Deep Learning Essentials
Deep Generative Models

Russ Salakhutdinov

Machine Learning Department
Carnegie Mellon University

Statistical Generative Models

+

$\boxed{\text{Model family, loss function, optimization algorithm, etc.}}$

Data

Learning

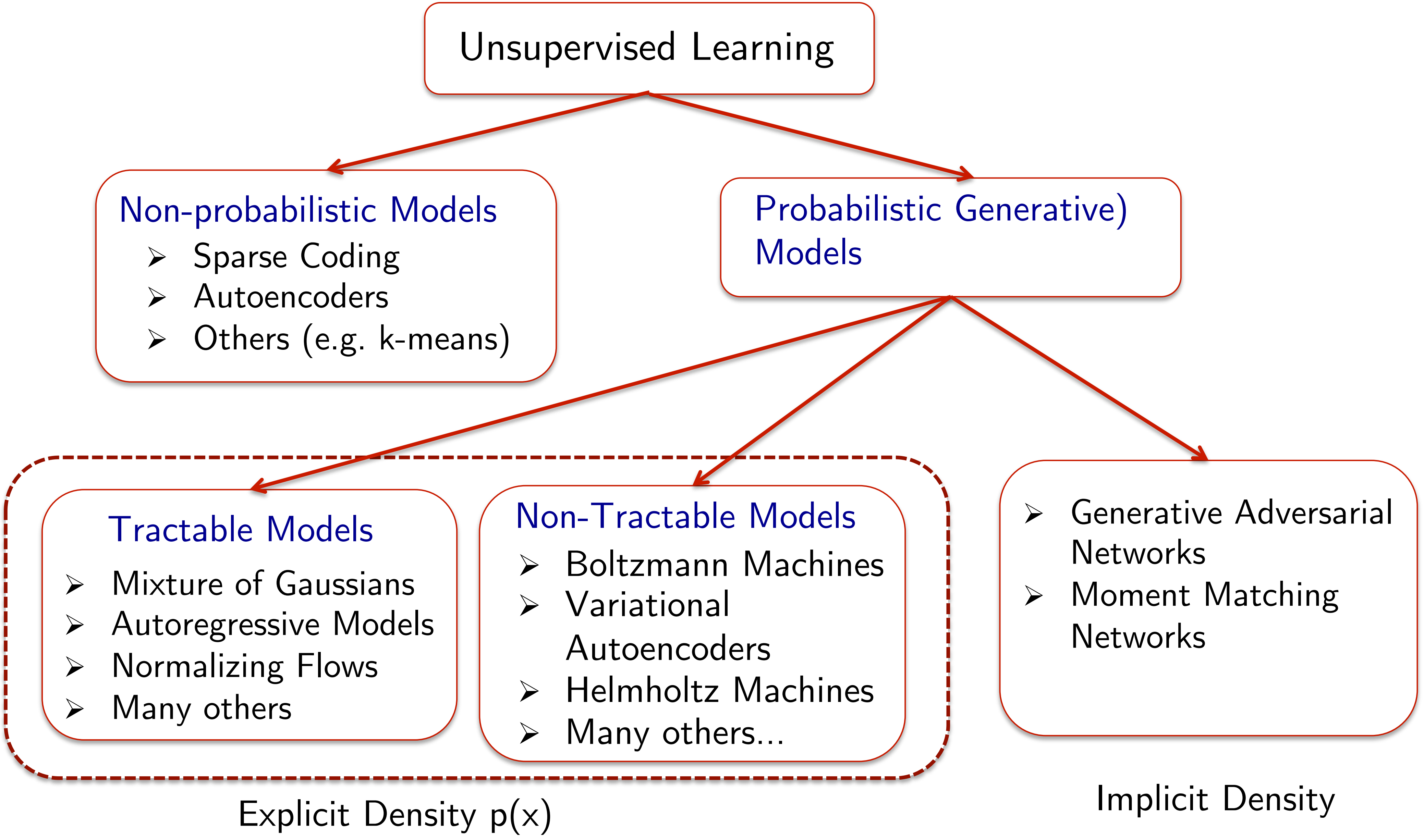
Prior Knowledge

Image x

$\longrightarrow$ A probability distribution p(x) $\longrightarrow$ probability p(x)

Sampling from p(x) **generates** new images:

Grover and Ermon, DGM Tutorial



Unsupervised Learning

Non-probabilistic Models
- Sparse Coding
- Autoencoders
- Others (e.g. k-means)

Probabilistic Generative) Models

Tractable Models
- Mixture of Gaussians
- Autoregressive Models
- Normalizing Flows
- Many others

Non-Tractable Models
- Boltzmann Machines
- Variational Autoencoders
- Helmholtz Machines
- Many others...

- Generative Adversarial Networks
- Moment Matching Networks

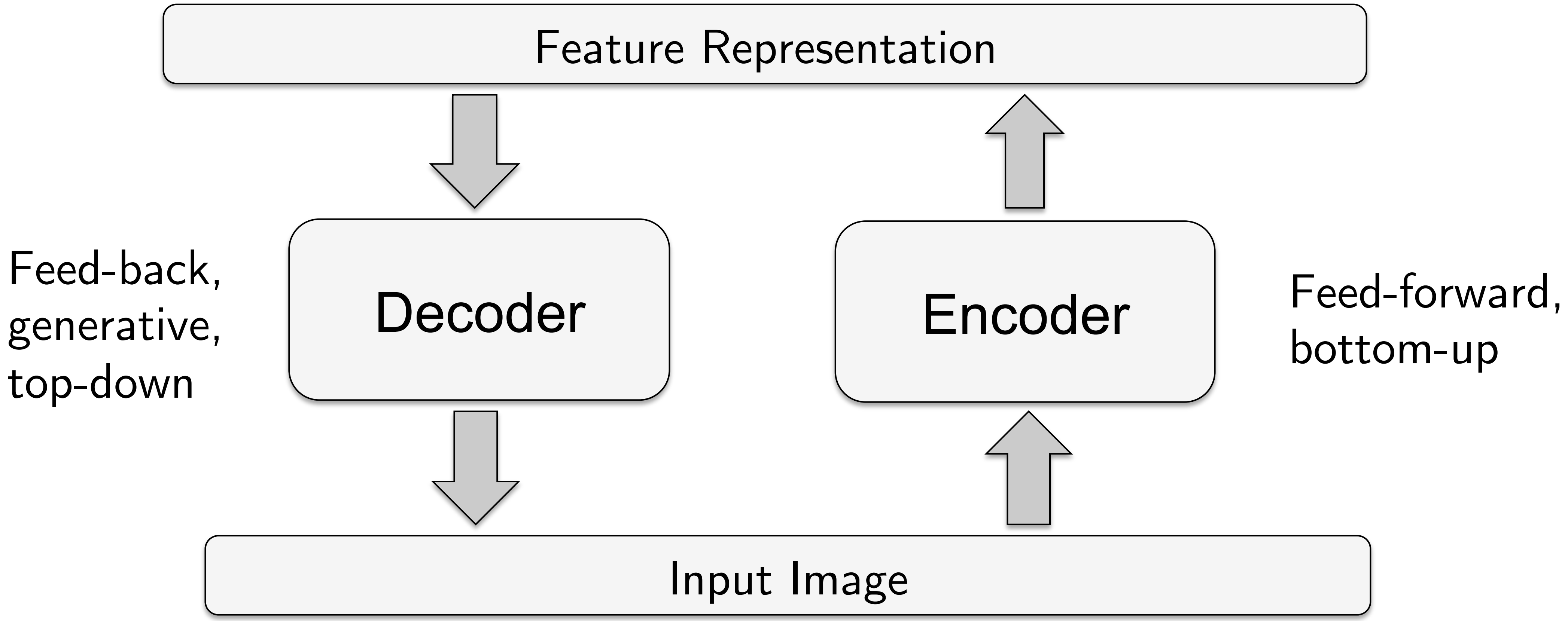
Explicit Density p(x)

Implicit Density

Talk Roadmap

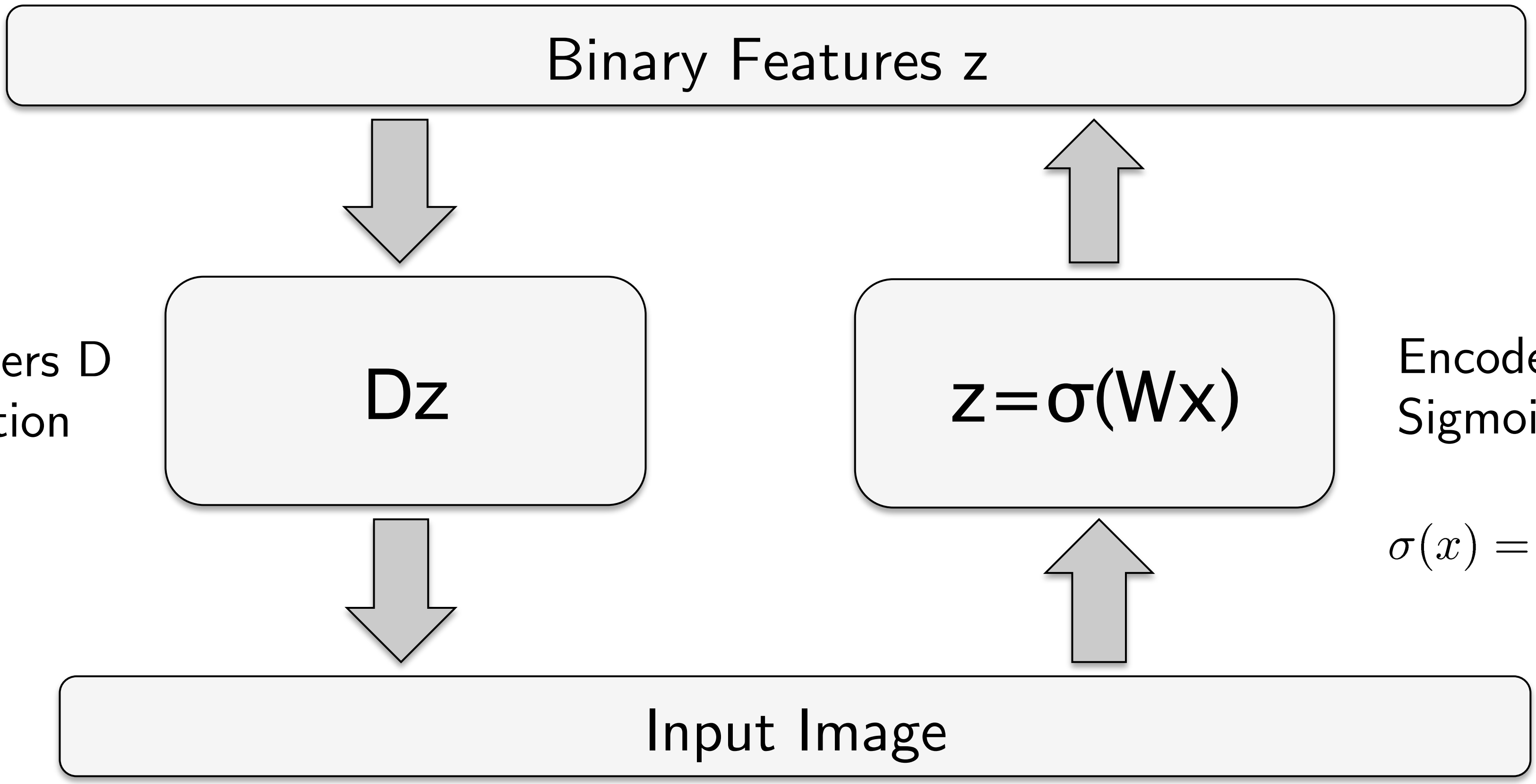
- Fully Observed Models
- Undirected Deep Generative Models
  - Restricted Boltzmann Machines (RBMs),
  - Deep Boltzmann Machines (DBMs)
- Directed Deep Generative Models
  - Variational Autoencoders (VAEs)
  - Normalizing Flows
- Generative Adversarial Networks (GANs)

Autoencoder



- Details of what goes insider the encoder and decoder matter
- Need constraints to avoid learning an identity.

Autoencoder

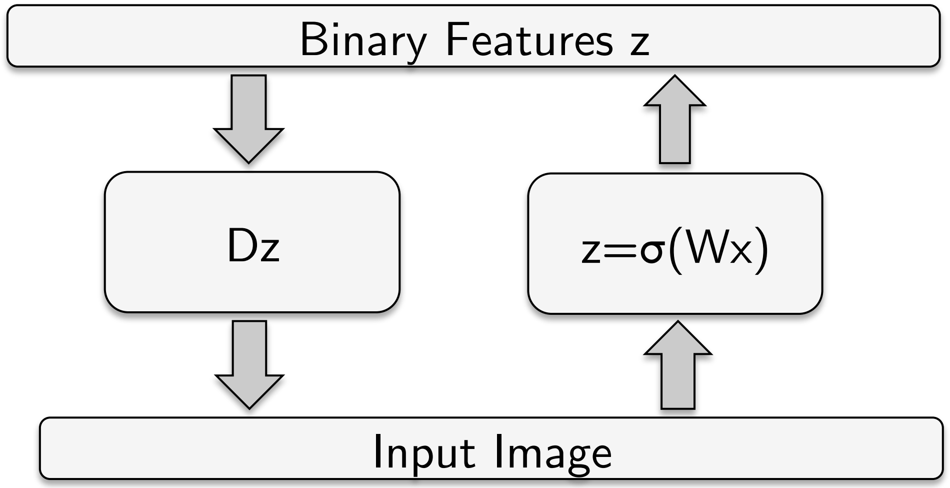


Decoder filters D
Linear function

Encoder filters W.
Sigmoid function

$$\sigma(x) = \frac{1}{1 + \exp(-x)}$$

Autoencoder

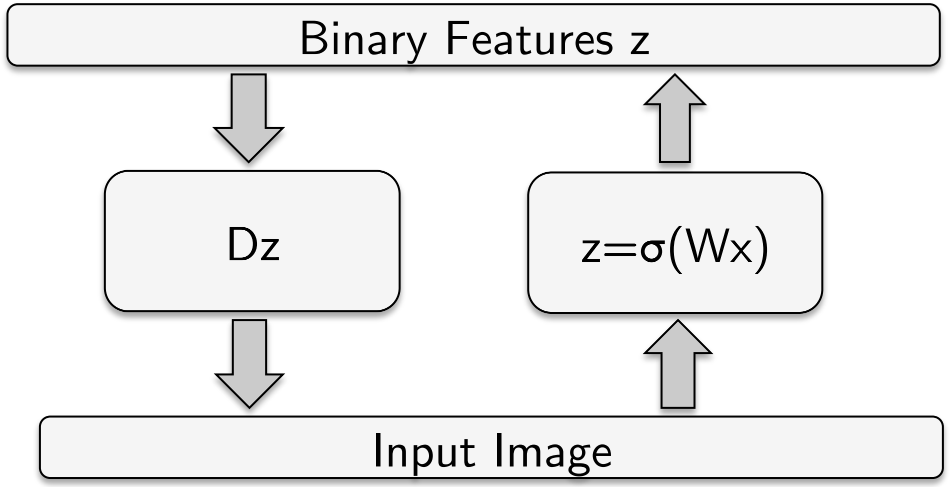


- An autoencoder with D inputs, D outputs, and K hidden units, with K<D
- Given an input x, its reconstruction is given by:

$$y_j(\mathbf{x}, W, D) = \underbrace{\sum_{k=1}^{K} D_{jk}\sigma}_{\text{Decoder}} \underbrace{\left(\sum_{i=1}^{D} W_{ki} x_i\right)}_{\text{Encoder}}, \quad j = 1, .., D.$$

$$y_j = \sum_{k=1}^{K} D_{jk} z_k \qquad z_k = \sigma\left(\sum_{i=1}^{D} W_{ki} x_i\right)$$

Autoencoder

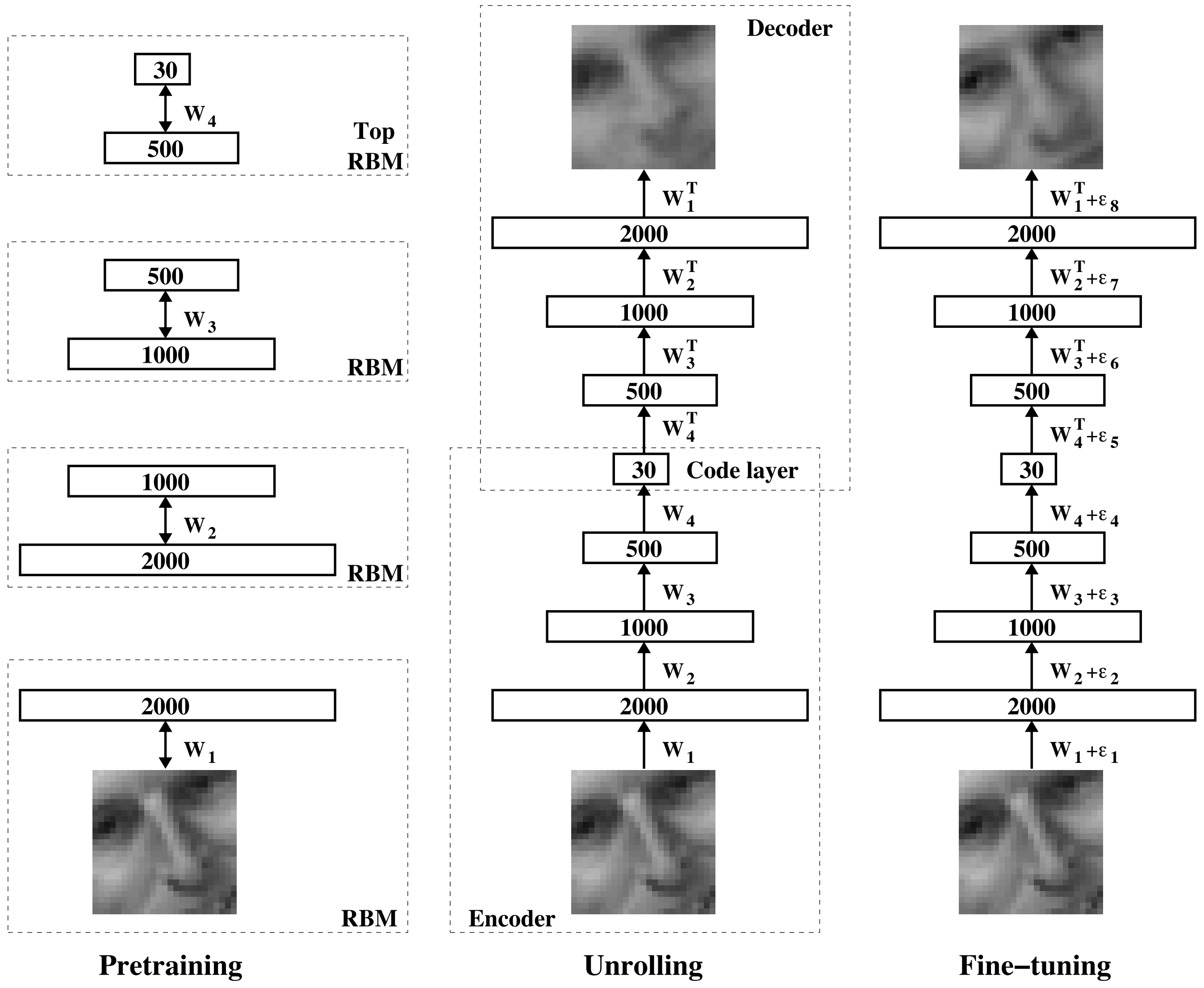


- An autoencoder with D inputs, D outputs, and K hidden units, with K<D.
- If the K hidden and output layers are linear, the hidden units will span the same space as the first k principal components

- We can determine the network parameters W and D by minimizing the reconstruction error:

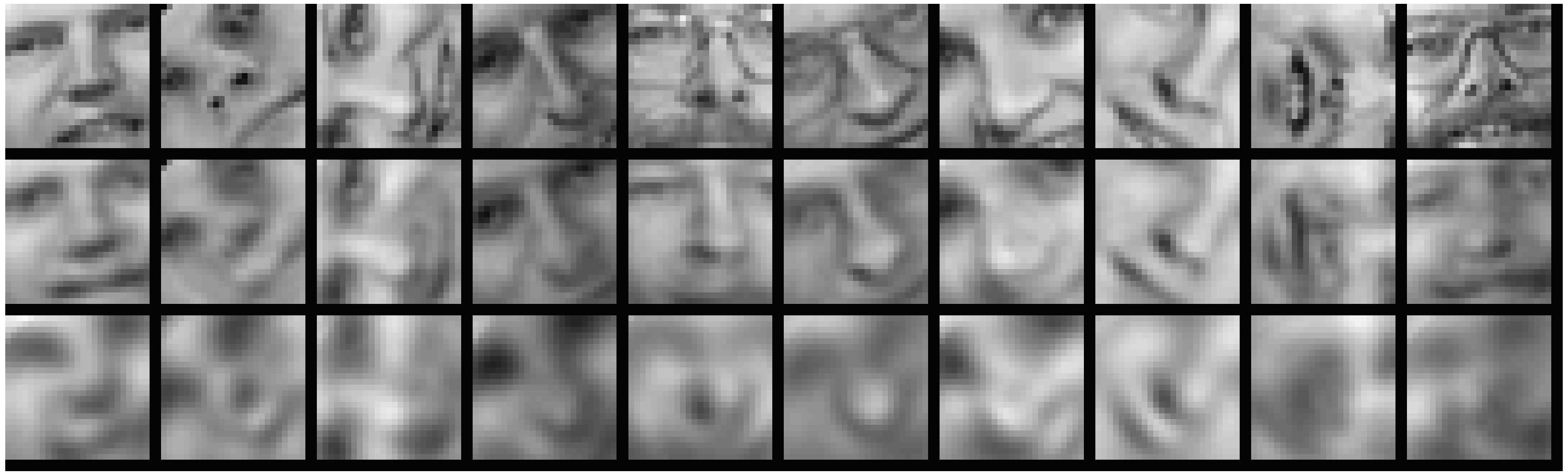
$$E(W, D) = \frac{1}{2}\sum_{n=1}^{N} ||y(\mathbf{x}_n, W, D) - \mathbf{x}_n||^2.$$

Deep Autoencoder



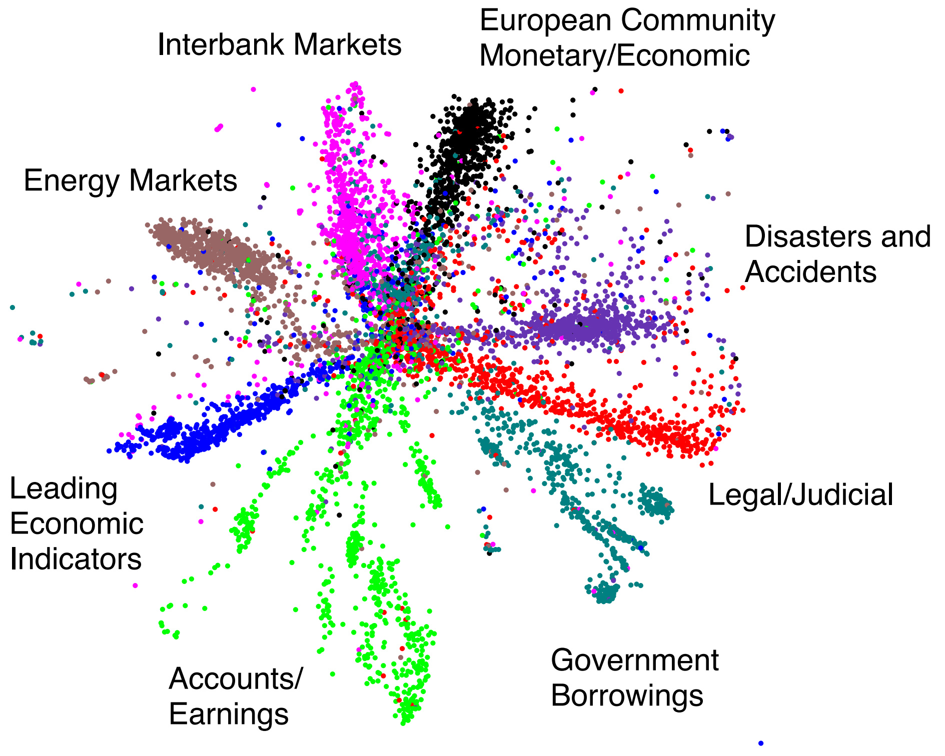
Deep Autoencoder

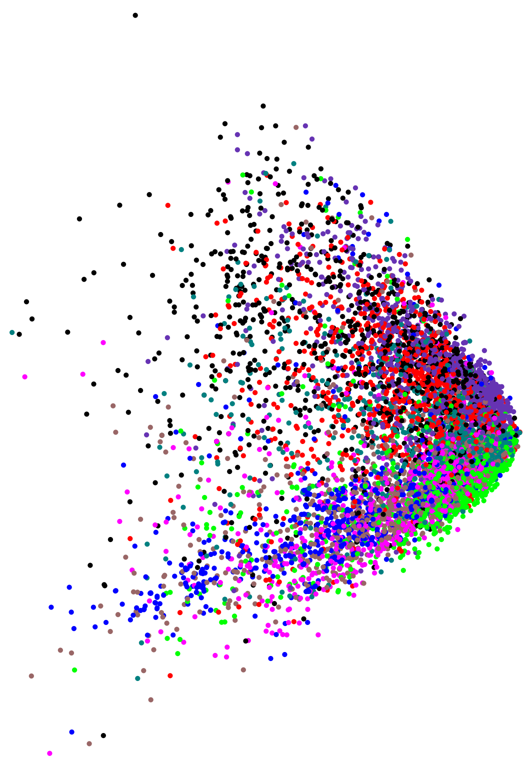
- 25x25 – 2000 – 1000 – 500 – 30 autoencoder to extract 30-D real-valued codes for Olivetti face patches.



- Top: Random samples from the test dataset
- Middle: Reconstructions by the 30-dimensional deep autoencoder
- Bottom: Reconstructions by the 30-dimentinoal PCA.

Deep Autoencoder: Information Retrieval





- The Reuters Corpus Volume II contains 804,414 newswire stories (randomly split into 402,207 training and 402,207 test)
- Bag-of-words" representation: each article is represented as a vector containing the counts of the most frequently used 2000 word

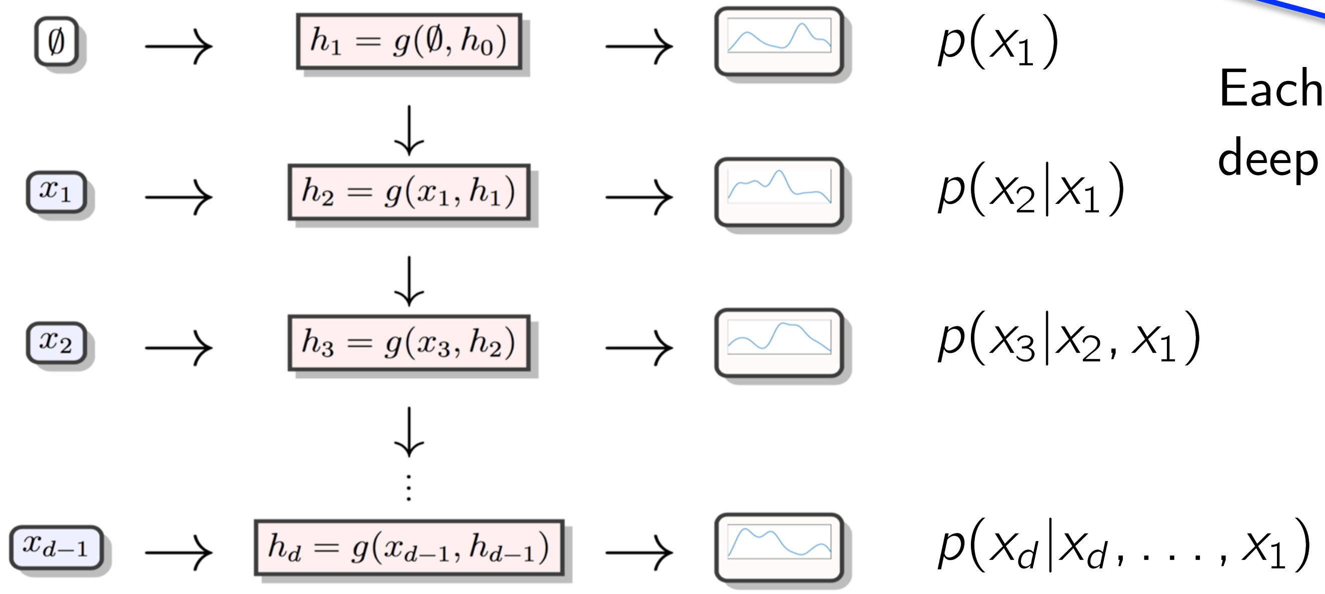
Talk Roadmap

- Fully Observed Models
- Undirected Deep Generative Models
  - Restricted Boltzmann Machines (RBMs),
  - Deep Boltzmann Machines (DBMs)
- Directed Deep Generative Models
  - Variational Autoencoders (VAEs)
  - Normalizing Flows
- Generative Adversarial Networks (GANs)

Fully Observed Models

- Density Estimation by Autoregression

$$p(x_1, \dots, x_d) = \prod_{i=1}^{d} p(x_i | x_{i-1}, \dots, x_1) \approx \prod_{i=1}^{d} p(x_i | g(x_{i-1}, \dots, x_1))$$



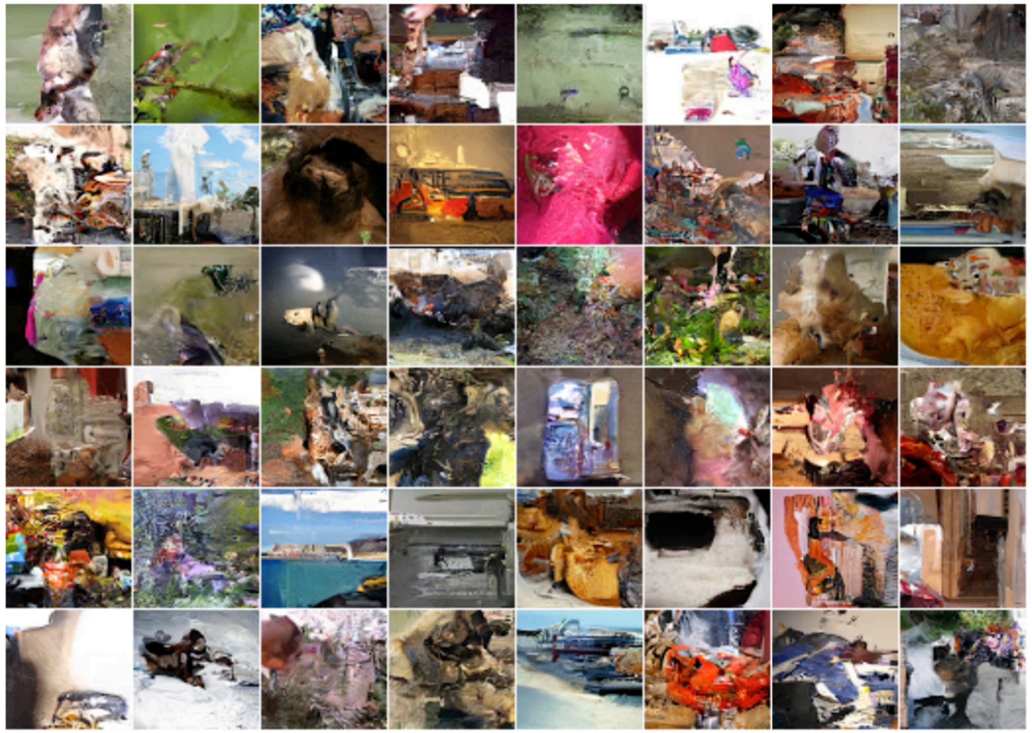
Each conditional can be a deep neural network

- Ordering of variables is crucial

NADE (Uria 2013), MADE (Germain 2017), MAF (Papamakarios 2017), PixelCNN (van den Oord, et al, 2016)

Fully Observed Models

- Density Estimation by Autoregression

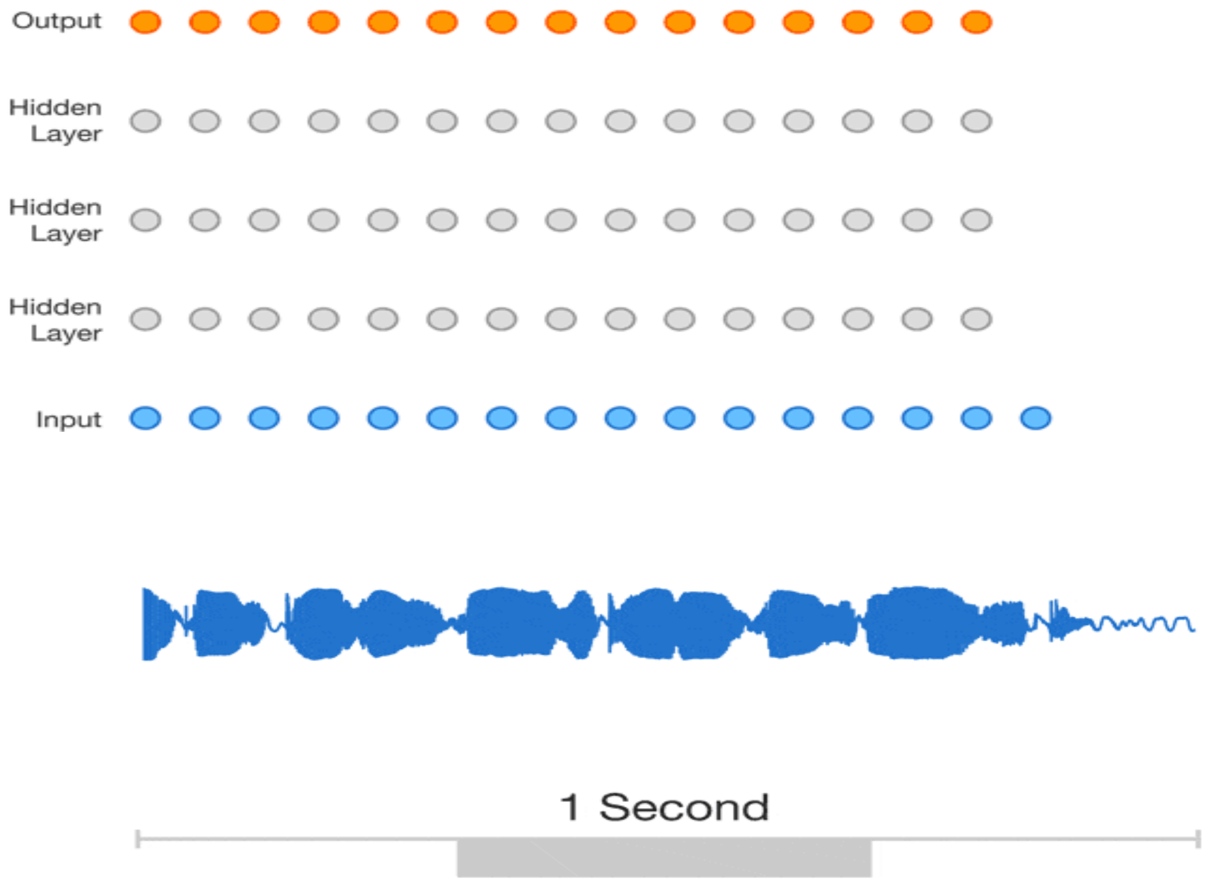


PixelCNN (van den Oord, et al, 2016)

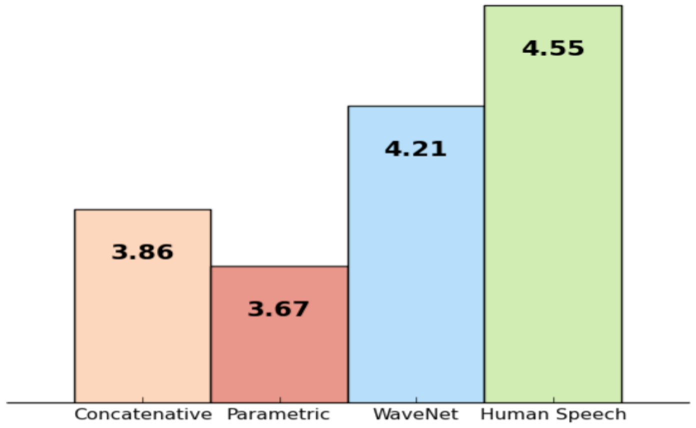
NADE (Uria 2013), MADE (Germain 2017), MAF (Papamakarios 2017), PixelCNN (van den Oord, et al, 2016)

WaveNet

- Generative Model of Speech Signals



Quality: Mean Opinion Scores

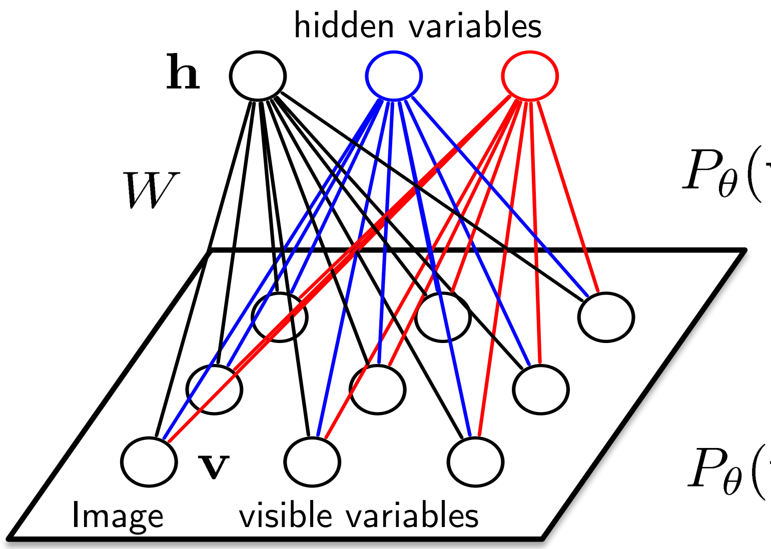


van den Oord et al, 2016

Talk Roadmap

- Fully Observed Models
- Undirected Deep Generative Models
  - Restricted Boltzmann Machines (RBMs),
  - Deep Boltzmann Machines (DBMs)
- Directed Deep Generative Models
  - Variational Autoencoders (VAEs)
  - Normalizing Flows
- Generative Adversarial Networks (GANs)

Restricted Boltzmann Machines



$$P_\theta(\mathbf{v},\mathbf{h}) = \frac{1}{\mathcal{Z}(\theta)} \exp\left(\sum_{i=1}^{D}\sum_{j=1}^{F} \overbrace{W_{ij}v_ih_j}^{\text{Pairwise}} + \sum_{i=1}^{D} \overbrace{v_ib_i}^{\text{Unary}} + \sum_{j=1}^{F} \overbrace{h_ja_j}^{\text{Unary}} \right)$$

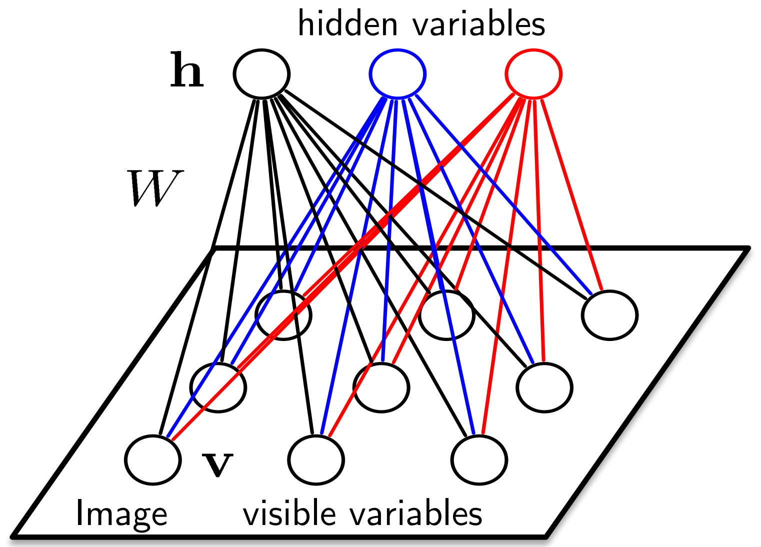
$$\theta = \{W, a, b\}$$

$$P_\theta(\mathbf{v}|\mathbf{h}) = \prod_{i=1}^{D} P_\theta(v_i|\mathbf{h}) = \prod_{i=1}^{D} \frac{1}{1+\exp(-\sum_{j=1}^{F} W_{ij}v_ih_j - b_i)}$$

- RBM is a Markov Random Field with
  - Stochastic binary visible variables $\mathbf{v} \in \{0,1\}^D$.
  - Stochastic binary hidden variables $\mathbf{h} \in \{0,1\}^F$.
  - Bipartite connections

Markov random fields, Boltzmann machines, log-linear models.

Maximum Likelihood Learning



$$P_\theta(\mathbf{v}) = \frac{P^*(\mathbf{v})}{\mathcal{Z}(\theta)} = \frac{1}{\mathcal{Z}(\theta)} \sum_{\mathbf{h}} \exp\left[\mathbf{v}^\top W\mathbf{h} + \mathbf{a}^\top\mathbf{h} + \mathbf{b}^\top\mathbf{v}\right]$$

- Maximize log-likelihood objective:

$$L(\theta) = \frac{1}{N}\sum_{n=1}^{N} \log P_\theta(\mathbf{v}^{(n)})$$

- Derivative of the log-likelihood:

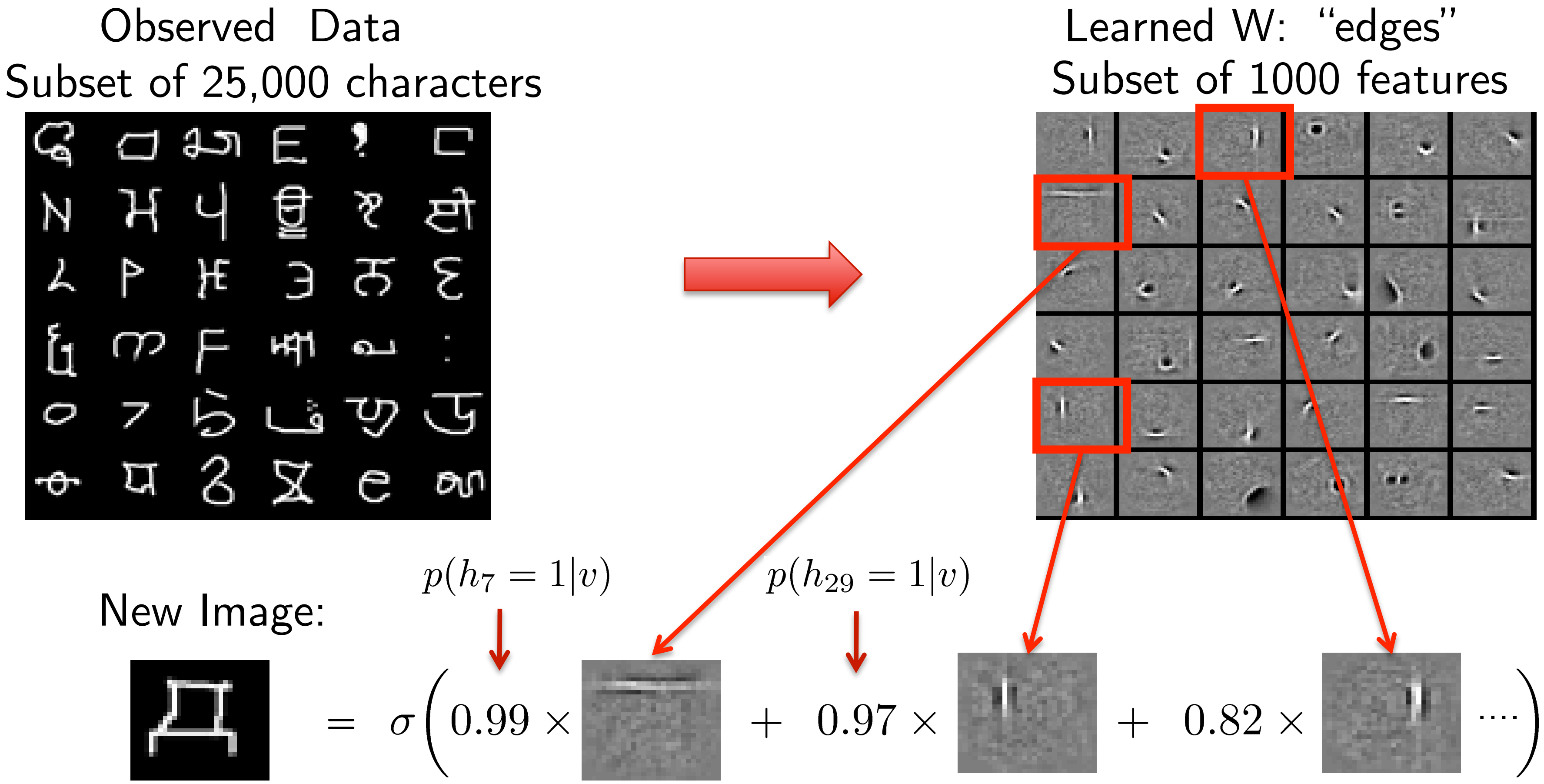
$$\frac{\partial L(\theta)}{\partial W_{ij}} = \frac{1}{N}\sum_{n=1}^{N} \frac{\partial}{\partial W_{ij}} \log\left(\sum_{\mathbf{h}} \exp\left[\mathbf{v}^{(n)\top} W\mathbf{h} + \mathbf{a}^\top\mathbf{h} + \mathbf{b}^\top\mathbf{v}^{(n)}\right]\right) - \frac{\partial}{\partial W_{ij}} \log\mathcal{Z}(\theta)$$

$$= \mathrm{E}_{P_{data}}[v_i h_j] - \underbrace{\mathrm{E}_{P_\theta}[v_i h_j]}_{\text{Difficult to compute: exponentially many configurations}}$$

$$P_{data}(\mathbf{v},\mathbf{h};\theta) = P(\mathbf{h}|\mathbf{v};\theta)P_{data}(\mathbf{v})$$

$$P_{data}(\mathbf{v}) = \frac{1}{N}\sum_n \delta(\mathbf{v} - \mathbf{v}^{(n)})$$

Learning Features



$$= \sigma\left(0.99 \times \square + 0.97 \times \square + 0.82 \times \square \cdots\right)$$

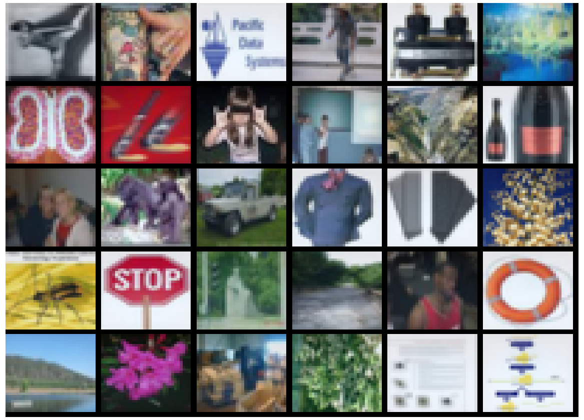
$p(h_7 = 1|v)$ $p(h_{29} = 1|v)$

$$\sigma(x) = \frac{1}{1+\exp(-x)}$$

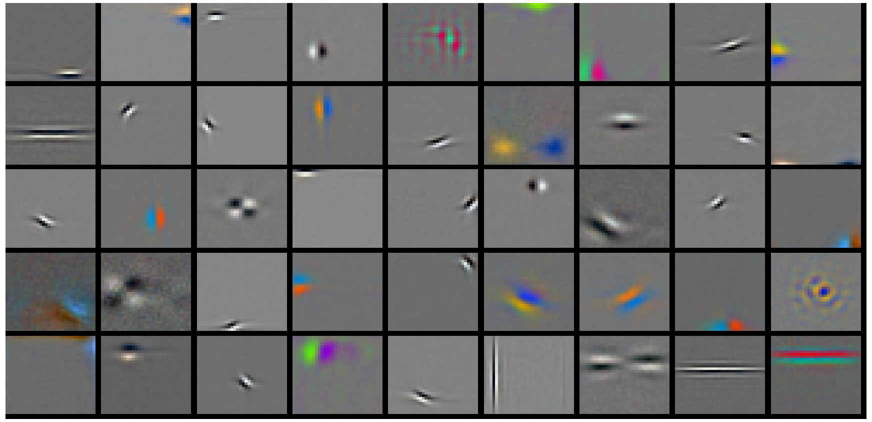
Logistic Function: Suitable for modeling binary images

RBMs for Real-valued & Count Data

4 million **unlabelled** images



Learned features (out of 10,000)



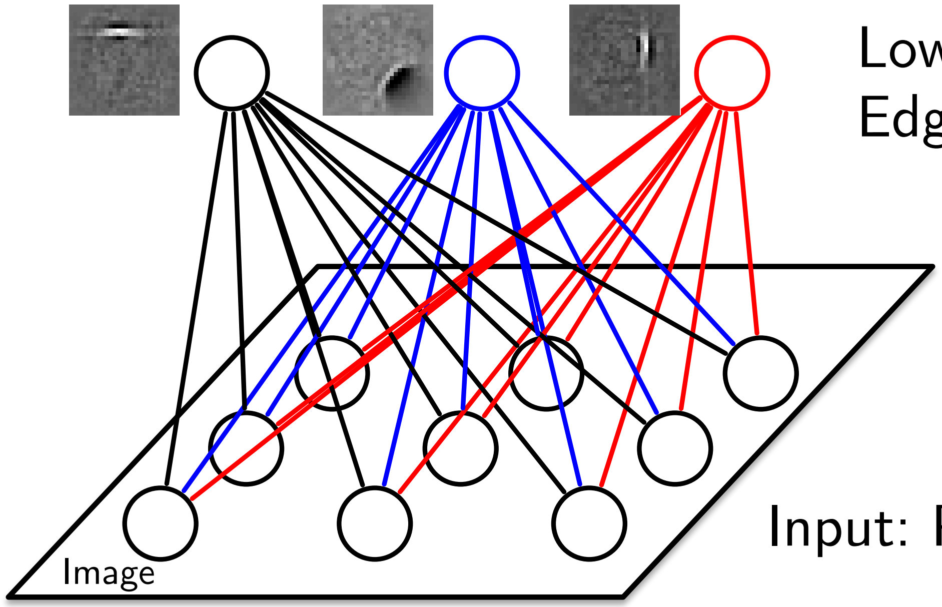
Reuters dataset:
804,414 **unlabeled**
newswire stories

Bag-of-Words

Learned features: ``topics''

russian	clinton	computer	trade	stock
russia	house	system	country	wall
moscow	president	product	import	street
yeltsin	bill	software	world	point
soviet	congress	develop	economy	dow

Deep Boltzmann Machines



Low-level features:
Edges

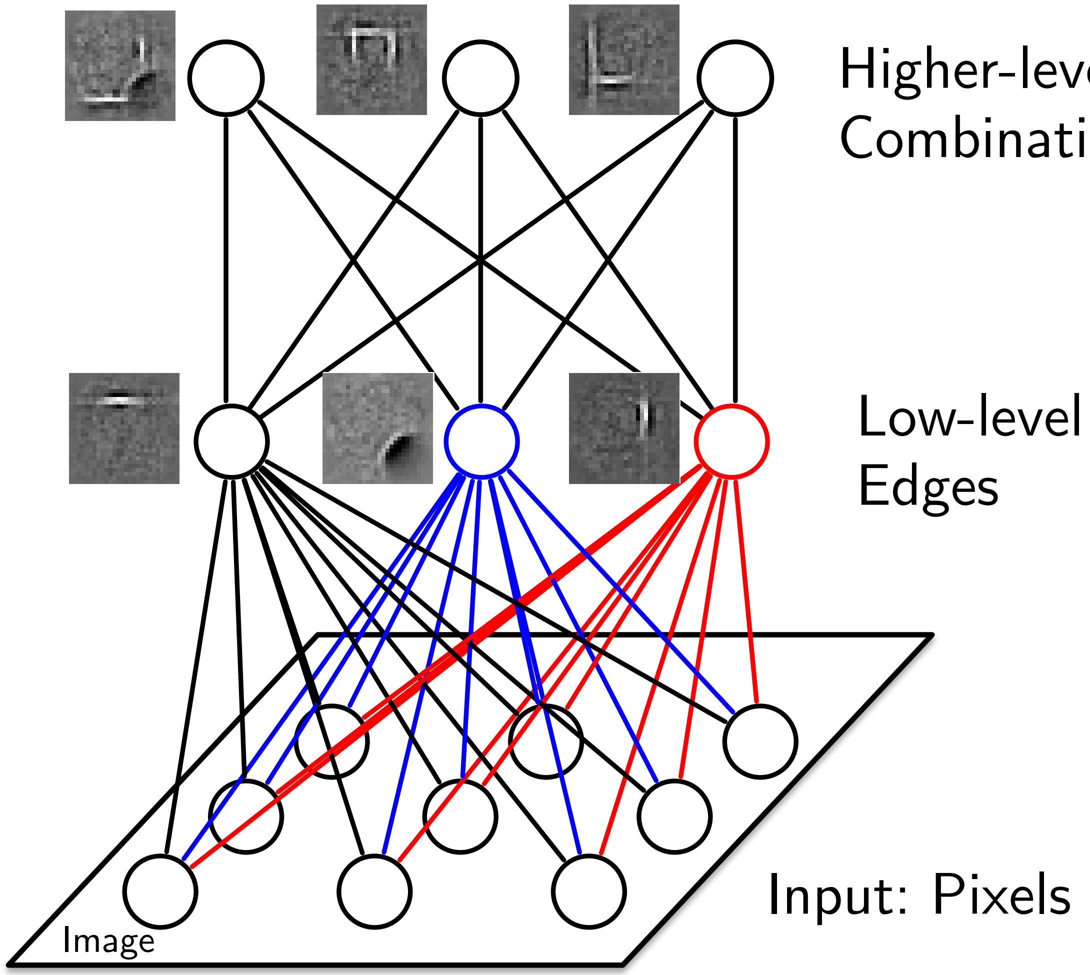
Input: Pixels

Built from **unlabeled** inputs.

(Salakhutdinov 2008, Salakhutdinov & Hinton 2009)

Deep Boltzmann Machines

Learn simpler representations, then compose more complex ones

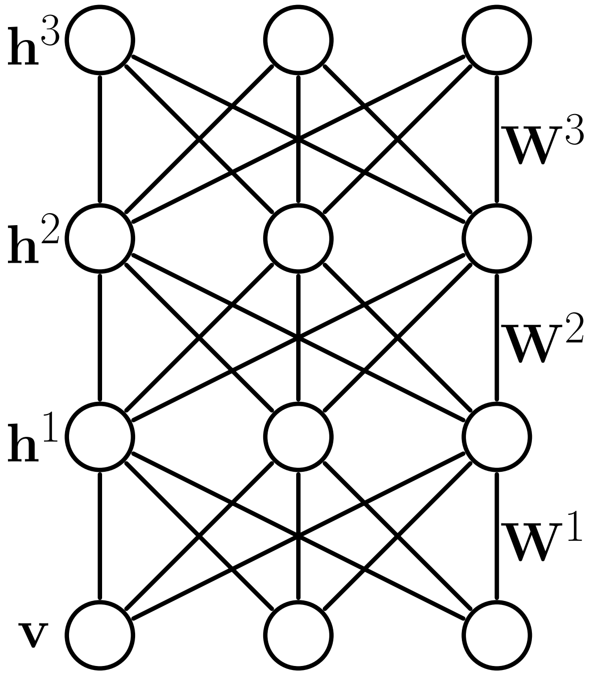


Built from **unlabeled** inputs.

(Salakhutdinov 2008, Salakhutdinov & Hinton 2009)

Model Formation

$$P_\theta(\mathbf{v}, \mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}) = \frac{1}{\mathcal{Z}(\theta)} \exp\left[\underbrace{\mathbf{v}^\top W^{(1)}\mathbf{h}^{(1)}}_{\text{Same as RBMs}} + \underline{\mathbf{h}^{(1)}{}^\top W^{(2)}\mathbf{h}^{(2)}} + \underline{\mathbf{h}^{(2)}{}^\top W^{(3)}\mathbf{h}^{(3)}}\right]$$



$\theta = \{W^1, W^2, W^3\}$ model parameters

- Dependencies between hidden variables
- All connections are undirected

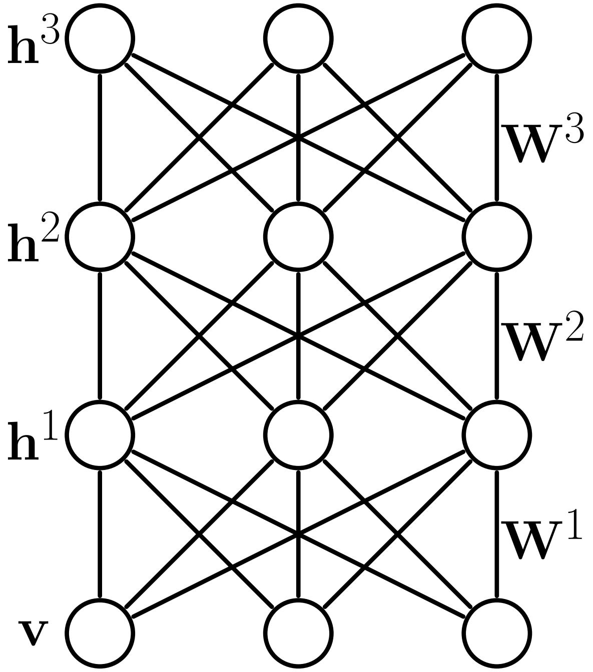
- Maximum Likelihood Learning:

$$\frac{\partial \log P_\theta(\mathbf{v})}{\partial W^1} = \mathbb{E}_{P_{data}}[\mathbf{v}\mathbf{h^1}^\top] - \mathbb{E}_{P_\theta}[\mathbf{v}\mathbf{h^1}^\top]$$

  - Both expectations are intractable

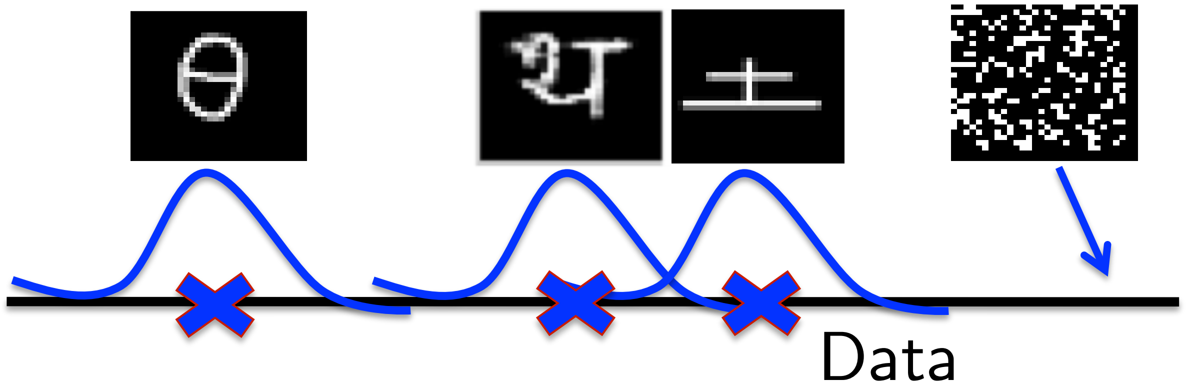
Approximate Learning

$$P_\theta(\mathbf{v}, \mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}) = \frac{1}{\mathcal{Z}(\theta)} \exp\left[\mathbf{v}^\top W^{(1)}\mathbf{h}^{(1)} + \mathbf{h}^{(1)\top} W^{(2)}\mathbf{h}^{(2)} + \mathbf{h}^{(2)\top} W^{(3)}\mathbf{h}^{(3)}\right]$$



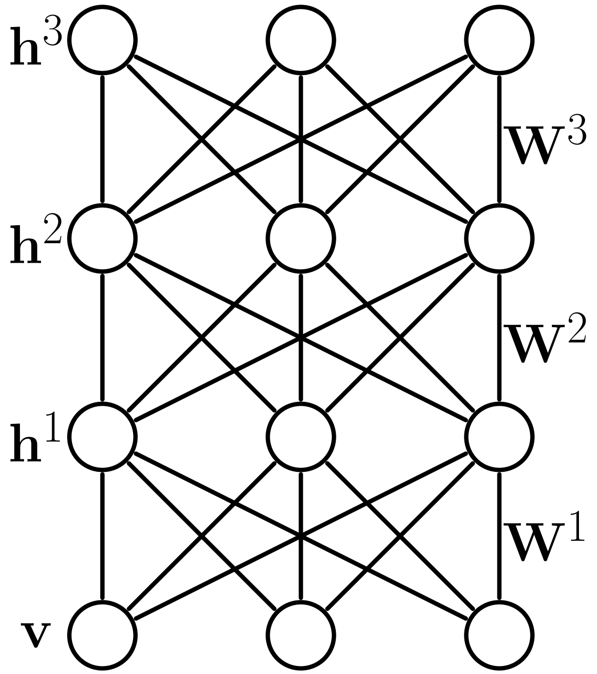
- Maximum Likelihood Learning:

$$\frac{\partial \log P_\theta(\mathbf{v})}{\partial W^1} = \mathbb{E}_{P_{data}}[\mathbf{v}\mathbf{h}^{\mathbf{1}\top}] - \mathbb{E}_{P_\theta}[\mathbf{v}\mathbf{h}^{\mathbf{1}\top}]$$



Approximate Learning

$$P_\theta(\mathbf{v}, \mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}) = \frac{1}{\mathcal{Z}(\theta)} \exp\left[\mathbf{v}^\top W^{(1)}\mathbf{h}^{(1)} + \mathbf{h}^{(1)\top} W^{(2)}\mathbf{h}^{(2)} + \mathbf{h}^{(2)\top} W^{(3)}\mathbf{h}^{(3)}\right]$$



- Maximum Likelihood Learning:

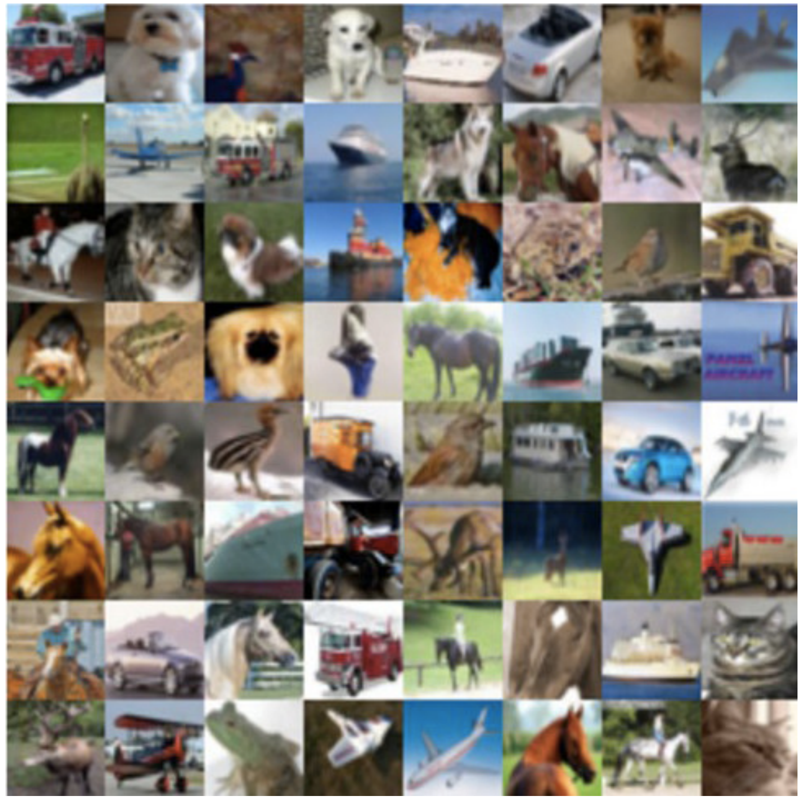
$$\frac{\partial \log P_\theta(\mathbf{v})}{\partial W^1} = \mathbb{E}_{P_{data}}[\mathbf{v}\mathbf{h}^{\mathbf{1}\top}] - \mathbb{E}_{P_\theta}[\mathbf{v}\mathbf{h}^{\mathbf{1}\top}]$$

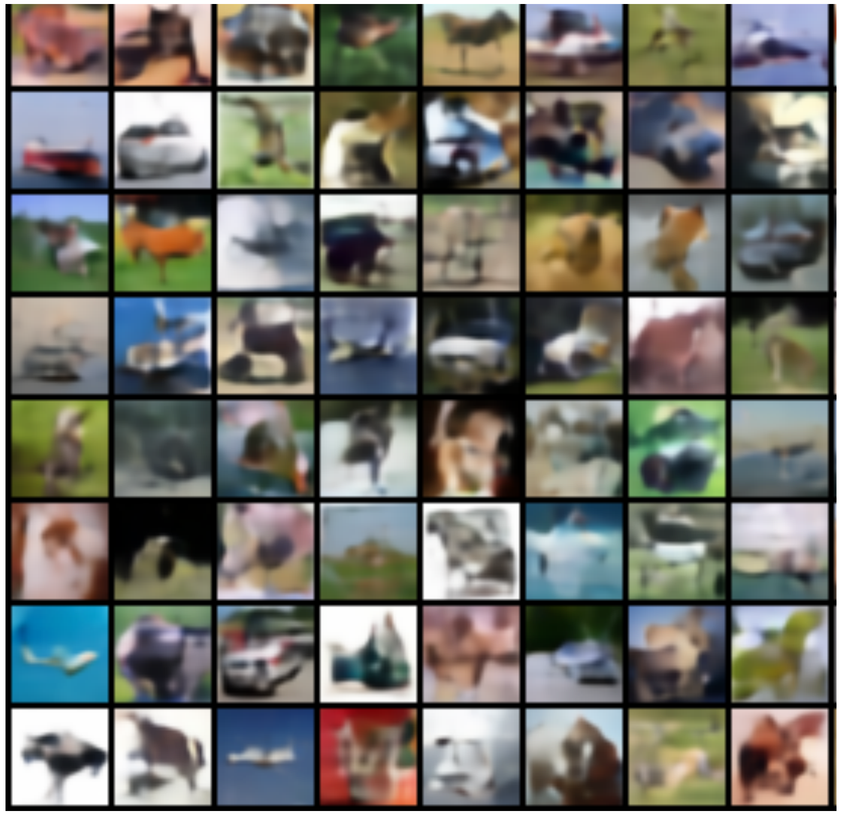
Variational Inference

Stochastic Approximation (MCMC-based)

Good Generative Model?

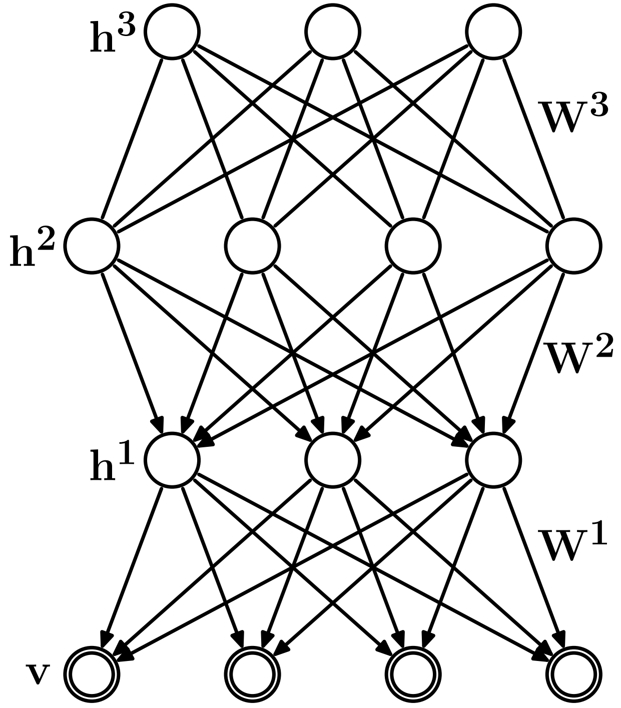
- CIFAR Dataset

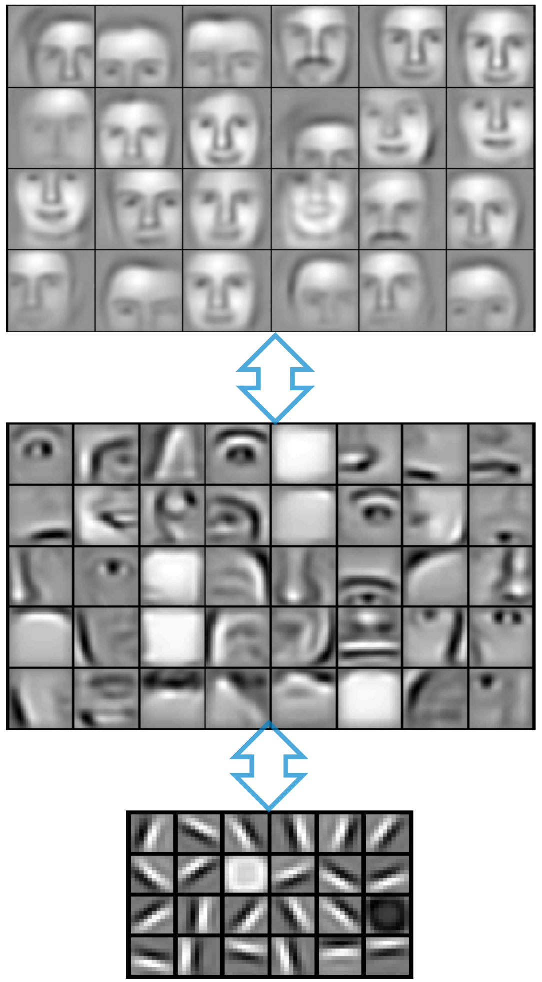

Training


Samples

Learning Part-Based Representations

Deep Belief Network





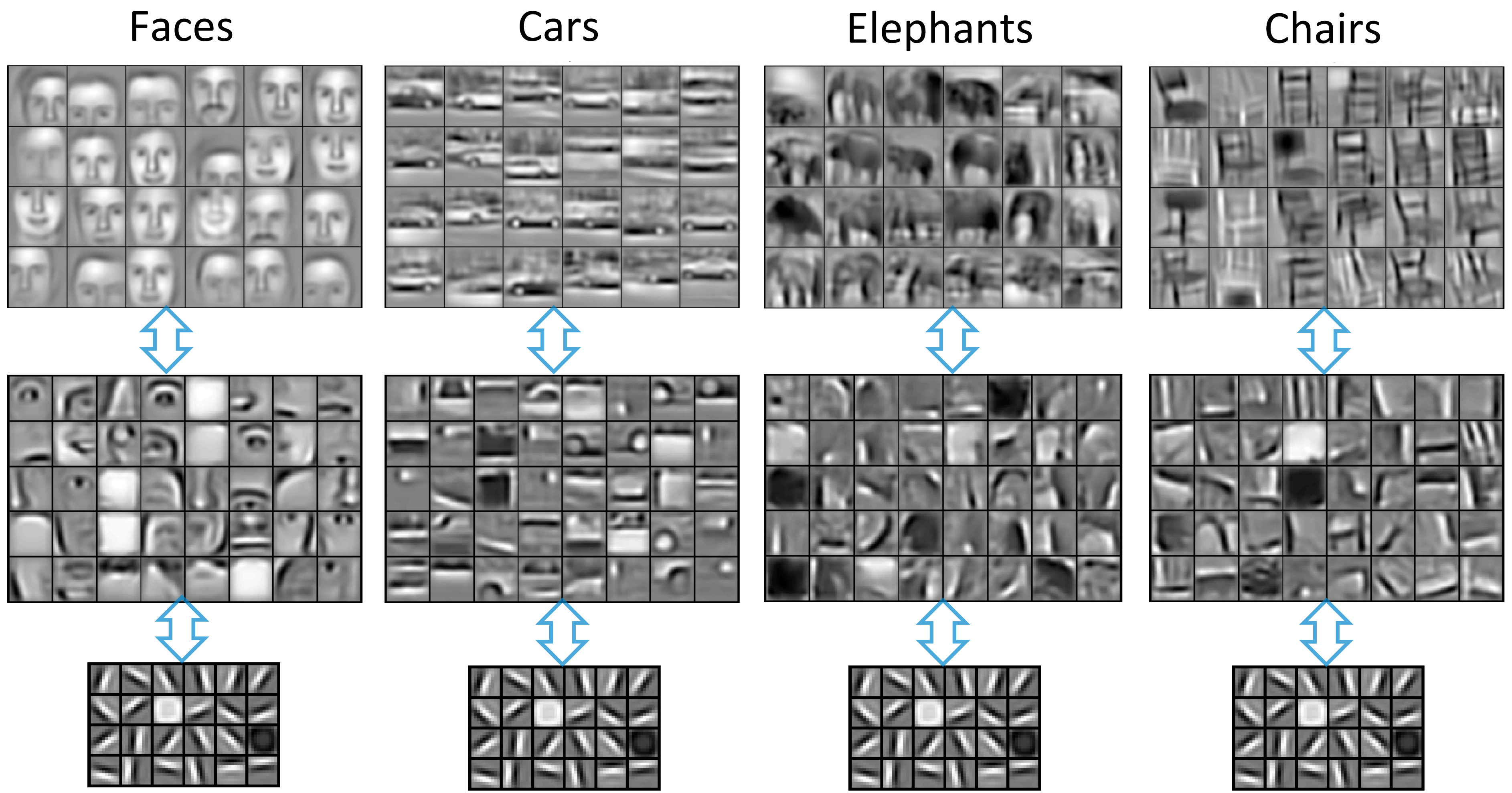
Groups of parts

Object Parts

Trained on face images.

Lee, Grosse, Ranganath, Ng, ICML 2009

Learning Part-Based Representations



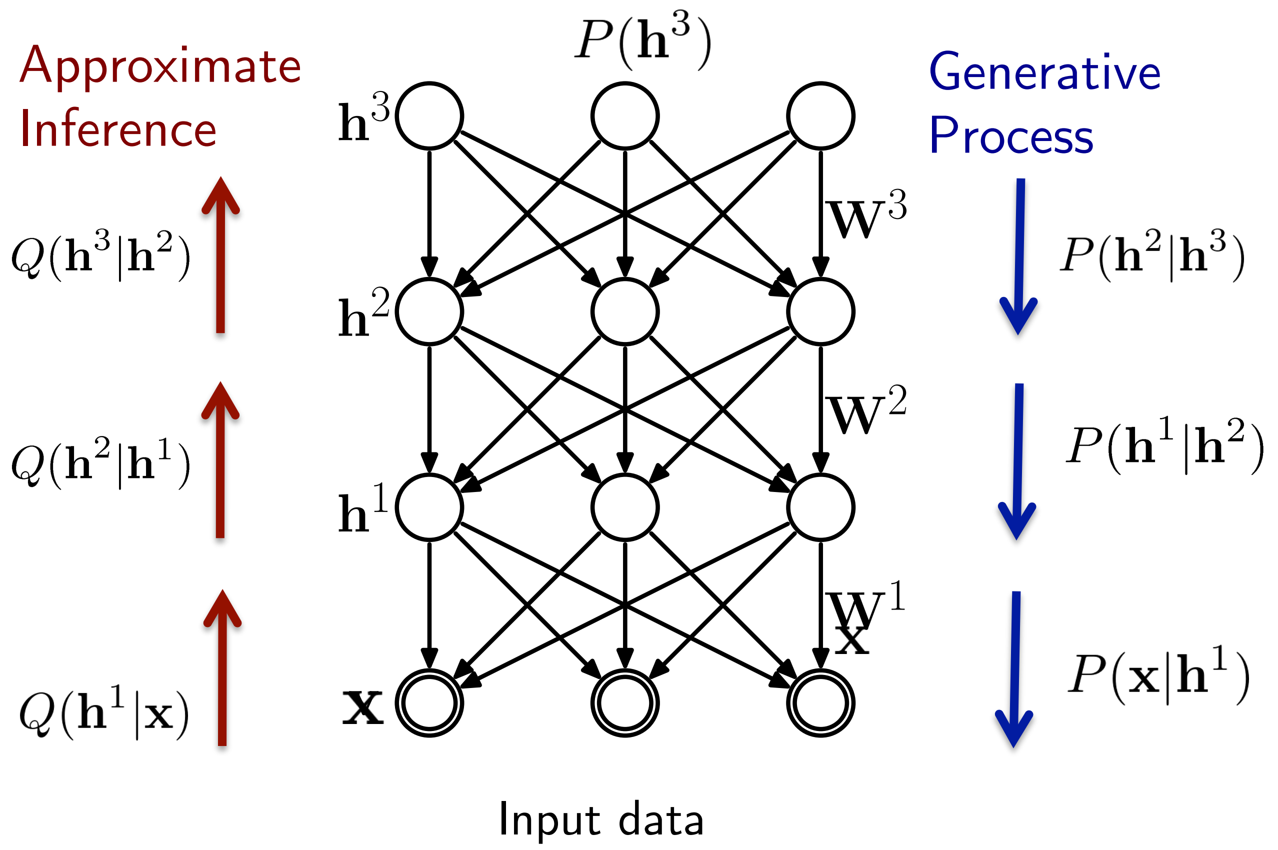
Lee, Grosse, Ranganath, Ng, ICML 2009

Talk Roadmap

- Fully Observed Models
- Undirected Deep Generative Models
  - Restricted Boltzmann Machines (RBMs),
  - Deep Boltzmann Machines (DBMs)
- Directed Deep Generative Models
  - Variational Autoencoders (VAEs)
  - Normalizing Flows
- Generative Adversarial Networks (GANs)

Helmholtz Machines

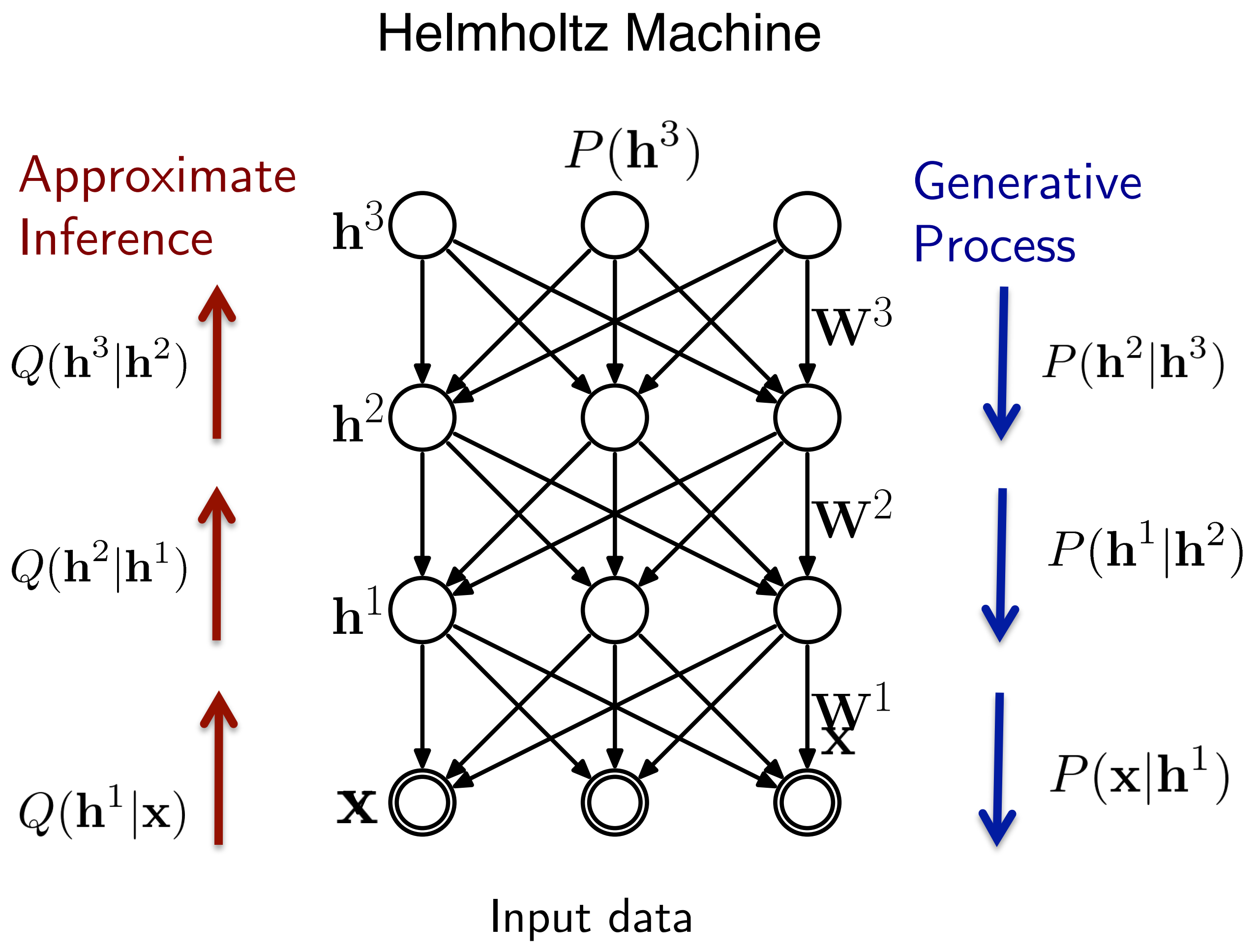
- Hinton, G. E., Dayan, P., Frey, B. J. and Neal, R., Science 1995

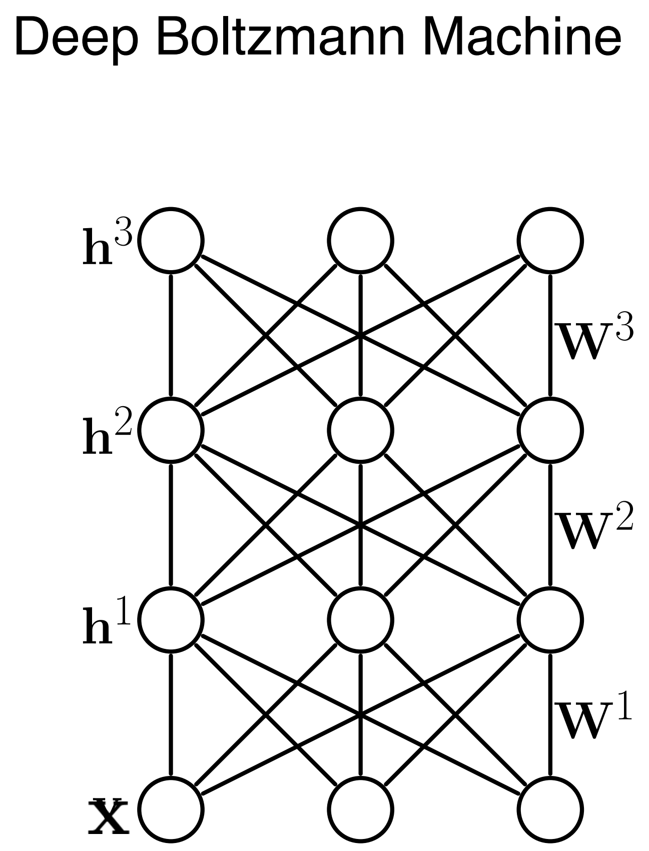


Approximate Inference
$Q(\mathbf{h}^3|\mathbf{h}^2)$
$Q(\mathbf{h}^2|\mathbf{h}^1)$
$Q(\mathbf{h}^1|\mathbf{x})$
$P(\mathbf{h}^3)$
$\mathbf{h}^3$
$\mathbf{h}^2$
$\mathbf{h}^1$
$\mathbf{x}$
$\mathbf{W}^3$
$\mathbf{W}^2$
$\mathbf{W}^1$
$\mathbf{x}$
Input data
Generative Process
$P(\mathbf{h}^2|\mathbf{h}^3)$
$P(\mathbf{h}^1|\mathbf{h}^2)$
$P(\mathbf{x}|\mathbf{h}^1)$

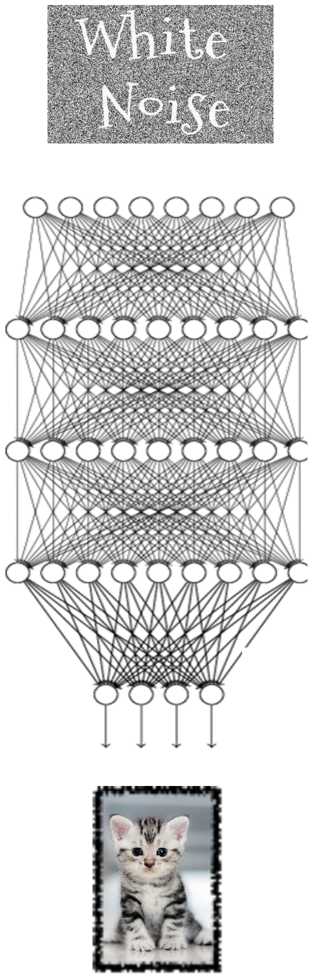
- Kingma & Welling, 2014
- Rezende, Mohamed, Daan, 2014
- Mnih & Gregor, 2014
- Bornschein & Bengio, 2015
- Tang & Salakhutdinov, 2013

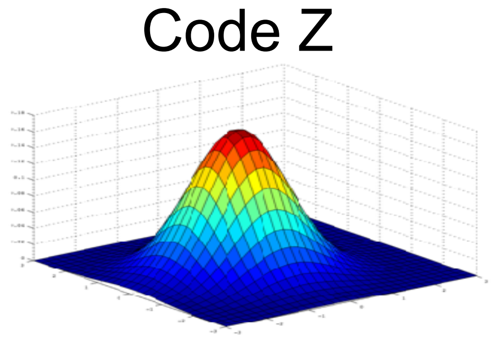
Helmholtz Machines





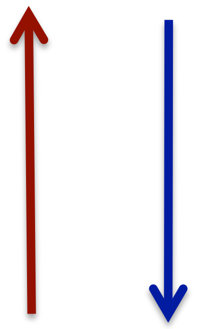
Deep Directed Generative Models

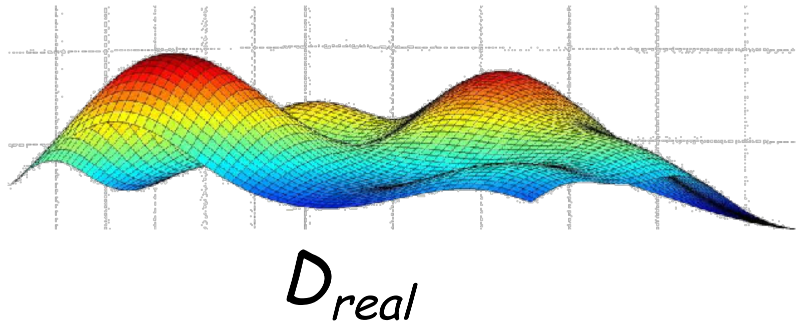




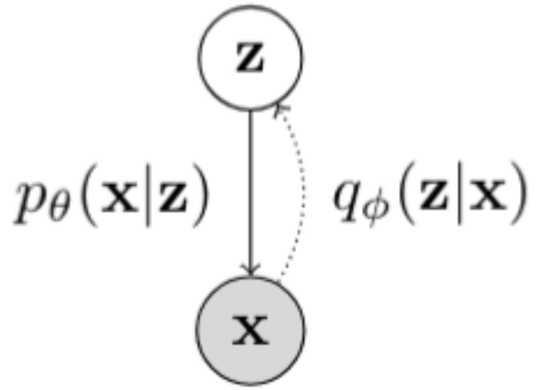
- Recognition
- Bottom-up
- Q(z|x)

- Generative
- Top-Down
- P(x|z)





- Latent Variable Models

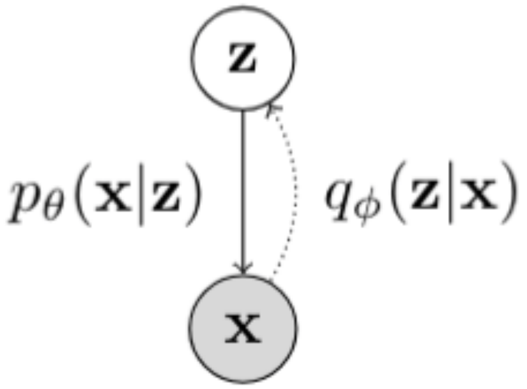

$\mathbf{z}$
$p_\theta(\mathbf{x}|\mathbf{z})$
$q_\phi(\mathbf{z}|\mathbf{x})$
$\mathbf{x}$

$$\log p_\theta(\mathbf{x}) = \log \int p_\theta(\mathbf{x},\mathbf{z})d\mathbf{z}$$

- Conditional distributions are parameterized by deep neural networks

Directed Deep Generative Models

- Directed Latent Variable Models with Inference Network


$\mathbf{z}$
$p_\theta(\mathbf{x}|\mathbf{z})$
$q_\phi(\mathbf{z}|\mathbf{x})$
$\mathbf{x}$

- Maximum log-likelihood objective

$$\max_{\theta} \sum_{\mathbf{x} \in \mathcal{D}} \log p_\theta(\mathbf{x})$$

- Marginal log-likelihood is intractable:

$$\log p_\theta(\mathbf{x}) = \log \int p_\theta(\mathbf{x}, \mathbf{z}) \mathrm{d}\mathbf{z}$$

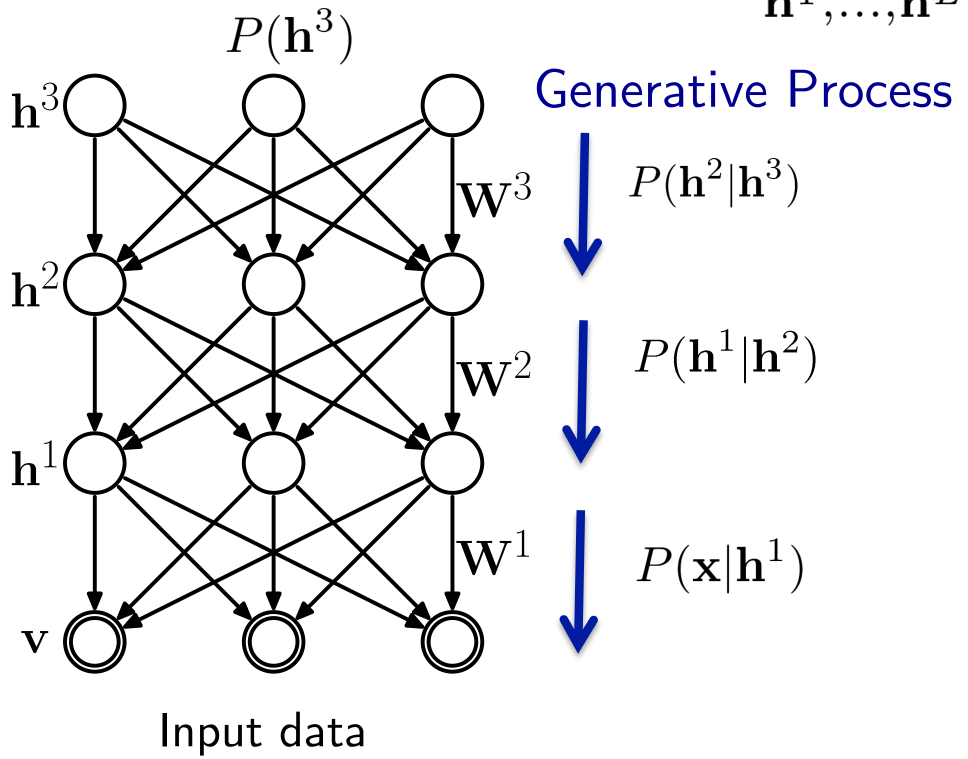
- Key idea: Approximate true posterior p(z|x) with a simple, tractable distribution q(z|x) (inference/recognition network).

Variational Autoencoders (VAEs)

- The VAE defines a generative process in terms of ancestral sampling through a cascade of hidden stochastic layers:

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{\mathbf{h}^1,\dots,\mathbf{h}^L} p(\mathbf{h}^L|\boldsymbol{\theta})p(\mathbf{h}^{L-1}|\mathbf{h}^L,\boldsymbol{\theta})\cdots p(\mathbf{x}|\mathbf{h}^1,\boldsymbol{\theta})$$

Each conditional term denotes a nonlinear relationship

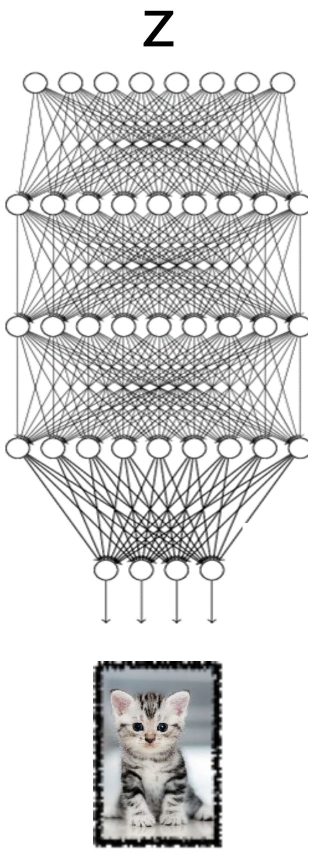


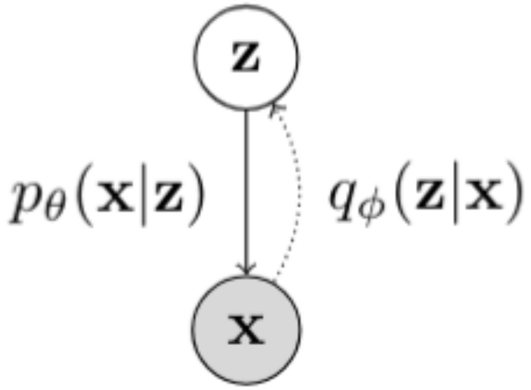
- L is the number of **stochastic** layers
- Sampling and probability evaluation is tractable for each $p(\mathbf{h}^\ell|\mathbf{h}^{\ell+1})$

Kingma and Welling, 2014

Variational Autoencoders (VAEs)

- Single stochastic (Gaussian) layer, followed by many deterministic layers




$\mathbf{z}$
$p_\theta(\mathbf{x}|\mathbf{z})$
$q_\phi(\mathbf{z}|\mathbf{x})$
$\mathbf{x}$

$$p(\mathbf{z}) = \mathcal{N}(0, I)$$

$$p_\theta(\mathbf{x}|\mathbf{z}) = \mathcal{N}\left(\mu(\mathbf{z}, \theta), \Sigma(\mathbf{z}, \theta)\right)$$

Deep neural network parameterized by θ.
(Can use different noise models)

$$q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}\left(\mu(\mathbf{x}, \phi), \Sigma(\mathbf{x}, \phi)\right)$$

Deep neural network parameterized by ϕ.

Kingma and Welling, 2014

Variational Bound

- VAE is trained to maximize the variational lower bound:

$$\begin{aligned}
\log p_\theta(\mathbf{x}) &= \log \int p_\theta(\mathbf{x}, \mathbf{z}) d\mathbf{z} = \log \int q_\phi(\mathbf{z}|\mathbf{x}) \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} d\mathbf{z} \\
&= \log \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \right] \\
&\geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \log \left[\frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} \right] \\
&= \log p_\theta(\mathbf{x}) - \mathrm{KL}\big(q_\phi(\mathbf{z}|\mathbf{x}) || p_\theta(\mathbf{z}|\mathbf{x})\big) = \mathcal{L}(\mathbf{x})
\end{aligned}$$

Variational Bound

- VAE is trained to maximize the variational lower bound:

$$
\begin{aligned}
\log p_\theta(\mathbf{x}) &= \log \int p_\theta(\mathbf{x},\mathbf{z})d\mathbf{z} = \log \int q_\phi(\mathbf{z}|\mathbf{x}) \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} d\mathbf{z} \\
&= \log \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] \\
&\geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \log \left[\frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] \\
&= \log p_\theta(\mathbf{x}) - \mathrm{KL}\big(q_\phi(\mathbf{z}|\mathbf{x})||p_\theta(\mathbf{z}|\mathbf{x})\big) = \mathcal{L}(\mathbf{x})
\end{aligned}
$$

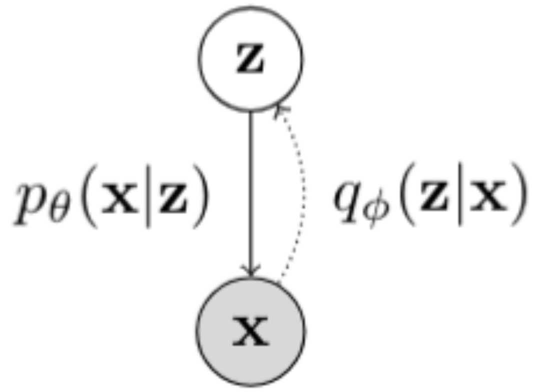
Tightness Condition:

$$q_\phi(\mathbf{z}|\mathbf{x}) = p_\theta(\mathbf{z}|\mathbf{x})$$

- Trading off the data log-likelihood and the KL divergence from the true posterior
- Hard to optimize the variational bound with respect to the q recognition network (high variance)
- Key idea of Kingma and Welling is to use reparameterization trick

Reparameterization

- Assume that the recognition distribution is Gaussian:


$\mathbf{z}$
$p_\theta(\mathbf{x}|\mathbf{z})$
$q_\phi(\mathbf{z}|\mathbf{x})$
$\mathbf{x}$

$$q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}\big(\mu(\mathbf{x},\phi), \Sigma(\mathbf{x},\phi)\big)$$

- Alternatively, we can express this in term of auxiliary variable:

$$\mathbf{z}(\epsilon,\mathbf{x},\phi) = \Sigma(\mathbf{x},\phi)^{1/2}\epsilon + \mu(\mathbf{x},\phi), \quad \epsilon \sim \mathcal{N}(0,I)$$

- The recognition distribution can be expressed as a deterministic mapping

$$\underbrace{\mathbf{z}(\epsilon,\mathbf{x},\phi)}_{\text{Deterministic Encoder}}$$

  - Distribution of ε does not depend on ϕ

Kingma and Welling, 2014

Computing Gradients

- The gradients of the variational bound w.r.t the recognition (similar w.r.t the generative) parameters:

$$
\begin{aligned}
\nabla_\phi \mathcal{L}(\mathbf{x}) &= \nabla_\phi \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}\right] \\
&= \nabla_\phi \mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)}\left[\log \frac{p_\theta(\mathbf{x},\mathbf{z}(\epsilon,\mathbf{x},\phi))}{q_\phi(\mathbf{z}(\epsilon,\mathbf{x},\phi)|\mathbf{x})}\right] \\
&= \mathbb{E}_{\epsilon\sim\mathcal{N}(0,I)}\left[\nabla_\phi \log \frac{p_\theta(\mathbf{x},\mathbf{z}(\epsilon,\mathbf{x},\phi))}{q_\phi(\mathbf{z}(\epsilon,\mathbf{x},\phi)|\mathbf{x})}\right]
\end{aligned}
$$

Autoencoder

Gradients can be computed by backprop

The mapping **z** is a deterministic neural net for fixed ε

VAE Assumptions

- Remember the variational bound:

$$\mathcal{L}(\mathbf{x}) = \log p(\mathbf{x}) - \mathrm{D}_{\mathrm{KL}}\left(q(\mathbf{h}|\mathbf{x}))||p(\mathbf{h}|\mathbf{x})\right)$$

- The variational assumptions must be approximately satisfied.

- The posterior distribution must be approximately factorial (common practice) and predictable with a feed-forward net.

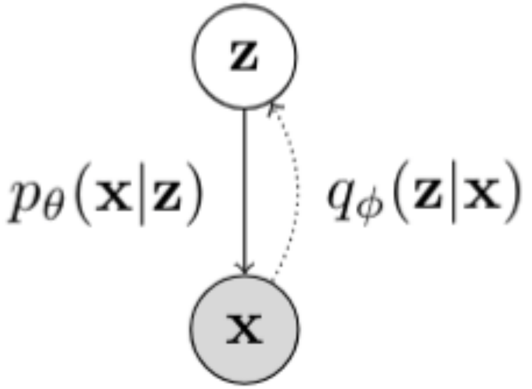
- We can relax these assumptions using a tighter lower bound on marginal log-likelihood.

Importance Weighted Autoencoder

- Improve VAE by using the following k-sample importance weighting of the log-likelihood:

$$\mathcal{L}_k(\mathbf{x}) = \mathbb{E}_{\mathbf{z}_1,\mathbf{z}_2,\ldots,\mathbf{z}_k \sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{1}{k}\sum_{i=1}^{k} \frac{p_\theta(\mathbf{x},\mathbf{z}_i)}{q_\phi(\mathbf{z}_i|\mathbf{x})}\right]$$

$$= \mathbb{E}_{\mathbf{z}_1,\mathbf{z}_2,\ldots,\mathbf{z}_k \sim q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{1}{k}\sum_{i=1}^{k} w_i\right]$$


$\mathbf{z}$
$p_\theta(\mathbf{x}|\mathbf{z})$
$q_\phi(\mathbf{z}|\mathbf{x})$
$\mathbf{x}$

- where multiple z are sampled from the recognition network.

unnormalized importance weights

- Can improve the tightness of the bound.

Burda et al., ICLR 2016,
Mnih & Rezende, ICML 2016

Tighter Lower Bound

- Using more samples can only improve the tightness of the bound.

- For all k, the lower bounds satisfy:

$$\log p(\mathbf{x}) \geq \mathcal{L}_{k+1}(\mathbf{x}) \geq \mathcal{L}_k(\mathbf{x})$$

- Moreover if $p(\mathbf{h}, \mathbf{x})/q(\mathbf{h}|\mathbf{x})$ is bounded, then:

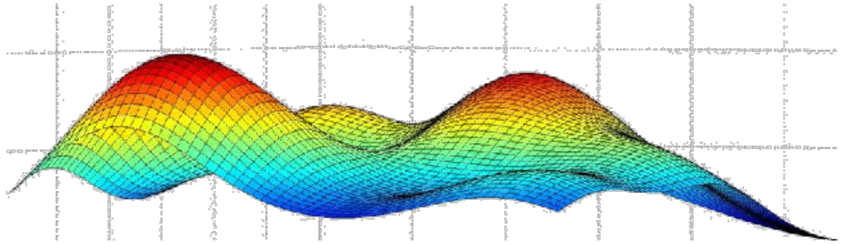
$$\mathcal{L}_k(\mathbf{x}) \to \log p(\mathbf{x}), \quad \text{as} \quad k \to \infty$$

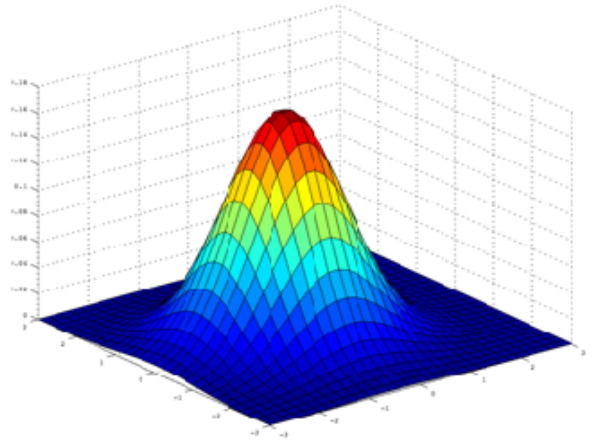
Talk Roadmap

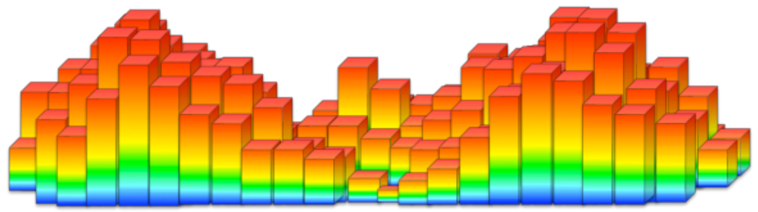
- Fully Observed Models
- Undirected Deep Generative Models
  - Restricted Boltzmann Machines (RBMs),
  - Deep Boltzmann Machines (DBMs)
- Directed Deep Generative Models
  - Variational Autoencoders (VAEs)
  - Normalizing Flows
- Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GAN)

- Implicit generative model for an unknown target density $p(x)$
- Converts sample from a known noise density $p_Z(z)$ to the target $p(x)$


Unknown target density $p(x)$ of data over domain $\mathcal{X}$, e.g. $\mathbb{R}^{32\times 32}$


Noise density $p_Z(z)$ over space $\mathcal{Z}$


Distribution of generated samples should follow target density $p(x)$

[Slide Credit: Manzil Zaheer]

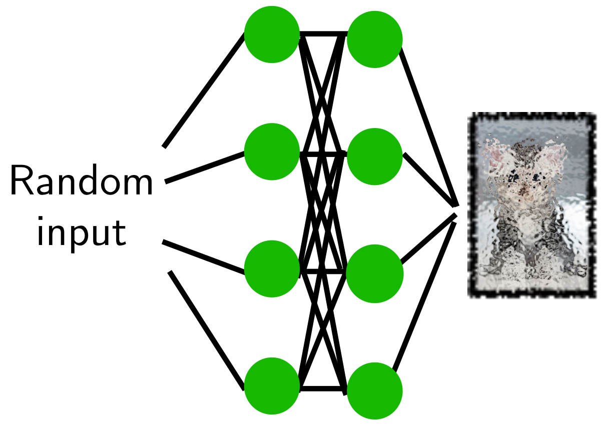
Goodfellow et al, 2014

GAN Formulation

- GAN consists of two components

Generator

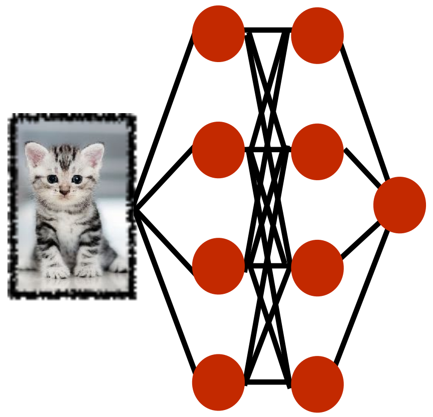
$G : \mathcal{Z} \to \mathcal{X}$



Goal: Produce samples indistinguishable from true data

Discriminator

$D : \mathcal{X} \to \mathbb{R}$



Goal: Distinguish true and generated data apart

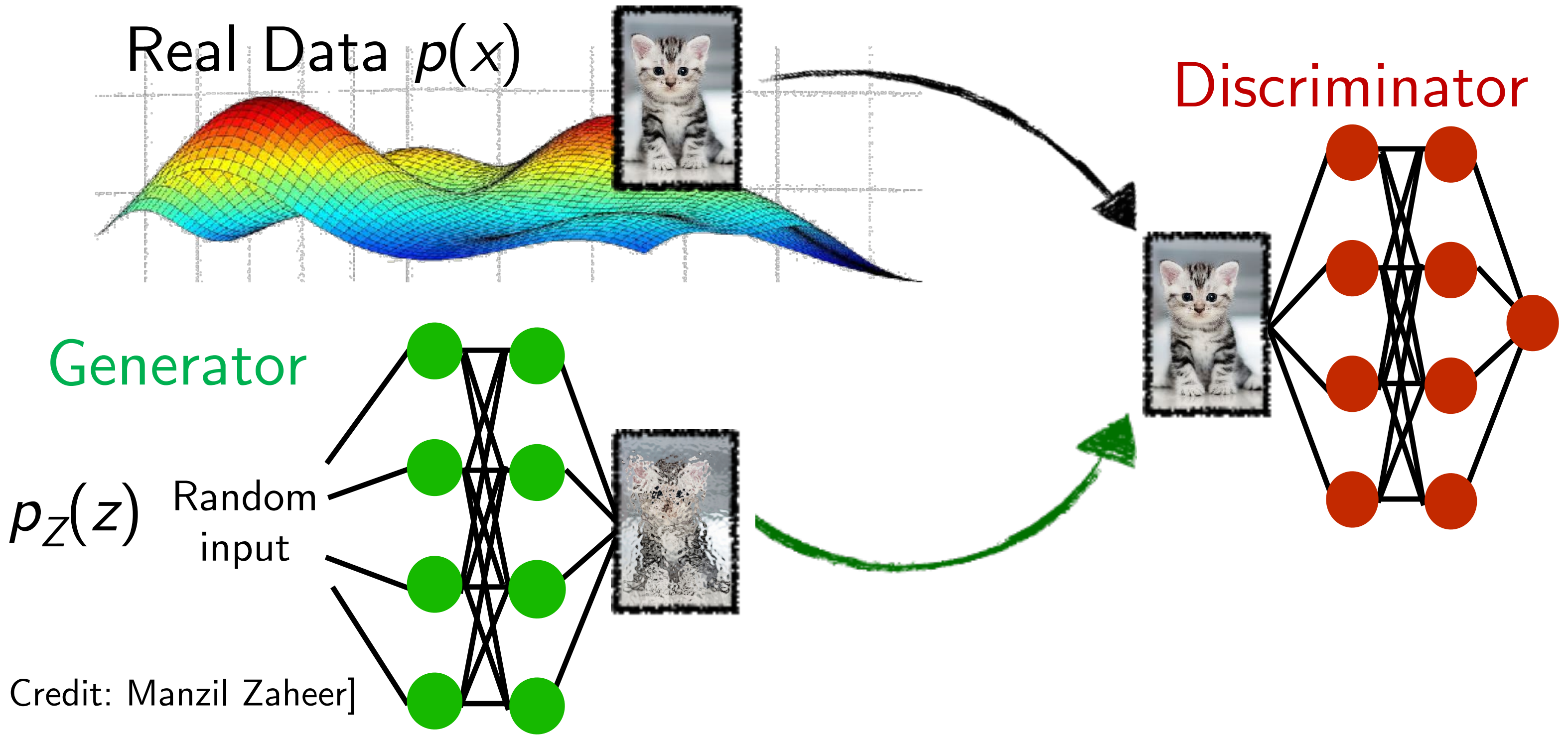
[Slide Credit: Manzil Zaheer]

Goodfellow et al, 2014

GAN Formulation: Discriminator

- Discriminator's objective: Tell real and generated data apart like a classifier

$$\max_{D} \mathbb{E}_{x\sim p}\left[\log D(x)\right] + \mathbb{E}_{z\sim p_Z}\left[\log\left(1 - D(G(z))\right)\right]$$



D outputs:

$D(x) = 1$ real

$D(x) = 0$ generated

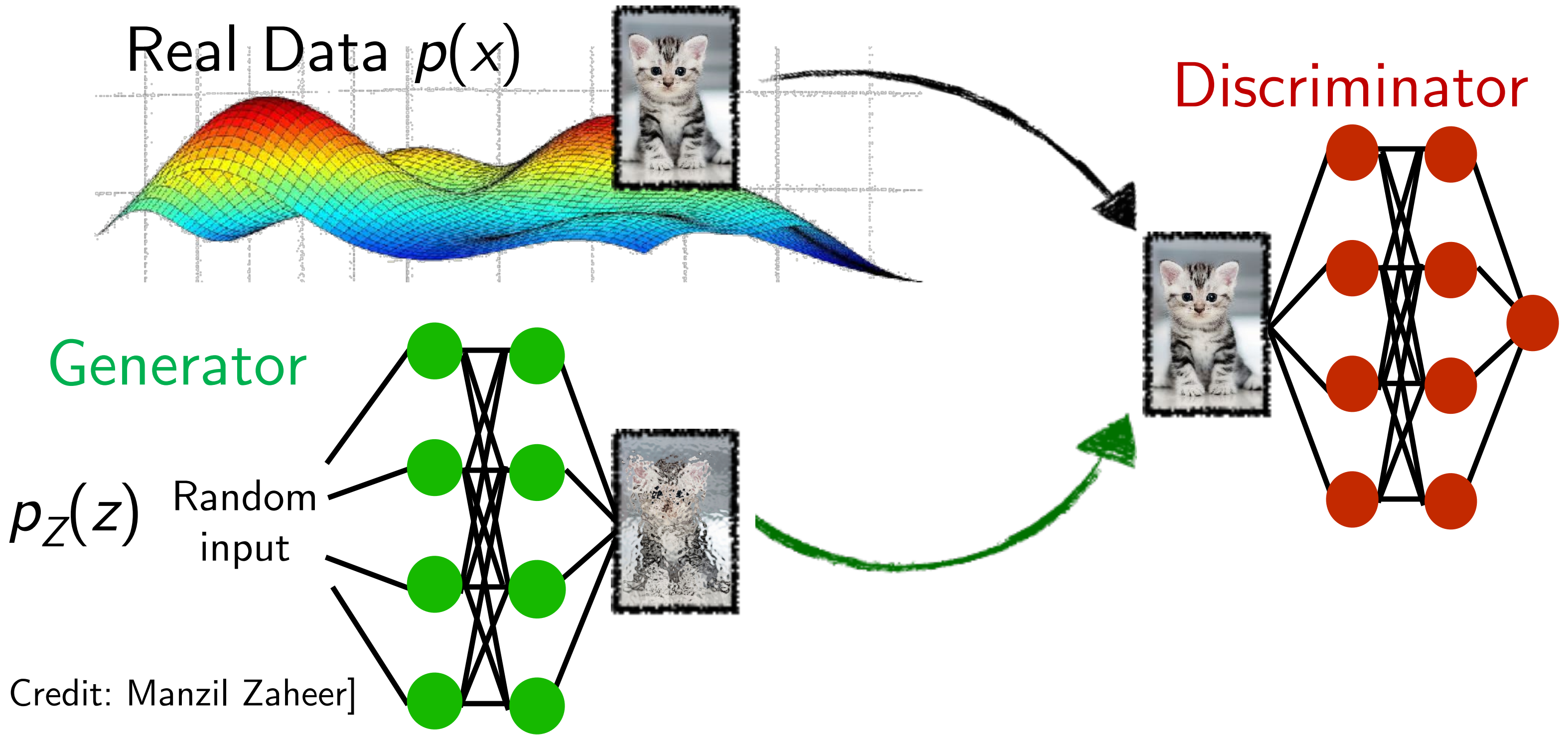
[Slide Credit: Manzil Zaheer]

Goodfellow et al, 2014

GAN Formulation: Generator

- Generator's objective: Fool the best discriminator

$$\min_{G} \max_{D} \mathbb{E}_{x\sim p}\left[\log D(x)\right] + \mathbb{E}_{z\sim p_Z}\left[\log\left(1 - D(G(z))\right)\right]$$



D outputs:

$D(x) = 1$ real

$D(x) = 0$ generated

[Slide Credit: Manzil Zaheer]

Goodfellow et al, 2014

GAN Formulation: Optimization

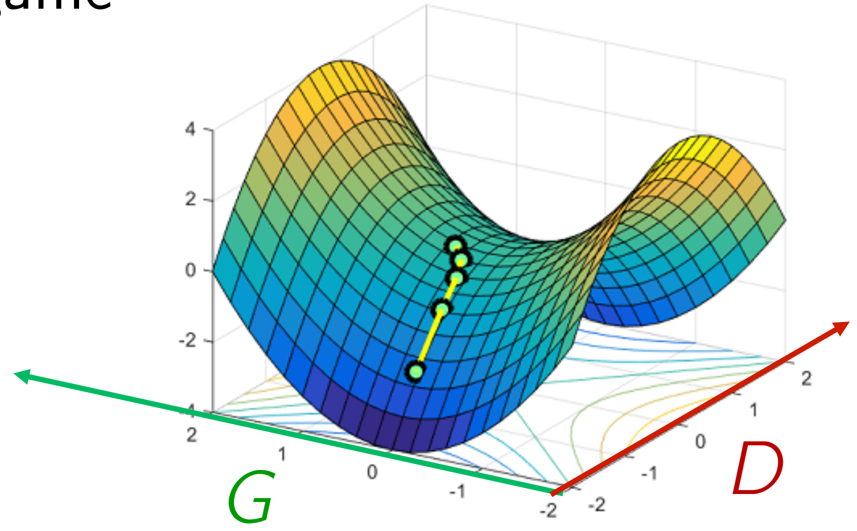
- Overall GAN optimization

$$\min_{G} \max_{D} V(G, D) = \mathbb{E}_{x \sim p}\left[\log D(x)\right] + \mathbb{E}_{z \sim p_Z}\left[\log\left(1 - D(G(z))\right)\right]$$

- The generator-discriminator are iteratively updated using SGD to find "equilibrium" of a "min-max objective" like a game

$$G \leftarrow G - \eta_G \nabla_G V(G, D)$$

$$D \leftarrow D - \eta_D \nabla_D V(G, D)$$



[Slide Credit: Manzil Zaheer]

Wasserstein GAN

- WGAN optimization

$$\min_{G} \max_{D} W(G, D) = \mathbb{E}_{x\sim p}\left[D(x)\right] - \mathbb{E}_{z\sim p_Z}\left[D(G(z))\right]$$

- Difference in expected output on real vs. generated images
  - Generator attempts to drive objective ≈ 0
- More stable optimization

Compare to training DBMs

$$\frac{\partial \log P_\theta(\mathbf{v})}{\partial W^1} = \mathbb{E}_{P_{data}}[\mathbf{v}\mathbf{h^1}^\top] - \mathbb{E}_{P_\theta}[\mathbf{v}\mathbf{h^1}^\top]$$

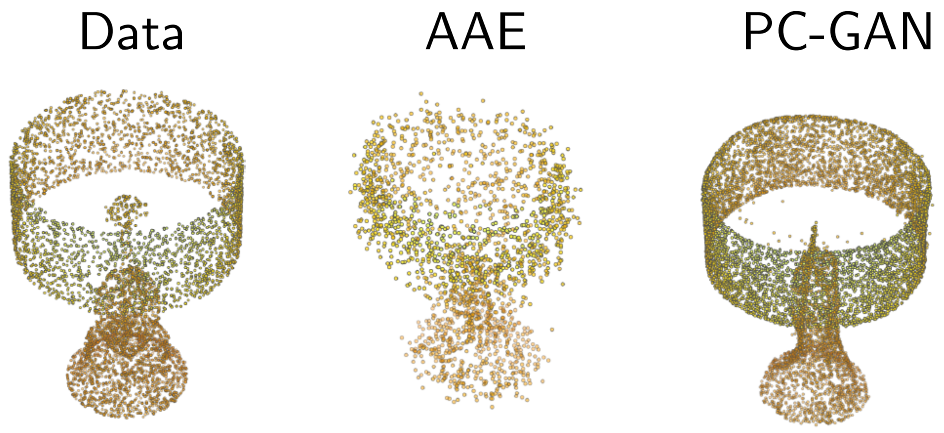
D outputs:

$D(x) = 1$ real

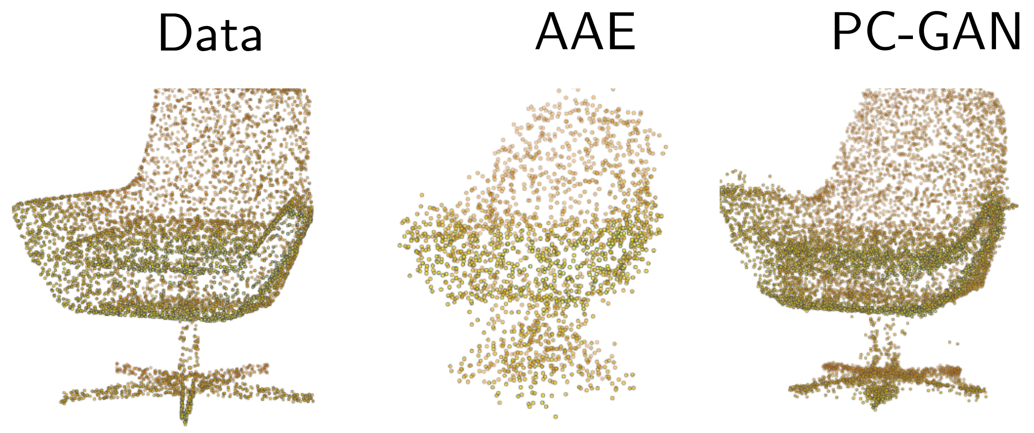
$D(x) = 0$ generated

Arjovsky et al., 2017

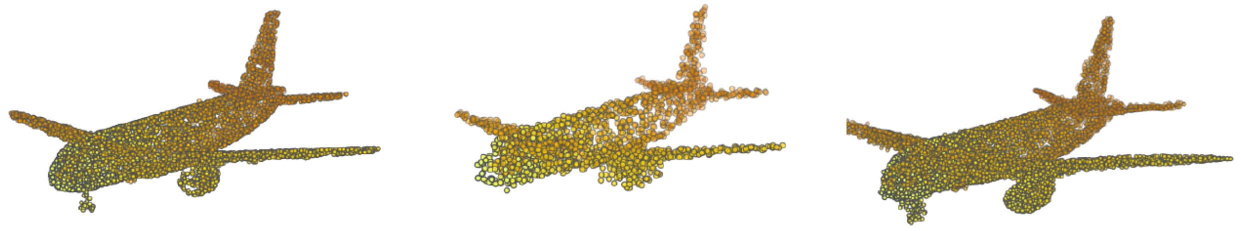
Modelling Point Cloud Data



(a) Lamp



(b) Chair

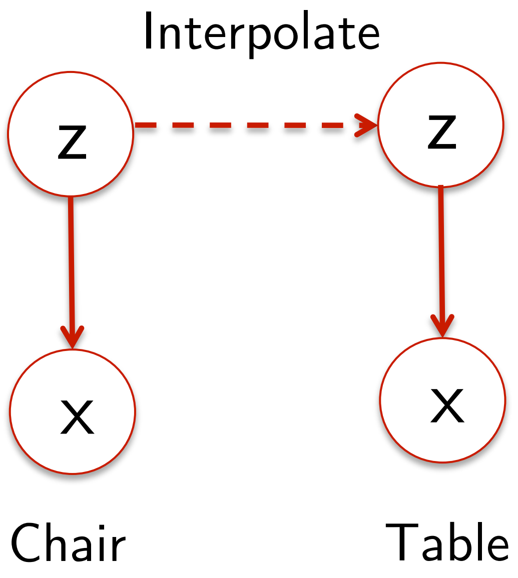


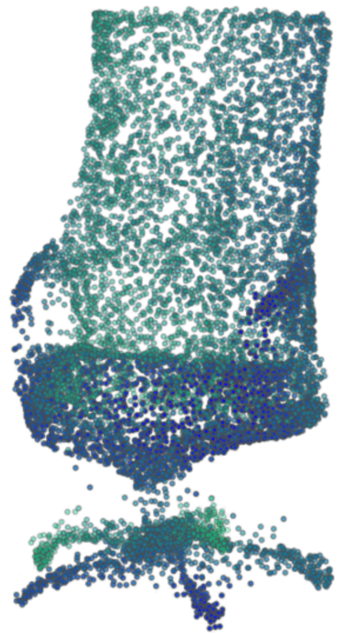
(c) Plane

(d) Guitar

Zaheer et al. Point Cloud GAN 2018

Interpolation in Latent Space

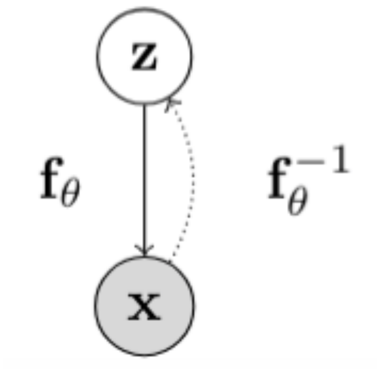




Zaheer et al. Point Cloud GAN 2018

Normalizing Flows

- Directed Latent Variable Invertible models



- The mapping between x and z is deterministic and invertible:

$$\begin{aligned}\mathbf{x} &= \mathbf{f}_\theta(\mathbf{z}) \\ \mathbf{z} &= \mathbf{f}_\theta^{-1}(\mathbf{x})\end{aligned}$$

- Use change-of-variables to relate densities between z and x

$$p_X(\mathbf{x};\theta) = p_Z(\mathbf{z}) \left| \det \frac{\partial \mathbf{f}_\theta^{-1}(\mathbf{x})}{\partial X} \right|_{X=\mathbf{x}}$$

Grover and Ermon DGM Tutorial, NICE (Dinh et al. 2014), Real NVP (Dinh et al. 2016)

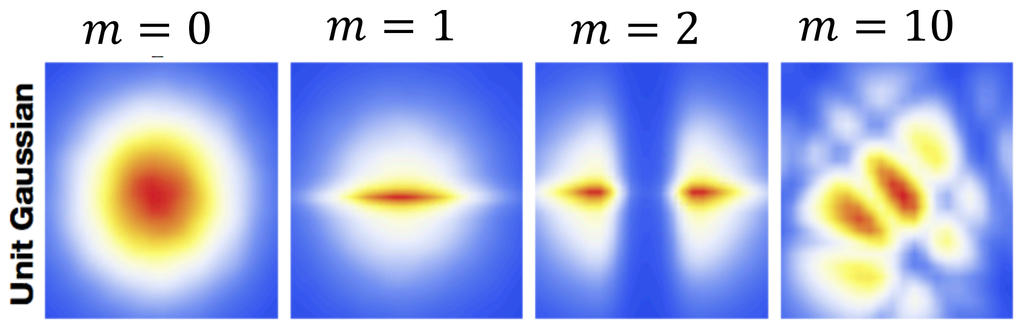
Normalizing Flows

- Invertible transformations can be composed:

$$\mathbf{x} = \mathbf{f}_\theta^M \circ \cdots \circ \mathbf{f}_\theta^1(\mathbf{z}^0); \qquad p_X(\mathbf{x};\theta) = p_{Z^0}(\mathbf{z}^0) \prod_{m=1}^{M} \left| \det \frac{\partial (\mathbf{f}_\theta^m)^{-1}}{\partial Z^m} \right|_{Z^m = \mathbf{z}^m}$$

- Planar Flows

$$f(\mathbf{z}) = \mathbf{z} + \mathbf{u} g(\mathbf{w}^\top \mathbf{z} + b)$$



Rezendre and Mohamed, 2016

Normalizing Flows

- Maximum log-likelihood objective

$$\max_{\theta} \log p_X(\mathcal{D};\theta) = \sum_{\mathbf{x}\in\mathcal{D}} \left(\log p_Z(\mathbf{z}) - \log \left| \det \frac{\partial (\mathbf{f}_\theta)^{-1}}{\partial X} \right|_{X=\mathbf{x}} \right)$$

- Exact log-likelihood evaluation via inverse transformations
- Sampling from the model

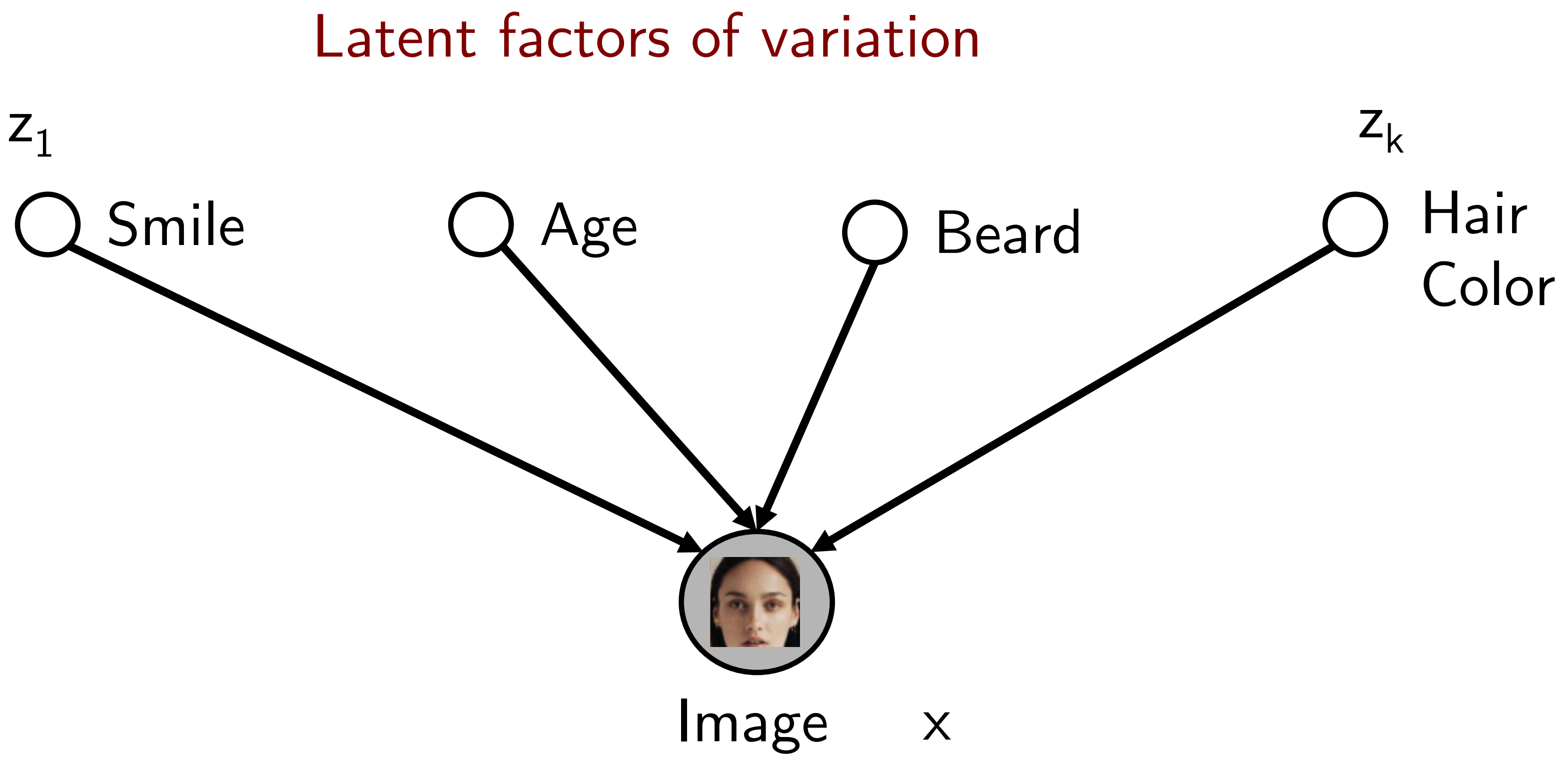
$$\mathbf{z} \sim p_Z(\mathbf{z}), \quad \mathbf{x} = \mathbf{f}_\theta(\mathbf{z})$$

- Inference over the latent representations:

$$\mathbf{z} = \mathbf{f}_\theta^{-1}(\mathbf{x})$$

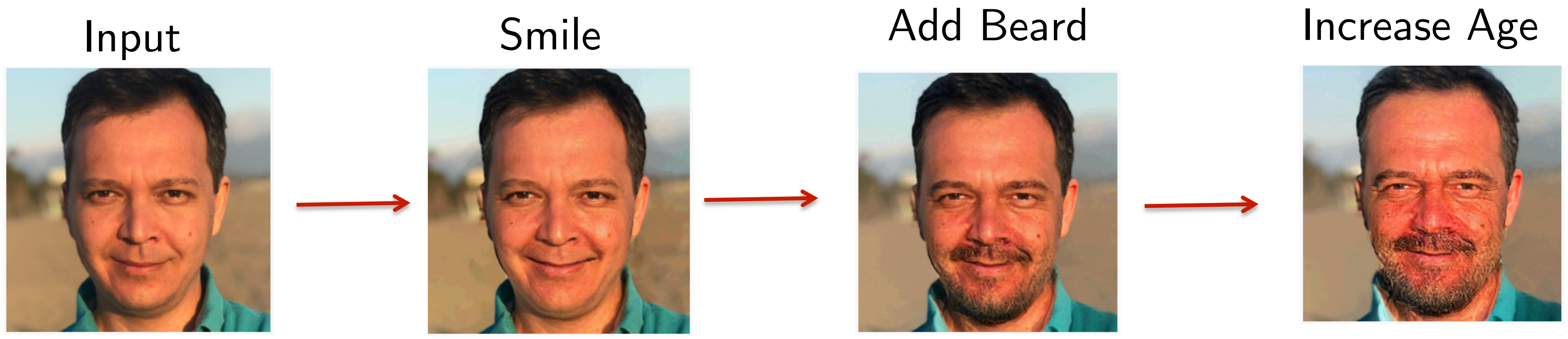
Example: GLOW

- Generative Flow with Invertible 1x1 Convolutions
  https://blog.openai.com/glow/



Kingma, Dhariwal, 2018

Example: GLOW



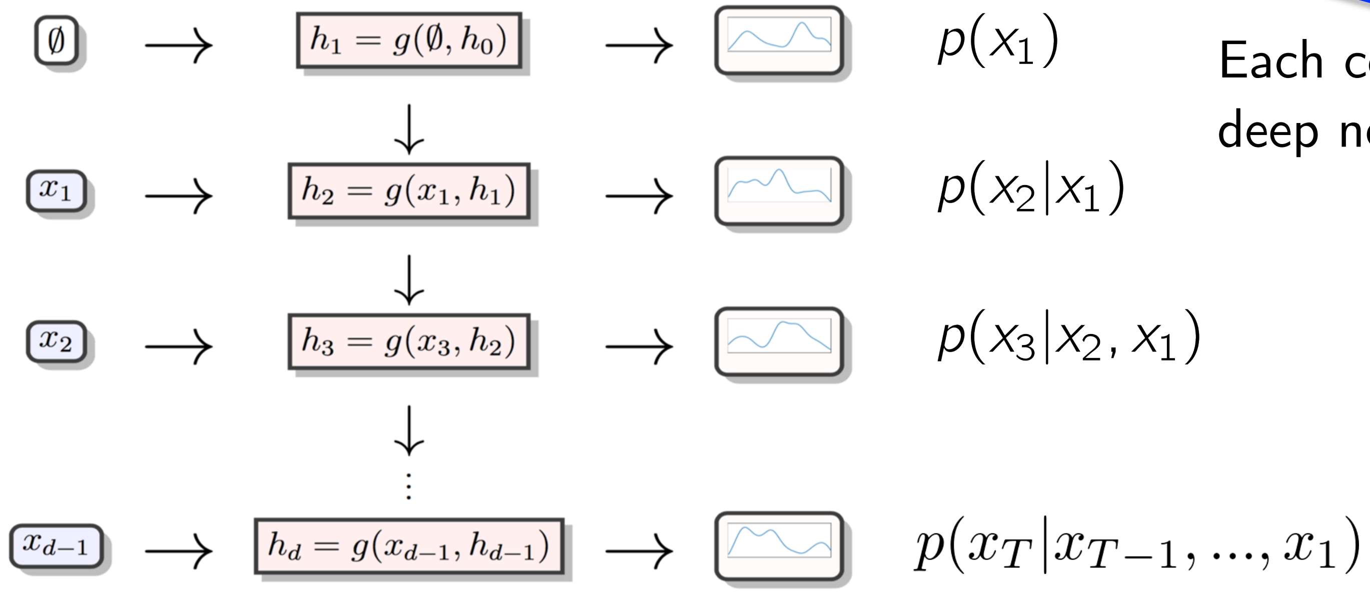
https://blog.openai.com/glow/

Fully Observed Models

- Given a sequence of length T: Density Estimation by Autoregression

$$p(x_1, x_2, ..., x_T) = \prod_{i=1}^{T} p(x_i | x_{i-1}, ..., x_1) \approx \prod_{i=1}^{T} p(x_i | g(x_{i-1}, ..., x_1))$$

Each conditional can be a deep neural network



NADE (Larochelle, 2013), MADE (Germain 2015), PixelCNN (van den Oord, et al, 2016)

Language Modeling

- Given a corpus of T sequential tokens (words) $\mathbf{x} = [x_1, x_2, ..., x_T]$, we model:

$$
\begin{aligned}
P_\theta(\mathbf{x}) &= \prod_{t=1}^{T} P_\theta(x_t \mid \mathbf{x}_{<t}) \\
&= \prod_{t=1}^{T} P(x_t \mid f_\theta(\mathbf{x}_{<t})) \\
&= \prod_{t=1}^{T} \frac{\exp\left(f_\theta(\mathbf{x}_{<t})^\top e_{x_t}\right)}{\sum_{x \in \mathcal{X}} \exp\left(f_\theta(\mathbf{x}_{<t})^\top e_x\right)}
\end{aligned}
$$

  - Here, we will focus on the choice f_θ

Generation:

In a series of conflicts from 1803-15 known as the Napoleonic Wars, various European powers formed five coalitions against the First French Empire. Like the wars sparked by the French Revolution (1789), these further revolutionized the formation, organization and training of European armies and led to an unprecedented militarization, mainly due to mass conscription. Under the leadership of Napoleon, French power rose quickly as the Grande Armée conquered most of Europe, and collapsed rapidly

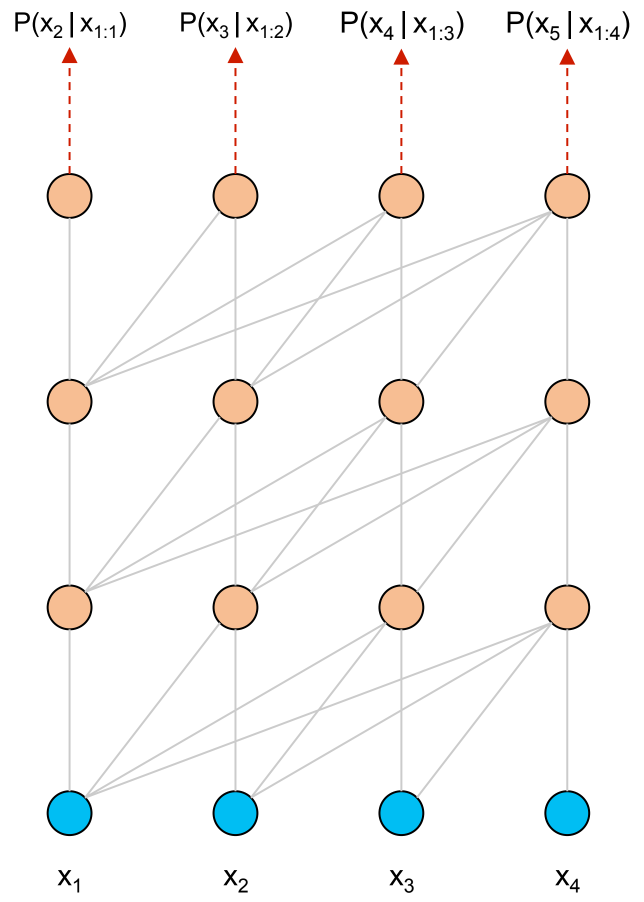
Reference:
after the disastrous invasion of Russia in 1812. Napoleon's empire ultimately suffered complete military defeat in the 1813 – 14 campaigns, resulting in the restoration of the Bourbon monarchy in France. Although Napoleon made a spectacular return in 1815, known as the Hundred Days, his defeat at the Battle of Waterloo, the pursuit of his army and himself, his abdication and banishment to the Island of Saint Helena concluded the Napoleonic Wars.
= = Danube campaign = =
From 1803-06 the Third Coalition fought the First French Empire and its client states (see table at right). Although several naval battles determined control of the seas, the outcome of the war was decided on the continent, predominantly in two major land operations in the Danube valley: the Ulm campaign in the upper Danube and the Vienna campaign, in the middle Danube valley. Political conflicts in Vienna delayed Austria's entry into the Third Coalition until 1805. After hostilities of the War of the Second Coalition ended in 1801, Archduke <unk> emperor's <unk> advantage of the subsequent years of peace to develop a military restructuring plan. He carefully put this plan into effect beginning in 1803 – 04, but implementation was incomplete in 1805 when Karl Mack, Lieutenant Field Marshal and Quartermaster-General of the Army, implemented his own restructuring. Mack bypassed Charles ' methodical approach. Occurring in the field, Mack's plan also undermined the overall command and organizational structure. Regardless, Mack sent an enthusiastic report to Vienna on the military's readiness. Furthermore, after misreading Napoleon's maneuvers in Württemberg, Mack also reported to Vienna on the weakness of French dispositions. His reports convinced the war party advising the emperor, Francis II, to enter the conflict against France, despite Charles ' own advice to the contrary. Responding to the report and rampant anti-French fever in Vienna, Francis dismissed Charles from his

XL Generation:

after the French conquest of Italy. This victory marked the beginning of the Second Coalition. Napoleon's rapid advance caused Austria – Russia, Britain and Russia to make peace negotiations. The Russian army under Prince Mikhail Mikhailovich Mikhailovich Chaffee, commander of the Imperial Russian Army in Switzerland, was able to drive off the Austrians in the Battle of Stockach (1805) and to take Pressburg (modern Ö3 Austria) from the Austrians. At an early stage the Austro-Hungarian fleet had been damaged in a storm off Cape Matapan; this allowed the Allies to launch a full-scale invasion of Italy in February 1805. The Battle of Austerlitz was the decisive French victory against Napoleon and the largest naval battle in history, involving more modern-day European navies. The French military situation worsened as Napoleon faced several threats related to his newly formed Army of Europe, particularly Russia itself, which was now on the verge of collapse. The Russian general and friend of Napoleon, Alexander, had been dissatisfied with the conduct of the Austrians; he was still in conflict with Emperor Napoleon, the French Republic's king, who had declared war on Austria during the summer of 1804. With the war over, Napoleon decided to launch a second invasion of Italy in October 1805.

Napoleon decided to launch a second invasion of Italy in October 1805.
= = Prelude = =
In July 1805, the French 1st Army entered southern Italy. The army, under the command of Marshal Marmont, were reinforced by a few battalions of infantry under Claude General Auguste de Marmont at the town of Philippsburg and another battalion at Belluno. On 17 September 1805, the army marched from Belluno towards Krems. By 29 September, they had reached Belluno and conducted its advance against a small Austrian force. By 31 September, the whole force had been reinforced by a brigade from the Army of Tyrol under the command of Pierre Augereau. The Austrians were now under the command of Marshal Jean Victor Marie Moreau, a member of the Directory. Moreau had taken command of the Austrian invasion force in the spring of 1805. His command included the VI Corps commanded by Jean Baptiste Drouet de Ney and the VI Corps commanded by Generals Jean Victor Marie Moreau and Joseph Souham. Ney's corps consisted of the III. Corps and VI. Corps, which consisted of the III Corps and VI. Corps, located in the Austrian Netherlands, was commanded by Friedrich Joseph, Count Baillet de Latour. Moreau's army consisted of six divisions and several associated brigades.

Vanilla Transformer Language Models



Forward Pass

Step 1: break the corpus into segments

$$\mathbf{x} = \left[\underbrace{(x_1, x_2, \ldots, x_L)}_{\text{segment } 1}, \cdots, \underbrace{(x_{(\tau-1)L+1}, x_{(\tau-1)L+2}, \ldots, x_{\tau T})}_{\text{segment } \tau}, \cdots \right]$$

$$= \left[\underbrace{(x_{1,1}, x_{1,2}, \ldots, x_{1,L})}_{\mathbf{s}_1}, \cdots, \underbrace{(x_{\tau,1}, x_{\tau,2}, \ldots, x_{\tau,L})}_{\mathbf{s}_\tau}, \cdots \right]$$

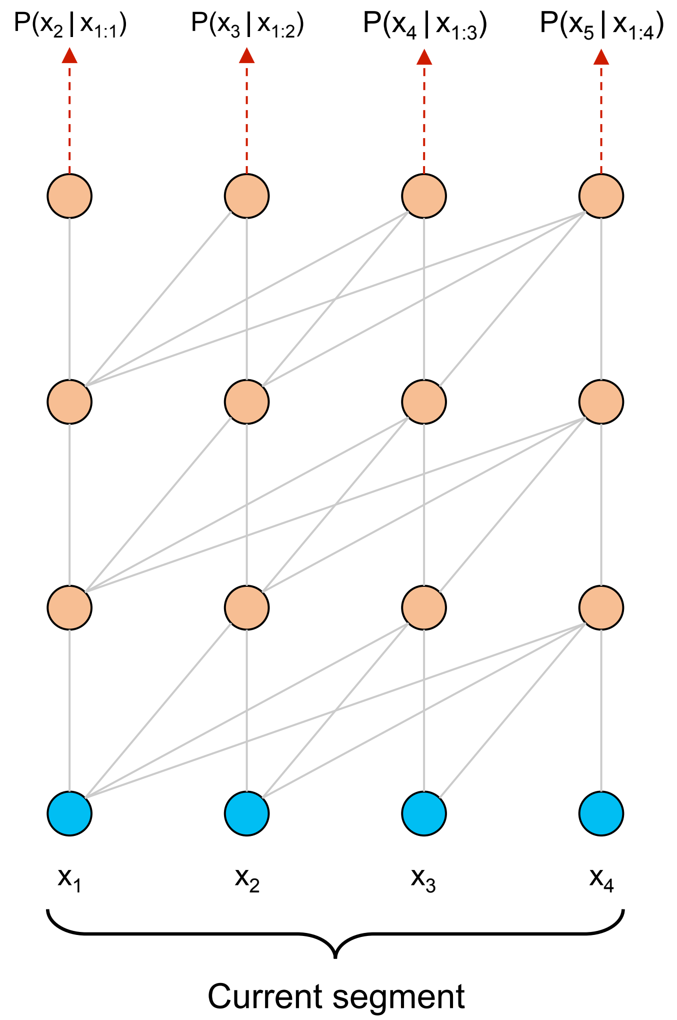
Step 2: Model each segment **independently** (limited memory)

$$P(\mathbf{x}) = \prod_\tau P(\mathbf{s}_\tau \mid \mathbf{s}_{<\tau}) \approx \prod_\tau P(\mathbf{s}_\tau) \qquad \text{(independence assumption)}$$

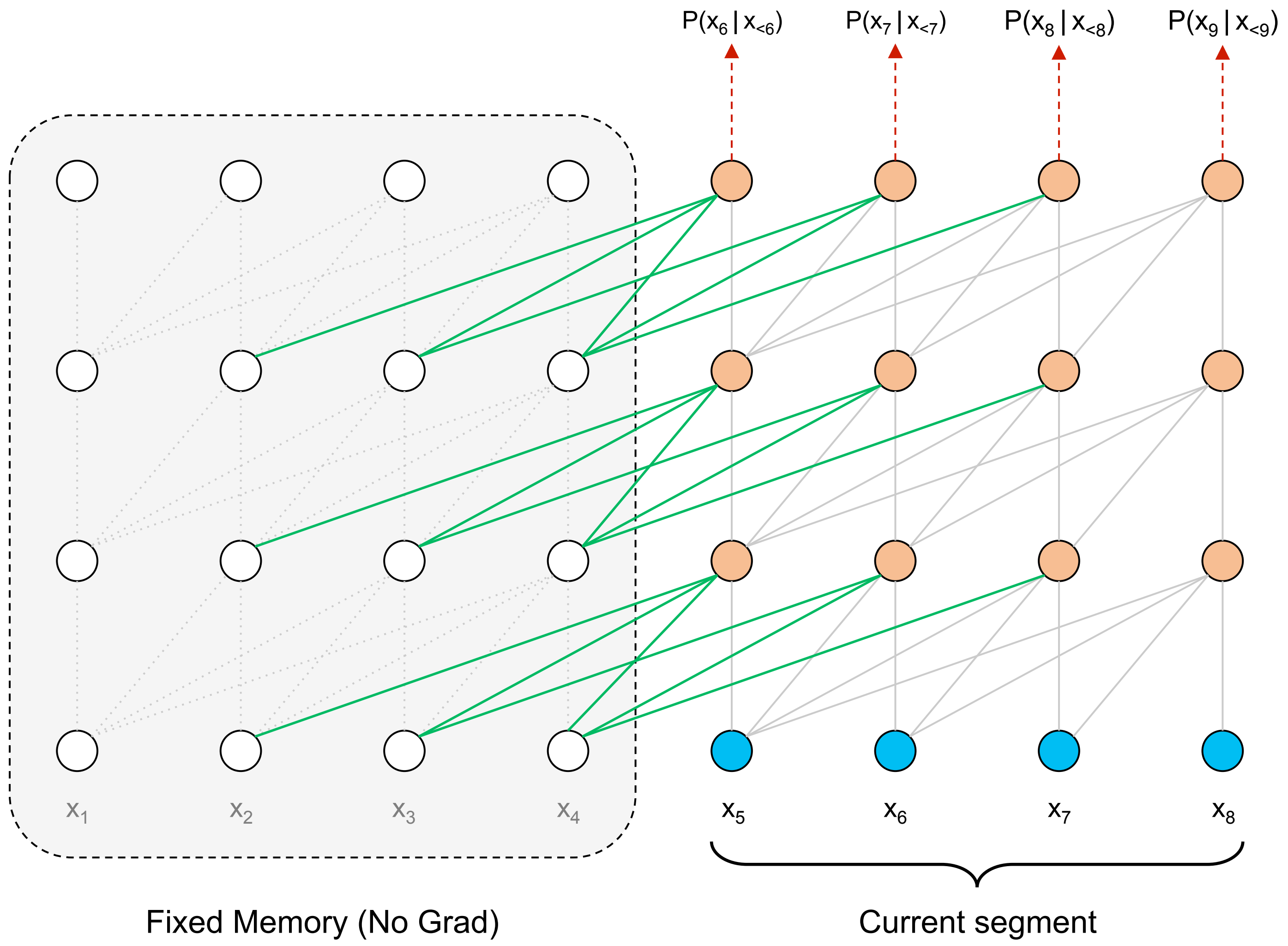
$$= \prod_\tau \prod_{i=1}^{L} P(x_{\tau,i} \mid \mathbf{x}_{\tau,<i}) = \prod_\tau \prod_{I=1}^{L} P(x_{\tau,i} \mid f(\mathbf{x}_{\tau,<I}))$$

Al-Rfou et al., 2016

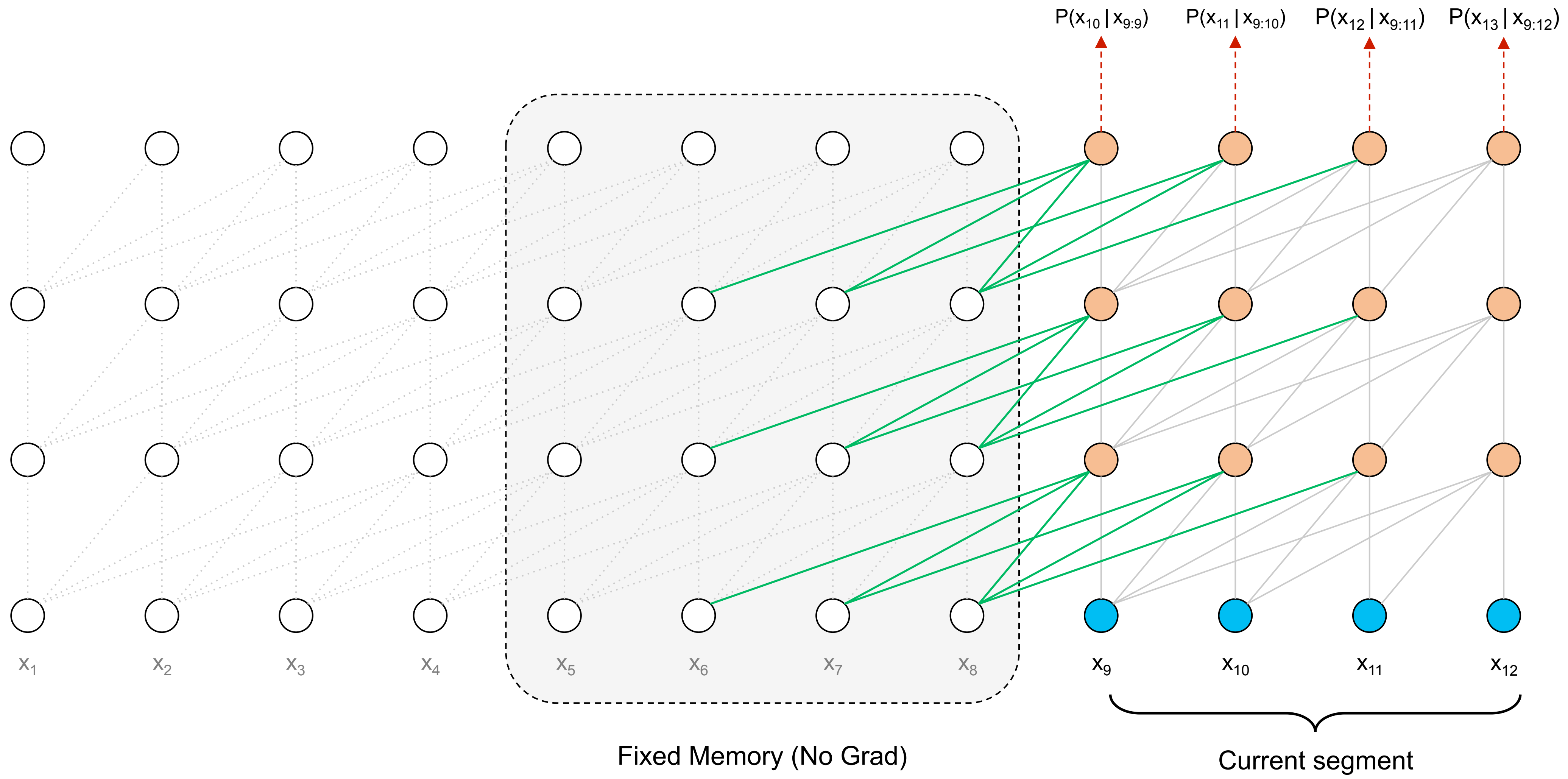
Training with Transformer-XL



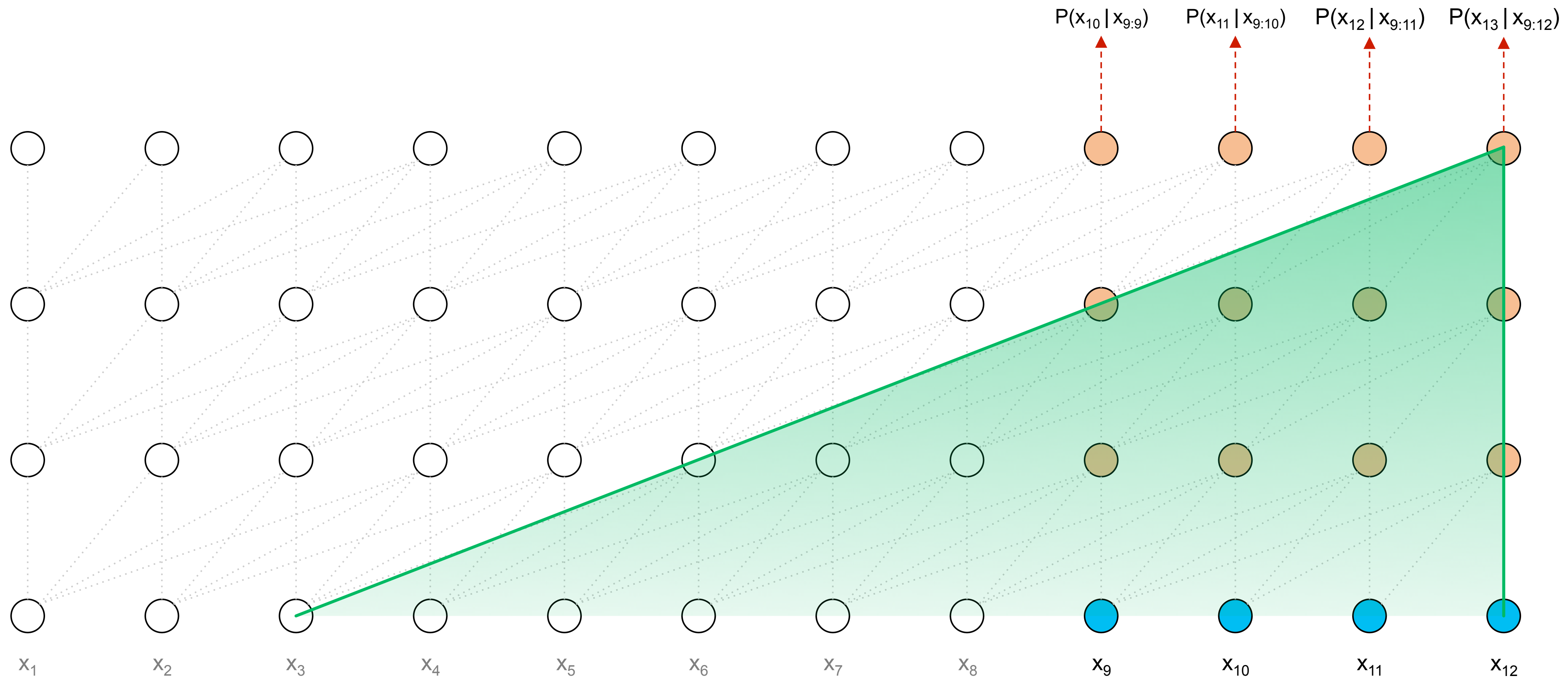
Training with Transformer-XL



Training with Transformer-XL

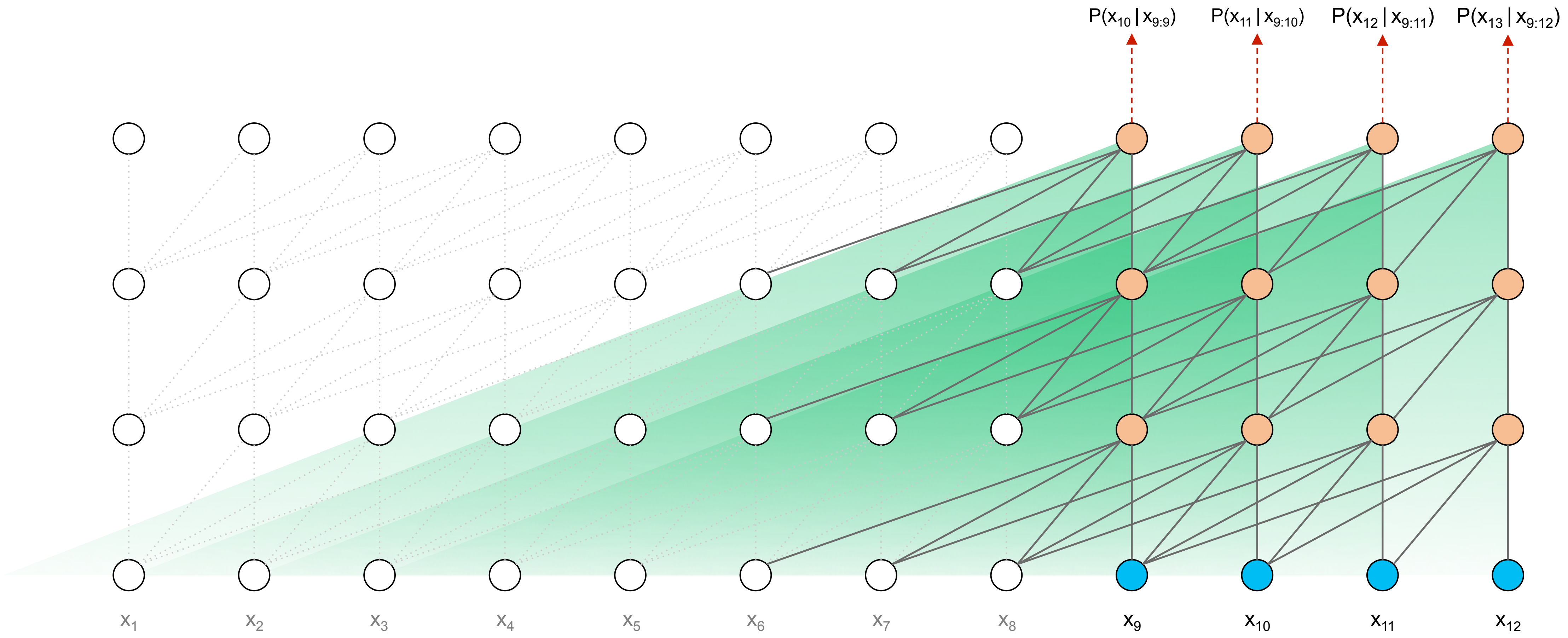


Modeling Much Longer Context



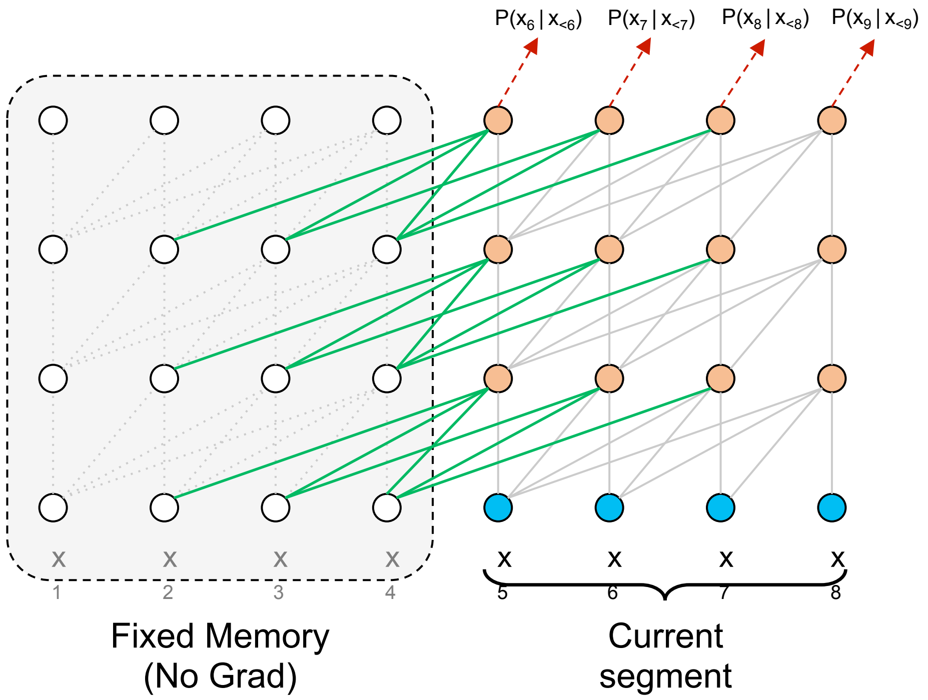
Extra-Long context span: linearly increasing w.r.t. both **segment length** and **number of layers**

Evaluation with Transformer-XL



Process the tokens in a segment in **one forward pass**, without any recomputing → 1800X faster

Transformer-XL



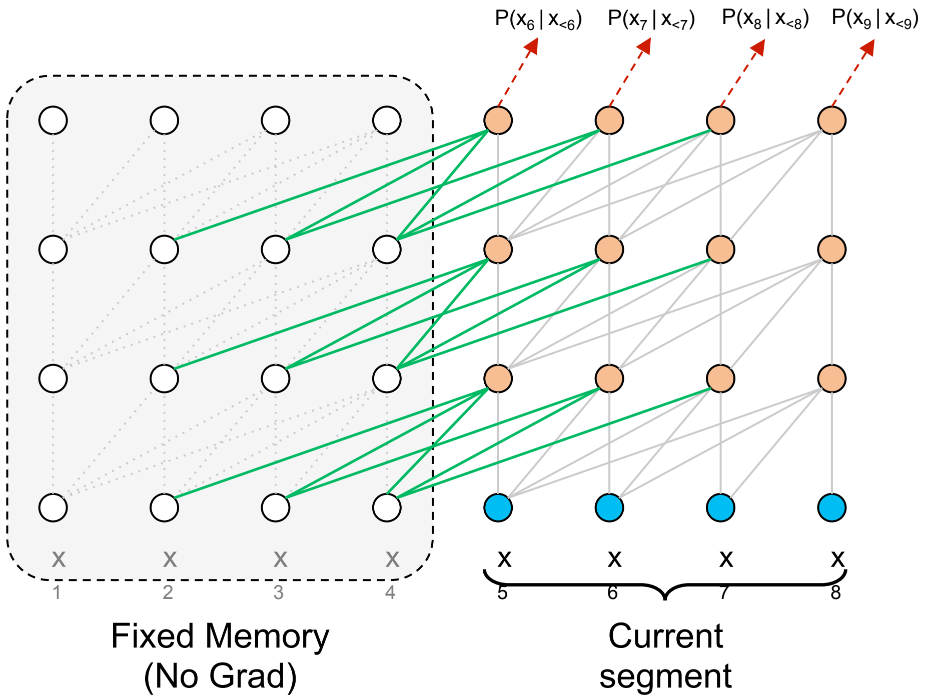
$$\tilde{\mathbf{h}}_\tau^{n-1} = [\mathrm{SG}(\mathbf{m}_\tau^{n-1}) \circ \mathbf{h}_\tau^{n-1}]$$

- $\mathbf{h}_\tau^n \in \mathbb{R}^{L\times d}$ n-th layer hidden state sequence produced for sequence τ
- $\mathbf{m}_\tau^{n-1}$ is memory cashed before segment τ
- SG stands for stop gradient
- $[\cdot \circ \cdot]$ stands for concatenation
- Incorporate extended context

$$\mathbf{q}_\tau^n = \mathbf{h}_\tau^{n-1}\mathbf{W}_q$$
$$\mathbf{k}_\tau^n = \tilde{\mathbf{h}}_\tau^{n-1}\mathbf{W}_k$$
$$\mathbf{v}_\tau^n = \tilde{\mathbf{h}}_\tau^{n-1}\mathbf{W}_v$$

$$\mathbf{h}_\tau^n = \text{Transformer-Layer}(\mathbf{q}_\tau^n, \mathbf{k}_\tau^n, \mathbf{v}_\tau^n)$$

Transformer-XL



$$\widetilde{\mathbf{h}}_{\tau}^{n-1} = \left[\mathrm{SG}(\mathbf{m}_{\tau}^{n-1}) \circ \mathbf{h}_{\tau}^{n-1}\right]$$

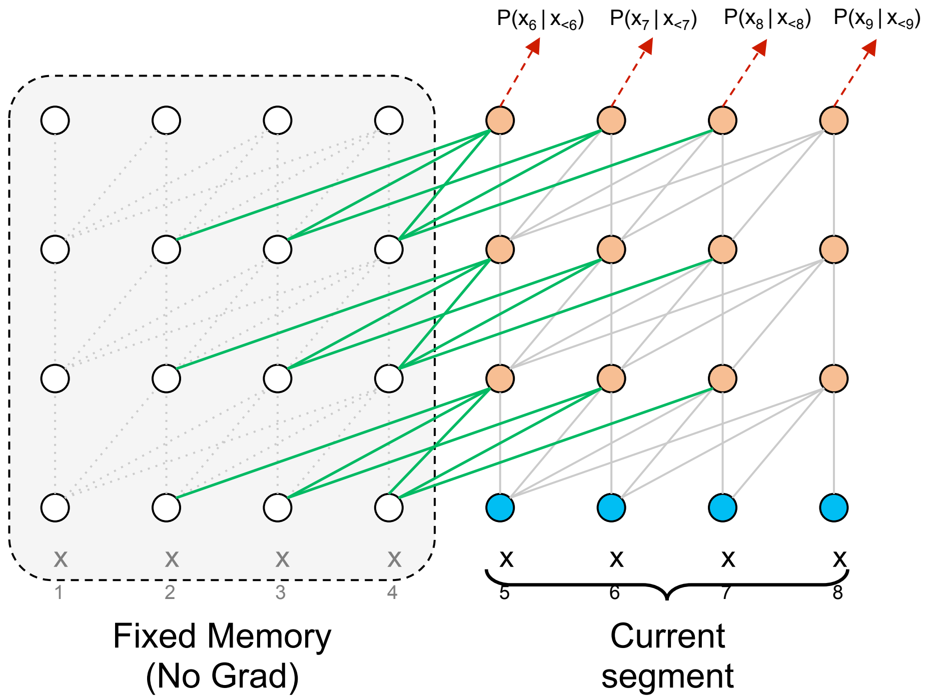
- $\mathbf{h}_{\tau}^{n} \in \mathbb{R}^{L \times d}$ n-th layer hidden state sequence produced for sequence τ
- $\mathbf{m}_{\tau}^{n-1}$ is memory cashed before segment τ
- SG stands for stop gradient
- $[\cdot \circ \cdot]$ stands for concatenation
- Incorporate extended context

$$\mathbf{q}_{\tau}^{n} = \mathbf{h}_{\tau}^{n-1}\mathbf{W}_{q}$$
$$\mathbf{k}_{\tau}^{n} = \tilde{\mathbf{h}}_{\tau}^{n-1}\mathbf{W}_{k}$$
$$\mathbf{v}_{\tau}^{n} = \tilde{\mathbf{h}}_{\tau}^{n-1}\mathbf{W}_{v}$$

Model parameters

$$\mathbf{h}_{\tau}^{n} = \text{Transformer-Layer}\left(\mathbf{q}_{\tau}^{n}, \mathbf{k}_{\tau}^{n}, \mathbf{v}_{\tau}^{n}\right)$$

Transformer-XL



$$\tilde{\mathbf{h}}_{\tau}^{n-1} = \left[\mathrm{SG}(\mathbf{m}_{\tau}^{n-1}) \circ \mathbf{h}_{\tau}^{n-1}\right]$$

- $\mathbf{h}_{\tau}^{n} \in \mathbb{R}^{L \times d}$ n-th layer hidden state sequence produced for sequence τ
- $\mathbf{m}_{\tau}^{n-1}$ is memory cashed before segment τ
- SG stands for stop gradient
- $[\cdot \circ \cdot]$ stands for concatenation
- Incorporate extended context

$$\mathbf{q}_{\tau}^{n} = \mathbf{h}_{\tau}^{n-1}\mathbf{W}_{q}$$
$$\mathbf{k}_{\tau}^{n} = \tilde{\mathbf{h}}_{\tau}^{n-1}\mathbf{W}_{k}$$
$$\mathbf{v}_{\tau}^{n} = \tilde{\mathbf{h}}_{\tau}^{n-1}\mathbf{W}_{v}$$

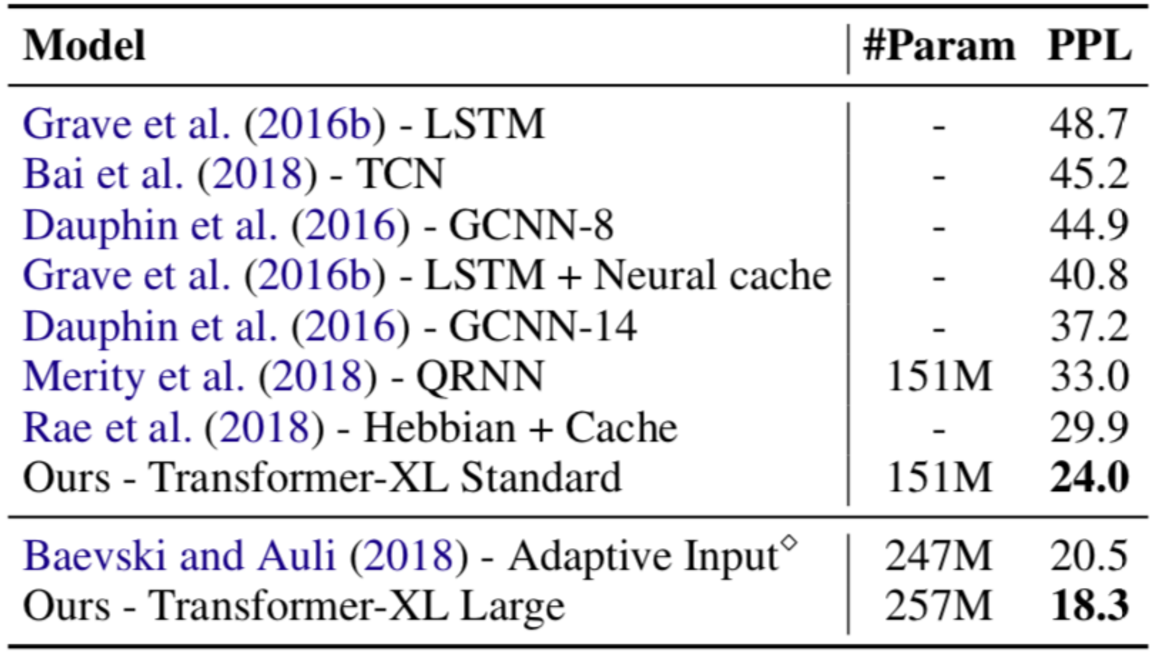
Model parameters

Extended context at layer n-1

$$\mathbf{h}_{\tau}^{n} = \text{Transformer-Layer}\left(\mathbf{q}_{\tau}^{n}, \mathbf{k}_{\tau}^{n}, \mathbf{v}_{\tau}^{n}\right)$$

Results

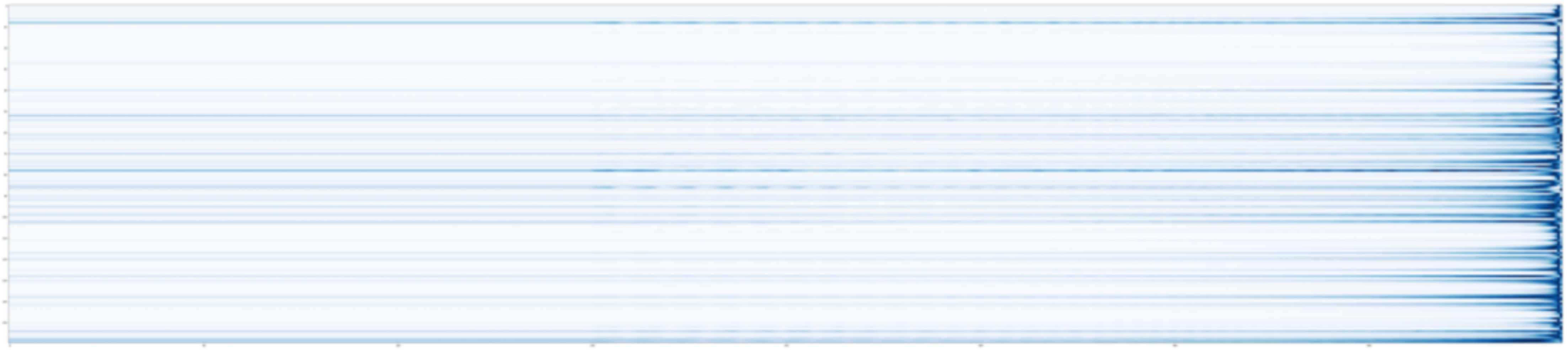
- WikiText-103 Test Corpus:
  - 103M tokens from 28K articles
  - Average length is 3.6K per article
- Training
  - Training segment length 400
  - Test segment length 1600
  - 16 layers



Model	#Param	PPL
Grave et al. (2016b) - LSTM	-	48.7
Bai et al. (2018) - TCN	-	45.2
Dauphin et al. (2016) - GCNN-8	-	44.9
Grave et al. (2016b) - LSTM + Neural cache	-	40.8
Dauphin et al. (2016) - GCNN-14	-	37.2
Merity et al. (2018) - QRNN	151M	33.0
Rae et al. (2018) - Hebbian + Cache	-	29.9
Ours - Transformer-XL Standard	151M	**24.0**
Baevski and Auli (2018) - Adaptive Input$^{\diamond}$	247M	20.5
Ours - Transformer-XL Large	257M	**18.3**

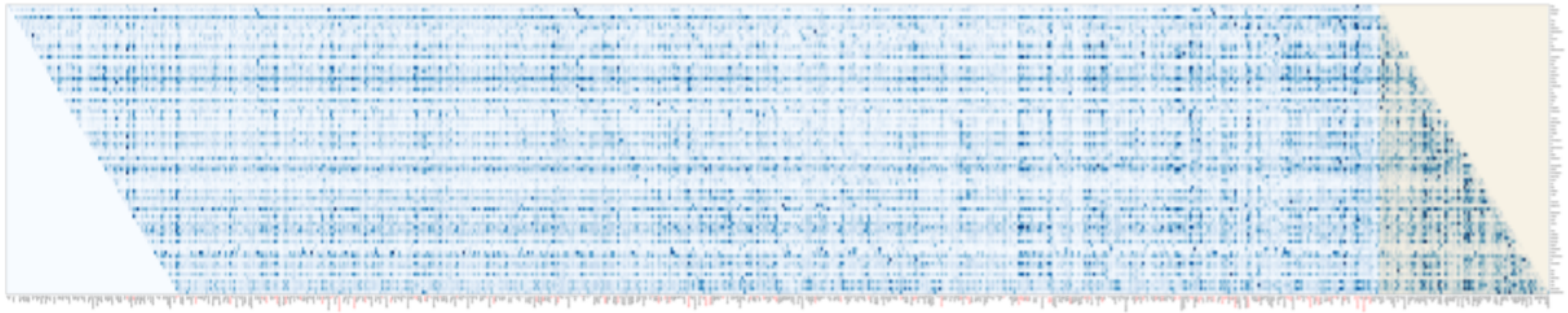
- Achieves State-of-the-Art on 5 publicly available datasets

Visualization of Attention:



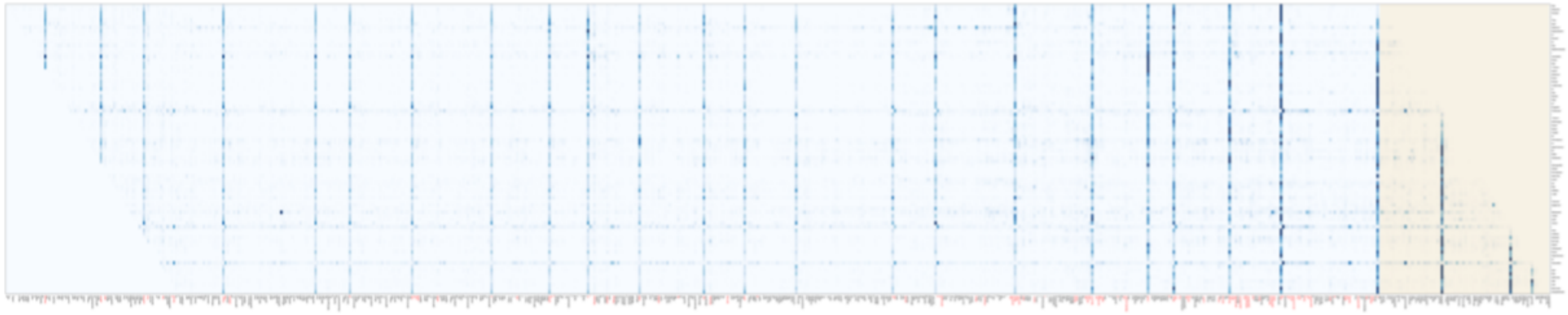
- Average attention over the previous 640 tokens,
- There are totally 160 attention weights across 16 layers

Visualization of Attention:



- Average attention over the previous 640 tokens,
- There are totally 160 attention weights across 16 layers

Visualization of Attention:



- Average attention over the previous 640 tokens,
- There are totally 160 attention weights across 16 layers

After finishing the 2010 season strong, the Dodgers named Kershaw as the Opening Day Starter for the 2011 season. On May 29, he pitched the second complete-game shutout of his career, striking out 10 while winning a two-hitter against the Florida Marlins, 8 – 0; he also had two singles and an RBI, scoring twice in the game. He produced his third career shutout on June 20, a two-hit, 11-strikeout effort against the Detroit Tigers. Kershaw became the first Dodgers starter to strike out the side in the 9th inning since Sandy Koufax's perfect game. In his next start, on June 26, Kershaw pitched another complete game (against the Los Angeles Angels of Anaheim). He became the first Dodger starter to have back-to-back complete game victories since Jeff Weaver in the 2005 season and the first Dodger to have double-digit strikeouts in consecutive starts since Chan-Ho Park in the 2000 season. He was awarded the National League Player of the Week award for the week of June 20 – 26 as a result of those two starts. Midway through June, Kershaw had amassed 32 career victories

XL Generation:

, tying him with Luis Castillo for the third best start total in the league and trailing only Johnny Bench and Eddie Mathews (37). The Dodgers signed Kershaw to a one-year deal on July 14, 2011, with a player option for 2012, in a deal worth just under $ 300,000. The Dodgers entered the 2011 season as the major league leaders, and Kershaw had the best record in the National League in saves (39), games started (78) and walks (120); he also had the lowest win / loss ratio in the league (1.87), and he had the worst on-base percentage (.349). He had the best ERA (2.00), most wins (16), or most shutouts (29) in his career (25), while leading the league in losses per nine innings pitched (4.01). Following the 2011 season, the Dodgers traded Kershaw to the Kansas City Royals on July 29 for pitcher Javier Torres.

= = = Kansas City Royals = = =

= = = = 2012 season = = = =

During spring training, Kershaw played very well. He was selected to spring training as a relief pitcher for the Royals for the 2012 season. After an injury to closer Javier Vázquez, he was activated on April 29 to replace Matt Holliday in the Royals ' starting rotation. In his only start with the Royals, on August 6, 2012, Kershaw struck out five batters in seven innings pitched to help the Royals to their first victory in franchise history. On September 27, 2012, it appeared Kershaw was going to pitch a complete game shutout against the Detroit Tigers, but did not manage to do so since the Tigers won 3 – 1. At the conclusion of the season, Kershaw was named Major League Baseball's Most Valuable Player, was chosen to the All-Star Game at Busch Stadium and was named to the All-Star Game as the starting pitcher at shortstop. The Royals announced on

Reference:

, a 3.15 ERA and 593 career strikeouts in 568.2 innings. According to the Elias Sports Bureau, Kershaw was the first 23-year-old pitcher to have that many victories, an ERA that low and an average of more than one strikeout per inning since ERA became an official statistic in 1910. Kershaw was selected to the National League team for the 2011 Major League Baseball All-Star Game, his first All-Star selection. In the month of July, Kershaw was 4 – 1 with a 2.02 ERA and NL-leading 45 strikeouts, earning him the National League Pitcher of the Month Award. On August 23, he struck out Matt Holliday of the St. Louis Cardinals for his 200th strikeout of the season and became the 10th Dodger pitcher to record back-to-back 200 strikeout seasons and the first since Chan-Ho Park did it in the 2001 season. Kershaw finished the 2011 season by leading the NL with 21 wins, 248 strikeouts and a 2.28 ERA, winning the NL pitching Triple Crown, the first Triple Crown winner since Jake Peavy of the 2007 San Diego Padres and the first Dodger since Sandy Koufax won it in the 1966 season. Justin Verlander of the Detroit Tigers won the American League Triple Crown the same season, marking the first major-league season since 1924 to feature Triple Crown-winning pitchers in both leagues. Kershaw's 21 wins were the most by a Dodger pitcher since Orel Hershiser won 23 during the 1988 season. His ERA was the lowest by a Dodger since Hershiser's 2.03 in the 1985 season, his strikeouts were the most by a Dodger since Koufax's 317 in 1966 and his 233 1 / 3 innings pitched were the most since Chan Ho Park pitched 234 in 2001. Since 1965 when Koufax did it, Peavy and Kershaw are only two pitchers in the National League have led the league in wins, strikeouts, ERA, and WHIP (walks plus hits per inning pitched). Kershaw also became just the second <unk> to

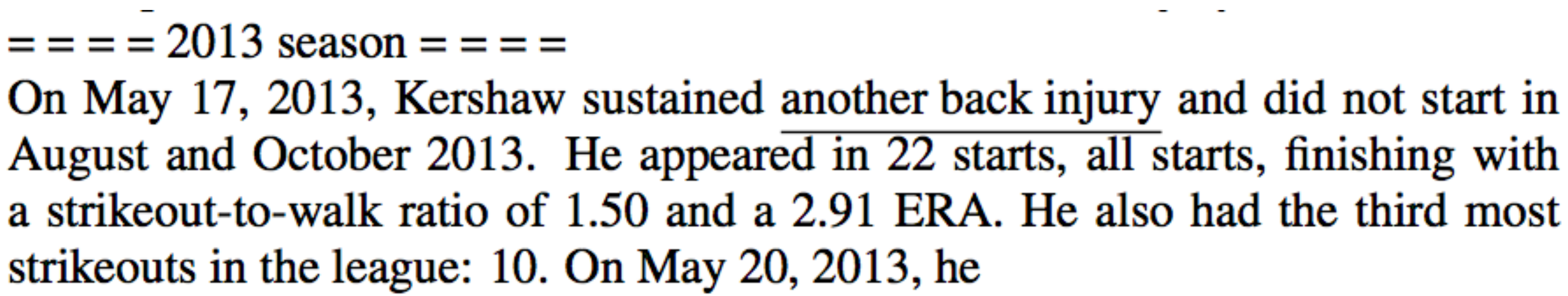
XL Generation:

, tying him with Luis Castillo for the third best start total in the league and trailing only Johnny Bench and Eddie Mathews (37). The Dodgers signed Kershaw to a one-year deal on July 14, 2011, with a player option for 2012, in a deal worth just under $ 300,000. The Dodgers entered the 2011 season as the major league leaders, and Kershaw had the best record in the National League in saves (39), games started (78) and walks (120); he also had the lowest win / loss ratio in the league (1.87), and he had the worst on-base percentage (.349). He had the best ERA (2.00), most wins (16), or most shutouts (29) in his career (25), while leading the league in losses per nine innings pitched (4.01). Following the 2011 season, the Dodgers traded Kershaw to the Kansas City Royals on July 29 for pitcher Javier Torres.

= = = Kansas City Royals = = =

= = = = 2012 season = = = =

During spring training, Kershaw played very well. He was selected to spring training as a relief pitcher for the Royals for the 2012 season. After an injury to closer Javier Vázquez, he was activated on April 29 to replace Matt Holliday in the Royals ’ starting rotation. In his only start with the Royals, on August 6, 2012, Kershaw struck out five batters in seven innings pitched to help the Royals



= = = = 2013 season = = = =

On May 17, 2013, Kershaw sustained another back injury and did not start in August and October 2013. He appeared in 22 starts, all starts, finishing with a strikeout-to-walk ratio of 1.50 and a 2.91 ERA. He also had the third most strikeouts in the league: 10. On May 20, 2013, he

Thank you