

1 Generative modelling

Learn  $p_{\text{model}} \approx p_{\text{data}}$ , sample from  $p_{\text{model}}$ .

- Explicit density:
  - Approximate:
    - \* Variational: VAE, Diffusion
    - \* Markov Chain: Boltzmann machine
  - Tractable:
    - \* Autoregressive: FVSBN/NADE/MADE, Pixel(C/R)NN, WaveNet/tcn, Autor. Transf.,
    - \* Normalizing Flows
- Implicit density:
  - Direct: Generative Adversarial Networks
  - MC: Generative Stochastic Networks

Autoencoder:  $X \rightarrow Z \rightarrow X$ ,  $g \circ f \approx \text{id}$ ,  $f$  and  $g$  are NNs. Optimal linear autoencoder is PCA. Undercomplete:  $|Z| < |X|$ , else overcomplete. Overcomp. is for denoising, inpainting. Latent space should be continuous and interpolable. Autoencoder spaces are neither, so they are only good for reconstruction.

2 Variational AutoEncoder (VAE)

Sample  $z$  from prior  $p_{\theta}(z)$ , to decode use conditional  $p_{\theta}(x | z)$  defined by a NN.

$D_{\text{KL}}(P||Q) := \int_x p(x) \log \frac{p(x)}{q(x)} dx$ : KL divergence, measure similarity of prob. distr.  $D_{\text{KL}}(P||Q) \neq D_{\text{KL}}(Q||P)$ ,  $D_{\text{KL}}(P||Q) \geq 0$   
Likelihood  $p_{\theta}(x) = \int_z p_{\theta}(x | z) p_{\theta}(z) dz$  is hard to max., let enc. NN be  $q_{\phi}(z | x)$ ,  $\log p_{\theta}(x^i) = \mathbb{E}_z [\log p_{\theta}(x^i | z)] - D_{\text{KL}}(q_{\phi}(z | x^i) || p_{\theta}(z)) + D_{\text{KL}}(q_{\phi}(z | x^i) || p_{\theta}(z | x^i))$ . Red is intractable, use  $\geq 0$  to ignore it; Orange is reconstruction loss, clusters similar samples; Purple makes posterior close to prior, adds cont. and interp. Orange - Purple is ELBO, maximize it.

$x \xrightarrow{\text{enc}} \mu_{z|x}, \Sigma_{z|x} \xrightarrow{\text{sample}} z \xrightarrow{\text{dec}} \mu_{x|z}, \Sigma_{x|z} \xrightarrow{\text{sample}} \hat{x}$   
Backprop through sample by reparametr.:  $z = \mu + \sigma \epsilon$ . For inference, use  $\mu$  directly.  
Disentanglement: features should correspond to distinct factors of variation. Can be done with semi-supervised learning by making  $z$  conditionally independent of given features  $y$ .

2.1  $\beta$ -VAE

Disentangle by  $\max_{\theta, \phi} \mathbb{E}_x [\mathbb{E}_{z \sim q_{\phi}} \log p_{\theta}(x | z)]$  s.t.  $D_{\text{KL}}(q_{\phi}(z | x) || p_{\theta}(z)) < \delta$ , with KKT:  $\max \text{Orange} - \beta \text{Purple}$ .

3 Autoregressive generative models

Autoregression: use data from the same input variable at previous time steps  
Discriminative:  $P(Y | X)$ , generative:  $P(X, Y)$ ,

maybe with  $Y$  missing. Sequence models are generative: from  $x_1 \dots x_{i+k}$  predict  $x_{i+k+1}$ .

Tabular approach:  $p(\mathbf{x}) = \prod_i p(x_i | \mathbf{x}_{<i})$ , needs  $2^{i-1}$  params. Independence assumption is too strong. Let  $p_{\theta_i}(x_i | \mathbf{x}_{<i}) = \text{Bern}(f_i(\mathbf{x}_{<i}))$ , where  $f_i$  is a NN. **Fully Visible Sigmoid Belief Networks**:  $f_i = \sigma(\alpha_0^{(i)} + \alpha^{(i)} \mathbf{x}_{<i}^T)$ , complexity  $n^2$ , but model is linear.

**Neural Autoregressive Density Estimator**: add hidden layer.  $\mathbf{h}_i = \sigma(\mathbf{b} + \mathbf{W}_{\cdot, <i} \mathbf{x}_{<i})$ ,  $\hat{x}_i = \sigma(c_i + \mathbf{V}_i \cdot \mathbf{h}_i)$ . Order of  $\mathbf{x}$  can be arbitrary but fixed. Train by max log-likelihood in  $O(TD)$ , can use 2nd order optimizers, can use **teacher forcing**: feed GT as previous output.

Extensions: Convolutional; Real-valued: conditionals by mixture of gaussians; Order-less and deep: one DNN predicts  $p(x_k | x_{i_1} \dots x_{i_j})$ .

**Masked Autoencoder Distribution Estimator**: mask out weights s.t. no information flows from  $x_d \dots$  to  $\hat{x}_d$ . Large hidden layers needed. Trains as fast as autoencoders, but sampling needs  $D$  forward passes.

**PixelRNN**: generate pixels from corner, dependency on previous pixels is by RNN (LSTM). **PixelCNN**: also from corner, but condition by CNN over context region (perceptive field)  $\Rightarrow$  parallelize. For conditionals use masked convolutions. Channels: model R from context, G from R + cont., B from G + R + cont. Training is parallel, but inference is sequential  $\Rightarrow$  slow. Use conv. stacks to mask correctly.

NLL is a natural metric for autoreg. models, hard to evaluate others.

**WaveNet**: audio is high-dimensional. Use dilated convolutions to increase perceptive field with multiple layers.

AR does not work for high res images/video, convert the images into a series of tokens with an AE: Vector-quantized VAE. The codebook is a set of vectors.  $x \xrightarrow{\text{enc}} z \xrightarrow{\text{codebook}} z_q \xrightarrow{\text{dec}} \hat{x}$ .

We can run an AR model in the latent space.

3.1 Attention

$\mathbf{x}_t$  is a convex combination of the past steps, with access to all past steps. For  $X \in \mathbb{R}^{T \times D}$ :  $K = XW_K, V = XW_V, Q = XW_Q$ . Check pairwise similarity between query and keys via dot product: let attention weights be  $\alpha = \text{Softmax}(QK^T / \sqrt{D})$ ,  $\alpha \in \mathbb{R}^{1 \times T}$ . Adding mask

$M$  to avoid looking into the future:

$$X = \text{Softmax} \left( \frac{(XW_Q)(XW_K)^T}{\sqrt{D}} + M \right) (XW_V)$$

Multi-head attn. splits  $W$  into  $h$  heads, then concatenates them. Positional encoding injects information about the position of the token. Attn. is  $O(T^2D)$ .