10807
Topics in Deep Learning

Russ Salakhutdinov

Machine Learning Department
rsalakhu@cs.cmu.edu
http://www.cs.cmu.edu/~rsalakhu/10807_2016/

Variational Autoencoders

Gaussian Policy: Continuous Actions

- Remember stochastic policy

$$\pi_\theta(s, a) = \mathbb{P}\left[a \mid s, \theta\right]$$

- In continuous action spaces, a Gaussian policy is natural
- Mean is a linear combination of state features

$$\mu(s) = \phi(s)^\top \theta$$

Nonlinear extension: replace $\phi(s)$ th a deep neural network with trainable weights w

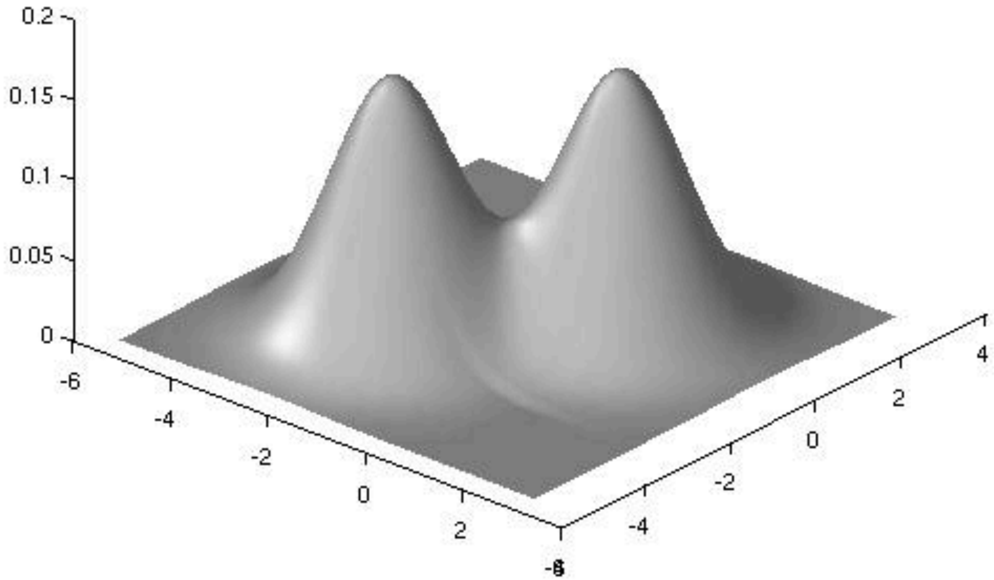
- Variance may be fixed σ_2, or can also parameterized
- Policy is Gaussian $a \sim \mathcal{N}(\mu(s), \sigma^2)$

Multimodal Outputs

- Remember stochastic policy

$$\pi_\theta(s, a) = \mathbb{P}\left[a \mid s, \theta\right]$$

- But what if stochastic policy has multiple modes?

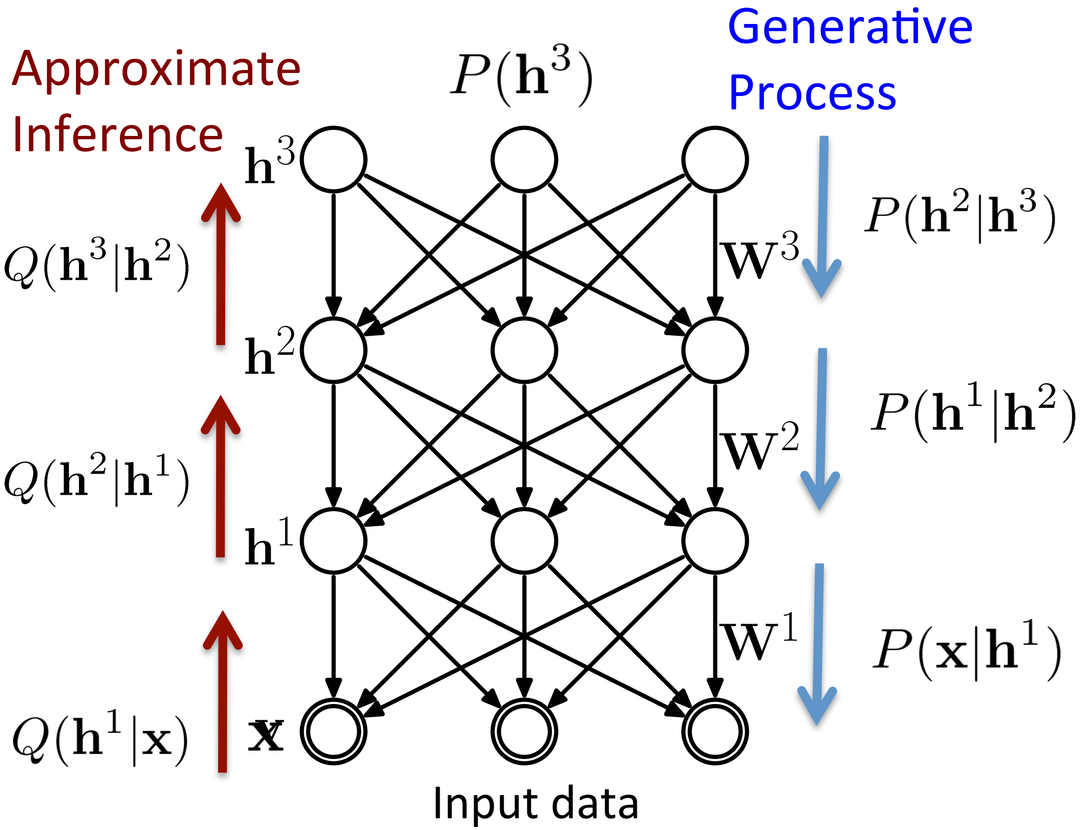


- Model-based RL: Dynamics of the environment can be multimodal.

$$P(s_{t+1}|a_t, s_t)$$

Helmholtz Machines

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



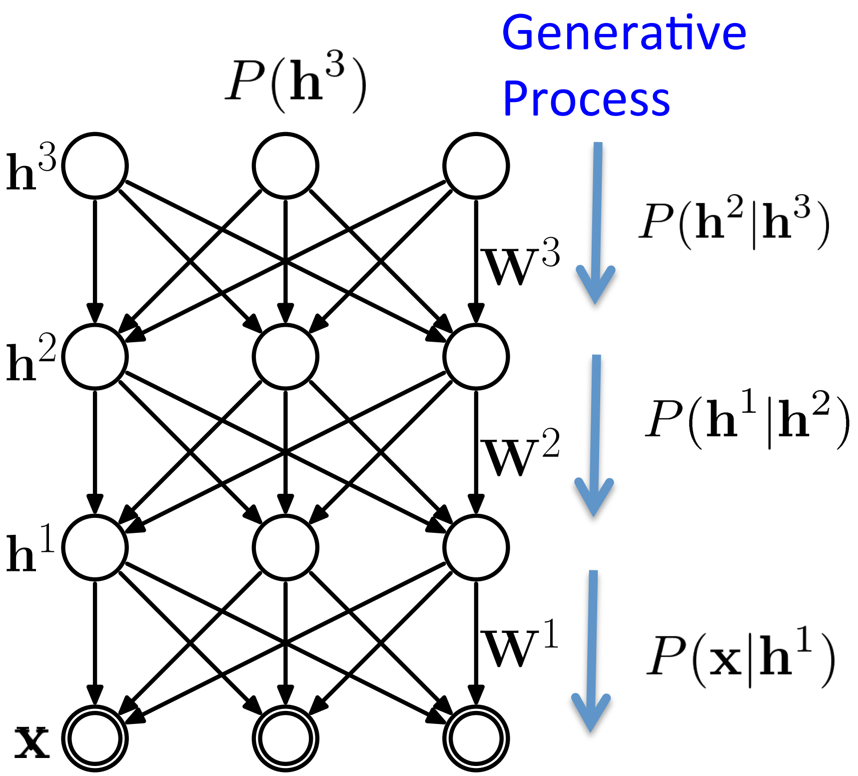
- Kingma & Welling, 2014
- Rezende, Mohamed, Daan, 2014
- Mnih & Gregor, 2014
- Bornschein & Bengio, 2015
- Tang & Salakhutdinov, 2013

Variational Autoencoders (VAEs)

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

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

Each term may denote a complicated nonlinear relationship



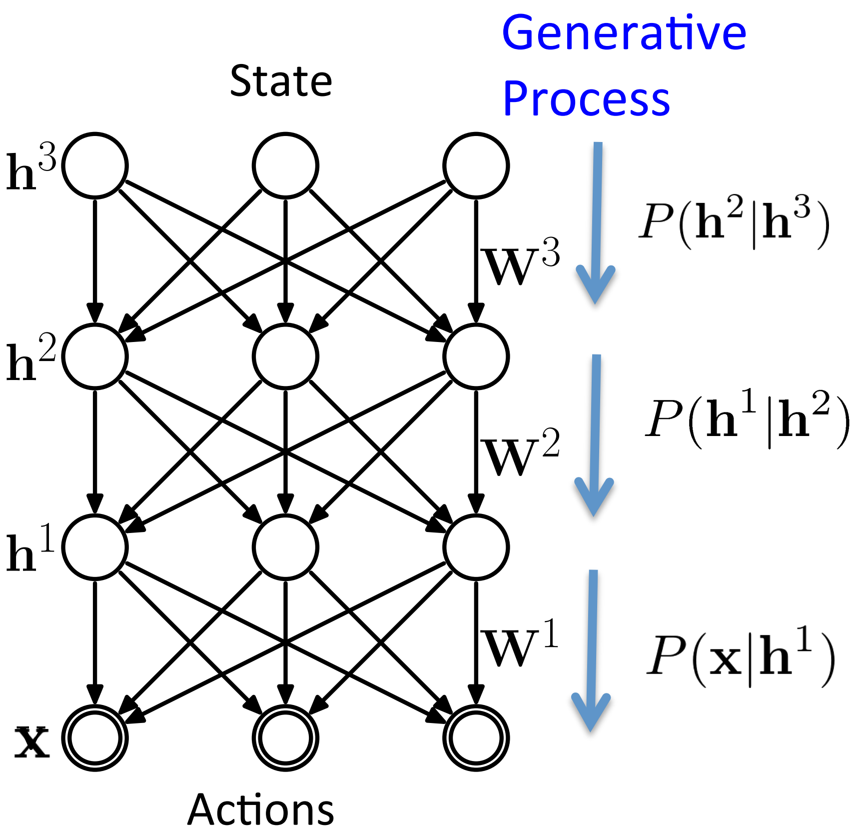
Input data

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

Variational Autoencoders (VAEs)

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

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



- Given state, we can generate a distribution over actions:

$$\pi_\theta(s, a) = \mathbb{P}\left[a \mid s, \theta\right]$$

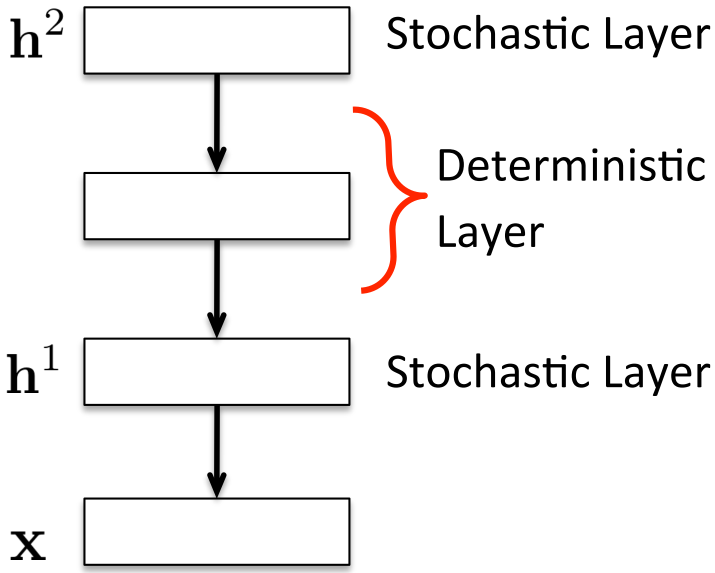
- Conditional VAE: neural networks with stochastic and deterministic layers

VAE: Example

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

$$p(\mathbf{x}|\boldsymbol{\theta}) = \sum_{\mathbf{h}^1,\mathbf{h}^2} p(\mathbf{h}^2|\boldsymbol{\theta})p(\mathbf{h}^1|\mathbf{h}^2,\boldsymbol{\theta})p(\mathbf{x}|\mathbf{h}^1,\boldsymbol{\theta})$$

This term denotes a one-layer neural net.



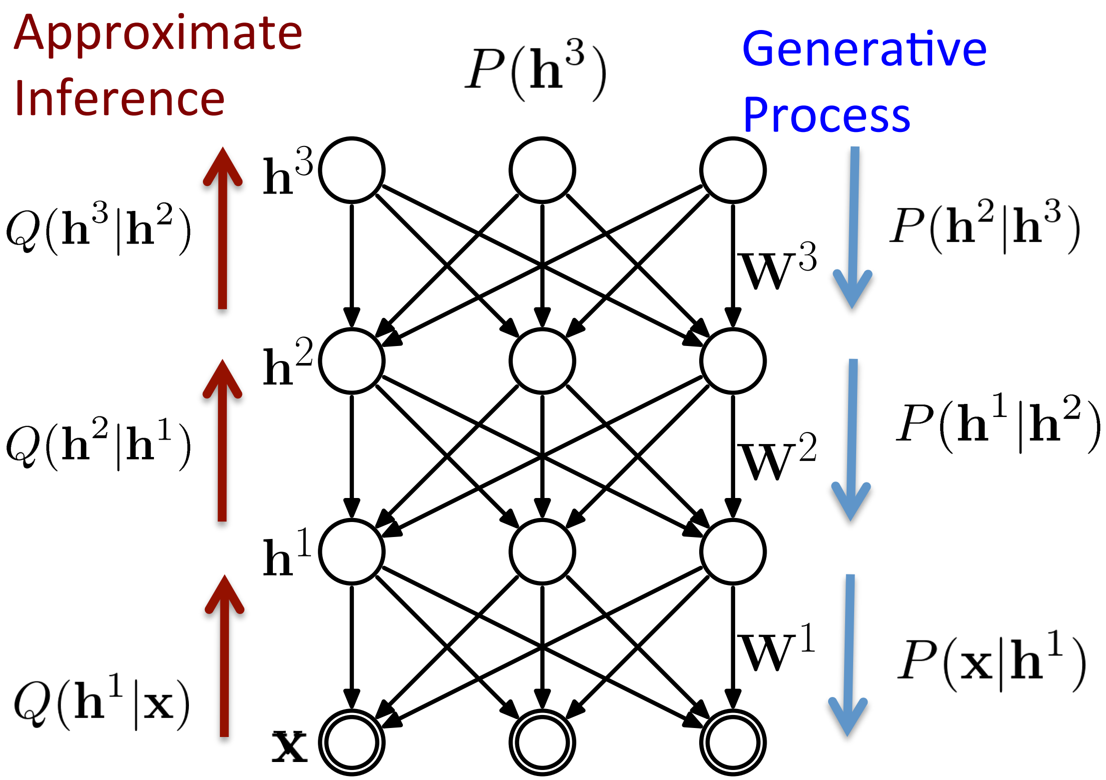
- $\boldsymbol{\theta}$ denotes parameters of VAE.
- L is the number of **stochastic** layers.
- Sampling and probability evaluation is tractable for each $p(\mathbf{h}^\ell|\mathbf{h}^{\ell+1})$.

Recognition Network

- The recognition model is defined in terms of an analogous factorization:

$$q(\mathbf{h}|\mathbf{x}, \boldsymbol{\theta}) = q(\mathbf{h}^1|\mathbf{x}, \boldsymbol{\theta})q(\mathbf{h}^2|\mathbf{h}^1, \boldsymbol{\theta}) \cdots q(\mathbf{h}^L|\mathbf{h}^{L-1}, \boldsymbol{\theta})$$

Each term may denote a complicated nonlinear relationship



Input data

- We assume that $\mathbf{h}^L \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$
- The conditionals:

  $p(\mathbf{h}^\ell|\ \mathbf{h}^{\ell+1})$

  $q(\mathbf{h}^\ell|\mathbf{h}^{\ell-1})$

  are Gaussians with diagonal covariances

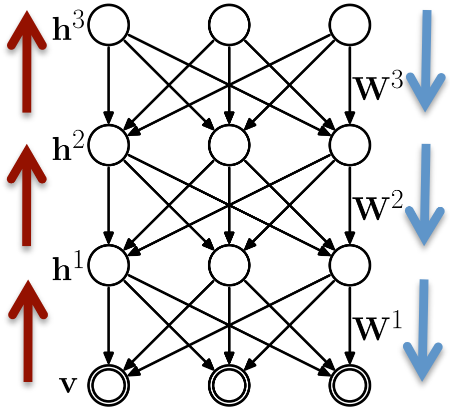
Variational Bound

- The VAE is trained to maximize the variational lower bound:

$$\log p(\mathbf{x}) = \log \mathbb{E}_{q(\mathbf{h}|\mathbf{x})}\left[\frac{p(\mathbf{x},\mathbf{h})}{q(\mathbf{h}|\mathbf{x})}\right] \geq \mathbb{E}_{q(\mathbf{h}|\mathbf{x})}\left[\log \frac{p(\mathbf{x},\mathbf{h})}{q(\mathbf{h}|\mathbf{x})}\right] = \mathcal{L}(\mathbf{x})$$

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

  - Trading off the data log-likelihood and the KL divergence from the true posterior.



Input data

- Hard to optimize the variational bound with respect to the recognition network (high-variance).

- Key idea of Kingma and Welling is to use reparameterization trick.

Reparameterization Trick

- Assume that the recognition distribution is Gaussian:

$$q(\mathbf{h}^\ell|\mathbf{h}^{\ell-1},\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}(\mathbf{h}^{\ell-1},\boldsymbol{\theta}),\boldsymbol{\Sigma}(\mathbf{h}^{\ell-1},\boldsymbol{\theta}))$$

  with mean and covariance computed from the state of the hidden units at the previous layer.

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

$$\boldsymbol{\epsilon}^\ell \sim \mathcal{N}(\mathbf{0},\boldsymbol{I})$$

$$\mathbf{h}^\ell\left(\boldsymbol{\epsilon}^\ell,\mathbf{h}^{\ell-1},\boldsymbol{\theta}\right) = \boldsymbol{\Sigma}(\mathbf{h}^{\ell-1},\boldsymbol{\theta})^{1/2}\boldsymbol{\epsilon}^\ell + \boldsymbol{\mu}(\mathbf{h}^{\ell-1},\boldsymbol{\theta})$$

Reparameterization Trick

- Assume that the recognition distribution is Gaussian:

$$q(\mathbf{h}^\ell|\mathbf{h}^{\ell-1},\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{\mu}(\mathbf{h}^{\ell-1},\boldsymbol{\theta}), \boldsymbol{\Sigma}(\mathbf{h}^{\ell-1},\boldsymbol{\theta}))$$

- Or

$$\boldsymbol{\epsilon}^\ell \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$

$$\mathbf{h}^\ell\left(\boldsymbol{\epsilon}^\ell, \mathbf{h}^{\ell-1}, \boldsymbol{\theta}\right) = \boldsymbol{\Sigma}(\mathbf{h}^{\ell-1},\boldsymbol{\theta})^{1/2}\boldsymbol{\epsilon}^\ell + \boldsymbol{\mu}(\mathbf{h}^{\ell-1},\boldsymbol{\theta})$$

- The recognition distribution $q(\mathbf{h}^\ell|\mathbf{h}^{\ell-1},\boldsymbol{\theta})$ can be expressed in terms of a deterministic mapping:

$$\underbrace{\mathbf{h}\left(\boldsymbol{\epsilon}, \mathbf{x}, \boldsymbol{\theta}\right)}_{\text{Deterministic Encoder}}, \quad \text{with} \quad \boldsymbol{\epsilon} = \underbrace{(\boldsymbol{\epsilon}^1, \ldots, \boldsymbol{\epsilon}^L)}_{\text{Distribution of } \boldsymbol{\epsilon} \text{ does not depend on } \boldsymbol{\theta}}$$

Computing the Gradients

- The gradient w.r.t the parameters: both recognition and generative:

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{h} \sim q(\mathbf{h}|\mathbf{x},\boldsymbol{\theta})} \left[\log \frac{p(\mathbf{x},\mathbf{h}|\boldsymbol{\theta})}{q(\mathbf{h}|\mathbf{x},\boldsymbol{\theta})} \right]$$

$$= \nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{\epsilon}^1,\ldots,\boldsymbol{\epsilon}^L \sim \mathcal{N}(\mathbf{0},\boldsymbol{I})} \left[\log \frac{p(\mathbf{x},\mathbf{h}(\boldsymbol{\epsilon},\mathbf{x},\boldsymbol{\theta})|\boldsymbol{\theta})}{q(\mathbf{h}(\boldsymbol{\epsilon},\mathbf{x},\boldsymbol{\theta})|\mathbf{x},\boldsymbol{\theta})} \right]$$

$$= \mathbb{E}_{\boldsymbol{\epsilon}^1,\ldots,\boldsymbol{\epsilon}^L \sim \mathcal{N}(\mathbf{0},\boldsymbol{I})} \left[\nabla_{\boldsymbol{\theta}} \log \frac{p(\mathbf{x},\mathbf{h}(\boldsymbol{\epsilon},\mathbf{x},\boldsymbol{\theta})|\boldsymbol{\theta})}{q(\mathbf{h}(\boldsymbol{\epsilon},\mathbf{x},\boldsymbol{\theta})|\mathbf{x},\boldsymbol{\theta})} \right]$$

Gradients can be computed by backprop

The mapping **h** is a deterministic neural net for fixed $\boldsymbol{\epsilon}$.

Computing the Gradients

- The gradient w.r.t the parameters: recognition and generative:

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{\mathbf{h} \sim q(\mathbf{h}|\mathbf{x},\boldsymbol{\theta})} \left[\log \frac{p(\mathbf{x},\mathbf{h}|\boldsymbol{\theta})}{q(\mathbf{h}|\mathbf{x},\boldsymbol{\theta})} \right] = \mathbb{E}_{\boldsymbol{\epsilon}^1,\ldots,\boldsymbol{\epsilon}^L \sim \mathcal{N}(\mathbf{0},\boldsymbol{I})} \left[\nabla_{\boldsymbol{\theta}} \log \frac{p(\mathbf{x},\mathbf{h}(\boldsymbol{\epsilon},\mathbf{x},\boldsymbol{\theta})|\boldsymbol{\theta})}{q(\mathbf{h}(\boldsymbol{\epsilon},\mathbf{x},\boldsymbol{\theta})|\mathbf{x},\boldsymbol{\theta})} \right]$$

- Approximate expectation by generating k samples from $\boldsymbol{\epsilon}$:

$$\frac{1}{k} \sum_{i=1}^{k} \nabla_{\boldsymbol{\theta}} \log w\left(\mathbf{x}, \mathbf{h}(\boldsymbol{\epsilon}_i, \mathbf{x}, \boldsymbol{\theta}), \boldsymbol{\theta}\right)$$

where we defined unnormalized importance weights:

$$w(\mathbf{x}, \mathbf{h}, \boldsymbol{\theta}) \;=\; p(\mathbf{x}, \mathbf{h}|\boldsymbol{\theta}) / q(\mathbf{h}|\mathbf{x}, \boldsymbol{\theta})$$

- VAE update: Low variance as it uses the log-likelihood gradients with respect to the latent variables.

VAE: Assumptions

- Remember the variational bound:

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

- The variational assumptions must be approximately satisfied.

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

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

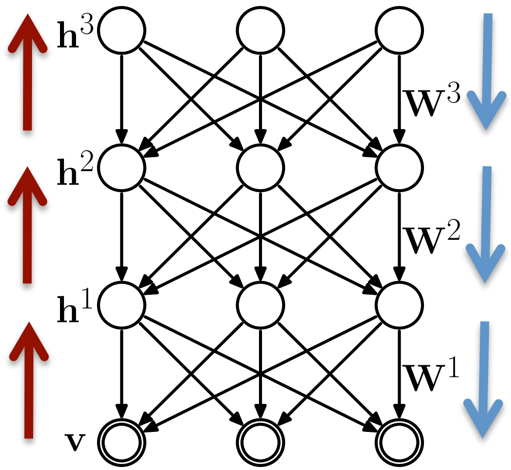
Importance Weighted Autoencoders

- Consider the following k-sample importance weighting of the log-likelihood:

$$\mathcal{L}_k(\mathbf{x}) = \mathbb{E}_{\mathbf{h}_1,\ldots,\mathbf{h}_k \sim q(\mathbf{h}|\mathbf{x})}\left[\log \frac{1}{k}\sum_{i=1}^{k} \frac{p(\mathbf{x},\mathbf{h}_i)}{q(\mathbf{h}_i|\mathbf{x})}\right]$$

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

unnormalized importance weights



Input data

where $\mathbf{h}_1, \ldots, \mathbf{h}_k$ are sampled from the recognition network.

Importance Weighted Autoencoders

- Consider the following k-sample importance weighting of the log-likelihood:

$$\mathcal{L}_k(\mathbf{x}) = \mathbb{E}_{\mathbf{h}_1,\ldots,\mathbf{h}_k \sim q(\mathbf{h}|\mathbf{x})} \left[\log \frac{1}{k} \sum_{i=1}^{k} \frac{p(\mathbf{x}, \mathbf{h}_i)}{q(\mathbf{h}_i|\mathbf{x})} \right]$$

- This is a lower bound on the marginal log-likelihood:

$$\mathcal{L}_k(\mathbf{x}) = \mathbb{E}\left[\log \frac{1}{k} \sum_{i=1}^{k} w_i\right] \leq \log \mathbb{E}\left[\frac{1}{k} \sum_{i=1}^{k} w_i\right] = \log p(\mathbf{x})$$

- Special Case of k=1: Same as standard VAE objective.

- Using more samples → Improves the tightness of the bound.

IWAEs vs. VAEs

- Draw k-samples form the recognition network $q(\mathbf{h}|\mathbf{x})$
  - or k-sets of auxiliary variables $\boldsymbol{\epsilon}$.
- Obtain the following Monte Carlo estimate of the gradient:

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}_k(\mathbf{x}) \approx \sum_{i=1}^{k} \widetilde{w}_i \boxed{\nabla_{\boldsymbol{\theta}} \log w(\mathbf{x}, \mathbf{h}(\boldsymbol{\epsilon}_i, \mathbf{x}, \boldsymbol{\theta}), \boldsymbol{\theta})}$$

- Compare this to the VAE's estimate of the gradient:

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}(\mathbf{x}) \approx \frac{1}{k} \sum_{i=1}^{k} \boxed{\nabla_{\boldsymbol{\theta}} \log w(\mathbf{x}, \mathbf{h}(\boldsymbol{\epsilon}_i, \mathbf{x}, \boldsymbol{\theta}), \boldsymbol{\theta})}$$

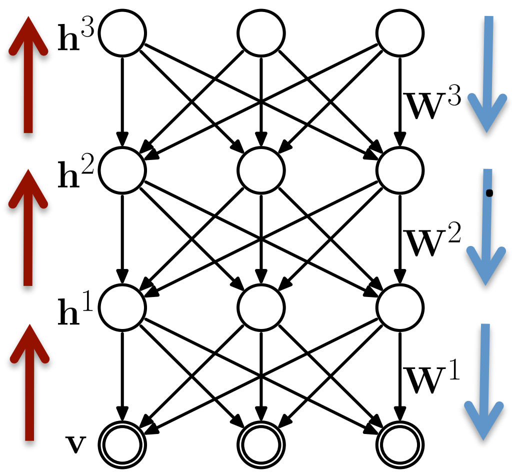
VAE: Intuition

- The gradient of the log weights decomposes:

$$\nabla_{\boldsymbol{\theta}} \log w(\mathbf{x}, \mathbf{h}(\boldsymbol{\epsilon}_i, \mathbf{x}, \boldsymbol{\theta}), \boldsymbol{\theta})$$

$$= \nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}, \underbrace{\mathbf{h}(\boldsymbol{\epsilon}_i, \mathbf{x}, \boldsymbol{\theta})}_{\text{Deterministic Encoder}} | \boldsymbol{\theta}) - \log q(\mathbf{h}(\boldsymbol{\epsilon}_i, \mathbf{x}, \boldsymbol{\theta}) | \mathbf{x}, \boldsymbol{\theta})$$

Deterministic decoder



Input data

First term:

- Decoder: encourages the generative model to assign high probability to each $\mathbf{h}^l | \mathbf{h}^{l+1}$.
- Encoder: encourages the recognition net to adjust its latent states h so that the generative network makes better predictions.

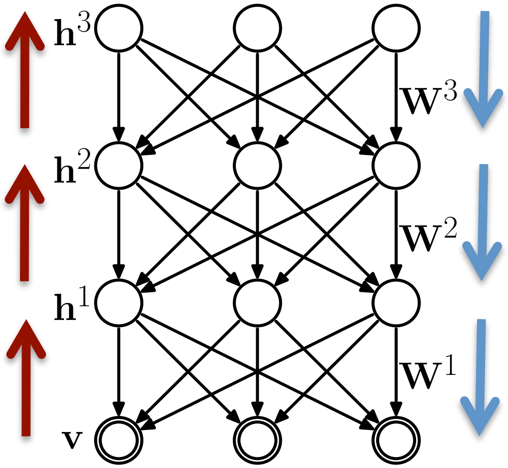
VAE: Intuition

- The gradient of the log weights decomposes:

$$\nabla_{\boldsymbol{\theta}} \log w(\mathbf{x}, \mathbf{h}(\boldsymbol{\epsilon}_i, \mathbf{x}, \boldsymbol{\theta}), \boldsymbol{\theta})$$

$$= \nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}, \underbrace{\mathbf{h}(\boldsymbol{\epsilon}_i, \mathbf{x}, \boldsymbol{\theta})}_{\text{Deterministic Encoder}} | \boldsymbol{\theta}) - \log q(\mathbf{h}(\boldsymbol{\epsilon}_i, \mathbf{x}, \boldsymbol{\theta}) | \mathbf{x}, \boldsymbol{\theta})$$

Deterministic decoder



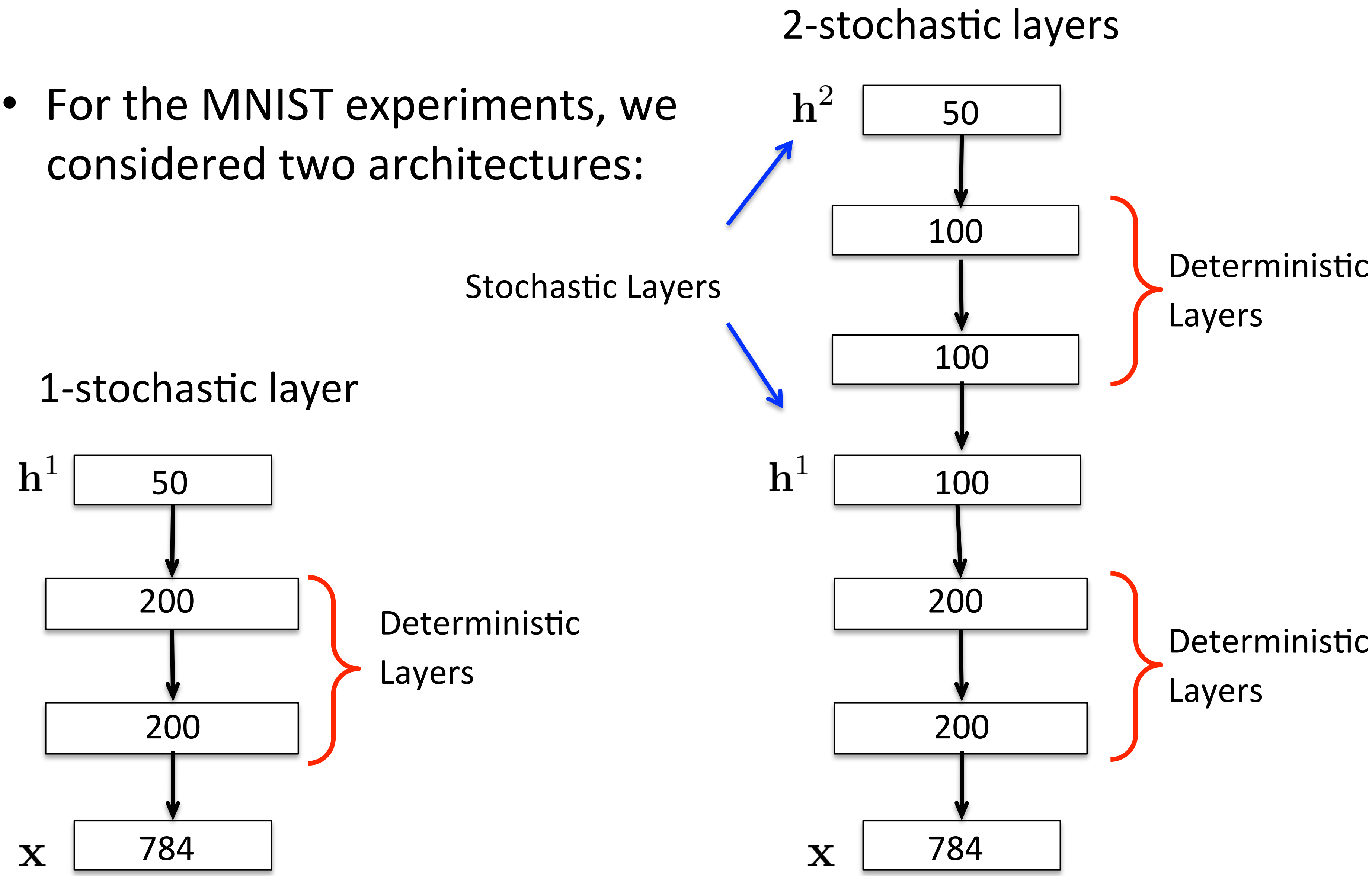
Input data

Second term:

- Encoder: encourages the recognition network to have a spread-out distribution over predictions.

Two Architectures

- For the MNIST experiments, we considered two architectures:



MNIST Results

		MNIST			
		VAE		IWAE	
# stoch. layers	k	NLL	active units	NLL	active units
1	1	86.76	19	86.76	19
	5	86.47	20	85.54	22
	50	86.35	20	84.78	25

MNIST Results

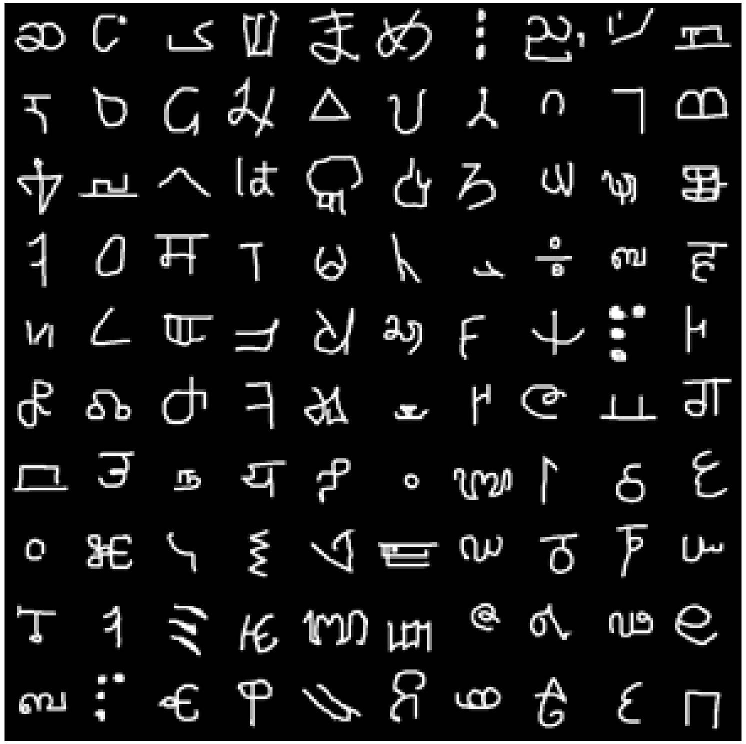
# stoch. layers	k	MNIST VAE NLL	MNIST VAE active units	MNIST IWAE NLL	MNIST IWAE active units
1	1	86.76	19	86.76	19
	5	86.47	20	85.54	22
	50	86.35	20	84.78	25
2	1	85.33	16+5	85.33	16+5
	5	85.01	17+5	83.89	21+5
	50	84.78	17+5	82.90	26+7

Good Generative Model?

Handwritten Characters

Good Generative Model?

Handwritten Characters



Good Generative Model?

Handwritten Characters

Simulated

Real Data

Good Generative Model?

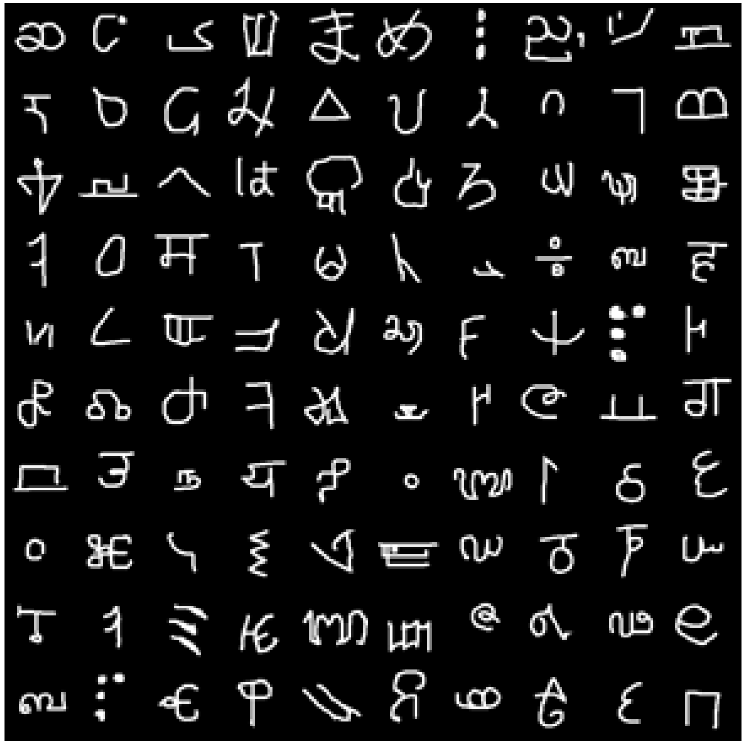
Handwritten Characters

Real Data

Simulated

Good Generative Model?

Handwritten Characters



Motivating Example

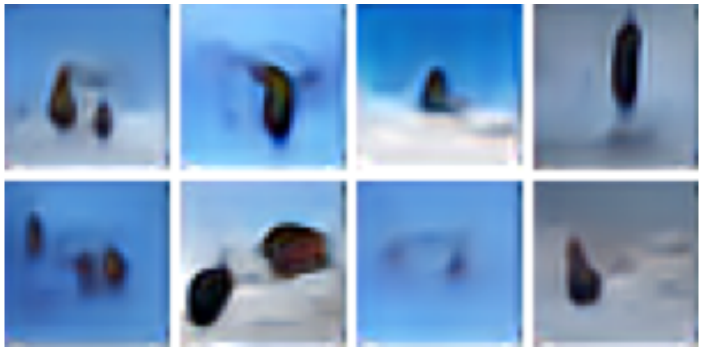
(Mansimov, Parisotto, Ba, Salakhutdinov, 2015)

- Can we generate images from natural language descriptions?

A **stop sign** is flying in blue skies

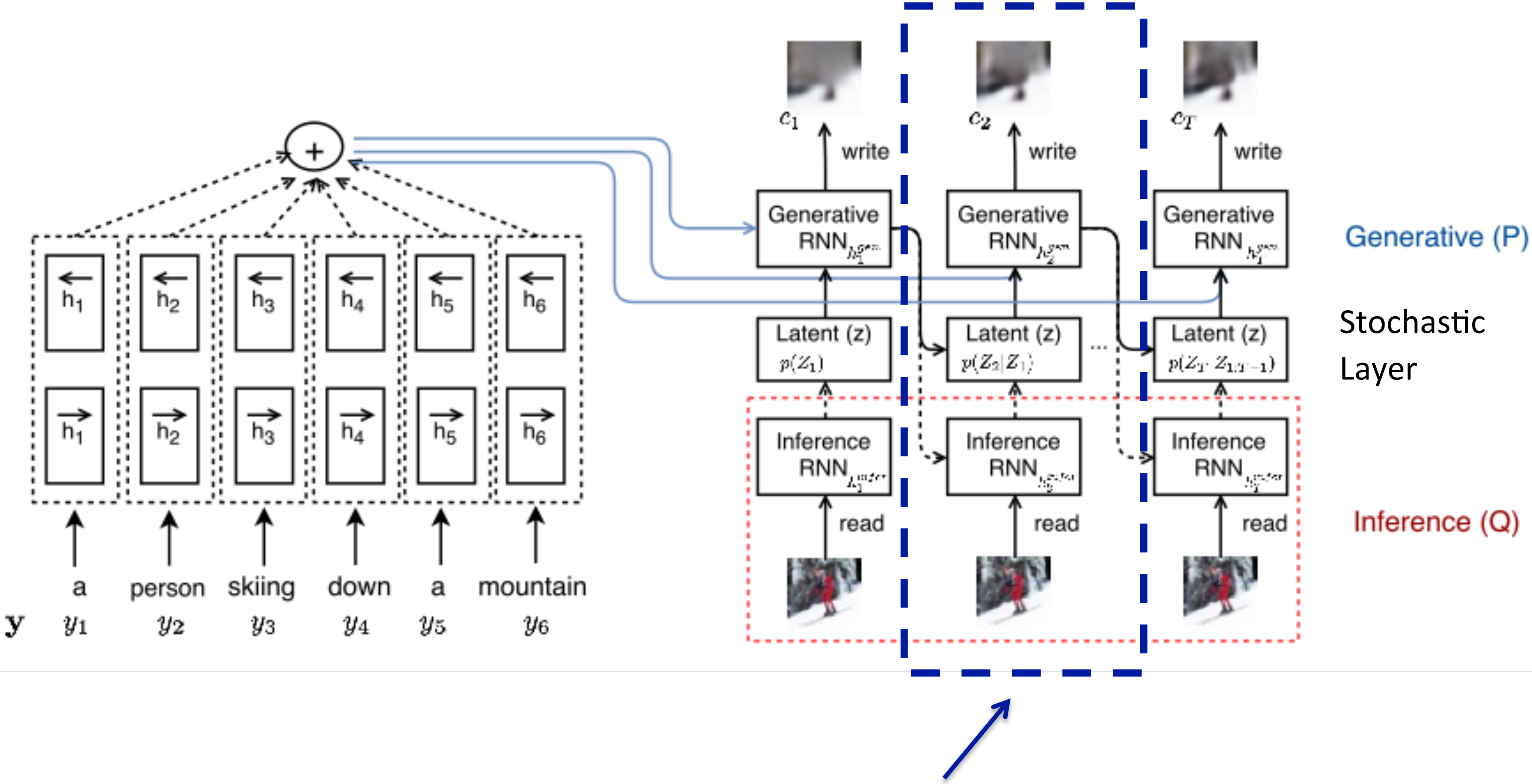
A **pale yellow school bus** is flying in blue skies

A **herd of elephants** is flying in blue skies



A **large commercial airplane** is flying in blue skies

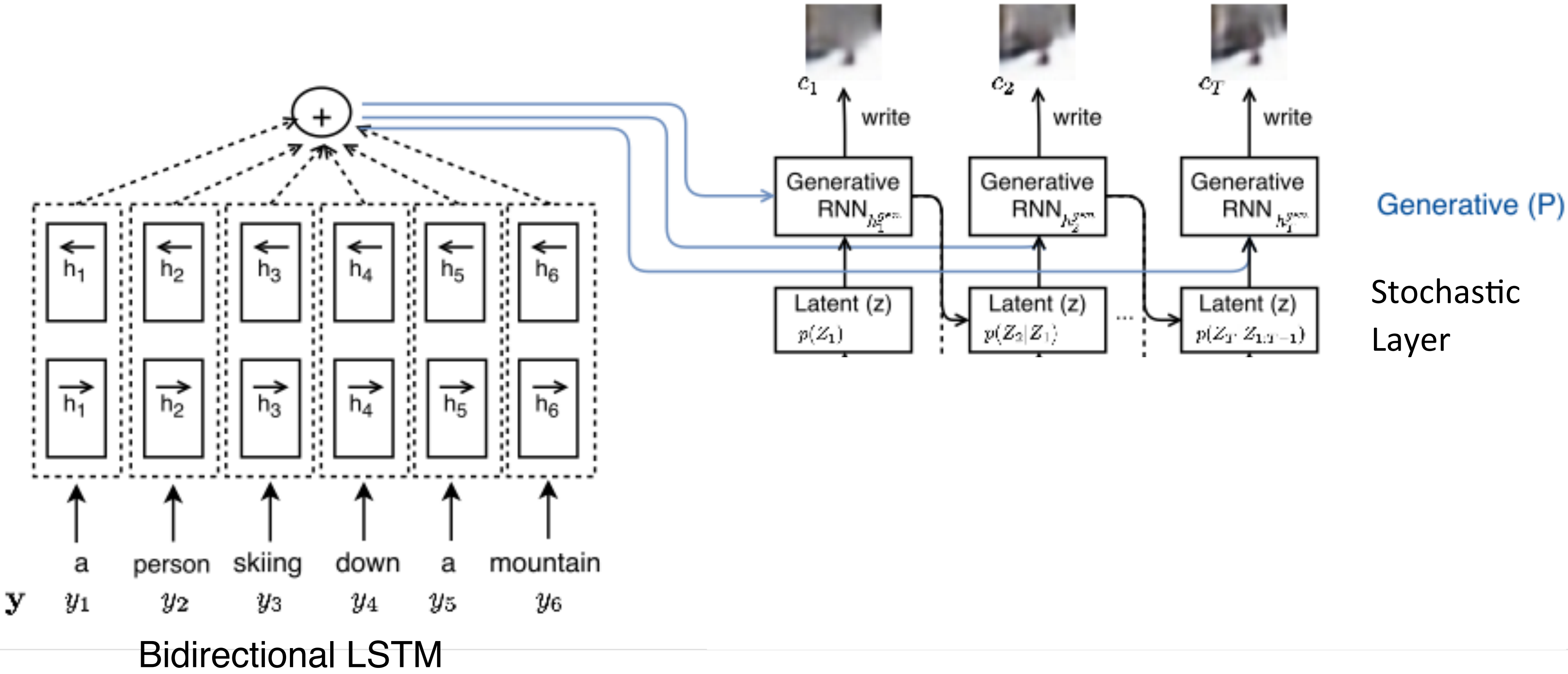
Overall Model



Variational Autoecnoder

Overall Model

(Mansimov, Parisotto, Ba, Salakhutdinov, 2015)

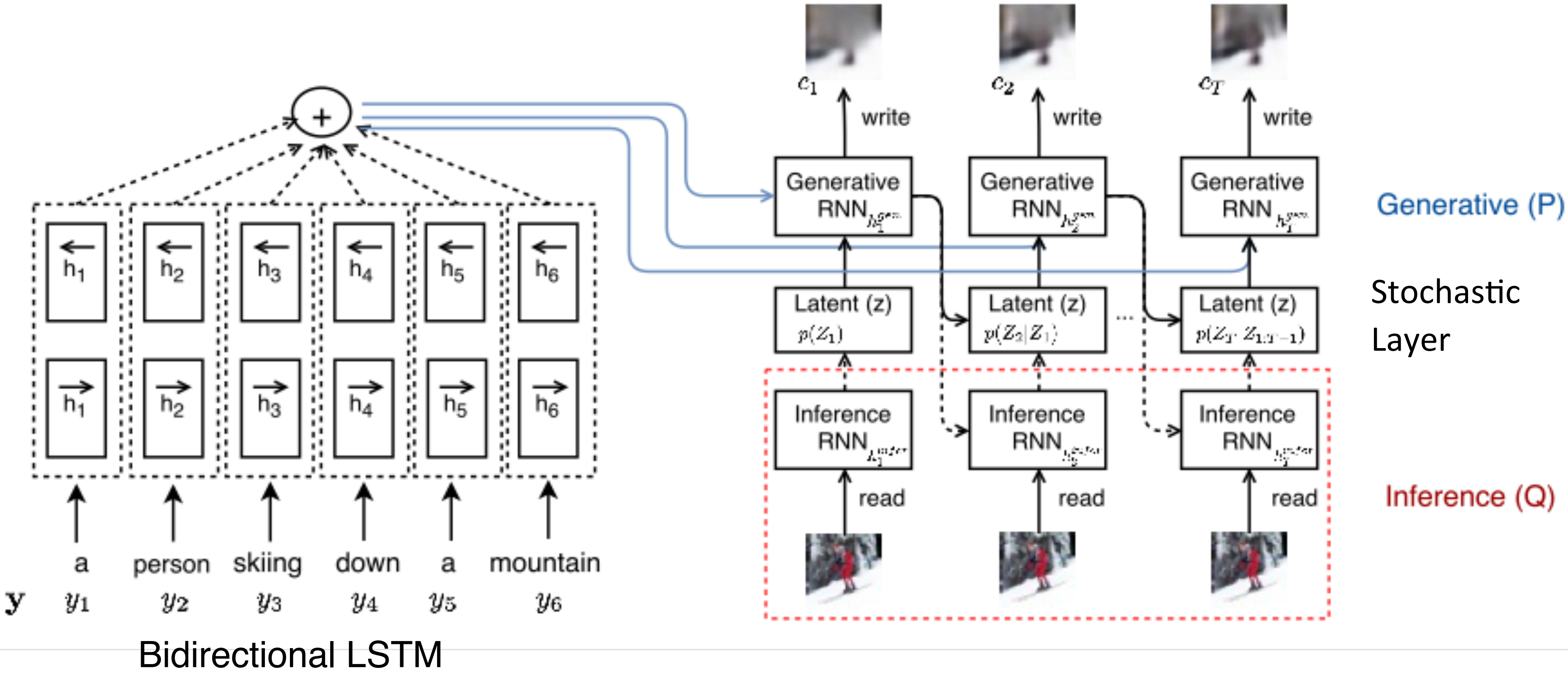


- Generative Model: Stochastic Recurrent Network, chained sequence of Variational Autoencoders, with a single stochastic layer.

Gregor et. al. 2015

Overall Model

(Mansimov, Parisotto, Ba, Salakhutdinov, 2015)

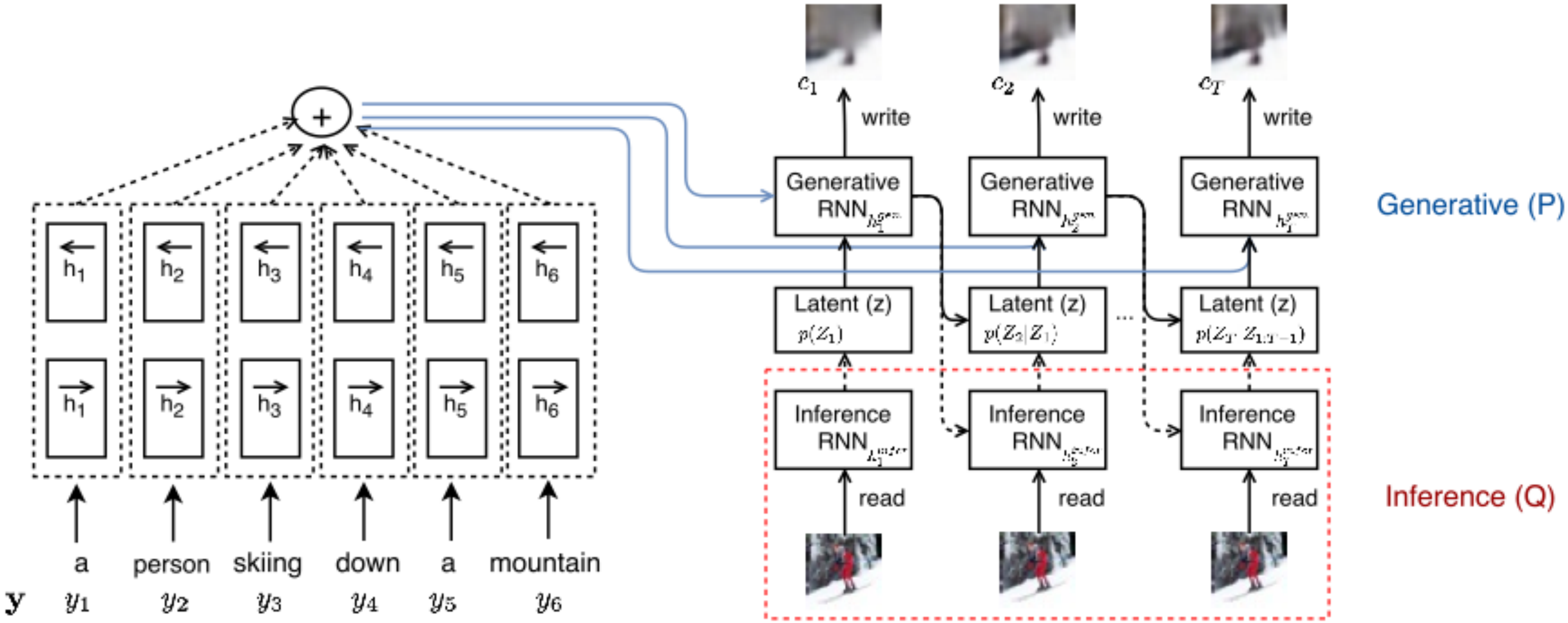


- Generative Model: Stochastic Recurrent Network, chained sequence of Variational Autoencoders, with a single stochastic layer.

- Recognition Model: Deterministic Recurrent Network.

Gregor et. al. 2015

Learning

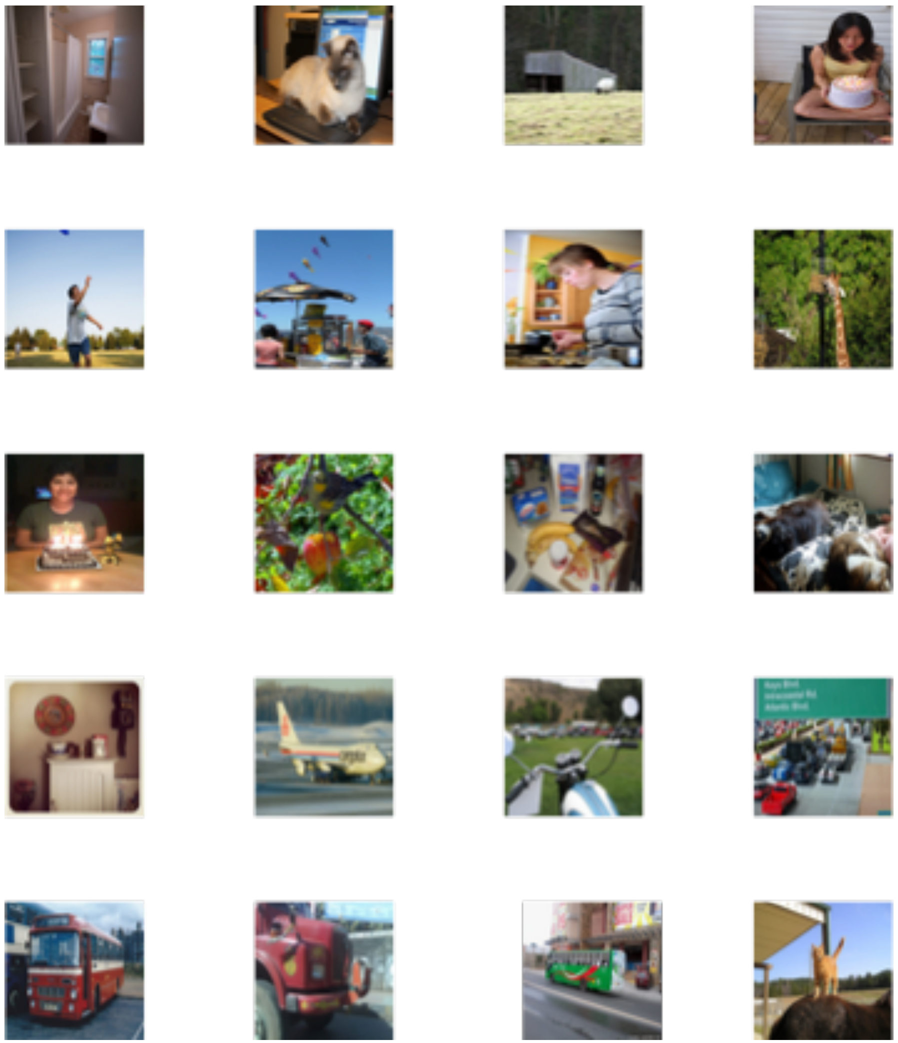


- Maximize the variational lower bound on the marginal log-likelihood of the correct image **x** given the caption **y**:

$$\mathcal{L} = \sum_{Z} Q(Z|\mathbf{x},\mathbf{y}) \log P(\mathbf{x}|Z,\mathbf{y}) - \mathrm{D}_{KL}\left(Q(Z|\mathbf{x},\mathbf{y})||P(Z|\mathbf{y})\right)$$

$$\leq \log P(\mathbf{x}|\mathbf{y})$$

MS COCO Dataset



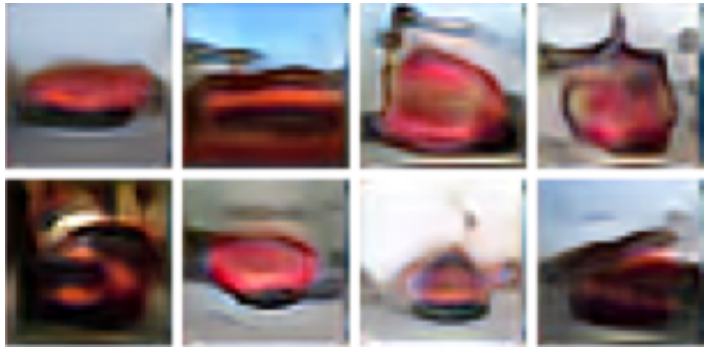
- Contains 83K images.
- Each image contains 5 captions.
- Standard benchmark dataset for many of the recent image captioning systems.

Lin et. al. 2014

Flipping Colors

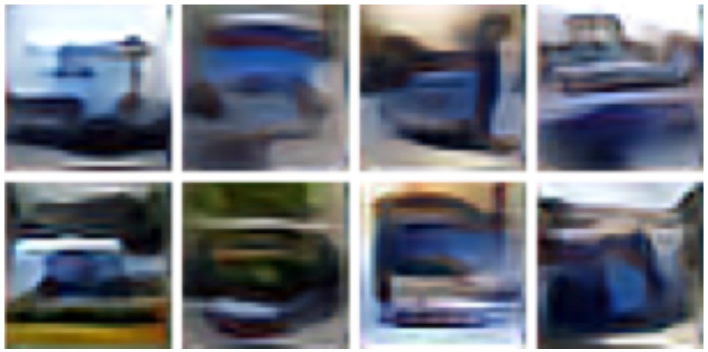
A **yellow school bus** parked in the parking lot

A **red school bus** parked in the parking lot



A **green school bus** parked in the parking lot

A **blue school bus** parked in the parking lot



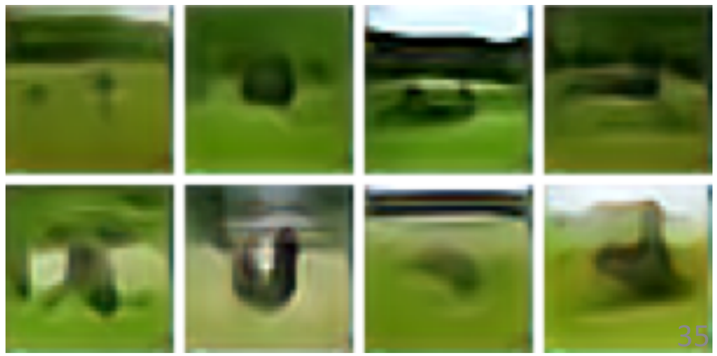
Flipping Backgrounds

A very large commercial plane flying **in clear skies**.

A very large commercial plane flying **in rainy skies**.

A herd of elephants walking across a **dry grass field**.

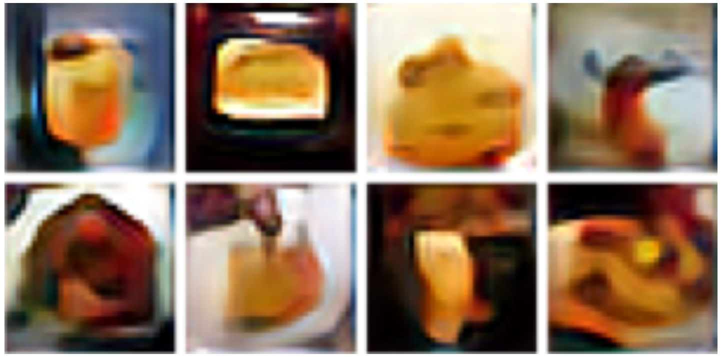
A herd of elephants walking across a **green grass field**.



Flipping Objects

The decadent chocolate desert is on the table.

A bowl of bananas is on the table..



A vintage photo of **a cat**.

A vintage photo of **a dog**.

Qualitative Comparison

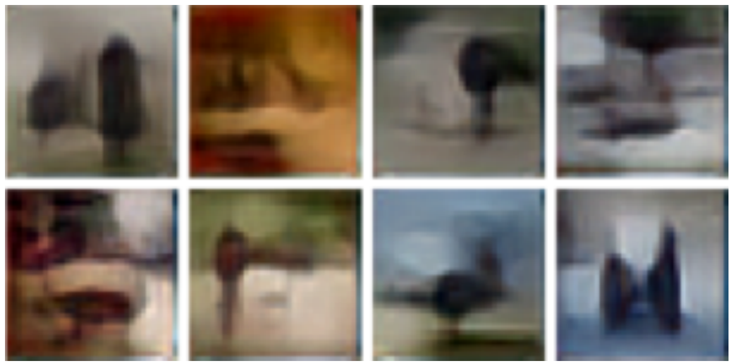
A group of people walk on a beach with surf boards

Our Model

LAPGAN (Denton et. al. 2015)

Conv-Deconv VAE

Fully Connected VAE



Novel Scene Compositions

A toilet seat sits open in the bathroom

A toilet seat sits open in the grass field

Ask Google?