EECS 182 Deep Neural Networks
Fall 2022 Anant Sahai

Discussion 12

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

(a) 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 $p(Y=k) \rightharpoonup 0$, the surprise of observing k approaches ∞
- if $p(Y=k) \rightharpoonup 1$, 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 Divervenge measures how much one distribution diverges from another. For two discrete probability distributions, p and q, it is defined as:

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

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.

Solution:

To show this, we need to show that:

$$D_{KL}(p||q) \neq D_{KL}(q||p)$$

$$D_{KL}(p||q) = 0.5 \times \log[\frac{0.5}{0.1}] + 0.5 \times \log[\frac{0.5}{0.9}]$$
$$D_{KL}(q||p) = 0.1 \times \log[\frac{0.1}{0.5}] + 0.9 \times \log[\frac{0.9}{0.1}]$$

hence D$_{KL}(p||q) \neq D_{KL}(q||p)$

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

Solution:

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

(c) Show that KL - divergence is always non-negative using Jensen's Inequality which states: E[$\log X$] $\leq \log E[X]$ and the fact that log is a concave function.

Solution: We will show that - $D_{KL}(p||q) \leq 0$ which implies that $D_{KL}(p||q) \geq 0$.

$$\begin{aligned}
-D_{KL}(p||q) &= -\Sigma_x \Big[p(x)\log[\frac{p(x)}{q(x)}]\Big] \\
&= \Sigma_x \Big[p(x)\log[\frac{q(x)}{p(x)}]\Big] \\
&\leq \log\Big[\Sigma_x p(x)[\frac{q(x)}{p(x)}]\Big] \\
&\leq \log\Big[\Sigma_x q(x)\Big] \\
&\leq \log\Big[1\Big] \\
&\leq 0
\end{aligned}$$

(d) Knowing that the equality in Jensen's inequality can only hold if X is a constant random variable, please state when is $D_{KL}(q||p) = 0$. ?

Solution: iff $p = q$

2. Simple Latent Variable Models

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:

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

to:

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

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?

Solution: No, we can't because, in the integral, it is intractable to compute $p(x \mid z)$ for every z. On the other hand, if we look at the posterior density given by $p(z \mid x) = \frac{p(x|z)p(z)}{p(x)}$, we can see that $p(x)$ is also intractable.

(b) We define the proxy likelihood given by:

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

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.

Solution:

$$\log p_\theta(x_i) = E_{z \sim q_\phi(z|x_i)}\Big[\log p_\theta(x_i)\Big]$$
$$= E_{z \sim q_\phi(z|x_i)}\Big[\log \tfrac{p_\theta(x_i|z)p_\theta(z)}{p_\theta(z|x_i)}\Big]$$
$$= E_{z \sim q_\phi(z|x_i)}\Big[\log \tfrac{p_\theta(x_i|z)p_\theta(z)}{p_\theta(z|x_i)} \tfrac{q_\phi(z|x_i)}{q_\phi(z|x_i)}\Big]$$
$$= E_{z \sim q_\phi(z|x_i)}\Big[\log p_\theta(x_i \mid z)\Big] - E_{z \sim q_\phi(z|x_i)}\Big[\log \tfrac{q_\phi(z|x_i)}{p_\theta(z)}\Big] + E_{z \sim q_\phi(z|x_i)}\Big[\log \tfrac{q_\phi(z|x_i)}{p_\theta(z|x_i)}\Big]$$

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

Because $D_{KL}(q_\phi(z \mid x_i)||p_\theta(z \mid x_i)) \geq 0$, and is not tractable due to $p_\theta(z \mid x_i)$ we can conclude that:
$\log p_\theta(x_i) \geq \mathcal{L}(x_i, \theta, \phi) = E_{z \sim q_\phi(z|x_i)}\Big[\log p_\theta(x_i \mid z)\Big] - D_{KL}(q_\phi(z \mid x_i)||p_\theta(z))$
Alternatively we could use Jensen's Inequality, which states, $\log E[X] \geq E[\log X]$ to show that:

$$\Sigma_{i=1}^N \log[p_\theta(x_i)] \geq \Sigma_{i=1}^N E_{q(z|x_i)}[\log\big(p_\theta(z)\big) - \log\big(p_q(z \mid x_i)\big) + \log\big(p_\theta(x_i \mid z)\big)]$$

That is:

We first write out the log-likelihood objective of a discrete latent variable model.

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

then,

$$\begin{aligned}
\Sigma_{i=1}^N \log[p_\theta(x_i)] &= \Sigma_{i=1}^N \Big(\Sigma_z \log[p_\theta(z)p_\theta(x_i \mid z)]\Big) \\
&= \Sigma_{i=1}^N \Big(\Sigma_z \log[\frac{q_\phi(z \mid x_i)}{q_\phi(z \mid x_i)} p_\theta(z)p_\theta(x_i \mid z)]\Big) \\
&= \Sigma_{i=1}^N \Big(\Sigma_z \log E_{q_\phi(z|x_i)}[\frac{1}{q_\phi(z \mid x_i)} p_\theta(z)p_\theta(x_i \mid z)]\Big) \\
\Sigma_{i=1}^N \log[p_\theta(x_i)] &\geq \Sigma_{i=1}^N E_{q(z|x_i)}[\log\big(p_\theta(z)\big) - \log\big(p_q(z \mid x_i)\big) + \log\big(p_\theta(x_i \mid z)\big)]
\end{aligned}$$

(c) To optimize the Variational Lower Bound derived in the previous problem, which distribution do we sample z from?

Solution: We sample from $q_\phi(z \mid x_i)$

(d) To be able to take a derivative through a sampling operation, we need to show how sampling can be done as a deterministic and continuous function of functions of parameters as well as an external independent source of randomness. Otherwise, it is hard to understand how things would change a little bit if the parameters changed a little bit. Such explicit representations of sampling are called "the reparameterization trick" in machine-learning communities. Assume we have a normal distribution for x with both means and variance parameterized by parameters θ and we would like to solve for:

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

Assuming that ϵ is an independent standard Normal $\mathcal{N}(0,1)$ random variable, write x as a function of ϵ and use that to compute the gradient of the objective function above.

Solution: We can first make the stochastic element in q independent of θ, and rewrite x as:

$$x = +\epsilon, \epsilon \sim \mathcal{N}(0, 1)$$

then:

$$E_q[x^2] = E_p[(\theta + \epsilon)^2]$$

where $p \sim \mathcal{N}(0, 1)$. Then we can write the derivative of $E_q[x^2]$ as:

$$\begin{aligned} \nabla_\theta E_q[x^2] &= \nabla_\theta E_p[(\theta + \epsilon)^2] \\ &= E_p[2(\theta + \epsilon)] \end{aligned}$$

(e) Describe step-by-step what happens during a forward pass during VAE training

Solution: For a forward pass, through which we run our minibatch of input data,

i. We pass this through our Encoder network ($q_\phi(z \mid x)$). Note this is specifically optimized through the second term in our lower bound loss function (ELBO) i. e $D_{KL}(q_\phi(z \mid x_i)||p_\theta(z \mid x_i))$ whose only goal is to make an approximation of our posterior distribution.

ii. We then sample z from $z|x \sim \mathcal{N}(\mu_{z|x}, \Sigma_{z|x})$. These are the samples of latent factors that we can infer from x

iii. We pass the obtained z through our Decoder network ($p_\theta(x \mid z)$). We then sample $\hat{x}$ from $x|z \sim \mathcal{N}(\mu_{x|z}, \Sigma_{x|z})$. Note that is handled specifically by the first term is our loss i. e $E_{z \sim q_\phi(z|x_i)}\Big[\log p_\theta(x_i \mid z)\Big]$ whose only goal is to maximize the likelihood of the original input being reconstructed.

iv. Once the compute the loss which is differentiable, we backpropagate and update parameters.

(f) Describe what the encoder and decoder of the VAE are doing to capture and encode this information into a latent representation of space z.

Solution:

i. **Encoder** - Encoder maps a high-dimensional input x (like the pixels of an image) and then (most often) outputs the parameters of a Gaussian distribution that specify the hidden variable z. In other words, they output $\mu_{z|x}$ and $\Sigma_{z|x}$. We will implement this as a deep neural network, parameterized by ϕ, which computes the probability $q_\phi(z|x)$. We could then sample from this distribution to get noisy values of the representation z.

ii. **Decoder** - Decoder maps the latent representation back to a high dimensional reconstruction, denoted as $\hat{x}$, and outputs the parameters to the probability distribution of the data. We will implement this as another neural network, parametrized by θ, which computes the probability $p_\theta(x|z)$. In the MNIST dataset example, if we represent each pixel as a 0 (black) or 1 (white), the probability distribution of a single pixel can be then represented using a Bernoulli distribution. Indeed, the decoder gets as input the latent representation of a digit z and outputs 784 Bernoulli parameters, one for each of the 784 pixels in the image.

(g) 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.?

Solution: We can now use only the Decoder network ($p_\theta(x \mid z)$). Here, instead of sampling z from the posterior that we had during training, we sample from our true generative process which is the prior that we had specified ($z \sim \mathcal{N}(0, I)$) and we proceed to use the network to sample $\hat{x}$ from there.

Contributors:

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

