Advanced Topics on Sequence Generation

Lili Mou
doublepower.mou@gmail.com
http://sei.pku.edu.cn/~moull12

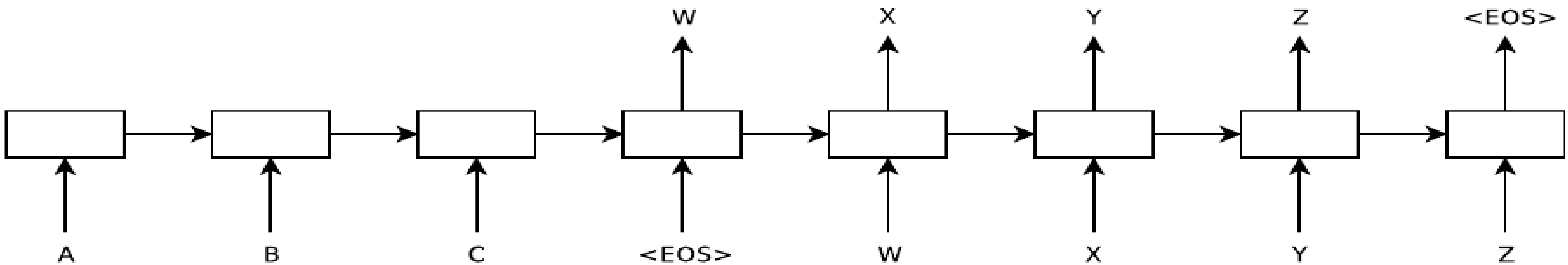
Outline

- **The Basics**
- BiRNN as Generators
- Reinforcement Learning
- Variational Autoencoder

The Basics

- Seq2seq
- Attention

Sequence to Sequence Generation



- Training phrase: X, Y, and Z are the ground truth (words in the corpus)
- Predicting phrase: X, Y, and Z are those generated by RNN
- Seq2seq model is essentially an LM (of XYZ) conditioned on another LM (of ABC)

Sutskever, Ilya, Oriol Vinyals, and Quoc V. Le. "Sequence to sequence learning with neural networks." NIPS. 2014.

The Attention Mechanism

- During sequence generation, the output sequence's hidden state $\boldsymbol{h}_t$ is related to
  - That of the last time step $\boldsymbol{h}_{t-1}$, and
  - A context vector ***c***, which is a combination of the input sequence's states

$$\boldsymbol{h}_t = \mathrm{RNN}(\boldsymbol{h}_{t-1}, \boldsymbol{c}) = f(W[\boldsymbol{h}_{t-1}; \boldsymbol{c}])$$

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).

Context Vector

The context vector $\boldsymbol{c}$ is a combination of the input sequence's states $$\boldsymbol{c} = \sum_i \alpha_i \boldsymbol{c}_i$$

where the coefficient α_i is related to

- The local context $\boldsymbol{c}_i$, and
- The last output state $\boldsymbol{h}_{t-1}$
- α_i is normalized

$$\alpha_i = \frac{\exp\{\tilde{\alpha}_i\}}{\sum_j \exp\{\tilde{\alpha}_j\}} \qquad \tilde{\alpha}_i = W[\boldsymbol{h}_{t-1}; \boldsymbol{c}_i]$$

But...

- Deep learning is far beyond CNNs, RNNs, etc.

Outline

- The Basics
- **BiRNN as Generators**
- Reinforcement Learning
- Variational Autoencoder

Bidirectional Recurrent Neural Networks as Generative Models

NIPS'15

Mathias Berglund
Aalto University, Finland

Tapani Raiko
Aalto University, Finland

Mikko Honkala
Nokia Labs, Finland

Leo Kärkkäinen
Nokia Labs, Finland

Akos Vetek
Nokia Labs, Finland

Juha Karhunen
Aalto University, Finland

- Motivation: Use LMs of both directions during language generation

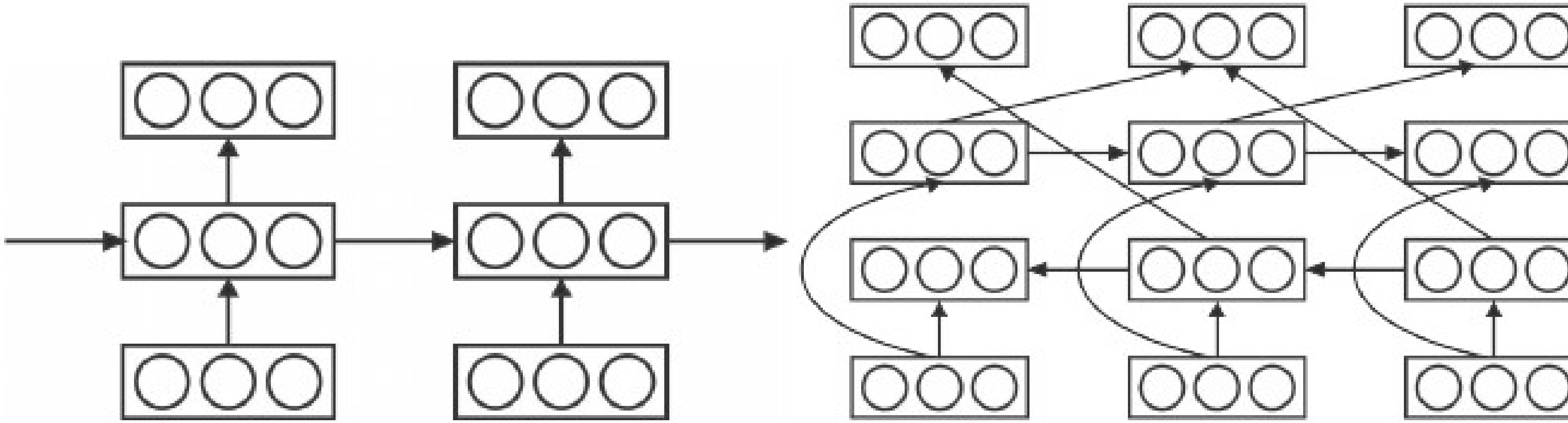


Figure 1: Structure of the simple RNN (left) and the bidirectional RNN (right).

The output $\mathbf{y}_t$ is traditionally determined by

$$P\left(\mathbf{y}_t \mid \{\mathbf{x}_d\}_{d\neq t}\right) = \phi\left(\mathbf{W}_{\mathrm{y}}^{\mathrm{f}}\mathbf{h}_t^{\mathrm{f}} + \mathbf{W}_{\mathrm{y}}^{\mathrm{b}}\mathbf{h}_t^{\mathrm{b}} + \mathbf{b}_y\right),$$

but we propose the use of

$$P\left(\mathbf{y}_t \mid \{\mathbf{x}_d\}_{d\neq t}\right) = \phi\left(\mathbf{W}_{\mathrm{y}}^{\mathrm{f}}\mathbf{h}_{t-1}^{\mathrm{f}} + \mathbf{W}_{\mathrm{y}}^{\mathrm{b}}\mathbf{h}_{t+1}^{\mathrm{b}} + \mathbf{b}_y\right)$$

where

$$\mathbf{h}_t^{\mathrm{f}} = \tanh\left(\mathbf{W}_{\mathrm{h}}^{\mathrm{f}}\mathbf{h}_{t-1}^{\mathrm{f}} + \mathbf{W}_{\mathrm{x}}^{\mathrm{f}}\mathbf{x}_t + \mathbf{b}_{\mathrm{h}}^{\mathrm{f}}\right)$$

$$\mathbf{h}_t^{\mathrm{b}} = \tanh\left(\mathbf{W}_{\mathrm{h}}^{\mathrm{b}}\mathbf{h}_{t+1}^{\mathrm{b}} + \mathbf{W}_{\mathrm{x}}^{\mathrm{b}}\mathbf{x}_t + \mathbf{b}_{\mathrm{h}}^{\mathrm{b}}\right).$$

Probabilistic Interpretation

- Unidirectional distributior $P_{\text{unidirectional}}(\mathbf{X}) = \prod_{t=1}^{T} P(\mathbf{x}_t \mid \{\mathbf{x}_d\}_{d=1}^{t-1})$
- Bidirectional distribution
  - Interpretation I (Generative Stochastic Networks, GSN)

    Asymptotic distribution when sampling $P_{\text{BRNN}}(\mathbf{x}_t \mid \{\mathbf{x}_d\}_{d \neq t})$

    Essentially the distribution defined by Gibbs sampling
  - Interpretation II (Neural Autoregressive Distribution Estimator, NADE)

    O_t: a permutation of time indexes 1..T

$$P_{\text{NADE}}(\mathbf{X} \mid o_d) = \prod_{d=1}^{T} P(\mathbf{x}_{o_d} \mid \{\mathbf{x}_{o_e}\}_{e=1}^{d-1})$$

Training and Inference

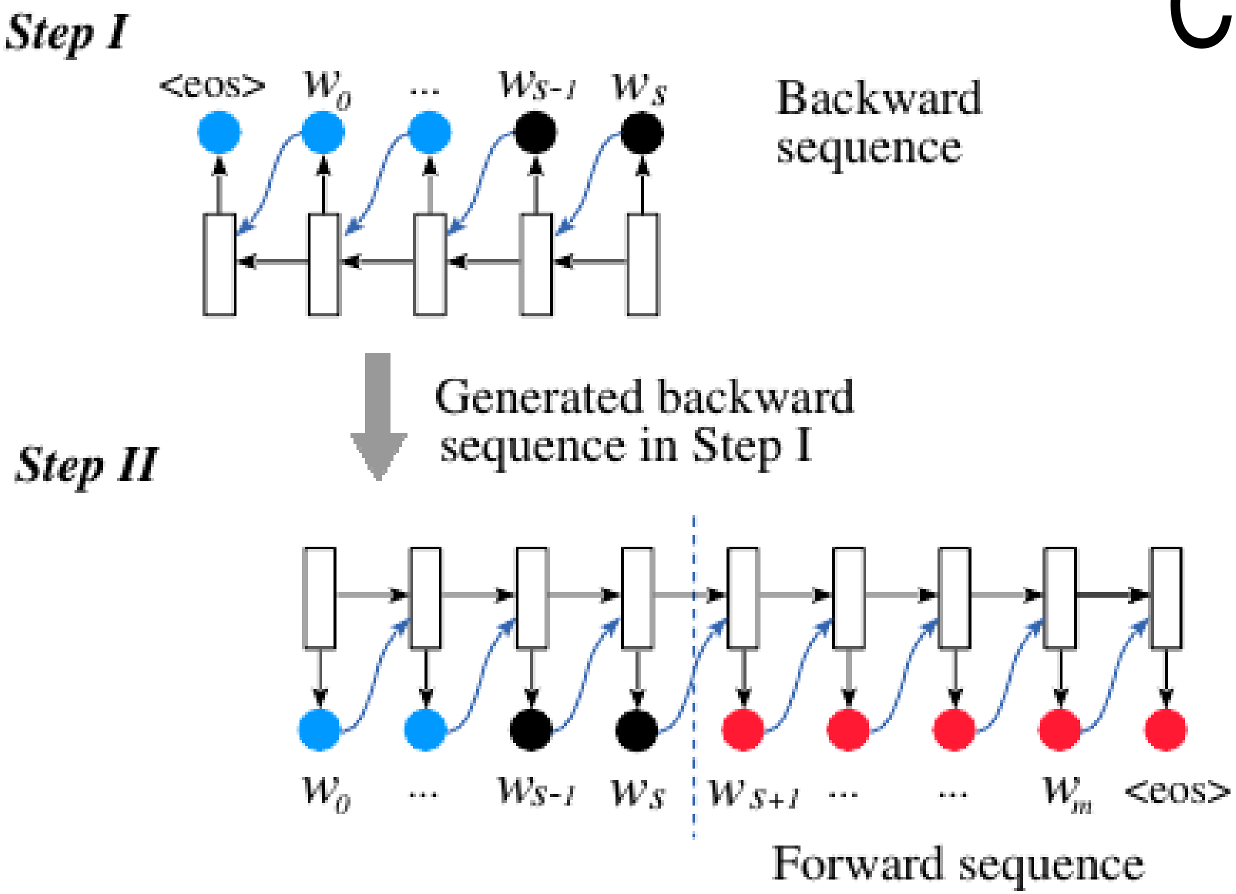
	Training	Inference
GSN	Assume the whole sentence is known	Gibbs sampling
NADE	Set some input to a missing value	Start from random initialization, and compute the prob. in a one-pass fashion
Bayesian MCMC		

$$
\begin{aligned}
& P_{\mathrm{RNN}}\left(\mathbf{x}_t = \mathbf{a} \mid \{\mathbf{x}_d\}_{d \neq t}\right) \\
& \quad \propto P_{\mathrm{RNN}}\left(\mathbf{x}_t = \mathbf{a} \mid \{\mathbf{x}_d\}_{d=1}^{t-1}\right) P_{\mathrm{RNN}}\left(\{\mathbf{x}_e\}_{e=t+1}^{T} \mid \mathbf{x}_t = \mathbf{a}, \{\mathbf{x}_d\}_{d=1}^{t-1}\right) \\
& \quad = \prod_{\tau=t}^{T} P_{\mathrm{RNN}}(\mathbf{x}_\tau \mid \{\mathbf{x}_d\}_{d=1}^{\tau-1})\Big|_{\mathbf{x}_t = \mathbf{a}}
\end{aligned}
$$

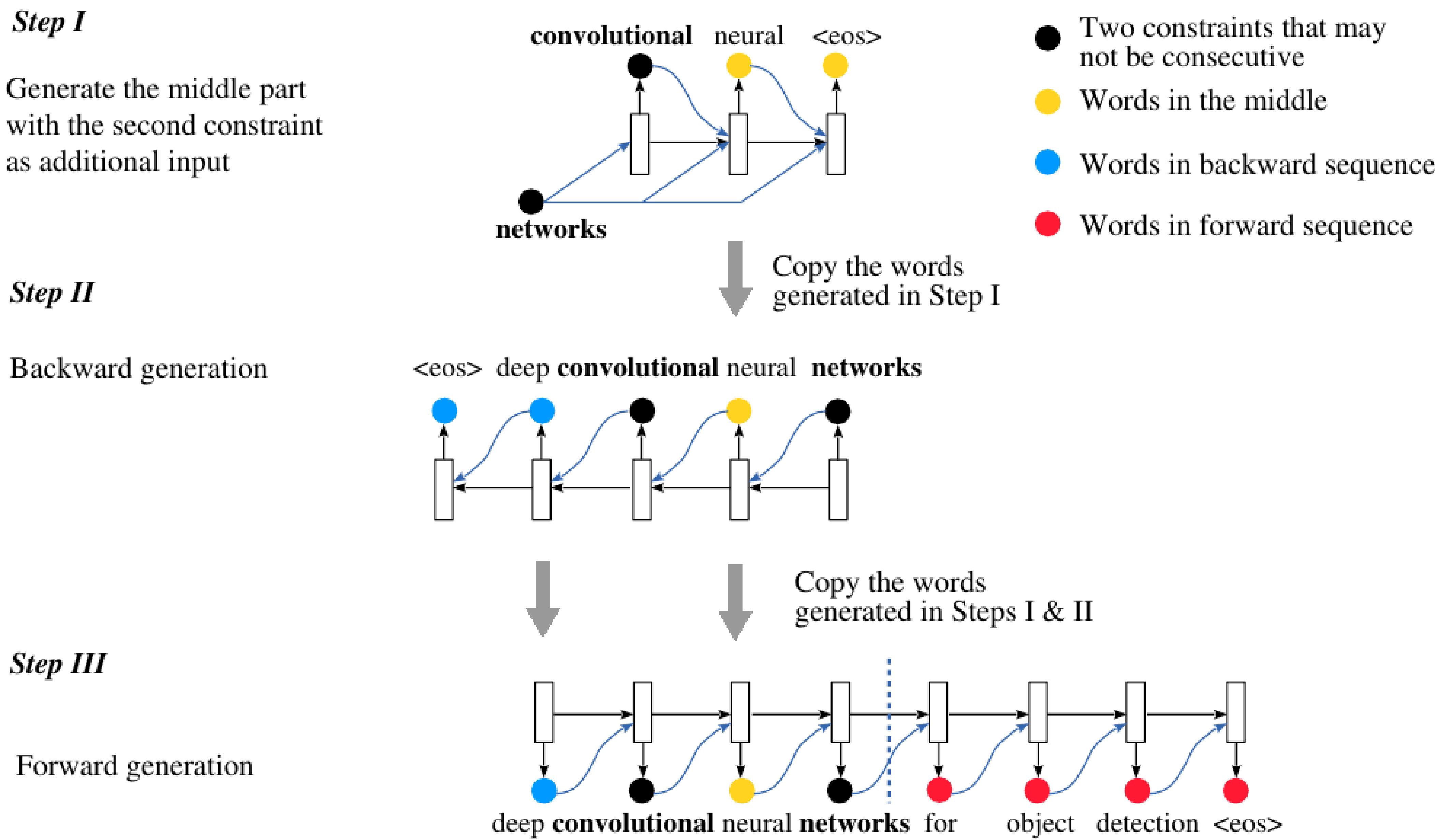
Pros and Cons

- Pros
  - 😋 The constraint of “sequence” is slacked
- Cons
  - 😿 Virtually impossible to be used in practice

C.f. our B/F LMs



stochastic **gradient** - based algorithm for <unk> - based convex optimization
gradient - free learning with a stochastic block model
on global convergence of **sub - gradient** descent for sparse graphs
a stochastic **gradient descent** algorithm for <unk> - <unk> systems
a **stochastic gradient descent** algorithm for <unk> - <unk> systems
deep **convolutional neural networks** for object detection
semi - supervised learning **with convolutional neural networks**
semantic image **segmentation with convolutional neural networks** for <unk>
efficient object **tracking with convolutional neural networks**
deep **convolutional neural networks for language**


Step I
Generate the middle part with the second constraint as additional input
convolutional neural <eos>
networks
Two constraints that may not be consecutive
Words in the middle
Words in backward sequence
Words in forward sequence
Copy the words generated in Step I
Step II
Backward generation
<eos> deep convolutional neural networks
Copy the words generated in Steps I & II
Step III
Forward generation
deep convolutional neural networks for object detection <eos>

a stochastic **gradient descent** algorithm for unk - unk systems
a **stochastic** gradient **descent** algorithm for unk - unk systems
on global convergence of **sub** - **gradient** descent for sparse graphs
convergence of the **gradient** descent algorithm for unk problems in the plane
stochastic **gradient** descent **for** non - convex optimization
stochastic gradient descent **for** non - convex optimization
the complexity of the unk **stochastic problem** and its effect
deep convolutional **neural networks** for object detection
deep **convolutional** neural **networks** for object detection
deep learning **for** unk
deep **convolutional** neural networks **with** unk features
convolutional neural **networks** for unk **detection**
object **detection** using deep **neural** networks
community **detection** in social **networks** : a survey

Outline

- The Basics
- BiRNN as Generators
- **Reinforcement Learning**
- Variational Autoencoder

Sequence-Level Training

- Motivation: We don't have the ground truth

  In a dialogue system, for example, “The nature of of open-domain conversations shows that a variety of replies are plausible, but some are more meaningful, and others are not.”

- Optimize the sequence generator as a whole in terms of external metrics

REINFORCE

Ranzato, Marc'Aurelio, et al. "Sequence Level Training with Recurrent Neural Networks." ICLR, 2016.

- Define an external cost function on a generated sequence
- Generate words by sampling
- Take the derivative of generated samples

$$L_\theta = -\sum_{w_1^g,\ldots,w_T^g} p_\theta(w_1^g,\ldots,w_T^g) r(w_1^g,\ldots,w_T^g) = -\mathbb{E}_{[w_1^g,\ldots w_T^g]\sim p_\theta} r(w_1^g,\ldots,w_T^g)$$

∂p(w)=p(w)∂log p(w) because p(w)=exp{log p(w)}

- $\partial J = \sum_{\mathbf{w}} [\partial p(\mathbf{w}|\ldots)]\, r(\mathbf{w}) = \sum_{\mathbf{w}} p(\mathbf{w})[\partial \log p(\mathbf{w})]\, r(\mathbf{w})$

$=$

$$\left(p_\theta(w_{t+1}|w_t^g, \mathbf{h}_{t+1}, \mathbf{c}_t) - \mathbf{1}(w_{t+1}^g)\right)$$

where $\mathbf{o}_t$ is the input to the softmax.

Tricks

- Redefine r as r – r_bar
  - r_bar: estimated average reward
- Combination of cross-entropy loss and external metrics
  - Simulated annealing

    First *n* words: cross entropy loss

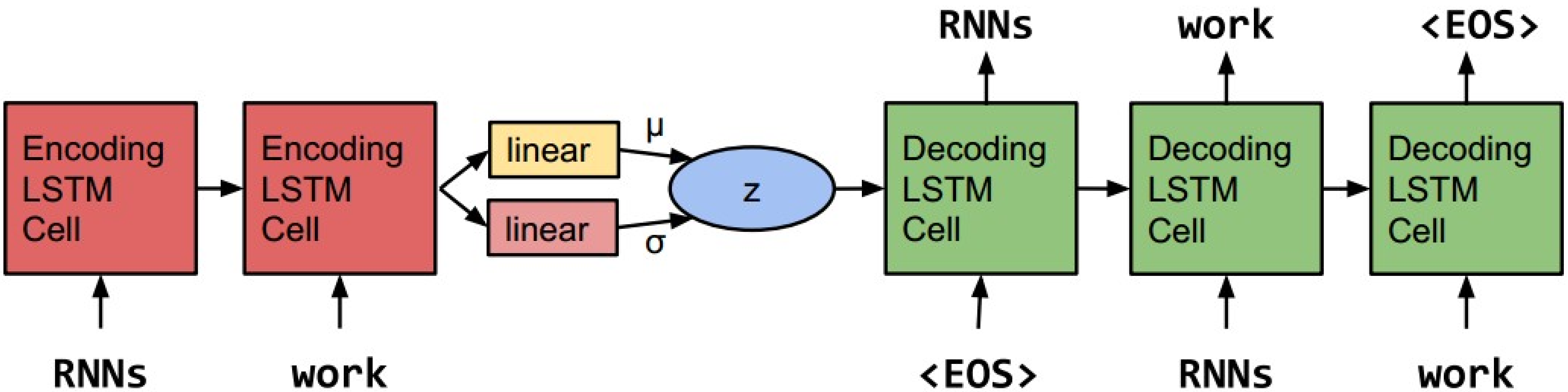
    Last N-n words: external reward

    Decrease n

Outline

- The Basics
- BiRNN as Generators
- Reinforcement Learning
- **Variational Autoencoder**

Variational Autoencoder



Penalize:

– Reconstruction error

– KL(posterior || prior)

A Very Brief (and maybe oversimplified) Introduction to VAE

- $\mathbf{x} = \{x_1, x_2, \cdots\}$ is observed variables
- $\mathbf{z} = \{z_1, z_2, \cdots\}$ is hidden/latent variables

$$\mathbf{z} \rightarrow \mathbf{x}$$

—Karl Marx

How God creates the world

Rain → Wet

- The marginal distribution of $\mathbf{x}$ is defined as

$$p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{z})\, d\mathbf{z} = \int p(\mathbf{z}) p(\mathbf{x}|\mathbf{z})\, d\mathbf{z}$$

or

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

where

$$KL(q(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z}|\mathbf{x})) = \int q(\mathbf{z}|\mathbf{x}) \log \left\{ \frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{x})} \right\} d\mathbf{z} \quad (\geq 0)$$

$$\mathcal{L}(\boldsymbol{\theta}, \phi; \mathbf{x}) = \int q(\mathbf{z}|\mathbf{x}) \log \left\{ \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})} \right\} d\mathbf{z}$$

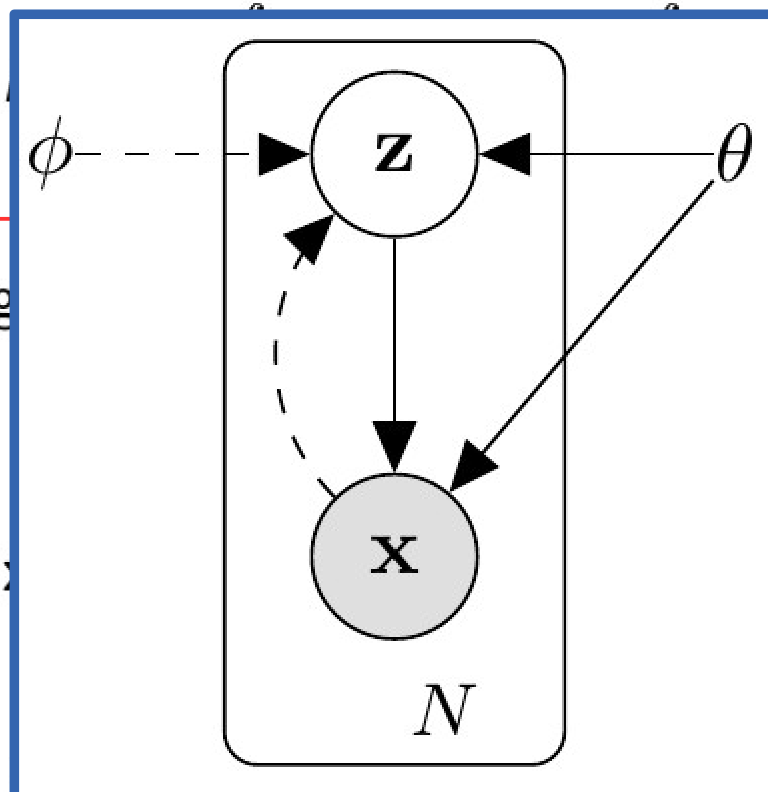
How man recognizes the world

Wet → Rain

How God creates the world

Rain → Wet

- The marginal distribution of $\mathbf{x}$ is defined as



$$)p(\mathbf{x}|\mathbf{z})\ d\mathbf{z}$$

or

$$\log \quad + \mathcal{L}(\boldsymbol{\theta}, \phi; \mathbf{x})$$

where

$$KL(q(\mathbf{z}|\quad \frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{x})} \Big\} \ d\mathbf{z} \quad (\geq 0)$$

$$\frac{(\mathbf{x},\mathbf{z})}{q(\mathbf{z})} \Big\} \ d\mathbf{z}$$

How man recognizes the world

Wet → Rain

- The marginal distribution of $\mathbf{x}$ is defined as

$$p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{z})\ \mathrm{d}\mathbf{z} = \int p(\mathbf{z})p(\mathbf{x}|\mathbf{z})\ \mathrm{d}\mathbf{z}$$

or

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

where

=log p(x)

$$KL(q(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z}|\mathbf{x})) = \int q(\mathbf{z}|\mathbf{x}) \log \left\{ \frac{q(\mathbf{z})}{p(\mathbf{z}|\mathbf{x})} \right\} \mathrm{d}\mathbf{z} \quad (\geq 0)$$

$$\mathcal{L}(\boldsymbol{\theta}, \phi; \mathbf{x}) = \int q(\mathbf{z}|\mathbf{x}) \log \left\{ \frac{p(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})} \right\} \mathrm{d}\mathbf{z}$$

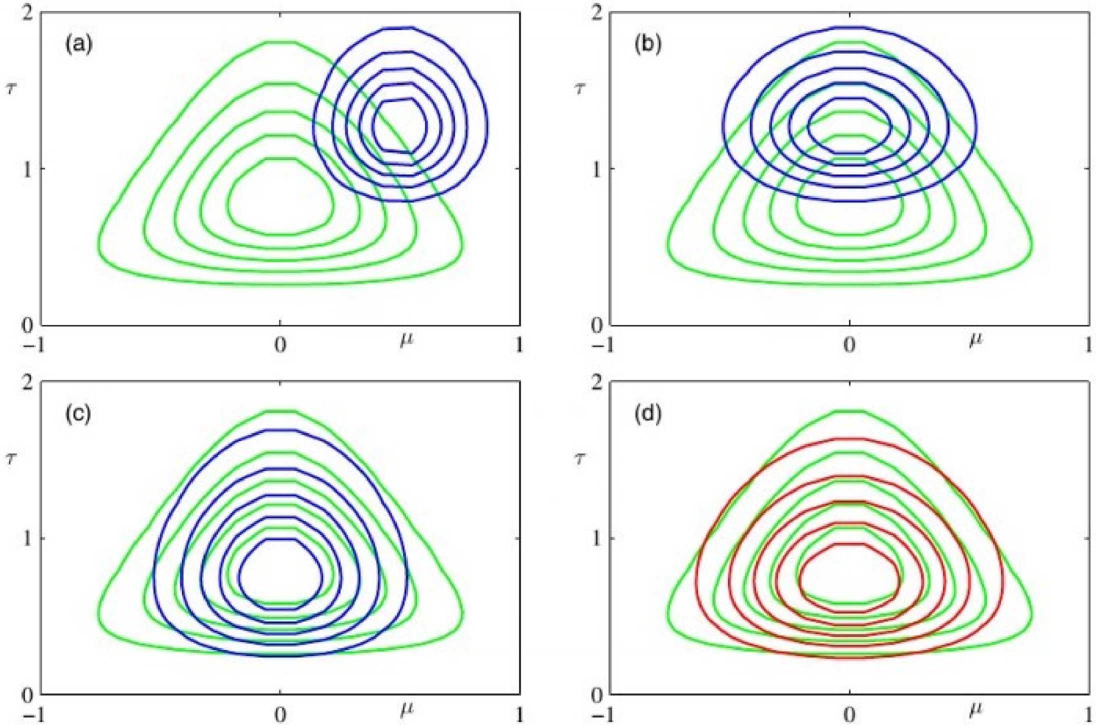
Factorized Posterior Assumption

Assumption

$$q(\boldsymbol{Z}) = \prod_{i=1}^{M} q_i(\boldsymbol{Z}_i)$$

Optimize $\mathcal{L}(q)$ w.r.t a group $\boldsymbol{Z}_j$ at a time

A case study of Gaussian-Gamma distribution



Variational Autoencoder

- Reparametrization

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

where $g(\cdot)$ is a deterministic function and $\epsilon \sim \mathcal{N}$

- Use NNs to recognize Z and then reconstruct X

$$\text{Input} \xrightarrow{\text{NN}} \phi \xrightarrow{\text{Sampling}} \text{Sample} \xrightarrow{\text{NN}} \text{X}$$

Variational Autoencoder

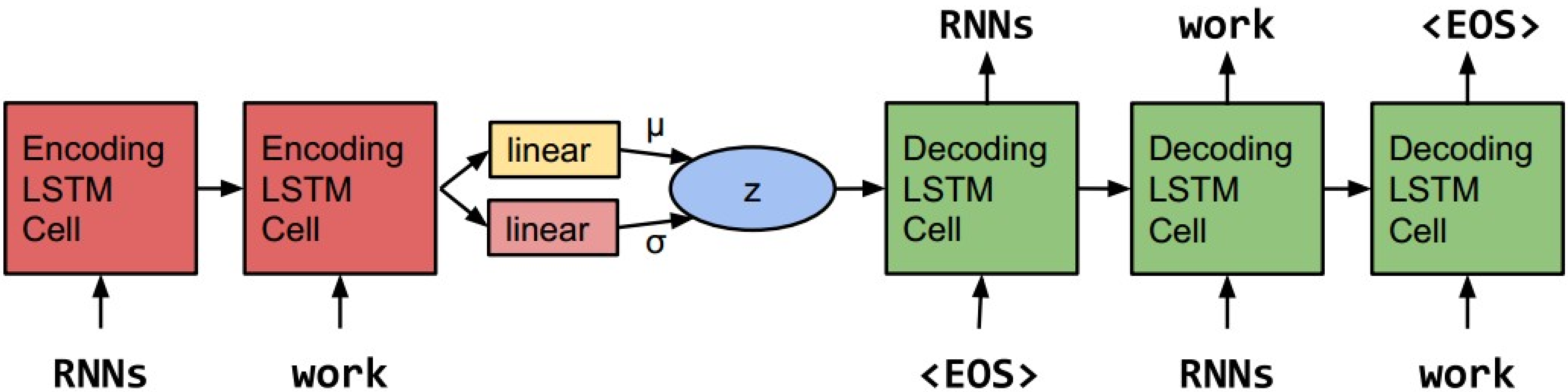
$$p(\mathbf{x}) \geq \mathcal{L}(\boldsymbol{\theta}, \phi; \mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[-\log q_\phi(\mathbf{z}|\mathbf{x}) + \log p_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})\right]$$

$$= -KL(q_\phi(\mathbf{z}|\mathbf{x}) \| p(\mathbf{z})) + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})\right]$$

$$\mathcal{L}(\boldsymbol{\theta}, \phi; \mathbf{x}^{(i)}) \simeq \frac{1}{2}\sum_{j=1}^{J}\left(1 + \log((\sigma_j^{(i)})^2) - (\mu_j^{(i)})^2 - (\sigma_j^{(i)})^2\right) + \frac{1}{L}\sum_{l=1}^{L}\log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}|\mathbf{z}^{(i,l)})$$

$$\text{Input} \xrightarrow{\text{NN}} \phi \xrightarrow{\text{Sampling}} \text{Sample} \xrightarrow{\text{NN}} \text{X}$$

Variational Autoencoder



$$\text{Input} \xrightarrow{\text{NN}} \phi \xrightarrow{\text{Sampling}} \text{Sample} \xrightarrow{\text{NN}} X$$

Penalize:

– Reconstruction error

– KL(posterior || prior)

A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues

Iulian V. Serban[*], Alessandro Sordoni[*], Ryan Lowe[◇], Laurent Charlin[◇], Joelle Pineau[◇], Aaron Courville[*] and Yoshua Bengio[‡]

Context	Response
jamaica blood!! <url> → lol. yes [name]!. [name]! → what's poppin in toronto homie	**VHRED:** nothing much. just chillin. what's good with you? **LSTM:** something!!!!!!! **HRED:** something!!!!!!
whhhhhhyyyy is it suddenly cold in tallahassee today?? this is florida! how can i dream of the beach when my fingers are icey cold?? → sittin on deck in sun looking at lake travis in austin! enjoying day b4 birthday	**VHRED:** the sun is shining!!! **LSTM:** i know!!!!!!!!! **HRED:** something!!!!!!!
reaches for [name]! → - approaches him warily - hey there buddy. - swings him into the air - how you doin? → tilts his head and your wariness and reaches for your face [name].	**VHRED:** - looks up at you and smiles - you're so cute. - kisses your cheek - **LSTM:** - smiles - i'm fine thank you. how are you? **HRED:** - grins, looking at you as i look at you -
i think i want an ipad2. but i already have an ibook. want vs. need. → if you have an ibook. then you deserve an ipad!	**VHRED:** that's what i'm thinking. i'm going to buy a new one. **LSTM:** i want one!!!! **HRED:** i don't know what to do with it. i don't know what to do.
ah echt? dan anrtwoord ik volgende keer via twitter als je me smst. → haha doe maar p. goed bezig hoor myrthe d → goed bezig? die snap ik even niet p	**VHRED:** haha, ik heb er geen zin in de something, maar ik heb er geen something meer something **LSTM:** haha something something something **HRED:** something something something something something something something something
i love katie waissel more than like, anything in life. → i wanna c the show again! i need <number> meet her again! n i want a b-day tweet nxt week doubt that will happen x	**VHRED:** me too. i'm going to have to wait for her to come back. i'm so excited. **LSTM:** i'm sure she will. i'm sure she'll be fine. **HRED:** i'm sure she'll be fine, i'm sure she'll be fine.
my math teacher over here talkn bout nickels nd i'm tryna make big money i need to be in calculus → lol	**VHRED:** i need to go to the gym and get my hair done lol **LSTM:** lol **HRED:** how u been

Thank you for listening!

Q&A and Discussion