Given an autoregressive model $P_\theta(x) = \prod_{i=1}^{n} P_{\text{neural}}(x_i \mid pa(x_i); \theta_i)$ $\quad pa(x_i) = x_1 \cdots x_{i-1}$ in an autoregressive model

dataset $D = \{x^{(1)} \cdots x^{(m)}\}$

$L(\theta) = \prod_{i=1}^{m} \prod_{j=1}^{n} P_{\text{neural}}(x_j^{(i)} \mid pa(x_j)^{(i)}; \theta_j)$

$\ell(\theta) = \sum_{j=1}^{n} \sum_{i=1}^{m} \log P_{\text{neural}}(x_j^{(i)} \mid pa(x_j)^{(i)}; \theta_j)$

$\Downarrow$

$\ell_j(\theta_j) = \sum_{i=1}^{m} \log P_{\text{neural}}(x_j^{(i)} \mid pa(x_j)^{(i)}; \theta_j)$

$\max_{\theta_j} \sum_{i=1}^{m} \log P_{\text{neural}}(x_j^{(i)} \mid pa(x_j)^{(i)}; \theta_j)$ decomposition of Bayesian Network

no close-form solution train each model $P(x_i \mid pa(x_i); \theta_i)$ separately

$$\nabla_{\theta_j} \ell_j(\theta_j) = \sum_{i=1}^{m} \nabla_{\theta_j} \log P_{\text{neural}}(x_j^{(i)} \mid pa(x_j)^{(i)}; \theta_j)$$
$$= m \cdot \frac{1}{m} \sum_{i=1}^{m} \nabla_{\theta_j} \log P_{\text{neural}}(x_j^{(i)} \mid pa(x_j)^{(i)}; \theta_j)$$
$$= m \cdot E_{x \sim \hat{p}_{\text{data}}} \left[\nabla_{\theta_j} \log P_{\text{neural}}(x_j \mid pa(x_j); \theta_j) \right]$$

} monte-carlo view of stochastic gradient descent

· Latent Variable Models

eg. human face images

the variability in images due to some high-level feature (eye color, facial expression, etc)

these factors are not explicitly given (latent)

latent variables correspond to high level features

if z chose properly $p(x|z)$ is much simpler than $p(x)$

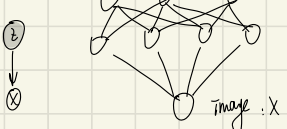
having trained those model, can identify features via $p(z|x)$

} ∵ hard to specify latent vars

· Deep Latent Variable Models

$z \sim \mathcal{N}(0, I)$

$x|z \sim \mathcal{N}(\mu_\theta(z), \Sigma_\theta(z))$ where $(\mu_\theta, \Sigma_\theta)$ are neural networks



z will hopefully corresponds to meaningful latent variations (features) unsupervised representation learning

features can be computed via $p(z|x)$

eg. Mixture of Gaussian is a shallow Latent Variable Model

- Variational Auto Encoder

$z \sim N(0, I)$

$x|z \sim N(\mu_\theta(z), \Sigma_\theta(z))$ where $\mu_\theta(\cdot)$, $\Sigma_\theta(\cdot)$ are neural networks } mixture of infinite Gaussians

$\mu_\theta(z) = \sigma(Az + c)$

$\Sigma_\theta(z) = \text{diag}(\exp(Bz + d))$

training is hard

$\max_\theta p(x) = \max_\theta \int_z p(z)\, p(x|z;\theta)\, dz$ infeasible