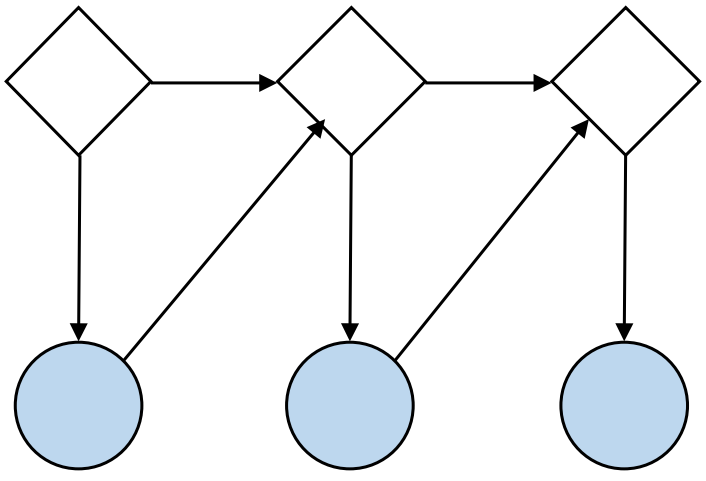
A Review on Recent Sequential Latent Variable Models

Presented by Zhe Gan

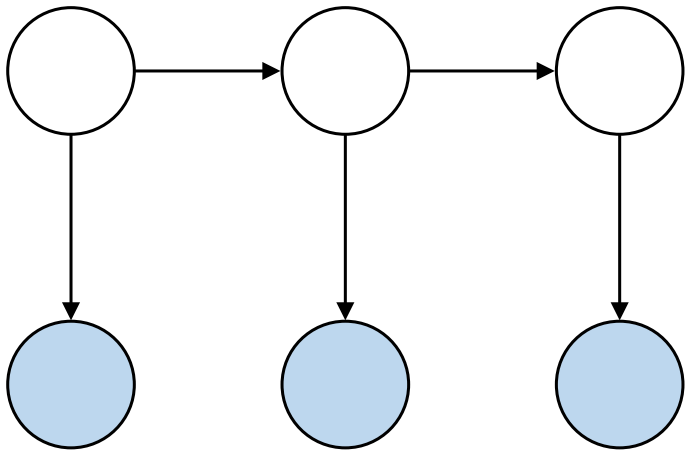
Duke University

October 7th, 2016

Motivation



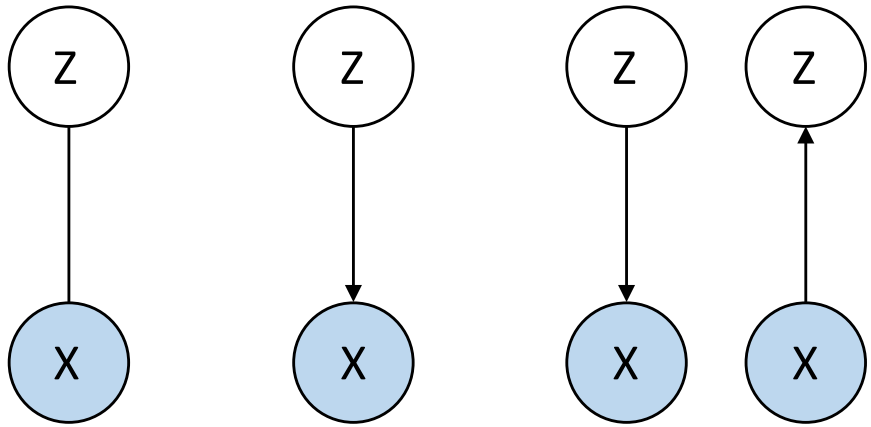
(a) RNN



(b) SSM

- RNNs are widely used in deep learning, e.g., LSTM, Seq2Seq Model, attention mechanism;
- SSMs are Bayesian probabilistic generative models, e.g., HMM, LDS (KF), EKPF;
- Can we leverage the advantage of both? – incorporate *stochastic* latent variable into RNN.

Deep Generative Models



(a) RBM (b) SBN (c) VAE

- RBM: *Undirected* graphical model, binary latent variable, inference by CD;
- SBN: *Directed* graphical model, binary latent variable, inference by Gibbs, MCEM, NVIL;
- VAE: *Directed* graphical model, usually Gaussian latent variable, inference by SGVB.

Literature Review

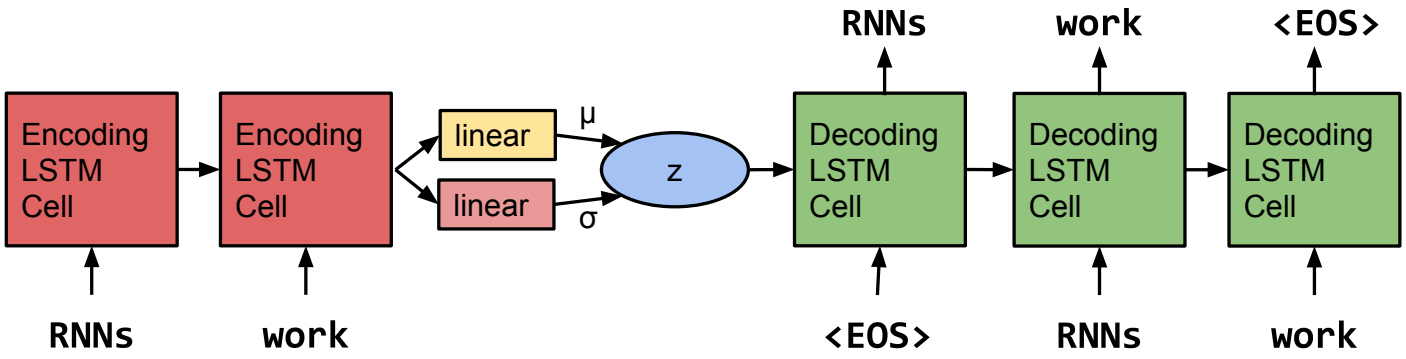
Category	Model	Conferences	Applications
RBM	TRBM [22, 19] RTRBM [20, 5, 17] FCRBM [21, 24] imCRBM [23]	NIPS 2006, AISTATS 2007 NIPS 2009, ICML 2012 & 2014 ICML 2009, NIPS 2011 CVPR 2010	Mocap, Bouncing ball plus Music, Weather Mocap, Facial expression Human pose tracking
SBN	TSBN, FTSBN [10, 18]	NIPS 2015, ICML 2016	plus Dynamical TM
VAE-I	VRAE [8] VRAE [6] NASM [16] VNMT [25]	ICLR 2015 workshop CoNLL 2016 ICML 2016 EMNLP 2016	Music Text modeling QA NMT
VAE-II	STORN [4] VRNN [7] NASMC [11] SRNN [9]	ICLR 2015 workshop NIPS 2015 NIPS 2015 NIPS 2016	Music Speech, Handwriting Music Speech, Music
SSM	DKF [14, 3, 13, 15] SVAE [12]	arXiv 2015 & 2016 NIPS 2016	EHR video of mouse behavior
	Poisson[2, 1, 26]	AISTATS 2015, NIPS 2016, ICDM 2016	Dynamical TM, Music

Overview

- Generating Sentences from a Continuous Space, CoNLL 2016 [6]
- Neural Variational Inference for Text Processing, ICML 2016 [16]
- A Recurrent Latent Variable Model for Sequential Data, NIPS 2015 [7]
- Sequential Neural Models with Stochastic Layers, NIPS 2016 [9]
- Neural Adaptive Sequential Monte Carlo, NIPS 2015 [11]

Generating Sentences from a Continuous Space

Model



- Objective:

$$\mathcal{L} = -KL(q_\phi(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})) + \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] \qquad (1)$$

- Direct implementation fails, the training always tries to yield models that consistently set $q(\boldsymbol{z}|\boldsymbol{x})$ equal to the prior $p(\boldsymbol{z})$.
- Solution: (i) KL cost annealing; (ii) Word dropout and historyless decoding.

Generating Sentences from a Continuous Space

Advantage

(Left) Traditional Autoencoder; (Right) Variational Autoencoder

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 .

Generating Sentences from a Continuous Space

Experiments

- Language modeling: perform roughly the same, or actually a little bit worse than RNNLM.

Model	Standard				Inputless Decoder			
	Train NLL	Train PPL	Test NLL	Test PPL	Train NLL	Train PPL	Test NLL	Test PPL
RNNLM	100 –	95	**100** –	**116**	135 –	600	135 –	> 600
VAE	98 (2)	100	101 (2)	119	120 (15)	300	**125** (15)	**380**

- Imputing missing words:

but now , as they parked out front and owen stepped out of the car , he could see _ _ _ _ _ _		
True: *that the transition was complete .*	**RNNLM:** *it , " i said .*	**VAE:** *through the driver 's door .*
you kill him and his _ _		
True: *men .*	**RNNLM:** *. "*	**VAE:** *brother .*
not surprising , the mothers dont exactly see eye to eye with me _ _ _ _		
True: *on this matter .*	**RNNLM:** *, i said .*	**VAE:** *, right now .*

Generating Sentences from a Continuous Space

Experiments

- Sampling from the prior

100% word keep	**75% word keep**
“ no , ” he said . *“ thank you , ” he said .*	*“ love you , too . ”* *she put her hand on his shoulder and followed him to the door .*
50% word keep	**0% word keep**
“ maybe two or two . ” *she laughed again , once again , once again , and thought about it for a moment in long silence .*	*i i hear some of of of* *i was noticed that she was holding the in in of the the in*

- Sampling from the posterior:

INPUT	**we looked out at the setting sun .**	**i went to the kitchen .**	**how are you doing ?**
MEAN	*they were laughing at the same time .*	*i went to the kitchen .*	*what are you doing ?*
SAMP. 1	*ill see you in the early morning .*	*i went to my apartment .*	*“ are you sure ?*
SAMP. 2	*i looked up at the blue sky .*	*i looked around the room .*	*what are you doing ?*
SAMP. 3	*it was down on the dance floor .*	*i turned back to the table .*	*what are you doing ?*

Neural Variational Inference for Text Processing

Document Modeling

Neural Variational Document Model (NVDM)

$$\mathcal{L} = \mathbb{E}_{q_\phi(\boldsymbol{h}|\mathbf{X})}[\sum_{i=1}^{N} \log p_\theta(\boldsymbol{x}_i|\boldsymbol{h})] - KL[q_\phi(\boldsymbol{h}|\mathbf{X})||p(\boldsymbol{h})] \quad (2)$$

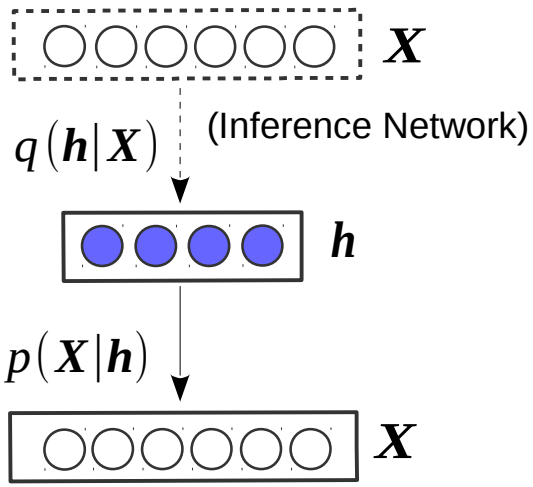


Figure 1. NVDM for document modelling.

- $\boldsymbol{h} \in \mathbb{R}^K$, $\mathbf{X} \in \mathbb{R}^{|V|}$, $\boldsymbol{x}_i \in \mathbb{R}^{|V|}$;
- $p_\theta(\boldsymbol{x}_i|\boldsymbol{h})$ is multinomial
- $q_\phi(\boldsymbol{h}|\mathbf{X})$ is Gaussian

Neural Variational Inference for Text Processing

Document Modeling

Model	Dim	20News	RCV1
LDA	50	1091	1437
LDA	200	1058	1142
RSM	50	953	988
docNADE	50	896	742
SBN	50	909	784
fDARN	50	917	724
fDARN	200	—-	598
NVDM	50	**836**	563
NVDM	200	852	**550**

(a) Perplexity on test dataset.

Word	**weapons**	**medical**	**companies**	**define**	**israel**	**book**
NVDM	guns	medicine	expensive	defined	israeli	books
	weapon	health	industry	definition	arab	reference
	gun	treatment	company	printf	arabs	guide
	militia	disease	market	int	lebanon	writing
	armed	patients	buy	sufficient	lebanese	pages
NADE	weapon	treatment	demand	defined	israeli	reading
	shooting	medecine	commercial	definition	israelis	read
	firearms	patients	agency	refer	arab	books
	assault	process	company	make	palestinian	relevent
	armed	studies	credit	examples	arabs	collection

(b) The five nearest words in the semantic space.

Space	***Religion***	***Encryption***	***Sport***	***Policy***
orbit	muslims	rsa	goals	bush
lunar	worship	cryptography	pts	resources
solar	belief	crypto	teams	charles
shuttle	genocide	keys	league	austin
moon	jews	pgp	team	bill
launch	islam	license	players	resolution
fuel	christianity	secure	nhl	mr
nasa	atheists	key	stats	misc
satellite	muslim	escrow	min	piece
japanese	religious	trust	buf	marc

Neural Variational Inference for Text Processing

Question Answering

Neural Answer Selection Model (NASM)

- A question $\boldsymbol{q}$ is associated with a set of answer sentences $\{\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n\}$, together with their judgements $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\}$, $\boldsymbol{y}_m = 1$, or 0.
- Each training data point can be treated as a triple $(\boldsymbol{q}, \boldsymbol{a}, \boldsymbol{y})$.

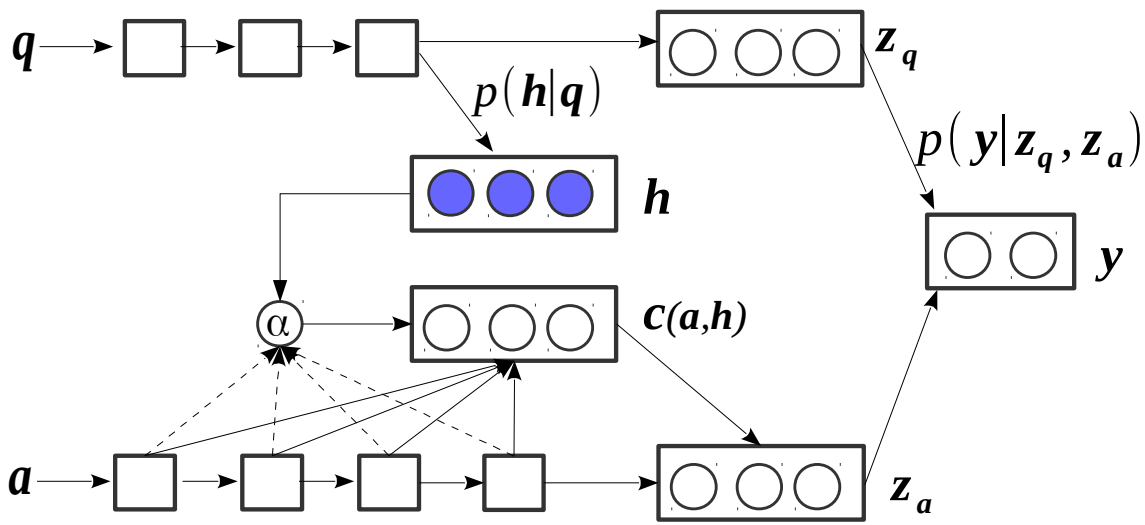


Figure 2. NASM for question answer selection.

Neural Variational Inference for Text Processing

Question Answering

- The proposed model employs two different LSTMs to embed question $\boldsymbol{q}$ and $\boldsymbol{a}$.
- Let $\boldsymbol{s}_q(j)$ and $\boldsymbol{s}_a(i)$ be the state outputs of the two LSTMs, i, j be the positions of the states.
- Objective:

$$\mathcal{L} = \mathbb{E}_{q_\phi(\boldsymbol{h}|\boldsymbol{q},\boldsymbol{a},\boldsymbol{y})}[\log p_\theta(\boldsymbol{y}|\boldsymbol{q},\boldsymbol{a},\boldsymbol{h})] - KL[q_\phi(\boldsymbol{h}|\boldsymbol{q},\boldsymbol{a},\boldsymbol{y})||p_\theta(\boldsymbol{h}|\boldsymbol{q})] \tag{3}$$

Neural Variational Inference for Text Processing

Question Answering

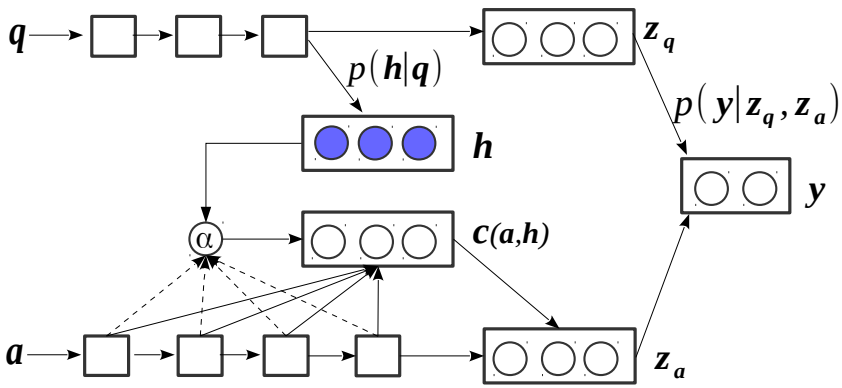


Figure 2. NASM for question answer selection.

(1) Inference Network $q_\phi(\boldsymbol{h}|\boldsymbol{q}, \boldsymbol{a}, \boldsymbol{y})$:

$$\boldsymbol{s}_q(|\boldsymbol{q}|) = f_q^{\text{LSTM}}(\boldsymbol{q}) \quad (30)$$

$$\boldsymbol{s}_a(|\boldsymbol{a}|) = f_a^{\text{LSTM}}(\boldsymbol{a}) \quad (31)$$

$$\boldsymbol{s}_y = \boldsymbol{W}_5\boldsymbol{y} + \boldsymbol{b}_5 \quad (32)$$

$$\boldsymbol{\gamma} = \boldsymbol{s}_q(|\boldsymbol{q}|)||\boldsymbol{s}_a(|\boldsymbol{a}|)||\boldsymbol{s}_y \quad (33)$$

$$\boldsymbol{\lambda}_\phi = \tanh(\boldsymbol{W}_6\boldsymbol{\gamma} + \boldsymbol{b}_6) \quad (34)$$

$$\boldsymbol{\pi}_\phi = \tanh(\boldsymbol{W}_7\boldsymbol{\lambda}_{\boldsymbol{\phi}} + \boldsymbol{b}_7) \quad (35)$$

$$\boldsymbol{\mu}_\phi = \boldsymbol{W}_8\boldsymbol{\pi}_{\boldsymbol{\phi}} + \boldsymbol{b}_8 \quad (36)$$

$$\log \boldsymbol{\sigma}_\phi = \boldsymbol{W}_9\boldsymbol{\pi}_{\boldsymbol{\phi}} + \boldsymbol{b}_9 \quad (37)$$

$$\boldsymbol{h} \sim \mathcal{N}(\boldsymbol{\mu}_\phi(\boldsymbol{q}, \boldsymbol{a}, \boldsymbol{y}), \text{diag}(\boldsymbol{\sigma}_\phi^2(\boldsymbol{q}, \boldsymbol{a}, \boldsymbol{y}))) \quad (38)$$

(2) Generative Model

$p_\theta(\boldsymbol{h}|\boldsymbol{q})$:

$$\boldsymbol{\lambda}_\theta = \tanh(\boldsymbol{W}_1\boldsymbol{s}_q(|\boldsymbol{q}|) + \boldsymbol{b}_1) \quad (39)$$

$$\boldsymbol{\pi}_\theta = \tanh(\boldsymbol{W}_2\boldsymbol{\lambda}_\theta + \boldsymbol{b}_2) \quad (40)$$

$$\boldsymbol{\mu}_\theta = \boldsymbol{W}_3\boldsymbol{\pi}_\theta + \boldsymbol{b}_3 \quad (41)$$

$$\log \boldsymbol{\sigma}_\theta = \boldsymbol{W}_4\boldsymbol{\pi}_\theta + \boldsymbol{b}_4 \quad (42)$$

$p_\theta(\boldsymbol{y}|\boldsymbol{q}, \boldsymbol{a}, \boldsymbol{h})$:

$$\boldsymbol{e}(i) = \boldsymbol{W}_\alpha^T \tanh(\boldsymbol{W}_h\boldsymbol{h} + \boldsymbol{W}_s\boldsymbol{s}_a(i)) \quad (43)$$

$$\alpha(i) = \frac{\boldsymbol{e}(i)}{\sum_j \boldsymbol{e}(j)} \quad (44)$$

$$\boldsymbol{c}(\boldsymbol{a}, \boldsymbol{h}) = \sum_i \boldsymbol{s}_a(i)\alpha(i) \quad (45)$$

$$\boldsymbol{z}_a(\boldsymbol{a}, \boldsymbol{h}) = \tanh(\boldsymbol{W}_a\boldsymbol{c}(\boldsymbol{a}, \boldsymbol{h}) + \boldsymbol{W}_n\boldsymbol{s}_a(|\boldsymbol{a}|)) \quad (46)$$

$$\boldsymbol{z}_q(\boldsymbol{q}) = \boldsymbol{s}_q(|\boldsymbol{q}|) \quad (47)$$

$$p_\theta(\boldsymbol{y} = 1|\boldsymbol{q}, \boldsymbol{a}, \boldsymbol{h}) = \sigma(\boldsymbol{z}_q^T\boldsymbol{M}\boldsymbol{z}_a + b) \quad (48)$$

Neural Variational Inference for Text Processing

Question Answering: Experiments

Source	Set	Questions	QA Pairs	Judgement
QASent	Train	1,229	53,417	automatic
	Dev	82	1,148	manual
	Test	100	1,517	manual
WikiQA	Train	2,118	20,360	manual
	Dev	296	2,733	manual
	Test	633	6,165	manual

Model	QASent		WikiQA	
	MAP	MRR	MAP	MRR
Published Models				
PV	0.5213	0.6023	0.5110	0.5160
Bigram-CNN	0.5693	0.6613	0.6190	0.6281
Deep CNN	0.5719	0.6621	—	—
PV + Cnt	0.6762	0.7514	0.5976	0.6058
WA	0.7063	0.7740	—	—
LCLR	0.7092	0.7700	0.5993	0.6068
Bigram-CNN + Cnt	0.7113	0.7846	0.6520	0.6652
Deep CNN + Cnt	0.7186	0.7826	—	—
Our Models				
LSTM	0.6436	0.7235	0.6552	0.6747
LSTM + Att	0.6451	0.7316	0.6639	0.6828
NASM	**0.6501**	**0.7324**	**0.6705**	**0.6914**
LSTM + Cnt	0.7228	0.7986	0.6820	0.6988
LSTM + Att + Cnt	0.7289	0.8072	0.6855	0.7041
NASM + Cnt	**0.7339**	**0.8117**	**0.6886**	**0.7069**

Q1	**how old was sue lyon when she made lolita**
A_{NASM}	**the actress who played lolita , sue lyon , was fourteen at the time of filming .**
A_{LSTM}	**the actress who played lolita , sue lyon , was fourteen at the time of filming .**
Q2	**how much is centavos in mexico**
A_{NASM}	**the peso is subdivided into 100 centavos , represented by " _UNK_ "**
A_{LSTM}	**the peso is subdivided into 100 centavos , represented by " _UNK_ "**
Q3	**what does a liquid oxygen plant look like**
A_{NASM}	**the blue color of liquid oxygen in a dewar flask**
A_{LSTM}	**the blue color of liquid oxygen in a dewar flask**

Figure 4. A visualisation of attention scores on answer sentences.

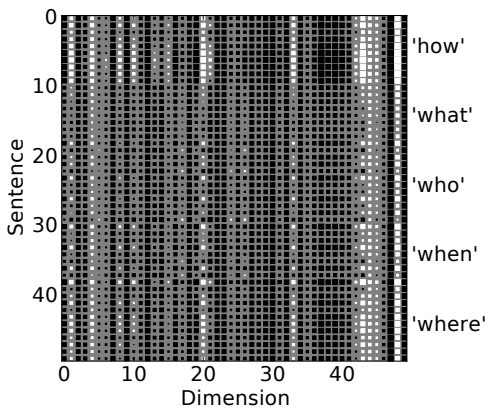
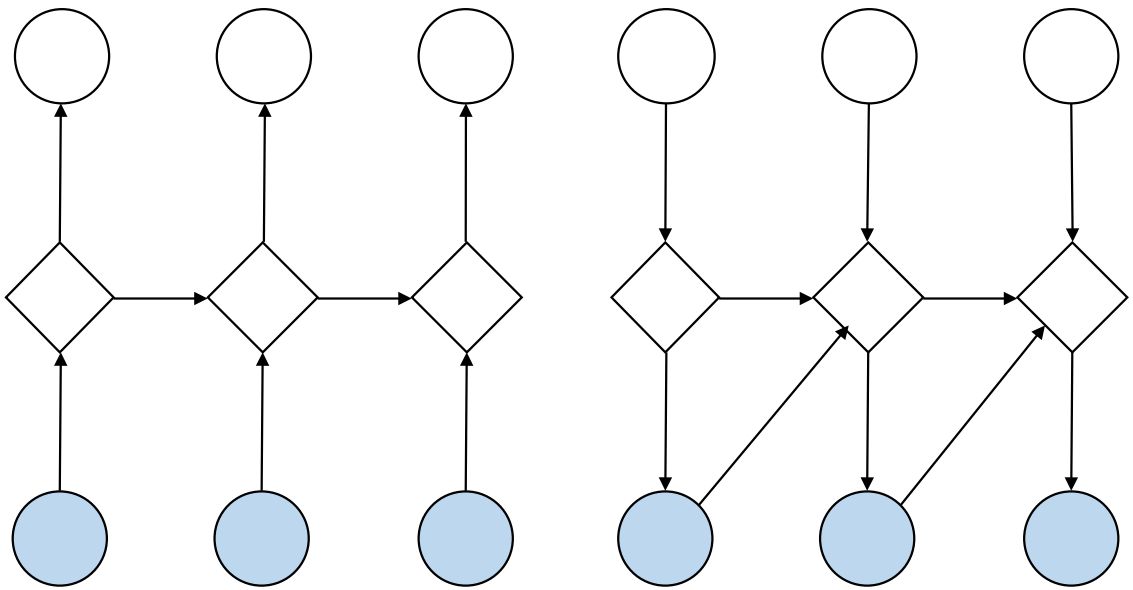


Figure 5. Hinton diagrams of the log standard deviations.

Learning Stochastic Recurrent Networks

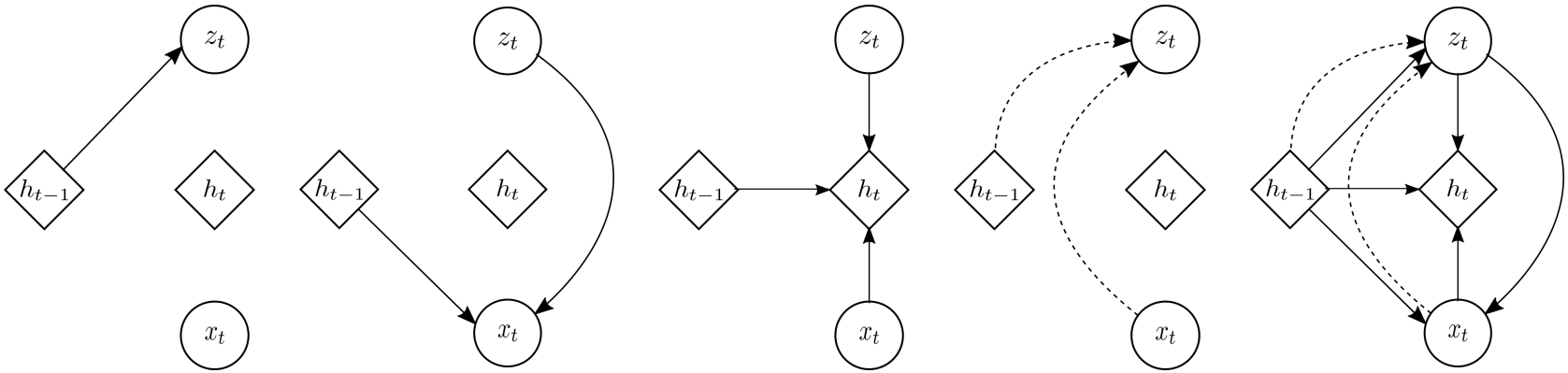
ICLR 2015 Workshop



(a) Inference Network (b) Generative Model

A Recurrent Latent Variable Model for Sequential Data

Model: Generation



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

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

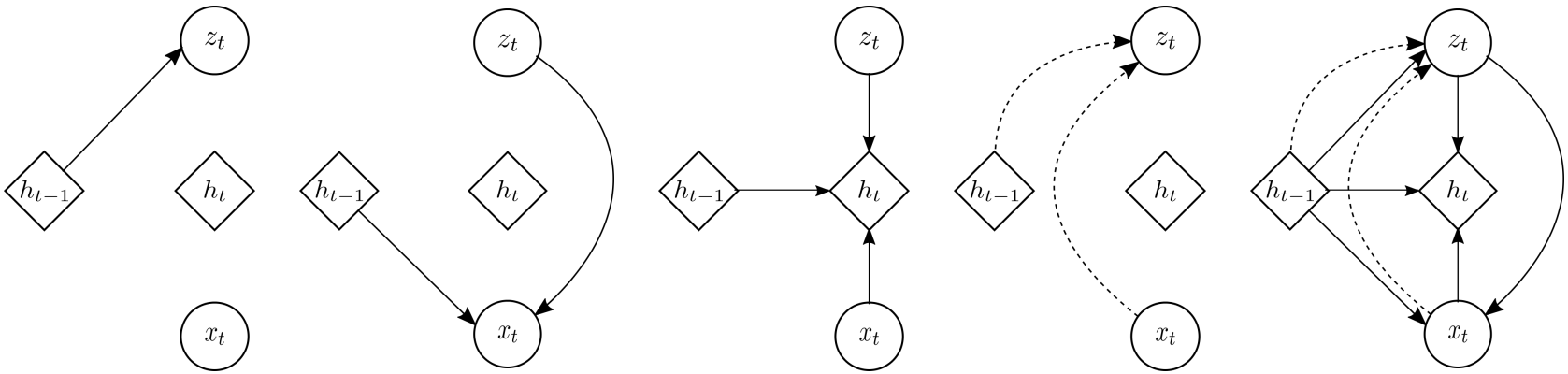
$$\boldsymbol{z}_t \sim \mathcal{N}(\boldsymbol{\mu}_{0,t}, \text{diag}(\boldsymbol{\sigma}_{0,t}^2)), \text{where} \quad [\boldsymbol{\mu}_{0,t}, \boldsymbol{\sigma}_{0,t}] = \phi_\tau^{prior}(\boldsymbol{h}_{t-1})$$

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

$$\boldsymbol{h}_t = f_\theta(\phi_\tau^{\boldsymbol{x}}(\boldsymbol{x}_t), \phi_\tau^{\boldsymbol{z}}(\boldsymbol{z}_t), \boldsymbol{h}_{t-1})$$

A Recurrent Latent Variable Model for Sequential Data

Model: Inference



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

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

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

A Recurrent Latent Variable Model for Sequential Data

Experiments

- Speech Modeling: train the model directly on raw audio signal, represented as a sequence of 200-d frames.
  - Blizzard: 300 hours of English spoken by a single female speaker.
  - TIMIT: 6300 English sentences, read by 630 speakers.
  - Onomatopoeia: 6738 non-linguistic human-made sounds, such as coughing, screaming, laughing and shouting, recorded from 51 vocie actors.
  - Accent: English paragraphs read by 2046 different English speakers.
- Handwriting Generation
  - IAM-OnDB: 13040 handwritten lines written by 500 writers.

A Recurrent Latent Variable Model for Sequential Data

Experiments

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	$\geq$ 8933 $\approx$ 9188	$\geq$ 28340 $\approx$ 29639	$\geq$ 19053 $\approx$ 19638	$\geq$ 3843 $\approx$ 4180	$\geq$ 1332 $\approx$ 1353
VRNN-Gauss	$\geq$ 9223 $\approx$ **9516**	$\geq$ 28805 $\approx$ **30235**	$\geq$ 20721 $\approx$ **21332**	$\geq$ 3952 $\approx$ 4223	$\geq$ 1337 $\approx$ 1354
VRNN-GMM	$\geq$ 9107 $\approx$ 9392	$\geq$ 28982 $\approx$ 29604	$\geq$ 20849 $\approx$ 21219	$\geq$ 4140 $\approx$ **4319**	$\geq$ 1384 $\approx$ **1384**

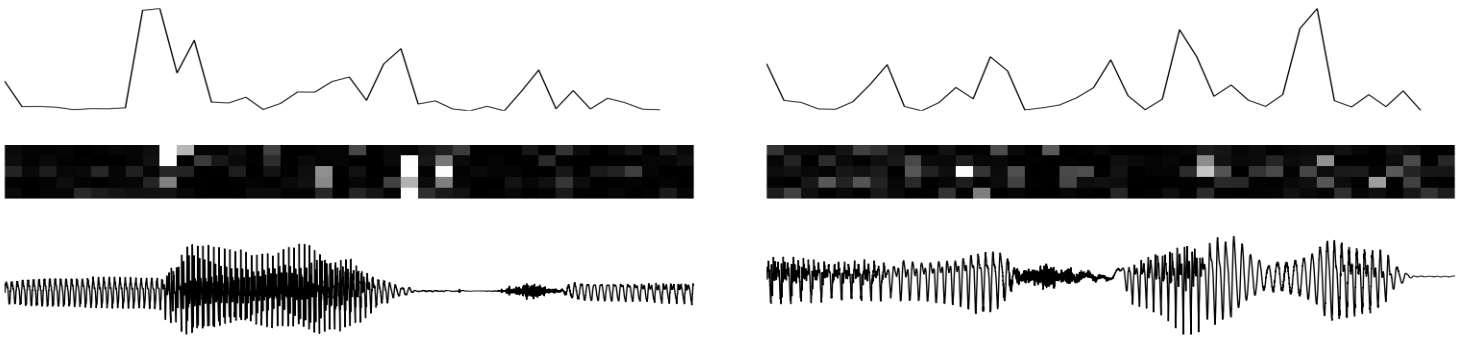
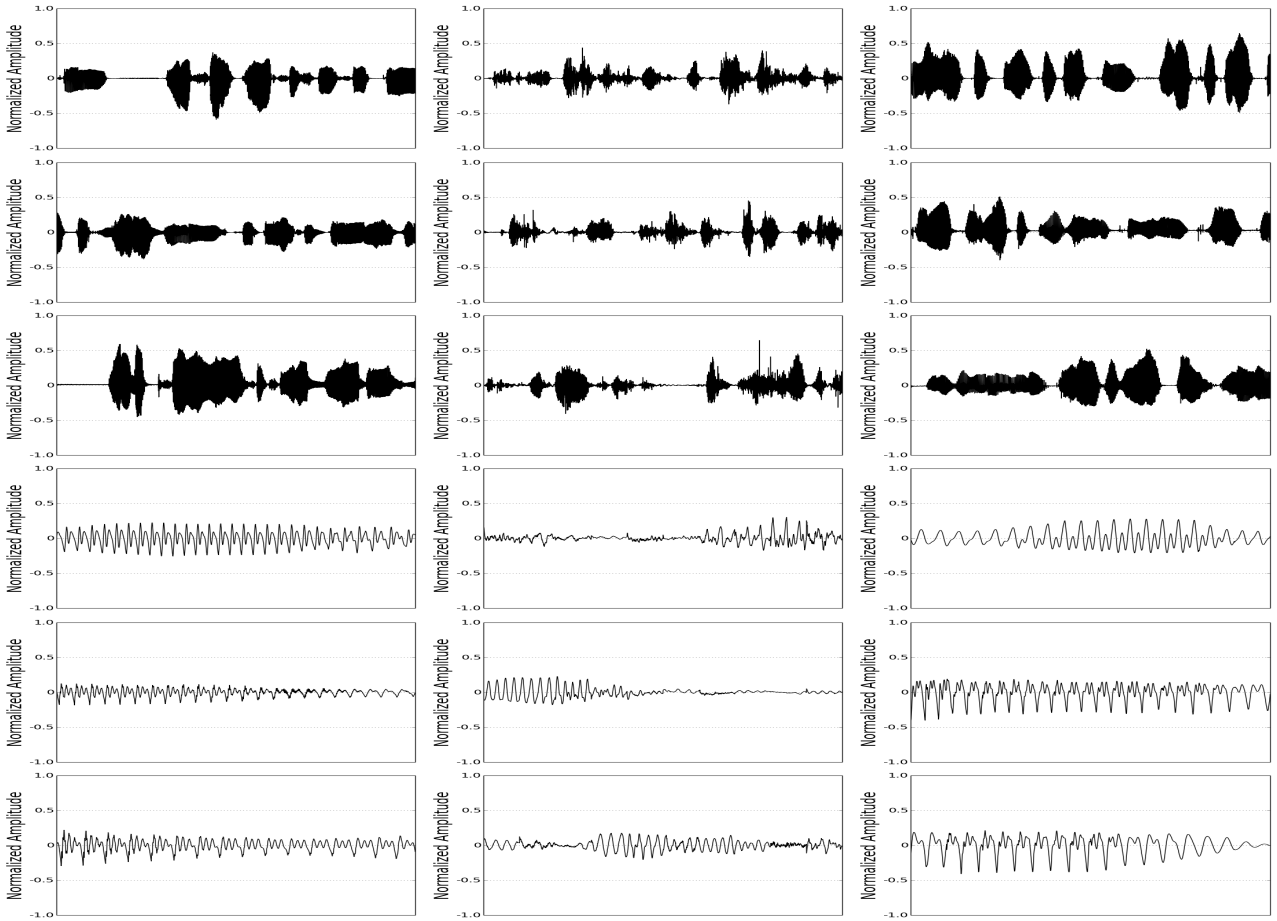


Figure 2: The top row represents the difference δ_t between $\boldsymbol{\mu}_{z,t}$ and $\boldsymbol{\mu}_{z,t-1}$. The middle row shows the dominant KL divergence values in temporal order. The bottom row shows the input waveforms.

A Recurrent Latent Variable Model for Sequential Data

Experiments

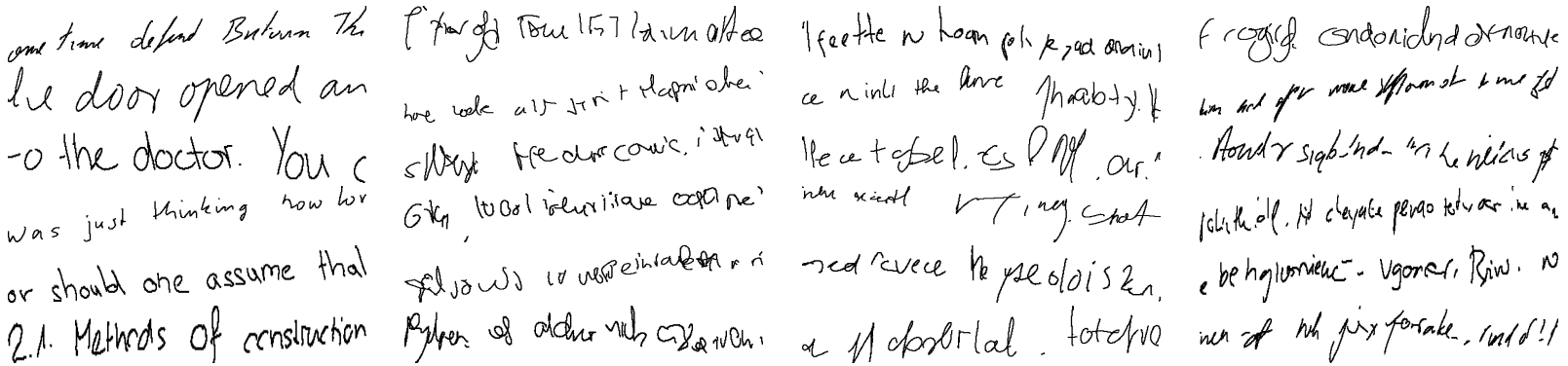


(a) Ground Truth (b) RNN-GMM (c) VRNN-Gauss

Figure 3: Examples from the training set and generated samples from RNN-GMM and VRNN-Gauss. Top three rows show the global waveforms while the bottom three rows show more zoomed-in waveforms. Samples from (b) RNN-GMM contain high-frequency noise, and samples from (c) VRNN-Gauss have less noise. We exclude RNN-Gauss, because the samples are almost close to pure noise.

A Recurrent Latent Variable Model for Sequential Data

Experiments

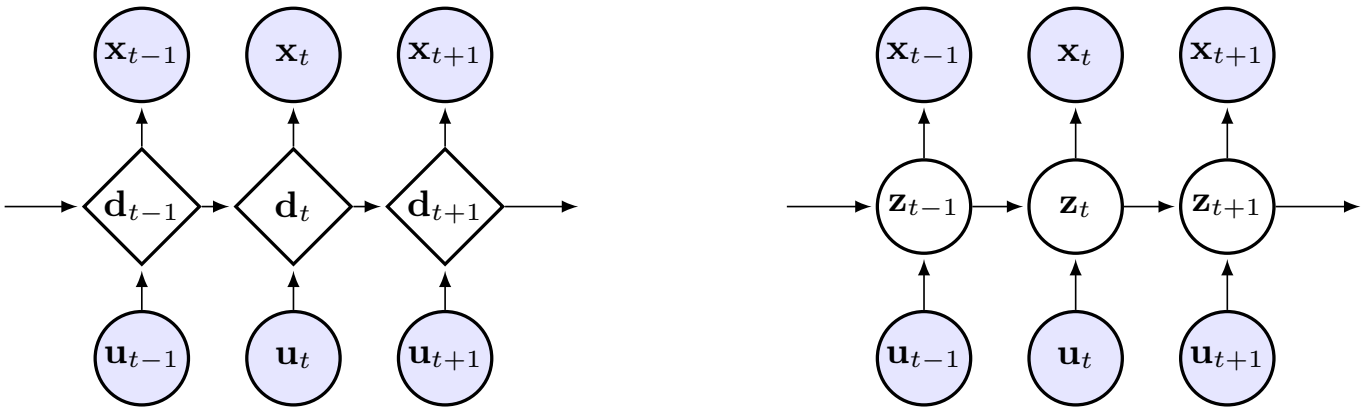


(a) Ground Truth (b) RNN-Gauss (c) RNN-GMM (d) VRNN-GMM

Figure 4: Handwriting samples: (a) training examples and unconditionally generated handwriting from (b) RNN-Gauss, (c) RNN-GMM and (d) VRNN-GMM. The VRNN-GMM retains the writing style from beginning to end while RNN-Gauss and RNN-GMM tend to change the writing style during the generation process. This is possibly because the sequential latent random variables can guide the model to generate each sample with a consistent writing style.

Sequential Neural Models with Stochastic Layers

Model



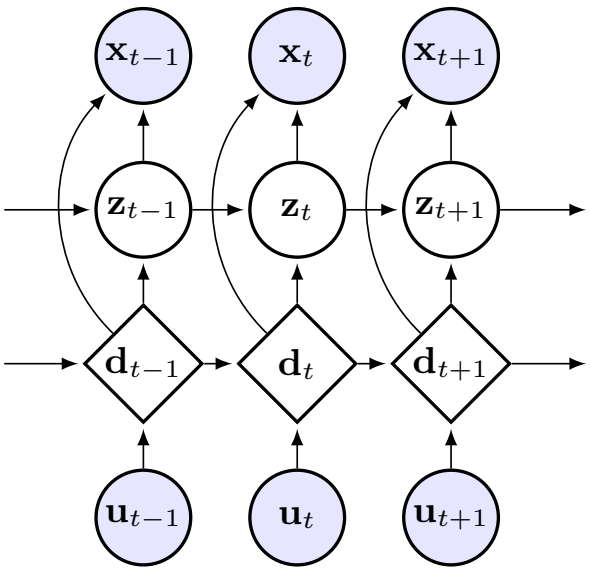
(a) RNN (b) SSM

Figure 1: Graphical models to generate $\mathbf{x}_{1:T}$ with a recurrent neural network (RNN) and a state space model (SSM). Diamond-shaped units are used for deterministic states, while circles are used for stochastic ones. For sequence generation, like in a language model, one can use $\mathbf{u}_t = \mathbf{x}_{t-1}$.

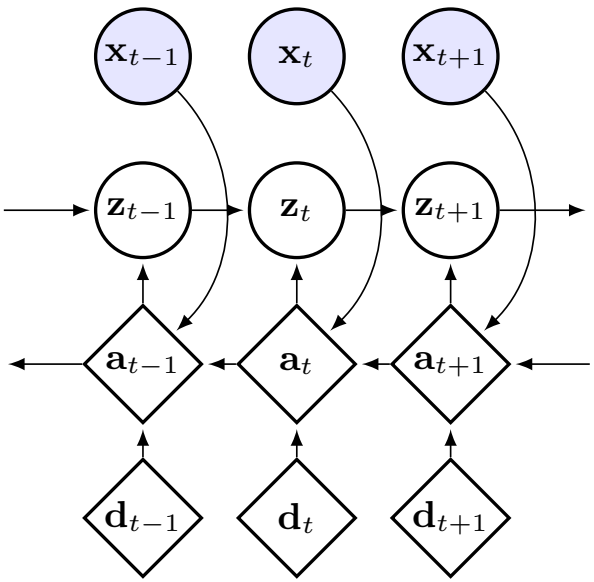
Sequential Neural Models with Stochastic Layers

Model

Like SSM, $\boldsymbol{z}_t$ directly depends on $\boldsymbol{z}_{t-1}$. The true posterior of $\boldsymbol{z}_t$ depends on $\boldsymbol{z}_{t-1}, \boldsymbol{d}_{t:T}, \boldsymbol{x}_{t:T}$.



(a) Generative model p_θ



(b) Inference network q_ϕ

Figure 2: A SRNN as a generative model p_θ for a sequence $\mathbf{x}_{1:T}$. Posterior inference of $\mathbf{z}_{1:T}$ and $\mathbf{d}_{1:T}$ is done through an inference network q_ϕ, which uses a backwards-recurrent state $\mathbf{a}_t$ to approximate the nonlinear dependence of $\mathbf{z}_t$ on future observations $\mathbf{x}_{t:T}$ and states $\mathbf{d}_{t:T}$; see Equation (7).

Sequential Neural Models with Stochastic Layers

Experiments

Models	Blizzard	TIMIT
SRNN (smooth+Res$_q$)	$\geq$**11991**	$\geq$ **60550**
SRNN (smooth)	$\geq$ 10991	$\geq$ 59269
SRNN (filt+Res$_q$)	$\geq$ 10572	$\geq$ 52126
SRNN (filt)	$\geq$ 10846	$\geq$ 50524
VRNN-GMM	$\geq$ 9107	$\geq$ 28982
	$\approx$ 9392	$\approx$ 29604
VRNN-Gauss	$\geq$ 9223	$\geq$ 28805
	$\approx$ 9516	$\approx$ 30235
VRNN-I-Gauss	$\geq$ 8933	$\geq$ 28340
	$\approx$ 9188	$\approx$ 29639
RNN-GMM	7413	26643
RNN-Gauss	3539	-1900

Table 1: Average log-likelihood per sequence on the test sets. For TIMIT the average test set length is 3.1s, while the Blizzard sequences are all 0.5s long. The non-SRNN results are reported as in [8]. *Smooth*: g_{ϕ_a} is a GRU running backwards; *filt*: g_{ϕ_a} is a feed-forward network; Res$_q$: parameterization with residual in (12).

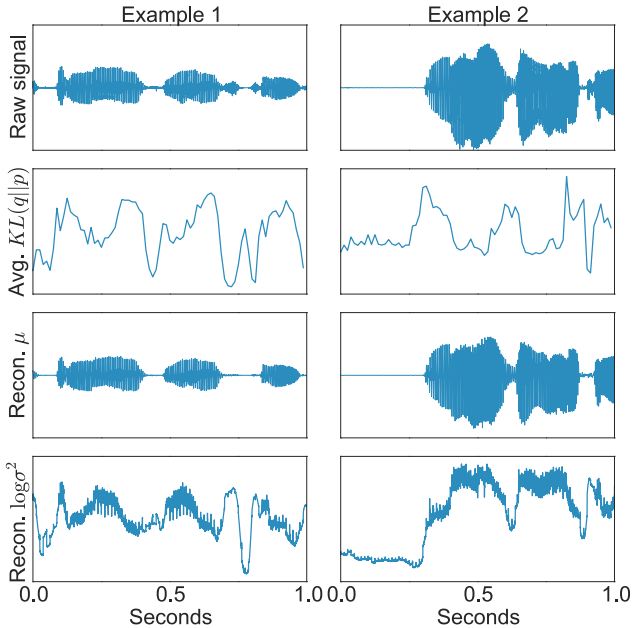


Figure 3: Visualization of the average KL term and reconstructions of the output mean and log-variance for two examples from the Blizzard test set.

Models	Nottingham	JSB chorales	MuseData	Piano-midi.de
SRNN (smooth+Res$_q$)	≥ -2.94	≥ -4.74	≥ -6.28	≥ -8.20
TSBN	≥ -3.67	≥ -7.48	≥ -6.81	≥ -7.98
NASMC	≈ -2.72	$\approx$ **-3.99**	≈ -6.89	≈ -7.61
STORN	≈ -2.85	≈ -6.91	≈ -6.16	≈ -7.13
RNN-NADE	$\approx$ **-2.31**	≈ -5.19	$\approx$ **-5.60**	$\approx$ **-7.05**
RNN	≈ -4.46	≈ -8.71	≈ -8.13	≈ -8.37

Table 2: Average log-likelihood on the test sets. The TSBN results are from [13], NASMC from [16], STORN from [3], RNN-NADE and RNN from [4].

Neural Adaptive Sequential Monte Carlo

Methods: Sequential Monte Carlo

Consider a probabilistic model

$$p(\boldsymbol{z}_{1:T}, \boldsymbol{x}_{1:T}) = p(\boldsymbol{z}_1)p(\boldsymbol{x}_1|\boldsymbol{z}_1)\prod_{t=2}^{T} p(\boldsymbol{z}_t|\boldsymbol{z}_{1:t-1})p(\boldsymbol{x}_t|\boldsymbol{z}_{1:t}, \boldsymbol{x}_{1:t-1})$$

The goal of SMC is to approximate the posterior as

$$p(\boldsymbol{z}_{1:T}|\boldsymbol{x}_{1:T}) \approx \sum_{n=1}^{N} \tilde{w}_t^{(n)} \delta(\boldsymbol{z}_{1:T} - \boldsymbol{z}_{1:T}^{(n)}) \tag{4}$$

through a weighted set of N sampled trajectories drawn from a proposal distribution $\{\boldsymbol{z}_{1:T}^{(n)}\}_{n=1:N} \sim q(\boldsymbol{z}_{1:T}|\boldsymbol{x}_{1:T})$,

$$q(\boldsymbol{z}_{1:T}|\boldsymbol{x}_{1:T}) = q(\boldsymbol{z}_1|\boldsymbol{x}_1)\prod_{t=2}^{T} q(\boldsymbol{z}_t|\boldsymbol{z}_{1:t-1}, \boldsymbol{x}_{1:t}) \tag{5}$$

Neural Adaptive Sequential Monte Carlo

Methods

How to get normalized importance weights:

$$w(\boldsymbol{z}_{1:T}^{(n)}) = \frac{p(\boldsymbol{z}_{1:T}^{(n)}, \boldsymbol{x}_{1:T})}{q(\boldsymbol{z}_{1:T}^{(n)}|\boldsymbol{x}_{1:T})},\ \tilde{w}(\boldsymbol{z}_{1:T}^{(n)}) = \frac{w(\boldsymbol{z}_{1:T}^{(n)})}{\sum_n w(\boldsymbol{z}_{1:T}^{(n)})} \propto \tilde{w}(\boldsymbol{z}_{1:T-1}^{(n)}) \frac{p(\boldsymbol{z}_T^{(n)}|\boldsymbol{z}_{1:T-1}^{(n)}) p(\boldsymbol{x}_T|\boldsymbol{z}_{1:T}^{(n)}, \boldsymbol{x}_{1:T-1})}{q(\boldsymbol{z}_T^{(n)}|\boldsymbol{z}_{1:T-1}^{(n)}, \boldsymbol{x}_{1:T})}$$

using Sequential Importance Sampling (SIS), or Sequential Importance Resampling (SIR).

Adapting proposals by descending the inclusive KL divergence

$$-\frac{\partial}{\partial \phi} \mathrm{KL}[p_\theta(\boldsymbol{z}_{1:T}|\boldsymbol{x}_{1:T})||q_\phi(\boldsymbol{z}_{1:T}|\boldsymbol{x}_{1:T})] \approx \sum_t \sum_n \tilde{w}_t^{(n)} \frac{\partial}{\partial \phi} \log q_\phi(\boldsymbol{z}_t^{(n)}|\boldsymbol{x}_{1:t}, \boldsymbol{z}_{1:t-1}^{A_{t-1}^{(n)}}).$$

Model parameter learning by maximum likelihood

$$\frac{\partial}{\partial \theta} \log[p_\theta(\boldsymbol{x}_{1:T})] \approx \sum_t \sum_n \tilde{w}_t^{(n)} \frac{\partial}{\partial \theta} \log p_\theta(\boldsymbol{x}_t, \boldsymbol{z}_t^{(n)}|\boldsymbol{x}_{1:t-1}, \boldsymbol{z}_{1:t-1}^{A_{t-1}^{(n)}}).$$

Neural Adaptive Sequential Monte Carlo

Methods

Algorithm 1 Stochastic Gradient Adaptive SMC (batch inference and learning variants)

Require: proposal: q_ϕ, model: p_θ, observations: $X = \{\boldsymbol{x}_{1:T_j}\}_{j=1:M}$, number of particles: N

repeat

$\{\boldsymbol{x}_{1:T_j}^{(j)}\}_{j=1:m} \leftarrow \text{NextMiniBatch}(X)$

$\{\boldsymbol{z}_{1:t}^{(i,j)}, \tilde{w}_t^{(i,j)}\}_{i=1:N,j=1:m,t=1:T_j} \leftarrow \text{SMC}(\theta, \phi, N, \{\boldsymbol{x}_{1:T_j}^{(j)}\}_{j=1:m})$

$\triangle\phi = \sum_j \sum_{t=1}^{T_j} \sum_i \tilde{w}_t^{(i,j)} \frac{\partial}{\partial \phi} \log q_\phi(\boldsymbol{z}_t^{(i,j)} | \boldsymbol{x}_{1:t}^{(j)}, \boldsymbol{z}_{1:t-1}^{A_{t-1}^{(i,j)}})$

$\triangle\theta = \sum_j \sum_{t=1}^{T_j} \sum_i \tilde{w}_t^{(i,j)} \frac{\partial}{\partial \theta} \log p_\theta(\boldsymbol{x}_t^{(j)}, \boldsymbol{z}_t^{(i,j)} | \boldsymbol{x}_{1:t-1}^{(j)}, \boldsymbol{z}_{1:t-1}^{A_{t-1}^{(i,j)}})$ (optional)

$\phi \leftarrow \text{Optimize}(\phi, \triangle\phi)$

$\theta \leftarrow \text{Optimize}(\theta, \triangle\theta)$ (optional)

until convergence

Dataset	LV-RNN (NASMC)	LV-RNN (Bootstrap)	STORN (SGVB)	FD-RNN	sRNN	RNN-NADE
Piano-midi-de	7.61	7.50	**7.13**	7.39	7.58	7.03
Nottingham	**2.72**	3.33	2.85	3.09	3.43	2.31
MuseData	6.89	7.21	**6.16**	6.75	6.99	5.60
JSBChorales	**3.99**	4.26	6.91	8.01	8.58	5.19

Takeaways

- Modeling:
  1. RNN + stochastic latent variable
  2. applications in mocap, music, text, speech, handwriting, etc.
  3. mixture RNN model?
- Training:
  1. SGVB
  2. NASMC
  3. combine SMC + VAE?

References I

H. Wallach A. Schein and M. Zhou.

Poisson gamma dynamical systems.
In *NIPS*, 2016.

Ayan Acharya, Joydeep Ghosh, and Mingyuan Zhou.

Nonparametric bayesian factor analysis for dynamic count matrices.
In *AISTATS*, 2015.

Evan Archer, Il Memming Park, Lars Buesing, John Cunningham, and Liam Paninski.

Black box variational inference for state space models.
arXiv preprint arXiv:1511.07367, 2015.

Justin Bayer and Christian Osendorfer.

Learning stochastic recurrent networks.
In *ICLR workshop*, 2015.

Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent.

Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription.
In *ICML*, 2012.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio.

Generating sentences from a continuous space.
In *CoNLL*, 2016.

Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio.

A recurrent latent variable model for sequential data.
In *NIPS*, 2015.

References II

Otto Fabius and Joost R van Amersfoort.

Variational recurrent auto-encoders.

In *ICLR workshop*, 2015.

Marco Fraccaro, Søren Kaae Sønderby, Ulrich Paquet, and Ole Winther.

Sequential neural models with stochastic layers.

In *NIPS*, 2016.

Zhe Gan, Chunyuan Li, Ricardo Henao, David E Carlson, and Lawrence Carin.

Deep temporal sigmoid belief networks for sequence modeling.

In *NIPS*, 2015.

Shixiang Gu, Zoubin Ghahramani, and Richard E Turner.

Neural adaptive sequential monte carlo.

In *NIPS*, 2015.

Matthew J Johnson, David Duvenaud, Alexander B Wiltschko, Sandeep R Datta, and Ryan P Adams.

Structured vaes: Composing probabilistic graphical models and variational autoencoders.

In *NIPS*, 2016.

Maximilian Karl, Maximilian Soelch, Justin Bayer, and Patrick van der Smagt.

Deep variational bayes filters: Unsupervised learning of state space models from raw data.

arXiv preprint arXiv:1605.06432, 2016.

Rahul G Krishnan, Uri Shalit, and David Sontag.

Deep kalman filters.

arXiv preprint arXiv:1511.05121, 2015.

References III

Rahul G Krishnan, Uri Shalit, and David Sontag.

Structured inference networks for nonlinear state space models.
arXiv preprint arXiv:1609.09869, 2016.

Yishu Miao, Lei Yu, and Phil Blunsom.

Neural variational inference for text processing.
In *ICML*, 2016.

Roni Mittelman, Benjamin Kuipers, Silvio Savarese, and Honglak Lee.

Structured recurrent temporal restricted boltzmann machines.
In *ICML*, 2014.

Jiaming Song, Zhe Gan, and Lawrence Carin.

Factored temporal sigmoid belief networks for sequence learning.
In *ICML*, 2016.

Ilya Sutskever and Geoffrey E Hinton.

Learning multilevel distributed representations for high-dimensional sequences.
In *AISTATS*, 2007.

Ilya Sutskever, Geoffrey E Hinton, and Graham W Taylor.

The recurrent temporal restricted boltzmann machine.
In *NIPS*, 2009.

Graham W Taylor and Geoffrey E Hinton.

Factored conditional restricted boltzmann machines for modeling motion style.
In *ICML*, 2009.

References IV

Graham W Taylor, Geoffrey E Hinton, and Sam T Roweis.
Modeling human motion using binary latent variables.
In *NIPS*, 2006.

Graham W Taylor, Leonid Sigal, David J Fleet, and Geoffrey E Hinton.
Dynamical binary latent variable models for 3d human pose tracking.
In *CVPR*, 2010.

Matthew D Zeiler, Graham W Taylor, Leonid Sigal, Iain Matthews, and Rob Fergus.
Facial expression transfer with input-output temporal restricted boltzmann machines.
In *NIPS*, 2011.

Biao Zhang, Deyi Xiong, and Jinsong Su.
Variational neural machine translation.
In *EMNLP*, 2016.

Yizhe Zhang, Yue Zhao, Lawrence David, Ricardo Henao, and Lawrence Carin.
Dynamic poisson factor analysis.
In *ICDM*, 2016.