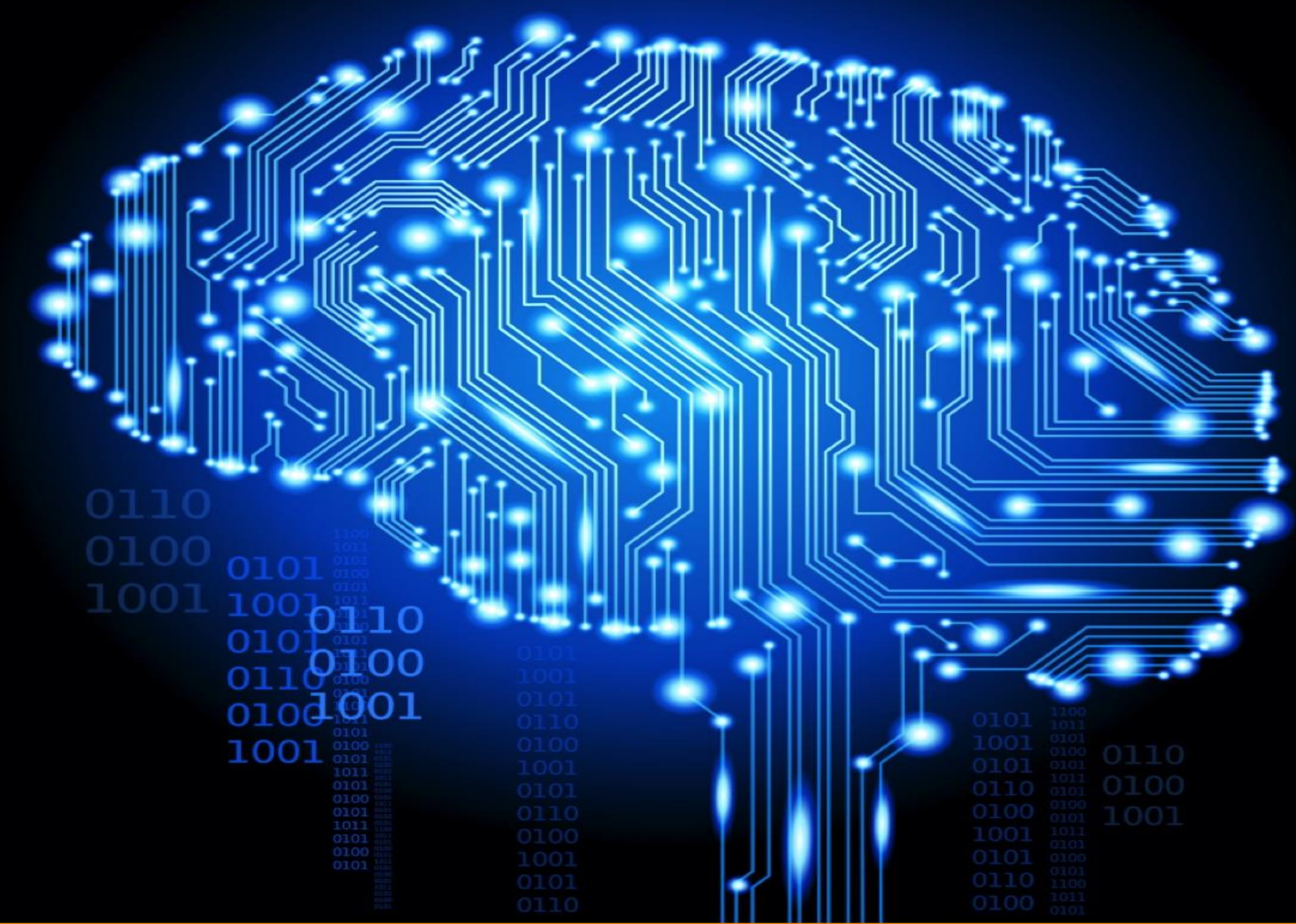


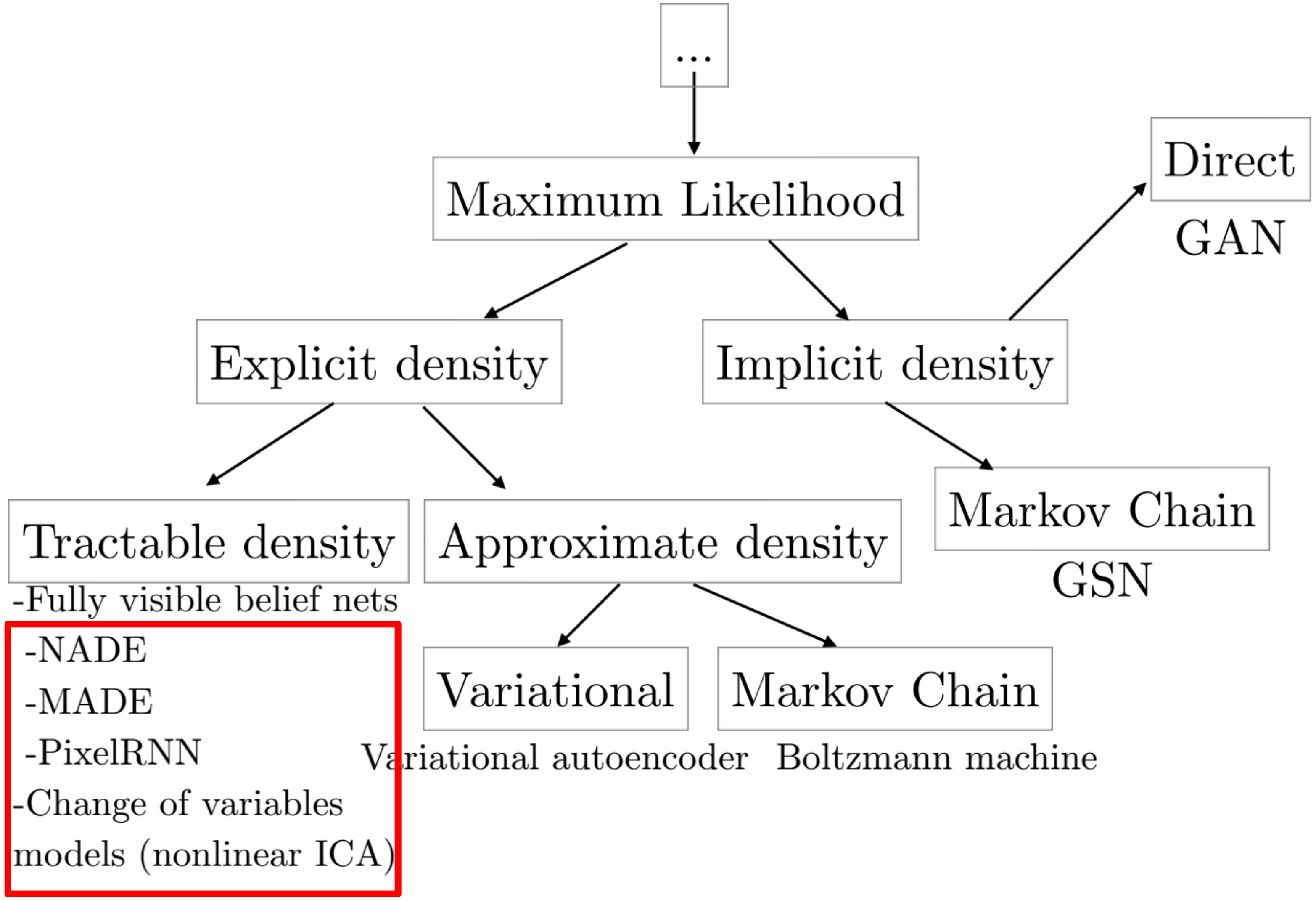
Lecture 11: Advanced Generative Models

Efstratios Gavves

Lecture overview

- Early autoregressive models
- Modern autoregressive models
- Normalizing flows
- Flow-based models

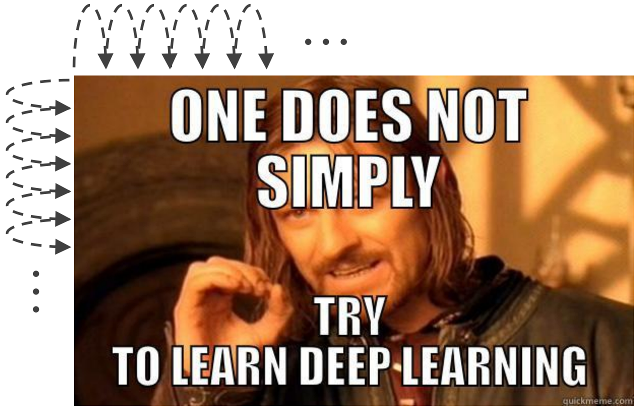
A map of generative models



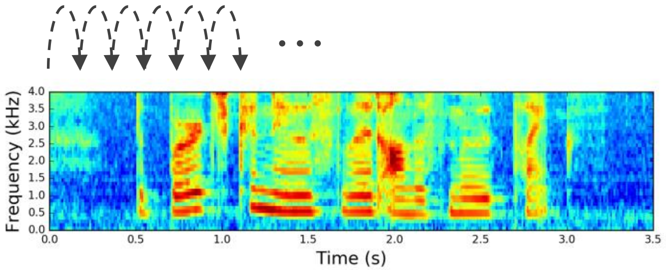
Beyond independent dimensions

- Often, in data there is either an order or we can make up an order
  - From a generation point of view, data dimensions depend on each other

<u>Images</u>



<u>Audio</u>



<u>Text</u>

...

'One does not simply try to learn deep learning'

Decomposing likelihood of sequential data

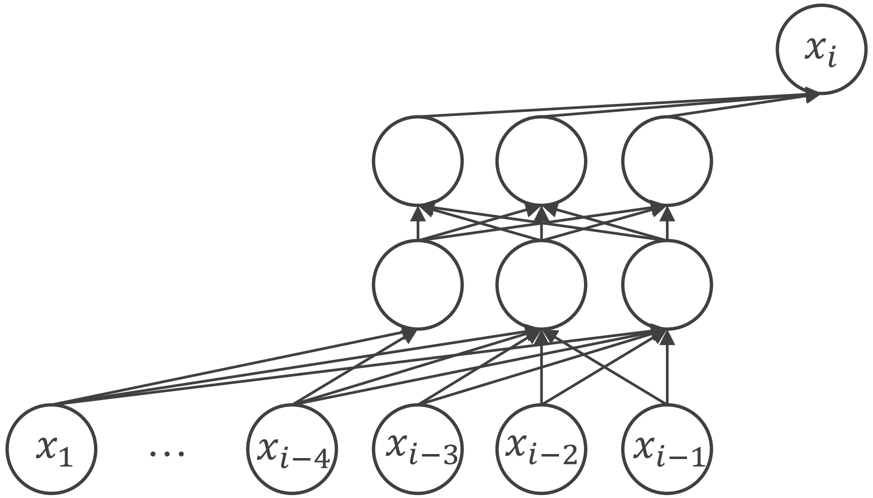
- If $\boldsymbol{x} = [x_1, x_2, \ldots, x_d]$ is sequential, $p(\boldsymbol{x})$ decomposes with chain rule of probabilities

$$p(\boldsymbol{x}) = p(x_1) \cdot p(x_2|x_1) \cdot p(x_3|x_1, x_2) \cdot \ldots \cdot p(x_d|x_1, \ldots, x_{d-1}) = \prod_{i=1}^{d} p(x_i|x_{<i})$$

- If $\boldsymbol{x}$ is *not* sequential, we can assume an artificial order
  - *e.g.*, the order with which pixels make (generate) an image
  - This can create artificial bias, however

Deep networks to model conditional likelihoods

- Model the conditional likelihoods with deep neural networks
  - Logistic regression (Frey et al., 1996), Neural nets (Bengio and Bengio, 2000)
  - *E.g.*, learn a deep net to generate one pixel at a time given past pixels
- The learning objective is to maximize the log-likelihood $\log p(\boldsymbol{x})$
  - If each conditional is tractable, $\log p(\boldsymbol{x})$ is tractable
  - Model conditional probabilities directly and witn no partition functions Z



Neural Autoregressive Density Estimation

- Inspired by RBMs but with tractable density estimation
  - Each conditional modelled with sigmoidal neural net like in RBMs

- Parameter matrix W maps past inputs $\boldsymbol{v}_{<i}$ to hidden feature $\boldsymbol{h}_i$

- Parameter matrix V generates pixel v_i given the hidden feature $\boldsymbol{h}_i$

$$p(v_i|\boldsymbol{v}_{<i}) = \sigma\big(b_i + (V^T)_{i,\cdot}\boldsymbol{h}_i\big)$$

$$\boldsymbol{h}_i = \sigma\big(\boldsymbol{c} + W_{\cdot,<i}\boldsymbol{v}_{<i}\big)$$

- Teacher forcing
  - During training use ground truth past inputs $\boldsymbol{v}_{<i}$
  - During testing use predicted past inputs $\hat{\boldsymbol{v}}_{<i}$

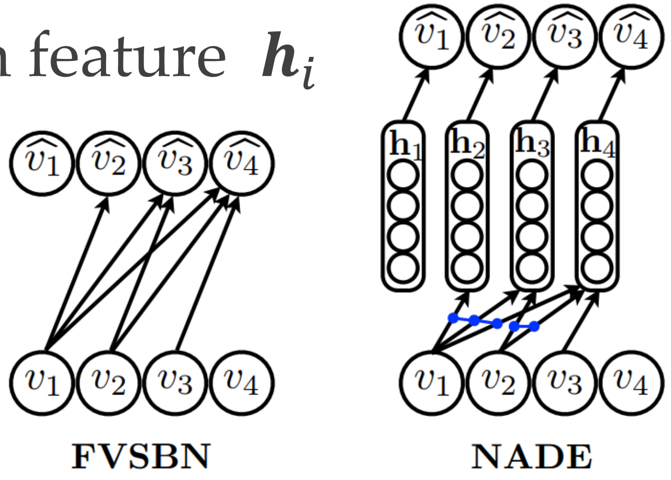


Figure 1: **(Left)** Illustration of a fully visible sigmoid belief network. **(Right)** Illustration of a neural autoregressive distribution estimator. $\hat{v}_i$ is used as a shorthand for $p(v_i = 1|\mathbf{v}_{<i})$. Arrows connected by a blue line correspond to connections with shared or tied parameters.

Larochelle and Murray, Neural Autoregressive Distribution Estimation

Masked Autoencoder for Distribution Estimation

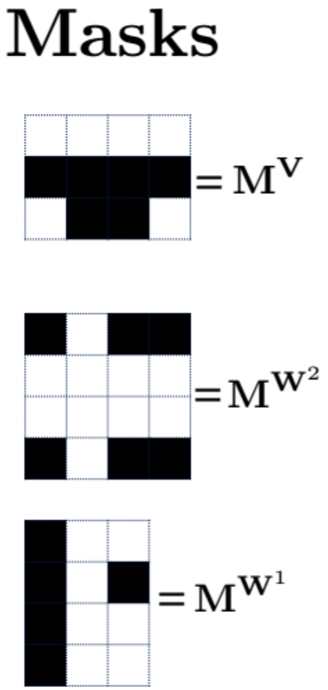
- Make an autoregressive autoencoder by setting each output x_i depend only on previous outputs $\boldsymbol{x}_{<i}$
  - In autoencoders the output dimensions depend on 'future' dimensions also

- Implement this by introducing a masking matrix M to multiply weights

$$h(\boldsymbol{x}) = g(\boldsymbol{b} + (W \odot M^W) \cdot \boldsymbol{x})$$
$$\hat{x} = \sigma(\boldsymbol{c} + (V \odot M^V) \cdot h(\boldsymbol{x}))$$

For the k-th neuron the mask column is $M_{k,d} = \begin{cases} 1 & m(k) \geq d \\ 0 & \text{otherwise} \end{cases}$

And $m(k)$ is a integer between 1 and $d - 1$



Germain, Gregor, Murray, Larochelle, Masked Autoencoder for Distribution Estimation

MADE architecture