Lecture 22 Generative Adversarial Neural Networks (GANs).

In this series of lectures we will discuss to two generative model frameworks that have gained wide appeal in the last years: Generative Adversarial Neural Networks (GANs) and variational Autoencoders (VAEs). These models are based on differential neural networks, therefore they will be trained by backpropagation as implemented, e.g., in high-level python packages such as Keras.

22.1 Limitations of maximising the likelihood.

The Kullback–Leibler (KL) divergence plays a central role in many generative models. It measures the similarity between two probability distributions $p(\vec{x})$ and $q(\vec{x})$. Given two distributions, there are two distinct KL divergences

$$D_{KL}(p\|q) = \int d\vec{x}\, p(\vec{x}) \ln \frac{p(\vec{x})}{q(\vec{x})}$$

$$D_{KL}(q\|p) = \int d\vec{x}\, q(\vec{x}) \ln \frac{q(\vec{x})}{p(\vec{x})}$$

These can be combined to construct the symmetric square metric

$$D_{JS}(p,q) = \frac{1}{2}\left[D_{KL}\left(p \,\Big\|\, \frac{p+q}{2}\right) + D_{KL}\left(q \,\Big\|\, \frac{p+q}{2}\right)\right]$$

which is called Jensen–Shannon divergence. An important property of the KL divergence is its positivity

$$D_{KL}(p\|q) \geq 0 \quad (\text{with equality if and only if } p(\vec{x}) = q(\vec{x}))$$

In ML, the two distributions we are interested in are

- the model distribution $p_\vartheta(\vec{x})$
- the data distribution $p_{data}(\vec{x})$

We want $p_\vartheta(\vec{x})$ to be as similar as possible to $p_{data}(\vec{x})$.

Example: let us consider two Gaussian distributions (1D)

$$p(x) = \frac{1}{\sqrt{2\pi}\,\sigma_p}\, e^{-(x-x_p)^2/2\sigma_p^2} \qquad q(x) = \frac{1}{\sqrt{2\pi}\,\sigma_q}\, e^{-(x-x_q)^2/2\sigma_q^2}$$

$$D_{KL}\left(p(x)\,\|\,q(x)\right) = \ln\frac{\sigma_q}{\sigma_p} + \frac{1}{2\sigma_q^2}\left\{(x_q-x_p)^2 + (\sigma_p^2-\sigma_q^2)\right\}$$

The two distributions are alike if the difference between their central values and standard deviations is small.

Maximizing the log-likelihood of the data under the model is the same as minimizing the KL divergence between the data distribution and the model distribution $D_{KL}(p_{data}\,\|\,p_\vartheta)$.

Let us rewrite

$$D_{KL}(p_{data}\,\|\,p_\vartheta) = \overbrace{\int d\vec{x}\, p_{data}(\vec{x}) \ln p_{data}(\vec{x})}^{-\text{ENTROPY}} - \int d\vec{x}\, p_{data}(\vec{x}) \ln p_\vartheta(\vec{x})$$

$$= -S[p_{data}] - \langle \ln p_\vartheta(\vec{x})\rangle_{data}$$

We have

$$\langle \ln p_\vartheta(\vec{x})\rangle_{data} = -S[p_{data}] - D_{KL}(p_{data}\,\|\,p_\vartheta)$$

The equivalence follows from the positivity of the KL divergence and the fact that the entropy of the data distribution is constant. The original formulation of GANs (Goodfellow) minimises an upper bound on the Jensen–Shannon divergence between the model distribution $p_\vartheta(\vec{x})$ and the data distribution $p_{data}(\bar{x})$.

Let us compare more closely the behaviour of the two KL divergences $D_{KL}(p_{data} \| p_\theta)$ and $D_{KL}(p_\theta \| p_{data})$. Both these KL-divergences measure similarities between p_{data} and p_θ, yet they are sensitive to very different things.

- $D_{KL}(p_\theta \| p_{data})$ is insensitive to setting $p_\theta \approx 0$ even when $p_{data} \neq 0$ whereas $D_{KL}(p_{data} \| p_\theta)$ punishes this harshly.
- $D_{KL}(p_{data} \| p_\theta)$ is insensitive to placing weight in the model distribution in regions where $p_{data} \approx 0$ whereas $D_{KL}(p_\theta \| p_{data})$ punishes this harshly.

In other words, $D_{KL}(p_{data} \| p_\theta)$ prefers model that have a high probability in regions with lots of training data points whereas $D_{KL}(p_\theta \| p_{data})$ punishes models for putting high probability where there is no data.

This suggests that the way likelihood-based methods can fail is by improperly "filling in" any low-probability density regions between peaks in the data distribution. The Jensen-Shannon divergence is sensitive to both placing weight where there is data and not placing weight where there is no data.

Now, $D_{KL}(p_\theta \| p_{data})$ is impossible to compute because we do not know $p_{data}(\bar{x})$. Conversely, $D_{KL}(p_{data} \| p_\theta)$ can be computed easily from the data using sampling. The idea of adversarial learning is to circumvent the aforementioned difficulty by using an adversarial learning procedure. It consists in training a discriminator network to distinguish between real data points and samples generated from the model.

By punishing the model for generating points that can be easily discriminated from the data, adversarial learning decreases the weight of regions in the model space that are far away from data points (regions that inevitably arise when maximising the likelihood).

22.2 Generative models and adversarial learning

The central idea of GANs is to construct two differentiable neural networks. The first neural network, usually a de-convolutional network approximates a generator function $G(\vec{z}, \vartheta_G)$ that takes as input a $\vec{z}$ sampled from some prior in the latent space, and outputs a $\vec{x}$ from the model. The second network approximates a discriminator function $D(\vec{x}, \vartheta_D)$ that is designed to distinguish between $\vec{x}$ from the data and samples generated by the model: $\vec{x} = G(\vec{z}, \vartheta_G)$. The scalar $D(\vec{x})$ represents the probability that $\vec{x}$ came from the data rather than the model p_{ϑ_G}. We train D to distinguish actual data points from synthetic examples and the generative network to fool the discriminative network.

To define the cost function for training, it is useful to define the functional

$$V(D, G) = \mathbb{E}_{X \sim p_{data}}\left(\log D(\vec{x})\right) + \mathbb{E}_{Z \sim p_{prior}}\left(\log\left[1 - D(G(\vec{z}))\right]\right)$$

We take the cost function for the discriminator and generator to be $C^{(G)} = - C^{(D)} = \frac{1}{2} V(D, G)$. This choice of cost functions corresponds to what is called a zero-sum game. Since the discriminator is maximised, we can write a cost function for the generator as

$$C(G) = \max_{D} V(G, D)$$

which can be rewritten as

$$C(G) = -\ln 4 + 2 D_{JS}(p_{data}, p_{\vartheta_G})$$