EECS 182 Deep Neural Networks
Spring 2023 Anant Sahai

Discussion 13

1. Entropy, Cross-Entropy, Kullback - Leibler (KL)-divergence

Entropy is a measure of expected surprise. For a given discrete Random variable Y, we know that from Information Theory that a measure the surprise of observing that Y takes the value k by computing:

$$\log \frac{1}{p(Y=k)} = -\log[p(Y=k)]$$

As given:

- if $p(Y=k) \to 0$, the surprise of observing k approaches ∞
- if $p(Y=k) \to 1$, the surprise of observing k approaches 0

The Entropy of the distribution of Y is then the expected surprise given by:

$$H(Y) = E_Y\Big[-\log\big(p(Y=k)\big)\Big] = -\Sigma_k\Big[p(Y=k)\log[p(Y=k)]\Big]$$

On the other hand, Cross-entropy is a measure building upon entropy, generally calculating the difference between two probability distributions p and q. it is given by:

$$\begin{aligned} H(p,q) &= E_{p(x)}\Big[\frac{1}{\log\big(q(x)\big)}\Big] \\ &= \Sigma_x\Big[p(x)\log[\frac{1}{q(x)}]\Big] \end{aligned}$$

Relative Entropy also known as KL Divergence measures how much one distribution diverges from another. For two discrete probability distributions, p and q, it is defined as:

$$D_{KL}(p||q) = \Sigma_x\Big[p(x)\log[\frac{p(x)}{q(x)}]\Big]$$

(a) Let's define the following probability distributions given by:

$$p(x) = \begin{cases} 1 & \text{with probability } 0.5 \\ -1 & \text{with probability } 0.5 \end{cases}$$

$$q(x) = \begin{cases} 1 & \text{with probability } 0.1 \\ -1 & \text{with probability } 0.9 \end{cases}$$

Show that KL-divergence is not symmetric and hence does not satisfy some intuitive attributes of distances.

(b) Re-write $D_{KL}(p||q)$ in term of the Entropy $H(p)$ and the cross entropy $H(p, q)$.

2. Reparameterization Trick

Formally, a latent variable model p is a probability distribution over observed variables x and latent variables z (variables that are not directly observed but inferred), $p_\theta(x, z)$. Because we know z is unobserved, using learning methods learned in class (like supervised learning methods) is unsuitable. Indeed, our learning problem of maximizing the log-likelihood of the data turns from:

$$\theta \leftarrow arg\max_\theta \frac{1}{N}\Sigma_{i=1}^N \log[p_\theta(x_i)]$$

to:

$$\theta \leftarrow arg\max_\theta \frac{1}{N}\Sigma_{i=1}^N \log[\int p_\theta(x_i \mid z)p(z)dz]$$

where $p(x)$ has become $\int p_\theta(x_i \mid z)p(z)dz$.

(a) State **whether or not we could directly maximize the likelihood above and why?**

(b) Instead of directly optimizing the likelihood of $p(x)$, we define the proxy likelihood as:

$$\mathcal{L}(x_i, \theta, \phi) = E_{z\sim q_\phi(z|x_i)}\Big[\log[p_\theta(x_i \mid z)]\Big] - D_{KL}\Big[q_\phi(z \mid x_i)||p(z)\Big]$$

This proxy term is a *lower bound* of the original likelihood. In order to optimize this variational lower bound, **which distribution do we sample from?**

(c) How do we take gradients through samples? To do we, we need to show how sampling can be done as a deterministic and continuous function of the model parameters θ and the independent source of randomness (ie. the *prior*). Such an explicit representation of sampling is called **reparameterization**. Consider the case where the data x is sampled from a normal distribution with its mean parameterized by parameters θ and variance of 1, with our objective being a quadratic function of x:

$$\min_{\theta} E_q[x^2]$$

Write x as a function of ϵ, a vector sampled from a standard Normal $\mathcal{N}(0,1)$, and compute the gradient of the expectation term above:

(d) Now consider a more generic case where we would like to optimize

$$\min_{\theta} E_v[\mathcal{L}(x)]$$

where x is sampled from a learnt latent function $f_\theta(u, v)$ that is dependent on u the input data and v the independent randomness. **Show that the gradient $\nabla_\theta E_v[\mathcal{L}(x)]$ can be estimated by samples of $\nabla_\theta f_\theta(u, v)$.** (*Hint*: the process of this question is very similar to the previous part.)

3. Latent Variable Models

(a) **Describe step-by-step what happens during a forward pass in VAE training.** Use the notation from the variational lower bound term (the "proxy likelihood") in the previous question, namely $q_\phi(z \mid x), p_\theta(z \mid x_i), D_{KL}(\cdot)$ ... etc.

(b) **Describe what the encoder and decoder of the VAE are *respectively* doing** to capture and encode this information into a latent representation of space z. **Is the latent space dimension smaller that the input space? How is the information bottleneck created in VAE as opposed to Autoencoder.**

(c) Once the VAE is trained, **how do we use it to generate a new fresh sample from the learned approximation of the data-generating distribution?**

(d) In the previous question we have used a proxy likelihood:

$$\mathcal{L}(x_i, \theta, \phi) = E_{z \sim q_\phi(z|x_i)}\Big[\log[p_\theta(x_i \mid z)]\Big] - D_{KL}\Big[q_\phi(z \mid x_i)||p(z)\Big]$$

Please show that $\mathcal{L}(x_i, \theta, \phi)$ is always a lower bound to the true log likelihood for x_i.

Hint: You can show that something is a lower bound by showing that adding a non-negative term to it gives the original quantity — remember, the KL divergence is always non-negative.

Contributors:

- Jerome Quenum.
- Anant Sahai.
- Kevin Li.
- Past CS282 Staff.

