Structured Model Learning
Variational Autoencoders

Boris Flach
Czech Technical University in Prague

Variational Autoencoders

So far we where considering neural networks as predictors, mapping input signals (features) to object states. Now, we reverse the task: we want to learn distributions on features (e.g. images) to be able to sample from them.

- Let $\mathcal{X}$ denote the image space and $\mathcal{Z}$ denote a noise space. We may assume e.g. $\mathcal{Z} = \mathbb{R}^n$. We fix a simple distribution $Z \sim \mathcal{N}(0, \mathbb{I})$ on it.
- Given a sample $\mathcal{S}^m = \{x^j \in \mathcal{X} \mid j = 1, \ldots, m\}$, we want to learn a parametrised conditional distribution $p_w(x \mid z)$ such that the likelihood of $\mathcal{S}^m$ w.r.t. to the joint model $p_w(x, z)$ is maximised

The task reads as

$$\frac{1}{m}\sum_{j=1}^{m} \log p_w(x^j) = \frac{1}{m}\sum_{j=1}^{m} \log \int p_w(x^j, z)\, dz \to \max_w \tag{1}$$

How to model $p_w(x \mid z)$:

- Consider a deterministic (convolutional) neural network, whose outputs are interpreted as parameters of a distribution on $\mathcal{X}$. E.g. $X \mid Z \sim \mathcal{N}(\mu_w(z), \mathrm{diag}(\sigma_w(z)))$.

Variational Autoencoders

- another variant is to consider $p_w(x \mid z)$ as a probabilistic neural network (see lecture 3.)

Let us now consider a summand in (1)

$$\log p_w(x) = \log \int p_w(x, z)\, dz$$

We can not compute this integral $\Rightarrow$ lower bound it in the same way as in the expectation maximisation algorithm

$$\log \int p_w(x, z)\, dz = \log \int \frac{q_v(z \mid x)}{q_v(z \mid x)}\, p_w(x, z)\, dz \geqslant \int q_v(z \mid x) \log \frac{p_w(x, z)}{q_v(z \mid x)}\, dz$$

This bound is tight if $q_v(z \mid x) = p_w(z \mid x)$, which is however hard to compute.

Main assumption of VAEs: the distributions $q_v(z \mid x)$ are modelled by a single neural network in one of the ways discussed above for $p(x|z)$.

Notice that this may result in gaps of the lower bound, if the requirement

$$\forall w\ \exists v(w) \text{ s.t. } q_{v(w)}(z \mid x) = p_w(z \mid x)$$

is not met.

Variational Autoencoders

With the assumptions made, the lower bound of the learning objective (1) reads

$$\frac{1}{m}\sum_{j=1}^{m}\Big[\int q_v(z \mid x^j)\log p_w(x^j \mid z) - KL(q_v(z \mid x^j) \parallel p(z))\Big] \to \max_{w,v}$$

- The KL-divergence can be computed in closed form if $q_v(z \mid x)$ is modelled as multivariate Gaussian
- the expectation $\mathbb{E}_{z\sim q_v}\log p_w(x \mid z)$ is approximated by a sample
- notice however that the first term must be differentiated w.r.t. w and v.
- How to differentiate a $z \sim q_v(z \mid x)$ w.r.t. v? By the reparametrisation trick discussed in previous lectures

Variational Autoencoders

Sophisticated versions of deep convolutional variational autoencoders achieve quite impressive results

Open problems: When using deep belief networks both for the "decoder" and the "encoder", we observe "latent variable collapse". This means that the last layers of the encoder model have the tendency to collapse into the decoder's prior, i.e.

$$q_v(z_t \mid z_{t-1}) \sim \mathcal{N}(0, \mathbb{I}).$$

We conjecture that the reason is that a layered belief network $q_v(z \mid x)$ can not in principle approximate reverse conditional probabilities of another layered belief network (the decoder).