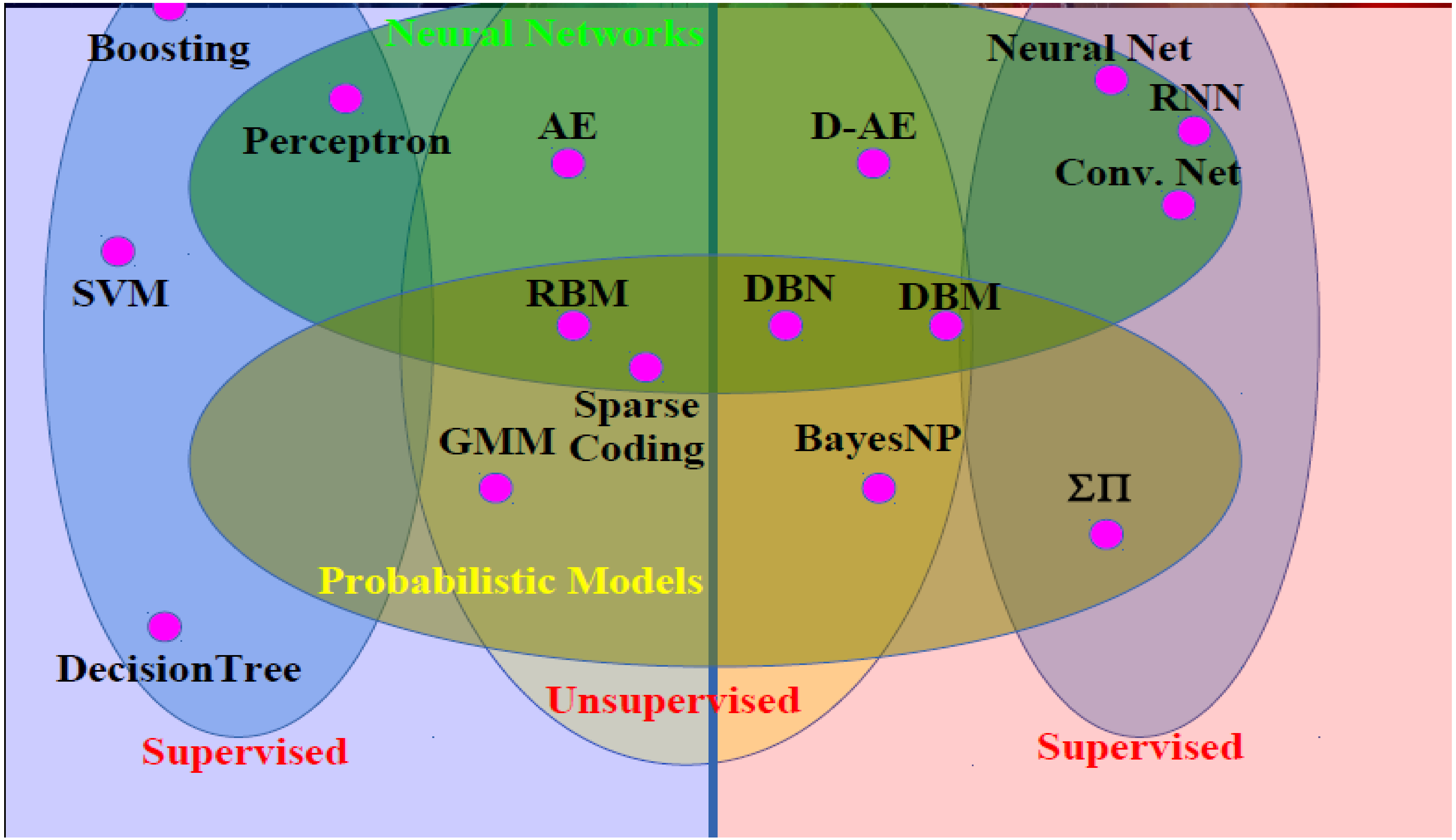
Marriage of Graphic Model and Neural Model

Dong Wang

2016/5/23

Content

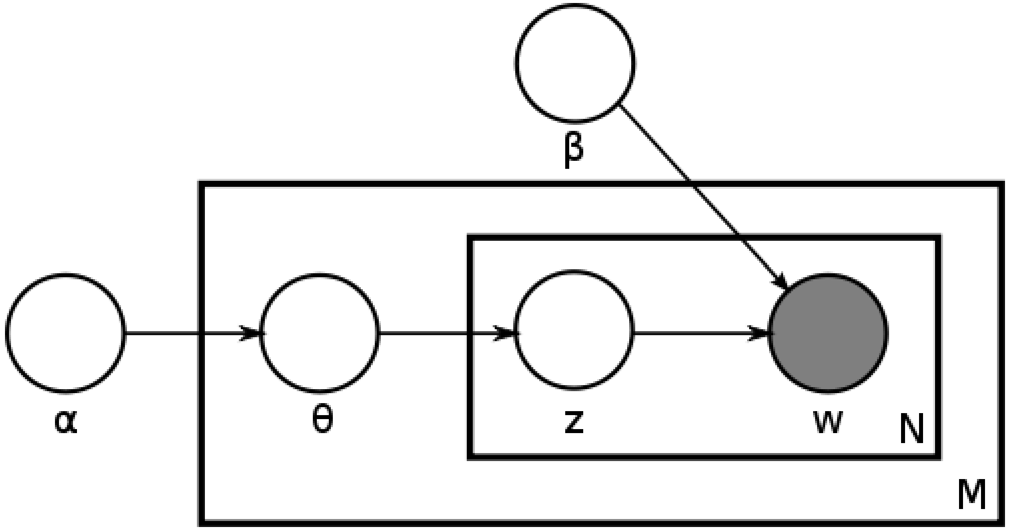
- What are they?
- Marriage 1: Variantional AE
- Marriage 2: Denoise AE

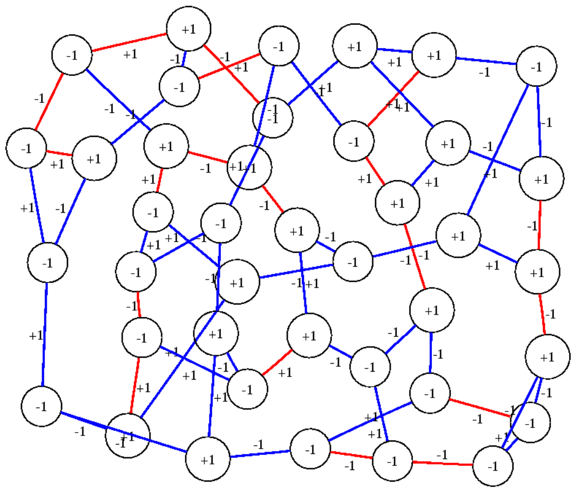


Yann LeCun, Marc'Aurelio Ranzato, Deep Learning Tutorial, ICML, Atlanta, 2013-06-16

Graphic models

- G=(V,E) represent joint probabilities of V, with conditionals or potentials represented by E
- Probabilistic variables
- Probabilistic inference



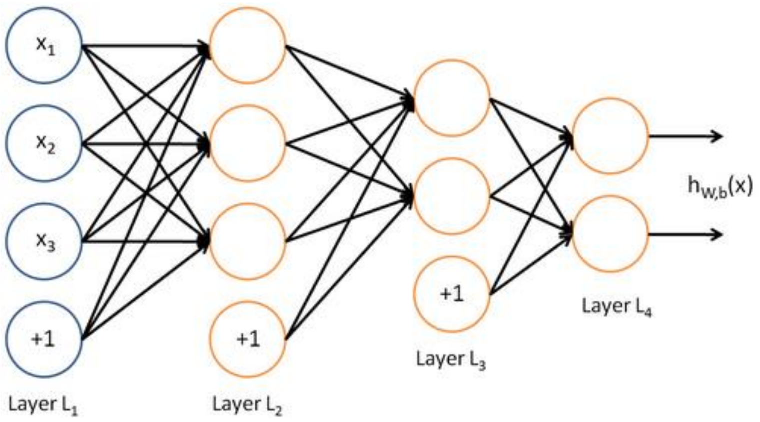


Graphical models and variational methods: Message-passing and relaxations , ICML-2008 Tutorial

http://www.eecs.berkeley.edu/~wainwrig/icml08/tutorial_icml08.html

Neural models

- G=(V,E) represents deterministic inference
- Probabilistic interpolation: Gaussian, Binomial, or MDN.
- Some ‘randomness’ on input, label, hidden units



Respective cons and pros

- Graphical model
  - Clear definition of facts and their relations
  - Easy to grow
  - Difficult in inference
- Neural models
  - Simple and Homogeneous units
  - Quick inference
  - Difficulty in training
  - Less probabilistic

Some models are in both...

- RBM, DBN, DBM, SGN...
- Clear probabilistic interpolation
- Homogeneous units

Content

- What are they?
- Marriage 1: Variantional AE
- Marriage 2: Denoise AE

How to marriage them in more depth?

- For Bayesian models, hope simpler inference
- For neural models, hope more randomness
- These two directions seem prefer the same architecture: stochastic NN.

Variational Bayesian

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) = D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}^{(i)})) + \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)})$$

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) \geq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)}) = \boxed{\mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}} \left[-\log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) + \boxed{\log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})} \right]$$

- Mean-field variational approximation
- q(z|x)=q(z1|x)q(z2|x)...

Variational Bayesian with Auto-encoder

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) = D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}^{(i)})) + \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}; \mathbf{x}^{(i)})$$

$$\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) \geq \mathcal{L}(\boldsymbol{\theta}, \phi; \mathbf{x}^{(i)}) = \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}\left[-\log q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) + \log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})\right]$$

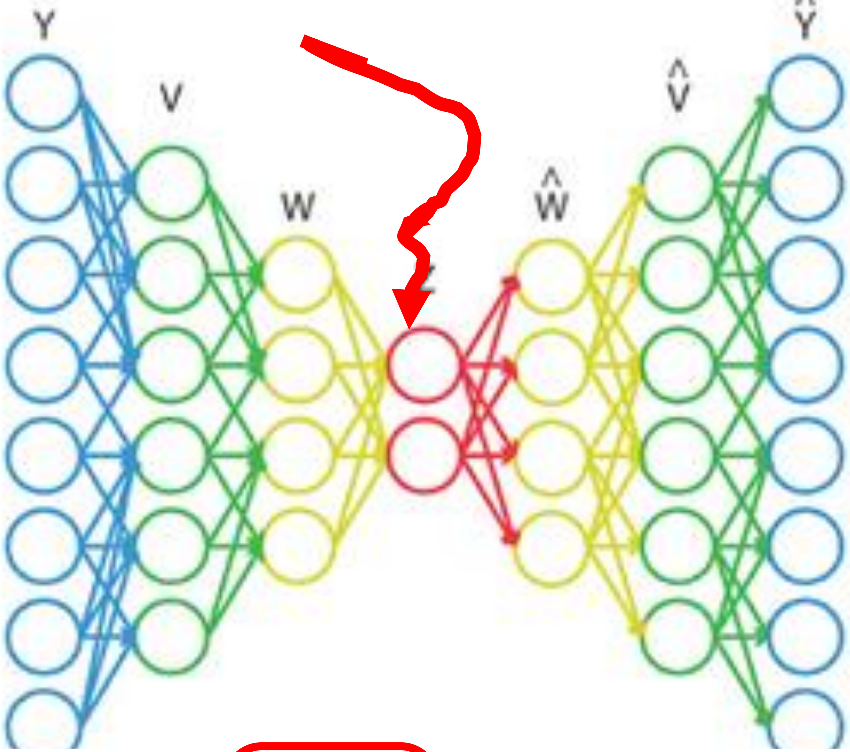
$$\mathcal{L}(\boldsymbol{\theta}, \phi; \mathbf{x}^{(i)}) = -D_{KL}(\boxed{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})}||p_{\boldsymbol{\theta}}(\mathbf{z})) + \mathbb{E}_{q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})}\left[\boxed{\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z})}\right]$$

$$\boxed{\widetilde{\mathbf{z}} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \mathbf{x})} \quad \text{with} \quad \boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$$

- Kingma, D. P. and Welling, M. (2014). Auto-encoding variational bayes. In ICLR.
- Danilo J. Rezende, Shakir Mohamed, Daan Wierstra, Stochastic Backpropagation and Approximate Inference in Deep Generative Models, ICMS 2014.

Variational Auto-encoder

$$\widetilde{\mathbf{z}} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \mathbf{x})$$



$$\boxed{\widetilde{\mathbf{z}} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}, \mathbf{x})} \quad \text{with} \quad \boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$$

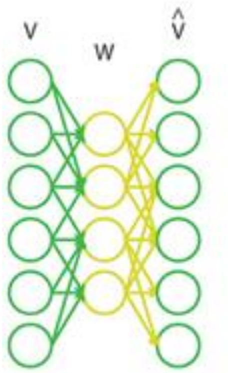
$$\widetilde{\mathcal{L}}^B(\boldsymbol{\theta}, \phi; \mathbf{x}^{(i)}) = -D_{KL}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}^{(i)})||p_{\boldsymbol{\theta}}(\mathbf{z})) + \boxed{\frac{1}{L}\sum_{l=1}^{L}}(\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}^{(i,l)}))$$

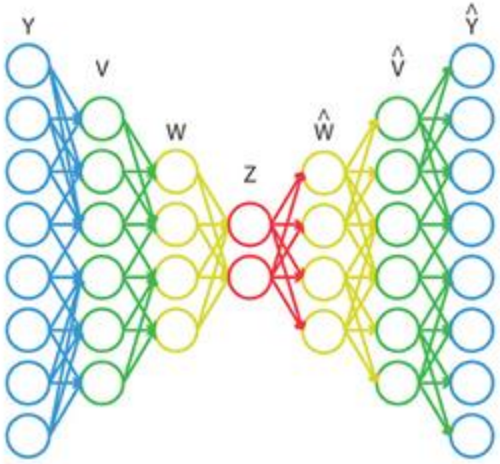
$$\text{where} \quad \mathbf{z}^{(i,l)} = g_{\boldsymbol{\phi}}(\boldsymbol{\epsilon}^{(i,l)}, \mathbf{x}^{(i)}) \quad \text{and} \quad \boldsymbol{\epsilon}^{(l)} \sim p(\boldsymbol{\epsilon})$$

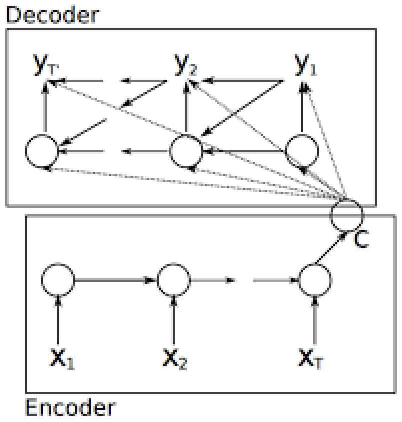
What have been changed?

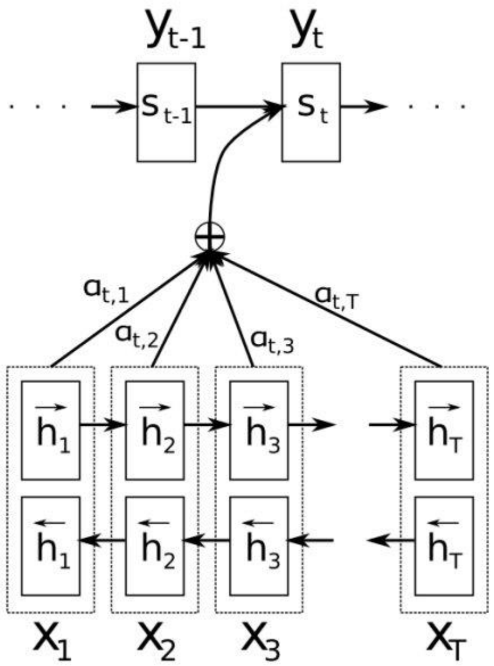
- Bayesian perspective
  - A encoder (parametric function) is used to map input x to code z, where the variation p(z) is simpler than p(x).
  - With x, p(z|x) keeps simple
  - With z, conditional probability p(x|z) is simpler than x.
  - All seems simpler!
  - Model training becomes parameter adjustment, using BP.
- Neural model perspective
  - Randomness
  - Can we BP? Using MCMC, on the simple p(x|z).
  - Seems a variational + MCMC

Extend to other encoder-decoder models









RNN with latent variable

$$\mathbf{z}_t \sim \mathcal{N}(\boldsymbol{\mu}_{0,t}, \mathrm{diag}(\boldsymbol{\sigma}^2_{0,t})) \qquad (5)$$

$$[\boldsymbol{\mu}_{0,t}, \boldsymbol{\sigma}_{0,t}] = \varphi_\tau^{\mathrm{prior}}(\mathbf{h}_{t-1})$$

$$\mathbf{x}_t \mid \mathbf{z}_t \sim \mathcal{N}(\boldsymbol{\mu}_{x,t}, \mathrm{diag}(\boldsymbol{\sigma}^2_{x,t}))$$

$$[\boldsymbol{\mu}_{x,t}, \boldsymbol{\sigma}_{x,t}] = \varphi_\tau^{\mathrm{dec}}(\varphi_\tau^{\mathbf{z}}(\mathbf{z}_t), \mathbf{h}_{t-1})$$

$$\mathbf{h}_t = f_\theta\left(\varphi_\tau^{\mathbf{x}}(\mathbf{x}_t), \varphi_\tau^{\mathbf{z}}(\mathbf{z}_t), \mathbf{h}_{t-1}\right)$$

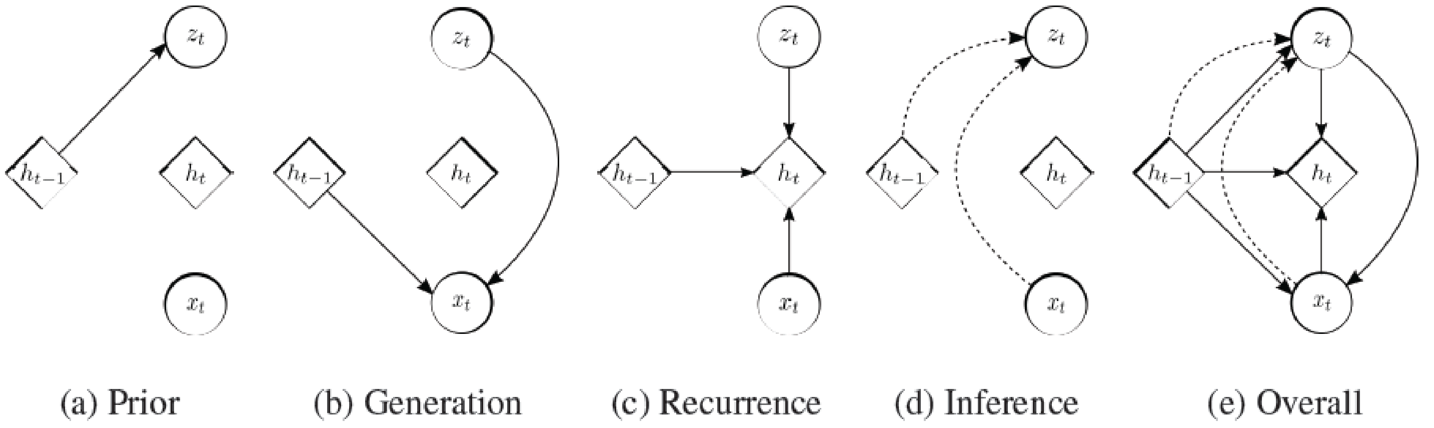


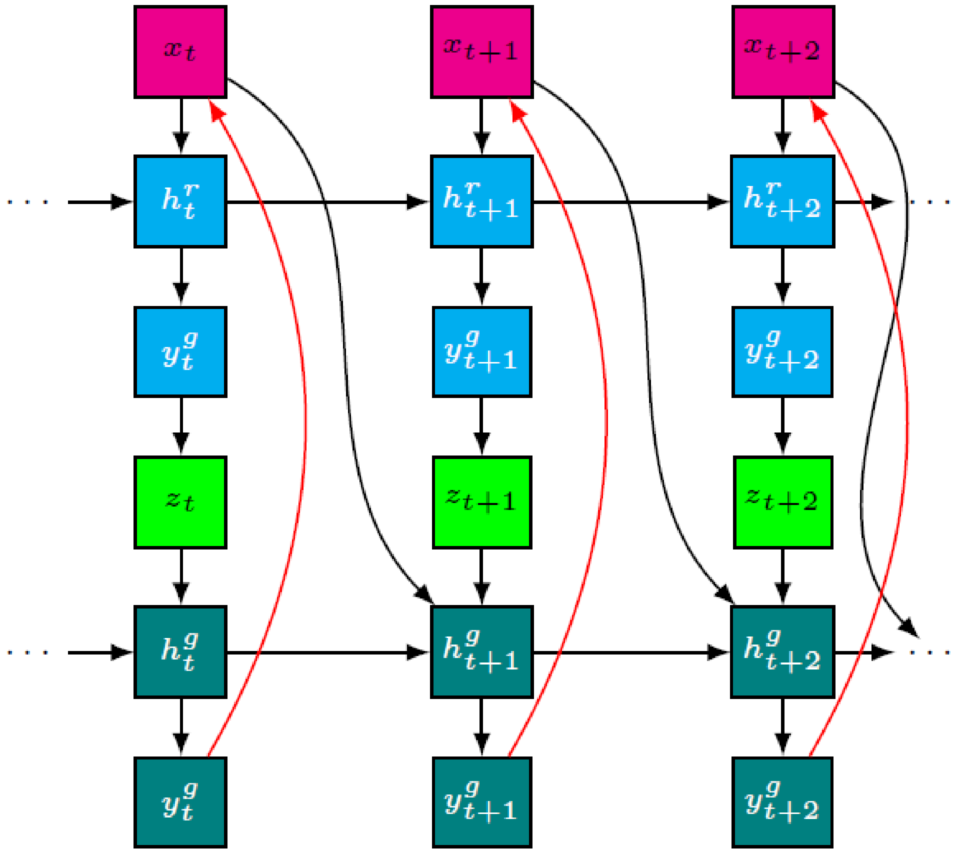
Figure 1: Graphical illustrations of each operation of the VRNN: (a) computing the conditional prior using Eq. (5); (b) generating function using Eq. (6); (c) updating the RNN hidden state using Eq. (7); (d) inference of the approximate posterior using Eq. (9); (e) overall computational paths of the VRNN.

$$p(\mathbf{x}_{\leq T}, \mathbf{z}_{\leq T}) = \prod_{t=1}^{T} p(\mathbf{x}_t \mid \mathbf{z}_{\leq t}, \mathbf{x}_{<t}) p(\mathbf{z}_t \mid \mathbf{x}_{<t}, \mathbf{z}_{<t}).$$

$$\mathbb{E}_{q(\mathbf{z}_{\leq T} \mid \mathbf{x}_{\leq T})}\left[\sum_{t=1}^{T}\left(-\mathrm{KL}(q(\mathbf{z}_t \mid \mathbf{x}_{\leq t}, \mathbf{z}_{<t}) \| p(\mathbf{z}_t \mid \mathbf{x}_{<t}, \mathbf{z}_{<t})) + \log p(\mathbf{x}_t \mid \mathbf{z}_{\leq t}, \mathbf{x}_{<t})\right)\right]$$

- Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A., and Bengio, Y. (2015). A recurrent latent variable model for sequential data. In NIPS, pages 2962–2970.

Stochastic Recurrent Networks (STORNs)



- Bayer, J. and Osendorfer, C. (2014). Learning stochastic recurrent networks. In NIPS Workshop on Advances in Variational Inference

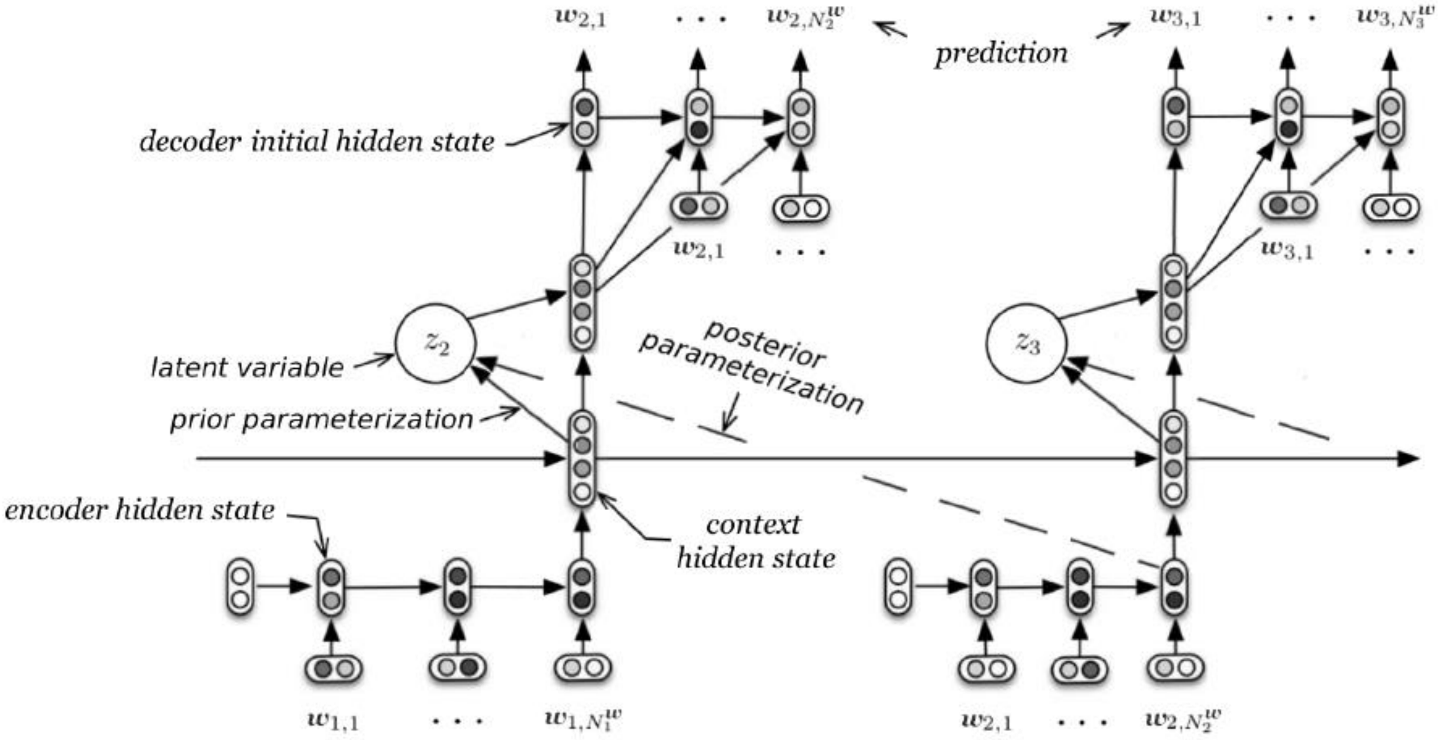
Variational Recurrent AE (VRAE)

$$
\begin{aligned}
h_{t+1} &= \tanh(W_{enc}^T h_t + W_{in}^T x_{t+1} + b_{enc}) \\
\mu_z &= W_\mu^T h_{end} + b_\mu \\
log(\sigma_z) &= W_\sigma^T h_{end} + b_\sigma
\end{aligned}
$$

$$
\begin{aligned}
h_0 &= \tanh(W_z^T z + b_z) \\
h_{t+1} &= \tanh(W_{dec}^T h_t + W_x^T x_t + b_{dec}) \\
x_t &= \mathrm{sigm}(W_{out}^T h_t + b_{out})
\end{aligned}
$$

- Fabius, O. and van Amersfoort, J. R. (2014). Variational recurrent auto-encoders. arXiv:1412.6581.
- Music generation

Variable Encoder-Decoder RNN



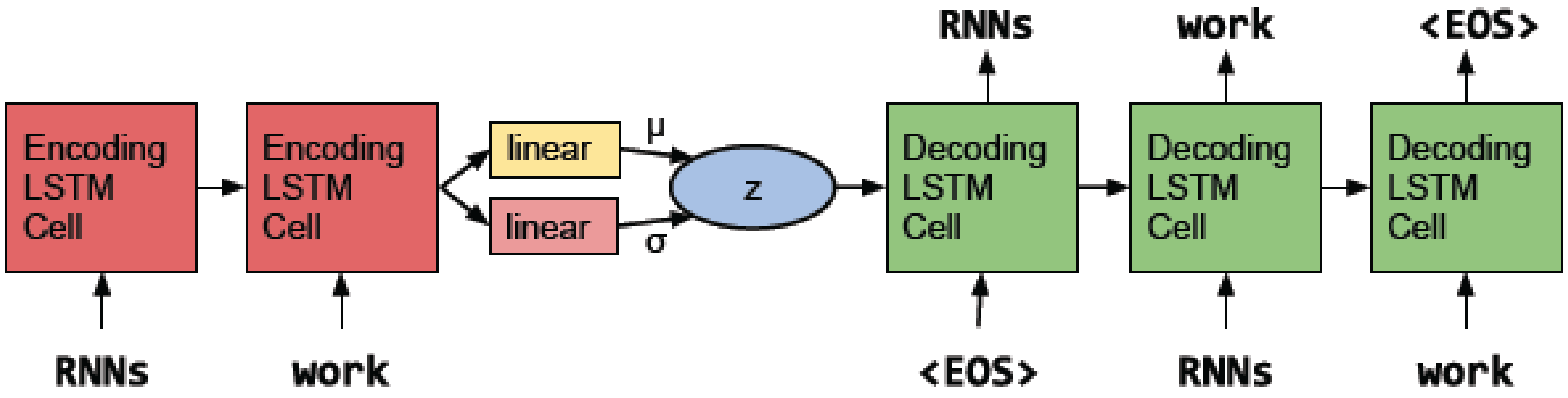
- Iulian Vlad Serban, Alessandro Sordoni, Ryan Lowe, Laurent Charlin, Joelle Pineau, Aaron Courville, Yoshua Bengio, **A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues, 2016/05/20**

Variational RNN LM

i went to the store to buy some groceries .
i store to buy some groceries .
i were to buy any groceries .
horses are to buy any groceries .
horses are to buy any animal .
horses the favorite any animal .
horses the favorite favorite animal .
horses are my favorite animal .

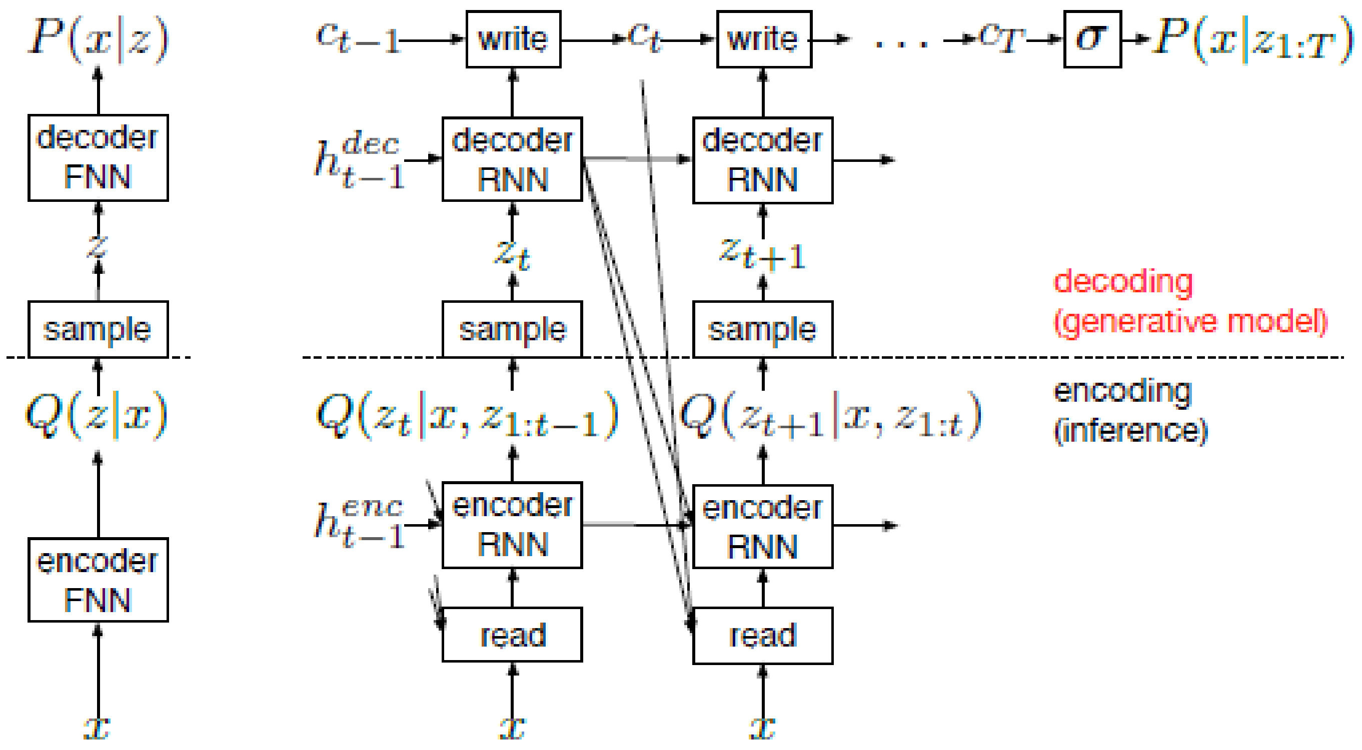
" i want to talk to you . "
"i want to be with you . "
"i do n't want to be with you . "
i do n't want to be with you .
she did n't want to be with him .

he was silent for a long moment .
he was silent for a moment .
it was quiet for a moment .
it was dark and cold .
there was a pause .
it was my turn .



- Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., and Bengio, S. (2015). Generating sentences from a continuous space. arXiv:1511.06349.

Variational image generation



$$\mathcal{L}^z = \sum_{t=1}^{T} KL(Q(Z_t|h_t^{enc})||P(Z_t))$$

$$\mathcal{L}^x = -\log D(x|c_T)$$

$$\mathcal{L} = \langle \mathcal{L}^x + \mathcal{L}^z \rangle_{z \sim Q}$$

- DRAW: A Recurrent Neural Network For Image Generation

Content

- What are they?
- Marriage 1: Variantional AE
- Marriage 2: Denoise AE

Denoise AE

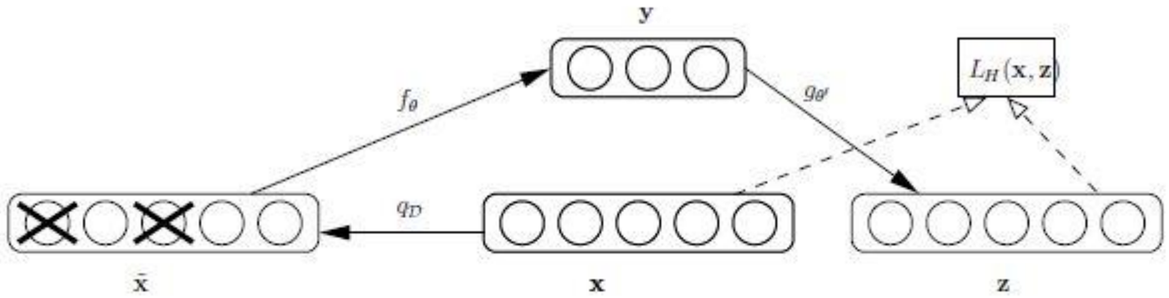
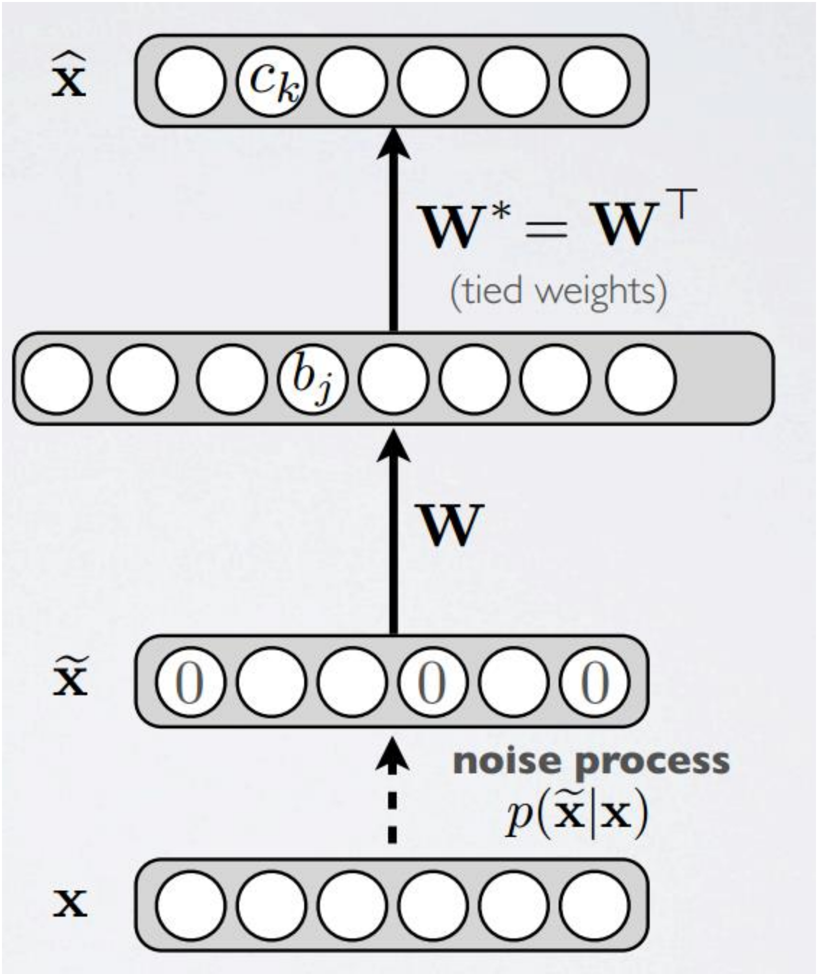
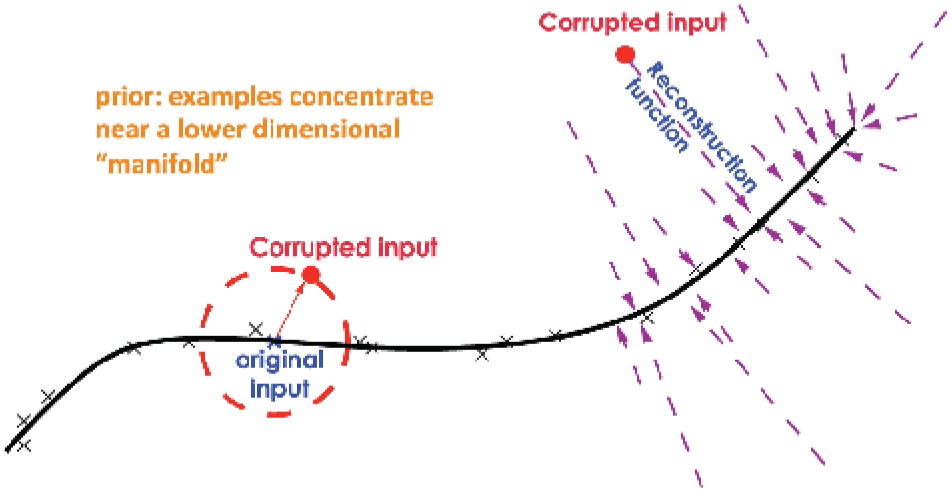


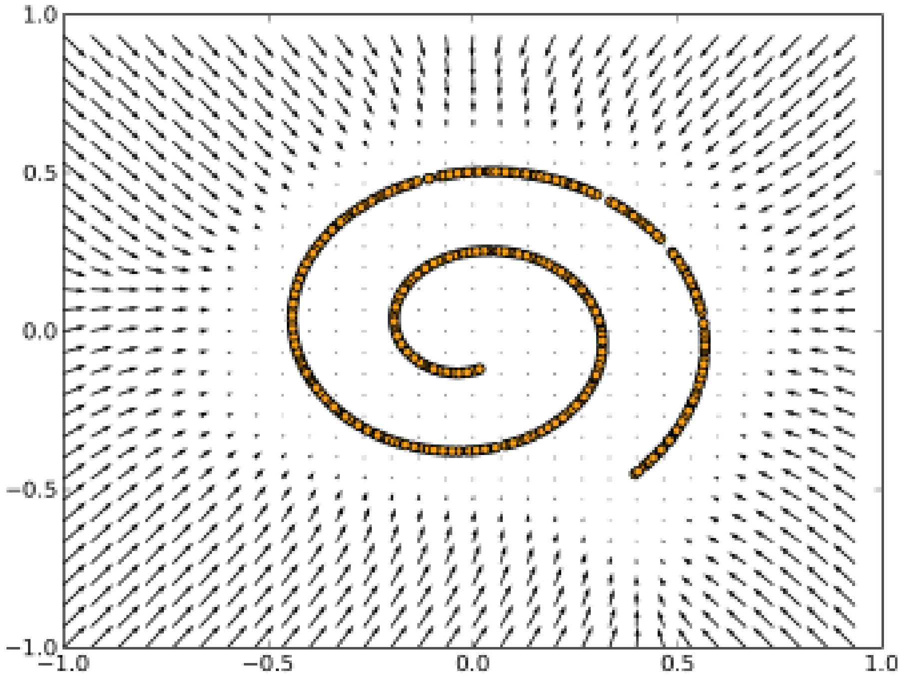
Figure 1. An example $\mathbf{x}$ is corrupted to $\tilde{\mathbf{x}}$. The autoencoder then maps it to $\mathbf{y}$ and attempts to reconstruct $\mathbf{x}$.



DAE learns scores (gradients)

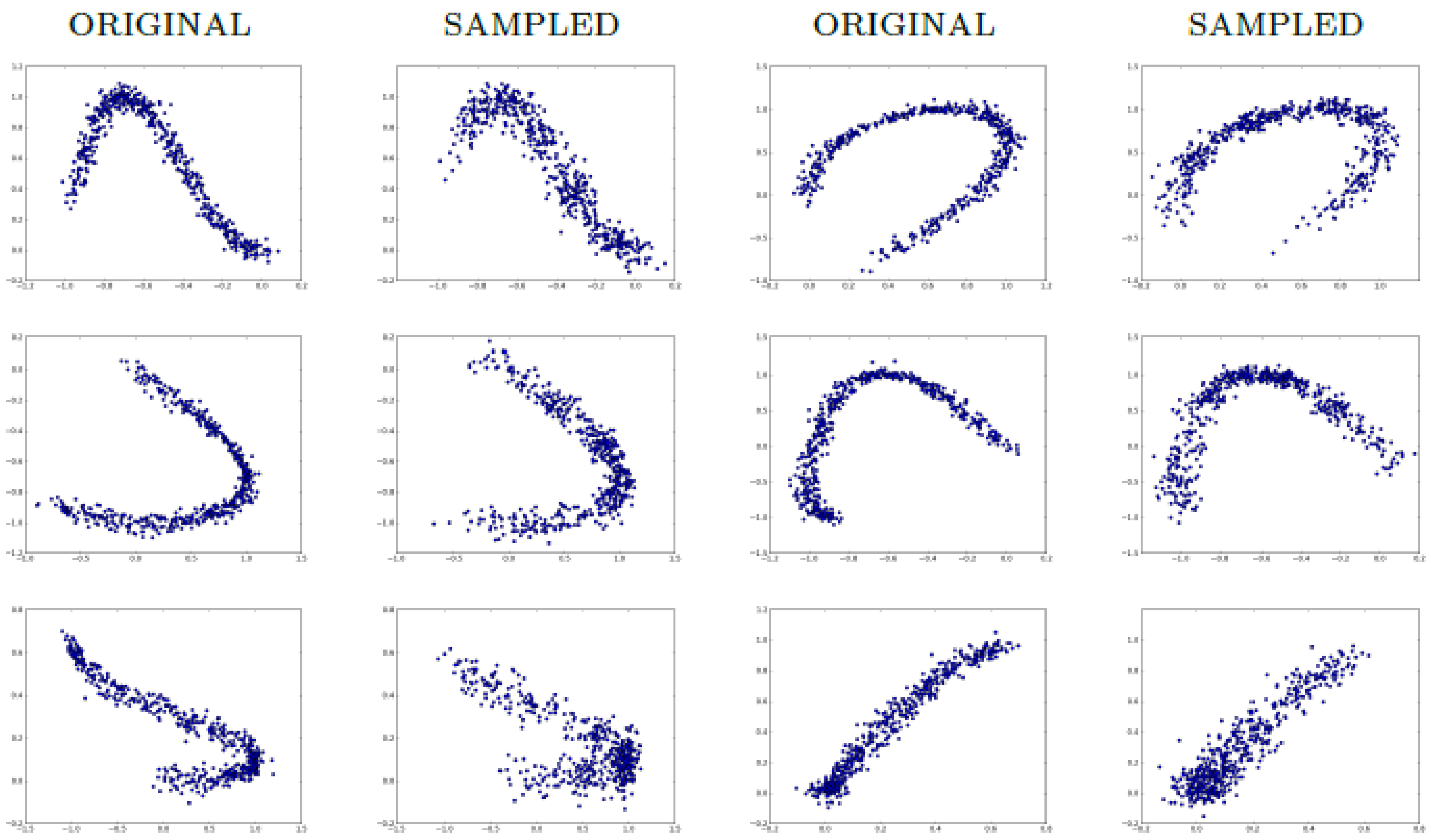
$$r^*_\sigma(x) = x + \sigma^2 \frac{\partial \log p(x)}{\partial x} + o(\sigma^2) \quad as \quad \sigma \to 0.$$





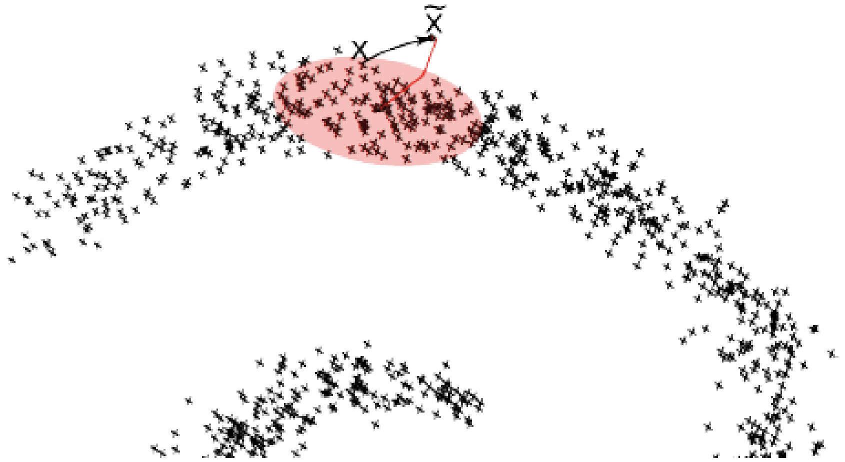
Guillaume Alain and Yoshua Bengio, What Regularized Auto-Encoders Learn from the Data Generating Distribution

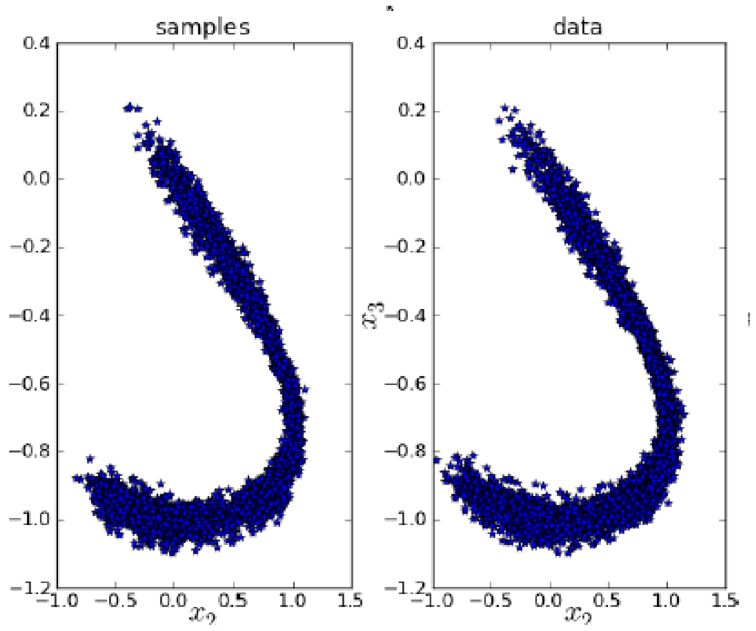
DAE can be used to sampling x



Any corruption + any cost

$$X_t \sim P_\theta(X|\tilde{X}_{t-1})$$
$$\tilde{X}_t \sim \mathcal{C}(\tilde{X}|X_t)$$





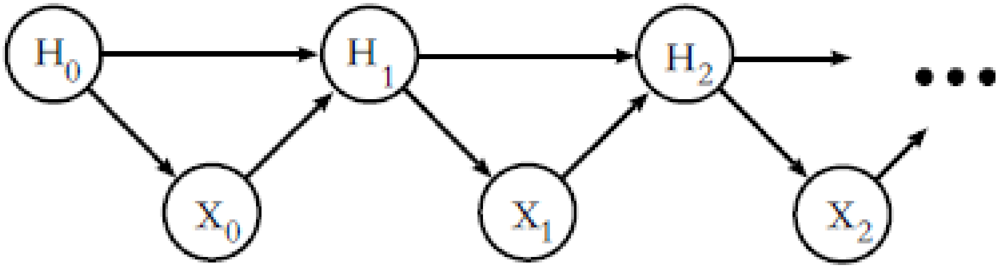
Theorem 1. *If $P_{\theta_n}(X|\tilde{X})$ is a consistent estimator of the true conditional distribution $\mathcal{P}(X|\tilde{X})$ and T_n defines an ergodic Markov chain, then as the number of examples $n \to \infty$, the asymptotic distribution $\pi_n(X)$ of the generated samples converges to the data-generating distribution $\mathcal{P}(X)$.*

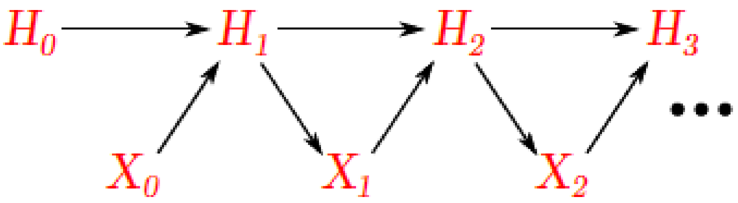
- Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent, Generalized Denoising Auto-Encoders as Generative Models.

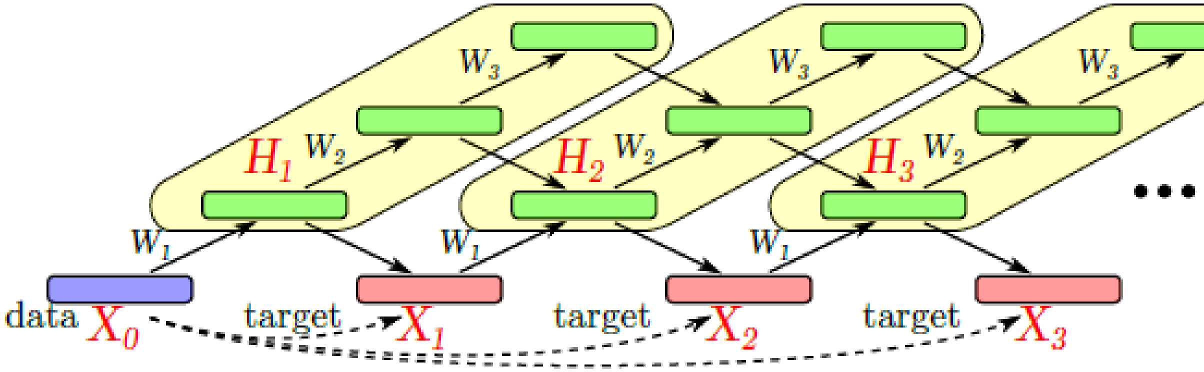
Introducing latent variables

$$
\begin{aligned}
H_{t+1} &\sim P_{\theta_1}(H|H_t, X_t) \\
X_{t+1} &\sim P_{\theta_2}(X|H_{t+1}).
\end{aligned}
$$

Yoshua Bengio et al., Deep Generative Stochastic Networks Trainable by Backprop.

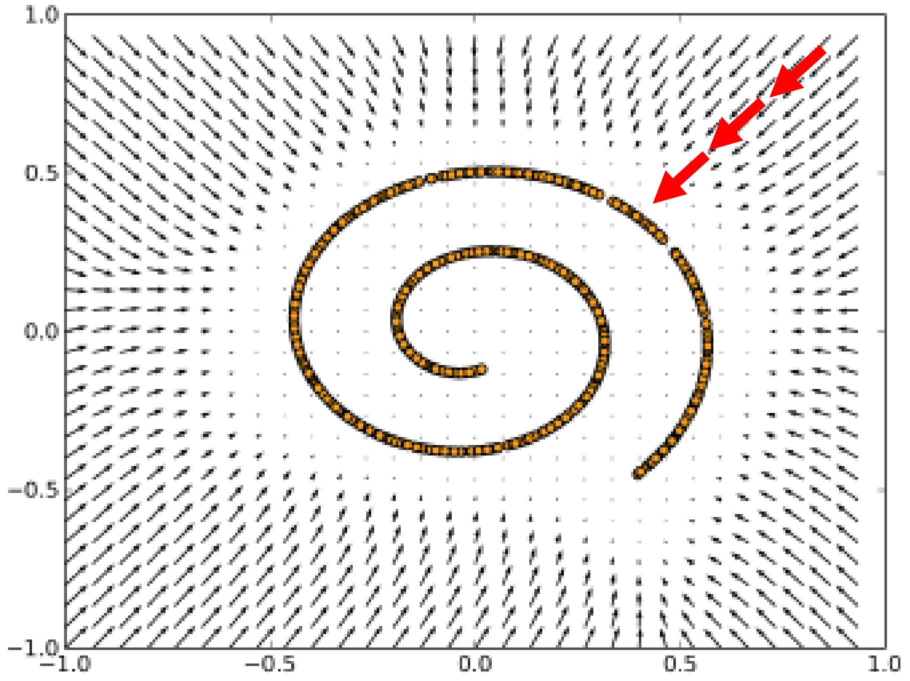






Multi-step generation

- Train DAE with random corruption
- Reconstruct iteratively until converge
- Equals to get stuck to minimum engergy, or max p(x)
- It can be proved that with symmetric corruption, the conergence is a stationary point.



Conclusions

- Graphical model and neural model are merging
- Both variational AE and denoise AE seem reasonable to recover data distribution
- If we treat variational AE as a corruption in the encoding phase, then seems it is a special denoise DAE. Is that true?
- How other regulations can be included in both training and decoding, e.g., rythm.