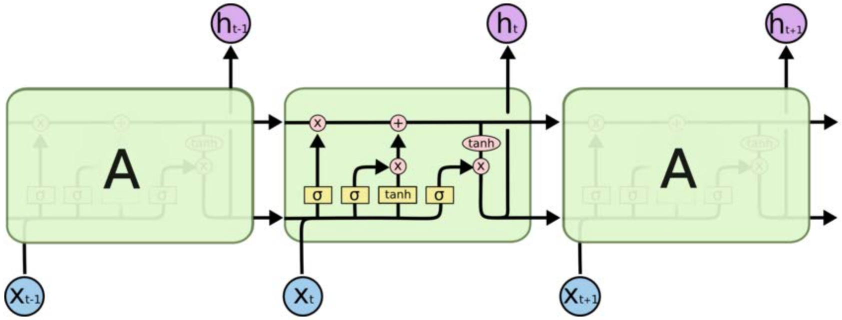
A Recurrent Latent Variable Model for Sequential Data

Chung, Junyoung, et al. "A recurrent latent variable model for sequential data."*arXiv preprint arXiv:1506.02216* (2015). (Accepted NIPS 2015)

2015.11.16 산업공학특론 발표

Sequence Learning

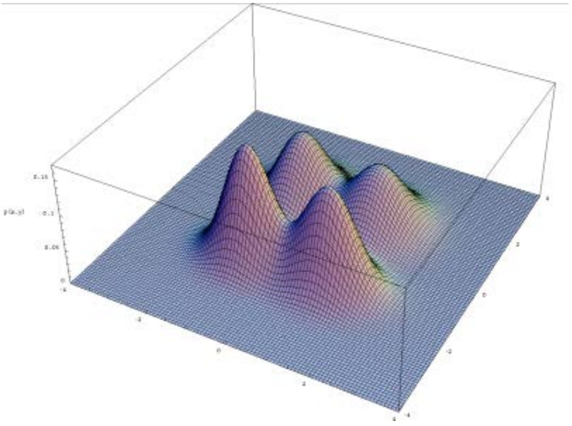
- Dynamic Bayesian networks(DBN)
  - Hidden Markov models(HMM)
  - Kalman filters
- Now the mainstream is recurrent neural network(RNN) based
  - DBNs are simple (HMM state space are single set of mutually exclusive states)
  - Training DBNs are hard (MCMC)
  - RNNs have a richly distributed internal state representation
  - RNNs have flexible non-linear transition functions
  - RNNs can be trained by Backpropagation



Sequence Learning

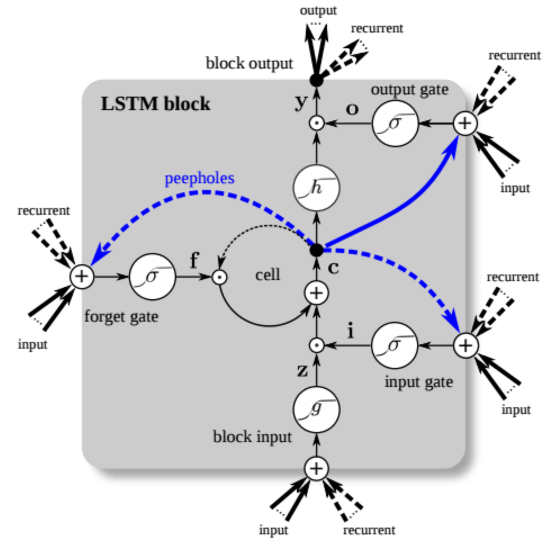
- DBN : (generative) probabilistic modeling
- DBNs hidden state is expressed in random variables(stochastic) : randomness of hidden variables
- RNNs are entirely deterministic

- For simple dependencies, variability is low.
- For data that has complex dependencies, additional variability should be incorporated into the model.
  - Hidden variables can explain the variability.



Recurrent Neural Networks

- Sequence modelling with RNN
- $h_t = f_\theta(x_t, h_{t-1})$
- f : deterministic non-linear transition function

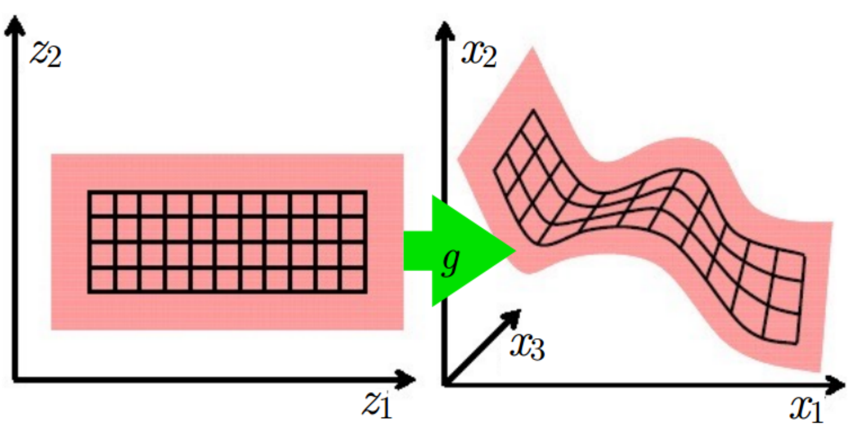


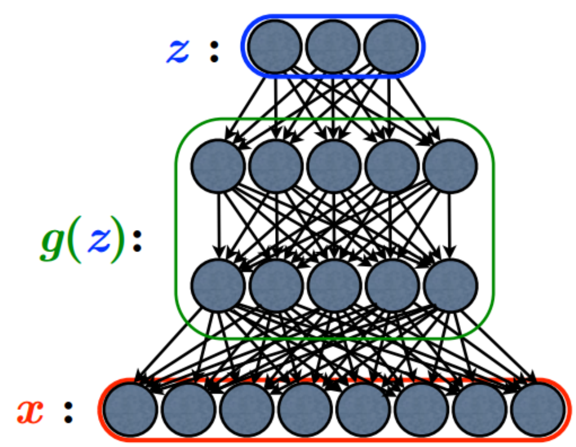
- Probability of a sequence $(x_1, x_2, \cdots, x_T)$
- $p(x_1, x_2, \cdots, x_T) = \prod_{t=1}^{T} p(x_t | x_{<t})$
- $p(x_t | x_{<t}) = g_\tau(h_{t-1})$

$$
\begin{aligned}
\mathbf{z}^t &= g(\mathbf{W}_z\mathbf{x}^t + \mathbf{R}_z\mathbf{y}^{t-1} + \mathbf{b}_z) && \textit{block input} \\
\mathbf{i}^t &= \sigma(\mathbf{W}_i\mathbf{x}^t + \mathbf{R}_i\mathbf{y}^{t-1} + \mathbf{p}_i \odot \mathbf{c}^{t-1} + \mathbf{b}_i) && \textit{input gate} \\
\mathbf{f}^t &= \sigma(\mathbf{W}_f\mathbf{x}^t + \mathbf{R}_f\mathbf{y}^{t-1} + \mathbf{p}_f \odot \mathbf{c}^{t-1} + \mathbf{b}_f) && \textit{forget gate} \\
\mathbf{c}^t &= \mathbf{i}^t \odot \mathbf{z}^t + \mathbf{f}^t \odot \mathbf{c}^{t-1} && \textit{cell state} \\
\mathbf{o}^t &= \sigma(\mathbf{W}_o\mathbf{x}^t + \mathbf{R}_o\mathbf{y}^{t-1} + \mathbf{p}_o \odot \mathbf{c}^t + \mathbf{b}_o) && \textit{output gate} \\
\mathbf{y}^t &= \mathbf{o}^t \odot h(\mathbf{c}^t) && \textit{block output}
\end{aligned}
$$

Latent variable model

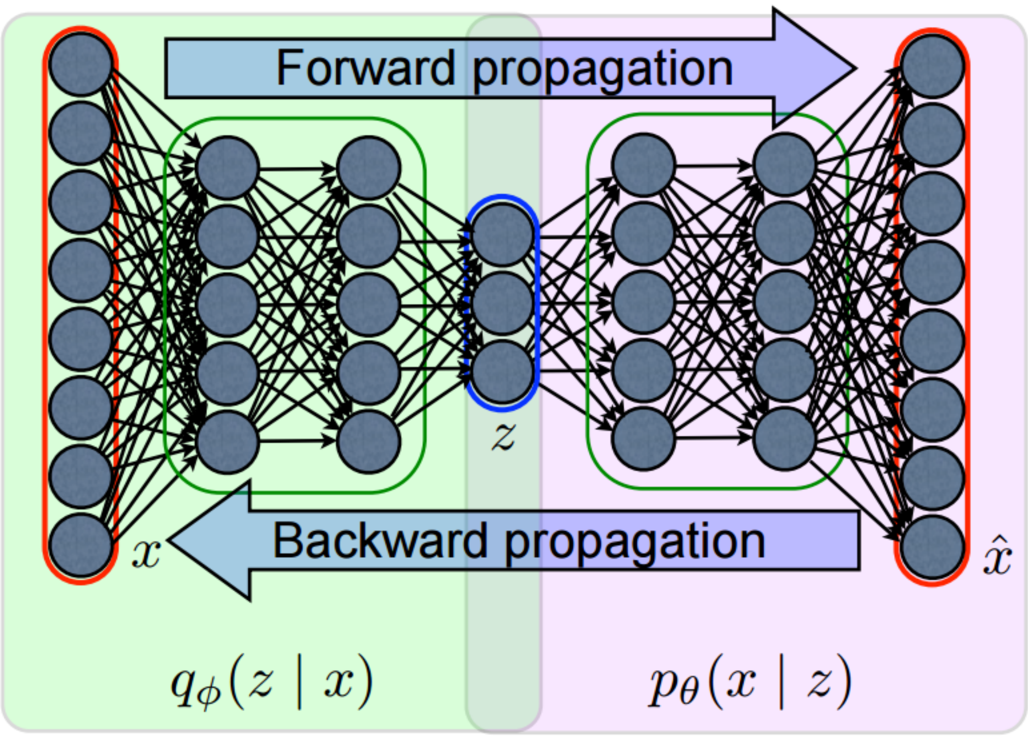
- Use neural networks as the (generative) transformation g from the latent space to the original feature space.
- For both training and inference, latent variable z must be inferred.
- This is a generative model : inferring the latent variable is hard
- The posterior $p_\theta(z|x)$ is intractable





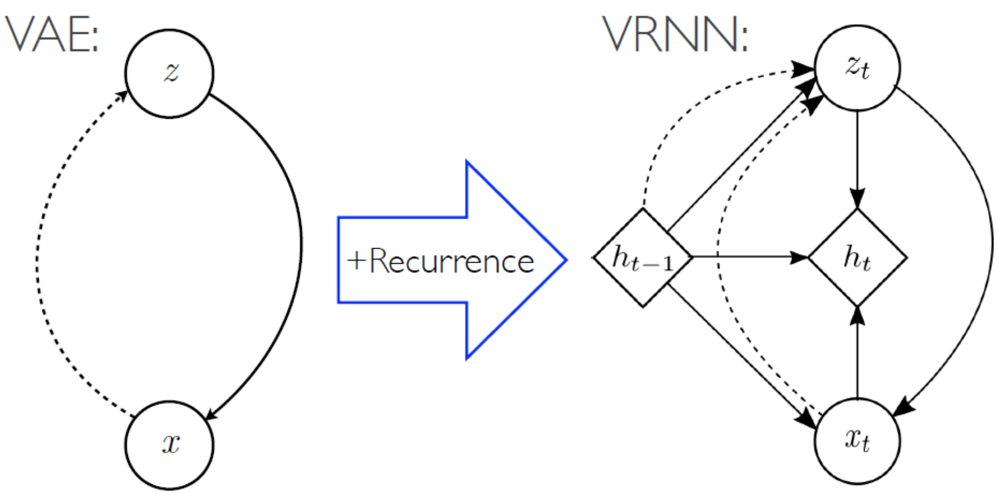
Variational Autoencoder

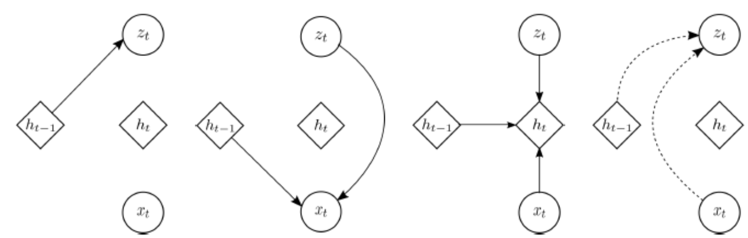
Objective function: $\mathcal{L}(\theta, \phi, x) = -D_{\mathrm{KL}}\left(q_\phi(z \mid x) \| p_\theta(z)\right) + \mathbb{E}_{q_\phi(z \mid x)}\left[\log p_\theta(x \mid z)\right]$



Variational Recurrent Neural Network(VRNN)

- VRNN is a recurrent application of the VAE at every time step
- Latent variables can model noise in a structured way.

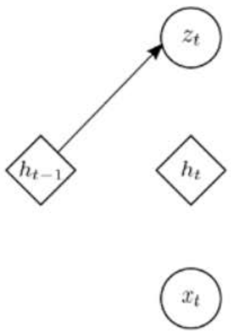




(a) Prior (b) Generation (c) Recurrence (d) Inference

VRNN : Generation

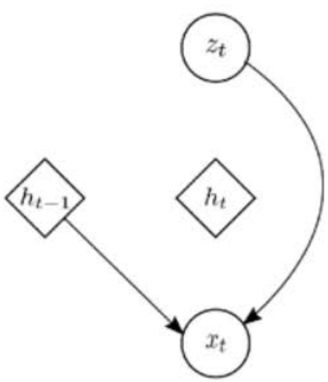
- Prior on z_t



$$\mathbf{z}_t \sim \mathcal{N}(\boldsymbol{\mu}_{0,t}, \mathrm{diag}(\boldsymbol{\sigma}_{0,t}^2)) \text{ , where } [\boldsymbol{\mu}_{0,t}, \boldsymbol{\sigma}_{0,t}] = \varphi_\tau^{\mathrm{prior}}(\mathbf{h}_{t-1})$$

Neural networks

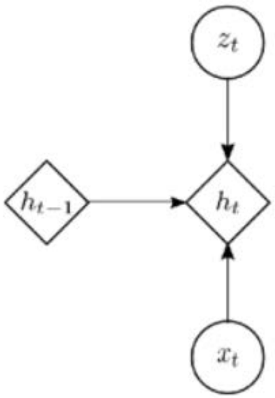
- Generation (decoding)



$$\mathbf{x}_t \mid \mathbf{z}_t \sim \mathcal{N}(\boldsymbol{\mu}_{x,t}, \mathrm{diag}(\boldsymbol{\sigma}_{x,t}^2)) \text{ , where } [\boldsymbol{\mu}_{x,t}, \boldsymbol{\sigma}_{x,t}] = \varphi_\tau^{\mathrm{dec}}(\varphi_\tau^{\mathbf{z}}(\mathbf{z}_t), \mathbf{h}_{t-1})$$

VRNN : Generation

- Recurrence



Neural networks

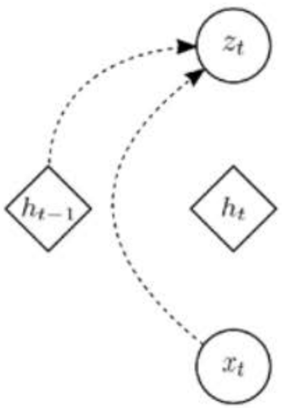
$$\mathbf{h}_t = f_\theta\left(\varphi_\tau^{\mathbf{x}}(\mathbf{x}_t), \varphi_\tau^{\mathbf{z}}(\mathbf{z}_t), \mathbf{h}_{t-1}\right)$$

- Objective function

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

VRNN : Inference

- Inference(encoding)



$$\mathbf{z}_t \mid \mathbf{x}_t \sim \mathcal{N}(\boldsymbol{\mu}_{z,t}, \mathrm{diag}(\boldsymbol{\sigma}^2_{z,t}))\text{ , where } [\boldsymbol{\mu}_{z,t}, \boldsymbol{\sigma}_{z,t}] = \varphi^{\mathrm{enc}}_{\tau}(\varphi^{\mathbf{x}}_{\tau}(\mathbf{x}_t), \mathbf{h}_{t-1})$$

Neural networks

- Approximate posterior for variational inference

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

VRNN : Learning

- Maximize the variational lower bound

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

- Use the method as in VAE

Experimental Results

- Data
  - Blizzard : 300 hours of English by a single female speaker
  - TIMIT : 6300 English sentences read by 630 speakers
  - Onomatopoeia : 6738 non-linguistic human-made sounds by 51 speakers
  - Accent : English paragraphs read by 2046 speakers
  - IAM-OnDB : 13040 handwritten lines by 500 writers
- Output function : the output layer parameterizes the following distribution
  - Gaussian distribution(Gauss)
  - Gaussian Mixture model(GMM)

$$\hat{y}_t = \left(\hat{e}_t, \{\hat{w}_t^j, \hat{\mu}_t^j, \hat{\sigma}_t^j, \hat{\rho}_t^j\}_{j=1}^M\right) = b_y + \sum_{n=1}^{N} W_{h^n y} h_t^n$$

$$\Pr(x_{t+1}|y_t) = \sum_{j=1}^{M} \pi_t^j \, \mathcal{N}(x_{t+1}|\mu_t^j, \sigma_t^j, \rho_t^j)$$

$$e_t = \frac{1}{1 + \exp(\hat{e}_t)} \implies e_t \in (0,1)$$

$$\pi_t^j = \frac{\exp\left(\hat{\pi}_t^j\right)}{\sum_{j'=1}^M \exp\left(\hat{\pi}_t^{j'}\right)} \implies \pi_t^j \in (0,1), \quad \sum_j \pi_t^j = 1$$

$$\mu_t^j = \hat{\mu}_t^j \implies \mu_t^j \in \mathbb{R}$$

$$\sigma_t^j = \exp\left(\hat{\sigma}_t^j\right) \implies \sigma_t^j > 0$$

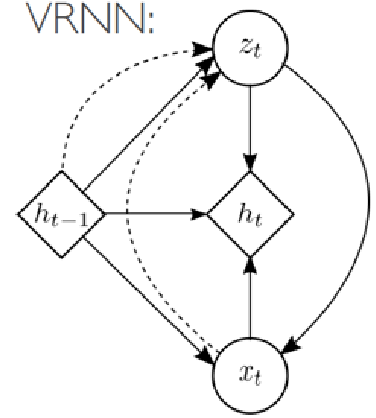
$$\rho_t^j = tanh(\hat{\rho}_t^j) \implies \rho_t^j \in (-1,1)$$

Experimental Results

- Model structure : single recurrent hidden layer with 2000 LSTM units
- VAE structure : 4
- Standard RNN
- $\varphi_\tau^x, \varphi_\tau^{dec}$

φ_τ^x → LSTM → φ_τ^{dec} → Output distribution parameterization

VRNN:



- VRNN
- $\varphi_\tau^x, \varphi_\tau^{dec}, \varphi_\tau^z, \varphi_\tau^{enc}, \varphi_\tau^{prior}$

$\varphi_\tau^x, \varphi_\tau^z$ → LSTM → $\varphi_\tau^{enc}, \varphi_\tau^{prior}$ → φ_τ^{dec} → Output Distribution parameterization

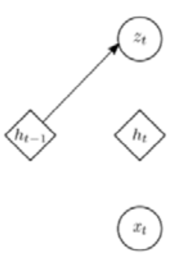
- τ : parameters of the output function(GMM or Gaussian)

Experimental Results

Table 1: Average log-likelihood on the test (or validation) set of each task.

Models	Speech modelling				Handwriting
	Blizzard	TIMIT	Onomatopoeia	Accent	IAM-OnDB
RNN-Gauss	3539	-1900	-984	-1293	1016
RNN-GMM	7413	26643	18865	3453	1358
VRNN-I-Gauss	≥ 8933 ≈ 9188	≥ 28340 ≈ 29639	≥ 19053 ≈ 19638	≥ 3843 ≈ 4180	≥ 1332 ≈ 1353
VRNN-Gauss	≥ 9223 $\approx \mathbf{9516}$	≥ 28805 $\approx \mathbf{30235}$	≥ 20721 $\approx \mathbf{21332}$	≥ 3952 ≈ 4223	≥ 1337 ≈ 1354
VRNN-GMM	≥ 9107 ≈ 9392	≥ 28982 ≈ 29604	≥ 20849 ≈ 21219	≥ 4140 $\approx \mathbf{4319}$	≥ 1384 $\approx \mathbf{1384}$

- RNN : standard RNN
- VRNN : variational RNN
- VRNN-I-Gauss : without conditional prior (standard normal prior)



$z_t \sim N(0,1)$

$$\mathbf{z}_t \sim \mathcal{N}(\boldsymbol{\mu}_{0,t}, \text{diag}(\boldsymbol{\sigma}^2_{0,t}))\text{ , where }[\boldsymbol{\mu}_{0,t}, \boldsymbol{\sigma}_{0,t}] = \varphi^{\text{prior}}_{\tau}(\mathbf{h}_{t-1})$$

Conclusion

- Propose a general framework for sequence modelling with latent random variables into a RNN.
- Introduction of latent random variables can provide significant improvements in modelling highly structured sequences.
- Temporal conditioning for latent random variables improves performance

Referenences

- Chung, Junyoung, et al. "A recurrent latent variable model for sequential data."*arXiv preprint arXiv:1506.02216* (2015). (Accepted NIPS 2015)
- Kingma, Diederik P., and Max Welling. "Auto-encoding variational bayes." *arXiv preprint arXiv:1312.6114* (2013).