=1}^{N} \log(1 + e^{-y_i w^T x_i}) + \frac{\lambda}{2}\|w\|^2 + constant
\end{aligned}
$$

Like, MLE, MAP provides a mechanism to derive the cost function. However, MAP can also model the uncertainty into the cost function which turns out to be the regularization factor used in deep learning.

Nash Equilibrium

In the game theory, the Nash Equilibrium is reached when no player will change its strategy after considering all possible strategy of opponents. i.e. in the Nash equilibrium, no one will change its decision even after we will all the player strategy to everyone. A game can have 0, 1 or multiple Nash Equilibria.

The Prisoner's Dilemma

In the prisoner's dilemma problem, police arrests 2 suspects but only have evidence to

charge them for a lesser crime with 1 month jail time. But if one of them confess, the other party will receive a 12 months jail time and the one confess will be released. Yet, if both confess, both will receive a jail time of 6 months. The first value in each cell is what Mary will get in jail time for each decision combinations while the second value is what Peter will get.

		Peter	
		Quiet	Confess
Mary	Quiet	-1, -1,	-12, 0
	Confess	0, -12	-6, -6

For Mary, if she thinks Peter will keep quiet, her best strategy will be confess to receive no jail time instead of 1 month.

		Peter	
		Quiet	Confess
Mary	Quiet	-1, -1,	-12, 0
	Confess	0, -12	-6, -6

On the other hand, if she thinks Peter will confess, her best strategy will be confess also to get 6 months jail time.

		Peter	
		Quiet	Confess
Mary	Quiet	-1, -1,	-12, 0
	Confess	0, -12	-6, -6

After knowing all possible actions, in either cases, Mary's best action is to confess. Similarly, Peter should confess also. Therefore (-6, -6) is the Nash Equilibrium even (-1, -1) is the least

jail time combined. Why (-1, -1) is not a Nash Equilibrium? Because if Mary knows Peter will keep quiet, she can switch to confess and get a lesser sentence which Peter will response by confessing the crime also. (Providing that Peter and Mary cannot co-ordinate their strategy.)

Jensen-Shannon Divergence

It measures how distinguishable two or more distributions are from each other.

$$JSD(X||Y) = H(\frac{X+Y}{2}) - \frac{H(X)+H(Y)}{2}$$