ACL Tutorial T6:
Deep Bayesian Natural Language Processing

Jen-Tzung Chien

National Chiao Tung University

jtchien@nctu.edu.tw

July 28, 2019

Table of Contents

Outline

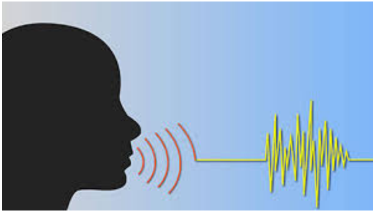
1. Deep Text Modeling
   - Natural language application
   - Probabilistic neural network
2. Deep Sequential Learning
3. Deep Stochastic Learning

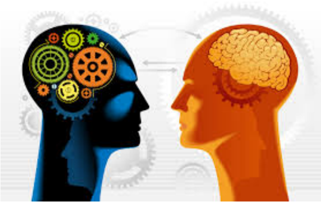
Outline

1. Deep Text Modeling
   - Natural language application
   - Probabilistic neural network
2. Deep Sequential Learning
3. Deep Stochastic Learning

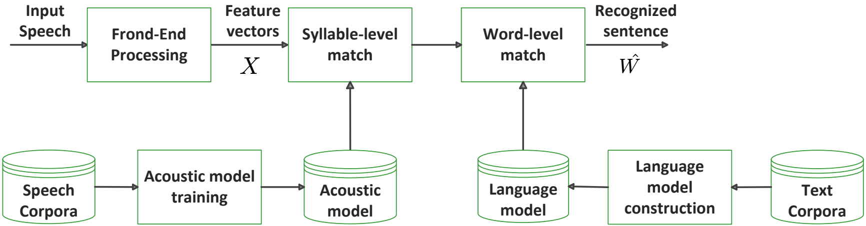
Speech and language

- Speech is the most natural way for communication
  - vocalized-form of communication
  - syntactic combination of lexicals
  - drawn from very large vocabularies
- Language is the ability to acquire and use complex systems of communication
  - natural language is a language used naturally by humans for communication





Speech recognition

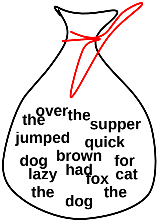


- Bayes decision rule

$$\hat{W} = \arg\max_{W} p(W|X) = \arg\max_{W} p(X|W)p(W)$$

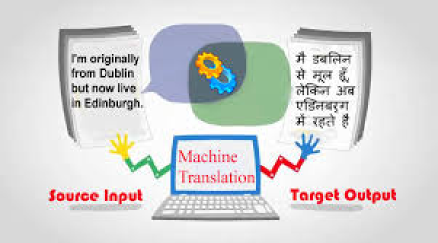
Document representation

- Document representation is developed for text analysis
- Topic-based text model
  - each document is treated as a bag of words
  - each document can exhibit multiple topics
- Symbolic model is required because
  - each topic is a multinomial variable
  - each document is represented by a multinomial mixture model
- Latent Dirichlet allocation (Blei et al., 2003) is popular to build the topic model



Machine translation

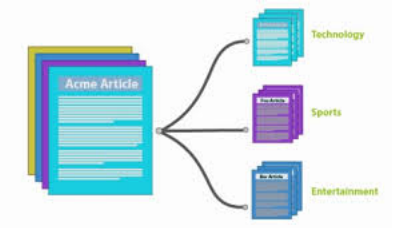
- Machine translation develops the algorithm to translate text or speech from one language to another
  - linguistic rules are helpful
  - statistical or corpus-based approach is popular



Information retrieval

- Document retrieval
  - ranking problem

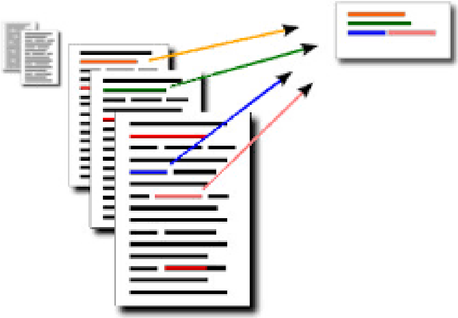
- Document categorization
  - classification problem



- Document representation or symbolic learning is a crucial issue

Document summarization

- Automatic summarization involves
  - a process of reducing a text document
  - a computer program in order to create a summary
  - the most important sentences of the original documents



- Selection of representative sentences is performed

Reading comprehension

- Reading comprehension is the ability to read text, process it, and understand its meaning
  - understanding of a text message
  - language skills: phonology, syntax, semantics, and pragmatics
  - affected by prior knowledge, ability to make inference

Information extraction

- Information extraction from news article

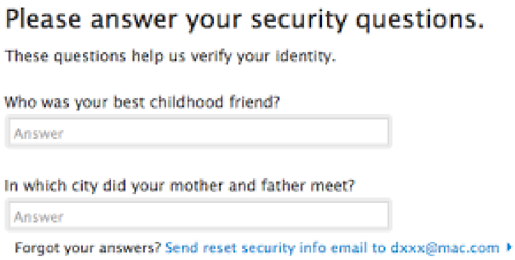
ShooterName: Scott Westerhuis
NumKilled: 6

A couple and four children found dead in their burning South Dakota home had been shot in an apparent murder-suicide, officials said Monday.
...
Scott Westerhuis's cause of death was "shotgun wound with manner of death as suspected suicide," it added in a statement.

(Narasimhan et al., 2016)

Question answering

- QA aims to answer the questions posted by humans in a natural language
  - takes natural language question as an input rather than keywords
  - keyword extraction is performed to identify the question type
  - "person" or "location" are retrieved from "who" or "where"
  - candidate answers are further classified
  - compact and meaningful answer is translated by parsing



Dialogue generation

A: Where are you going? (1) B: I'm going to the police station. (2) A: I'll come with you. (3) B: No, no, no, no, you're not going anywhere. (4) A: Why? (5) B: I need you to stay here. (6) A: I don't know what you are talking about. (7) ...
A: How old are you? (1) B: I'm 16. Why are you asking? (2) A I thought you were 12. (3) B: What made you think so? (4) A: I don't know what you are talking about. (5) B: You don't know what you are saying. (6) ... ...

(Li et al., 2016)

Dialogue with question clarification

Task 1: Question Paraphrase (AQ):
kb: Larry Crowne directed_by Tom Hanks
kb: Forrest Gump starred_actors Tom Hanks, Robin Wright, Gary Sinise
kb: Forrest Gump directed_by Robert Zemeckis
T/S : Conversation History.
T : Which movvie did Tom Hanks sttar in ?
S : What do you mean ?
T : I mean which film did Tom Hanks appear in.
T : Which movvie did Tom Hanks sttar in ?
S : Forrest Gump
T : That's correct. (+)

(Li et al., 2016)

Text understanding and reasoning

- Synthetic tasks in bAbI project (Weston et al., 2015) used to evaluate the learning algorithms for
  - text understanding and reasoning
  - question answering problem
  - categorization of different kinds of questions
- 20 tasks in bAbI dataset (https://research.fb.com/projects/babi)
  - single, two or three supporting facts
  - yes/no question
  - counting
  - lists/sets
  - simple negation
  - indefinite knowledge
- Children's book test (Hill et al., 2016)
  - measure how well a text model can exploit wider linguistic context
  - in each question, the first 20 sentences form the context, and a word is removed from the 21st sentence, which becomes the query

Outline

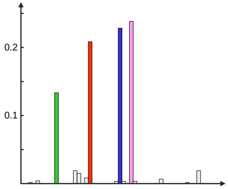
1. Deep Text Modeling
   - Natural language application
   - Probabilistic neural network
2. Deep Sequential Learning
3. Deep Stochastic Learning

Probabilistic model

Most likely words from top topics

sequence	measured	residues	computer
region	average	binding	methods
pcr	range	domains	number
identified	values	helix	two
fragments	different	cys	principle
two	size	regions	design
genes	three	structure	access
three	calculated	terminus	processing
cdna	two	terminal	advantage
analysis	low	site	important

Topic proportions



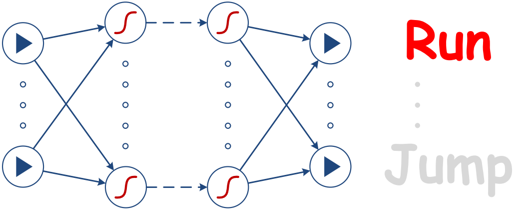
Abstract with the most likely topic assignments

Statistical approaches help in the determination of significant configurations in protein and nucleic acid sequence data. Three recent statistical methods are discussed: (i) score-based sequence analysis that provides a means for characterizing anomalies in local sequence text and for evaluating sequence comparisons; (ii) quantile distributions of amino acid usage that reveal general compositional biases in proteins and evolutionary relations; and (iii) r-scan statistics that can be applied to the analysis of spacing of sequence markers.

$$p(\text{word}) = \sum_{\text{topic}} p(\text{word} \mid \text{topic})\, p(\text{topic})$$

Neural network

- Deep structured/hierarchical learning
- Multiple layers of nonlinear processing units
- High-level abstraction is learned



Probabilistic Model $+$ **Neural Network**

Modern machine learning

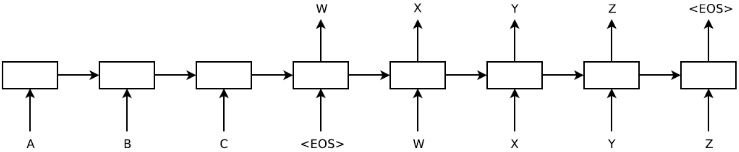
	Probabilistic Models	Neural Nets
Structure	Top-down	Bottom-up
Representation	Intuitive	Distributed
Interpretation	**Easy**	**Harder**
Semi/unsupervised	**Easier**	**Harder**
Incorp. domain knowl.	**Easy**	**Hard**
Incorp. constraint	**Easy**	**Hard**
Incorp. uncertainty	**Easy**	**Hard**
Learning	Many algorithms	Back-propagation
Inference/decode	**Harder**	**Easier**
Evaluation on	int. quantity	**End performance**

Outline

Outline

Seq2Seq learning: *encoder-decoder network*

- Traditional DNN was sensibly encoded with vectors with a fixed dimensionality
- Many important problems are best expressed with sequences whose lengths are unknown *a priori*
- An input sequence "ABC" is encoded and decoded to produce "WXYZ" as the output sequence (Sutskever et al., 2014)



- LSTM architecture is applied to deal with this problem

Sequence learning

- RNN can not deal with sequential learning with input and output sequences in different lengths
- Sequence to sequence learning is performed by
  - first, map the input sequence to a fixed-sized vector using on RNN
  - second, map the vector to the target sequence using another RNN
- LSTM is used to estimate $p(y_1, \ldots, y_{T'} | x_1, \ldots, x_T)$ where $\{x_1, \ldots, x_T\}$ is an input sequence and $\{y_1, \ldots, y_{T'}\}$ is its output sequence whose length T' may differ from T
- LSTM language model is calculated by

$$p(y_1, \ldots, y_{T'} | x_1, \ldots, x_T) = \prod_{t=1}^{T'} p(y_t | v, y_1, \ldots, y_{t-1})$$

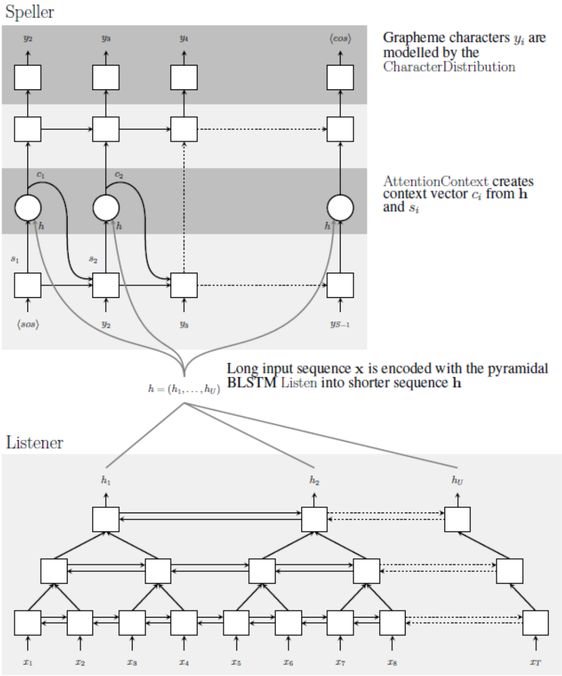
- LSTM computes this probability by obtaining the fixed dimensional v of $\{x_1, \ldots, x_T\}$ given by the last hidden state of LSTM

Learning via LSTM

- Each sentence ends with a symbol <EOS>, which enables the model to define a distribution over sequences of all possible lengths
- Two LSTMs are used (Sutskever et al., 2014)
  - one for the input sequence and another for the output sequence
  - number of parameters is increased
  - computational cost is negligible
  - natural to train LSTM on multiple language pairs simultaneously
- Deep LSTM outperformed shallow LSTM. Four-layer LSTM was chosen
- Reverse the order of the words of an input sentence

Listen, attend and spell

- Traditional acoustic, pronunciation and language models were trained separately based on different objectives
- This disjoint training issue was tackled by designing models that are trained end-to-end from speech signals directly to word transcripts
  - connectionist temporal classification
  - sequence to sequence model with attention
- Listen, attend and spell are introduced (Chan et al., 2015)
- Encoder is a listener while decoder is a speller
- Bidirectional LSTM is used in encoder and decoder
- Attention model is used to extract the relevant information from a small number of time steps


Speller
Grapheme characters y_i are modelled by the CharacterDistribution
AttentionContext creates context vector c_i from $\mathbf{h}$ and s_i
$h = (h_1, \ldots, h_U)$
Long input sequence $\mathbf{x}$ is encoded with the pyramidal BLSTM Listen into shorter sequence $\mathbf{h}$
Listener
y_2 y_3 y_4 $\langle eos \rangle$
c_1 c_2 s_1 s_2 h
$\langle sos \rangle$ y_2 y_3 y_{S-1}
h_1 h_2 h_U
x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_T

Machine translation

- Sequence to sequence translation model (Sutskever et al., 2014)
  - compresses all the information into a fixed length vector $\mathbf{s}_0$
  - degrades as the length of input sentence increases

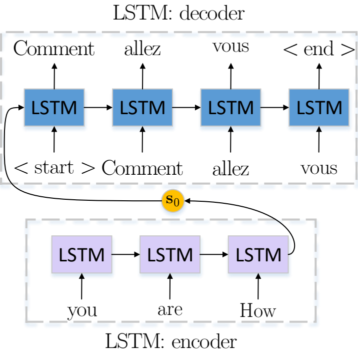
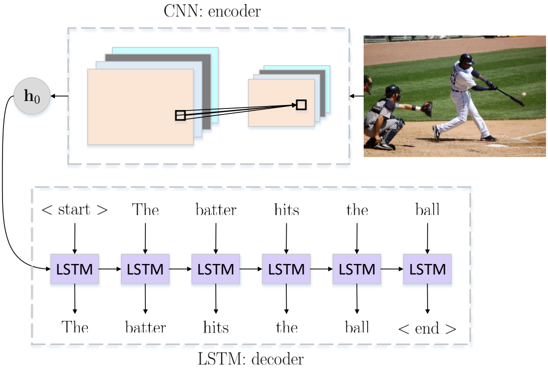


Image caption

- It is challenging to describe the content of an image which
  - captures the objects in an image
  - expresses the relations between objects
- An end-to-end system (Vinyals et al., 2015) is built with
  - CNN encoder
  - LSTM decoder



Machine translation with attention

- Attention mechanism was merged in a sequence to sequence model (Bahdanau et al., 2015)
  - alignment model
  - translation model

$$\mathbf{c}_i = \sum_{j=1}^{T_x} \alpha_{ij} \mathbf{h}_j$$

- Compute attention weights

$$\alpha_{ij} = \frac{\exp e_{ij}}{\sum_{k=1}^{T_x} \exp e_{ik}}$$

where $e_{ij} = \text{Score}(\mathbf{s}_{i-1}, \mathbf{h}_j)$

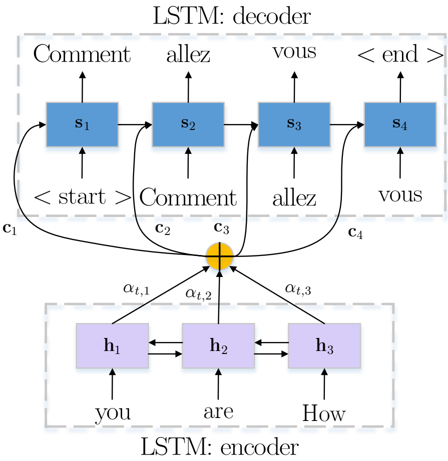
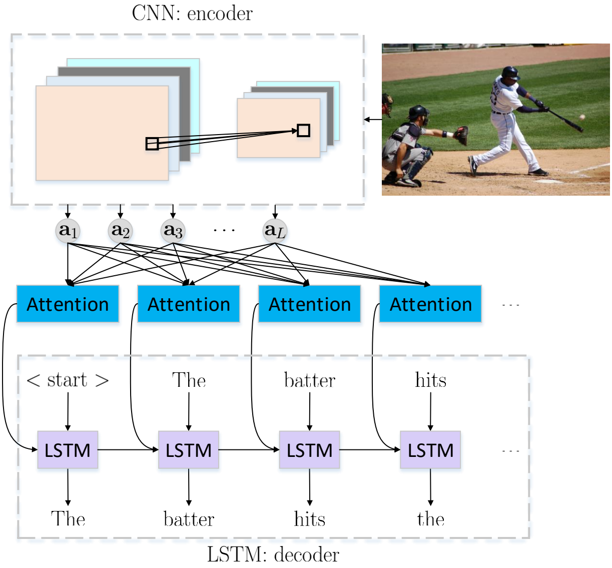


Image caption with attention



Results on MS COCO dataset

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

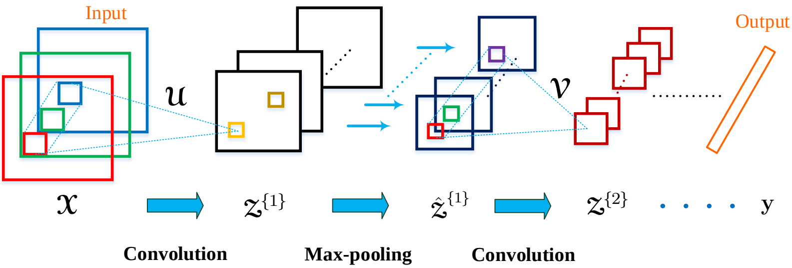
A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

Outline

1. Deep Text Modeling
2. Deep Sequential Learning
   - Sequence to sequence learning
   - Convolutional neural network
   - Dilated neural network
   - Memory-augmented neural network
3. Deep Stochastic Learning

- Two-dimensional CNN (Krizhevsky et al., 2012)

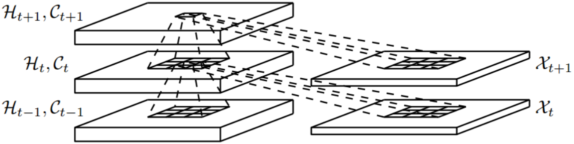


Convolutional LSTM

- Spatiotemporal correlation is captured for weather forecasting (Xingjian et al., 2015)

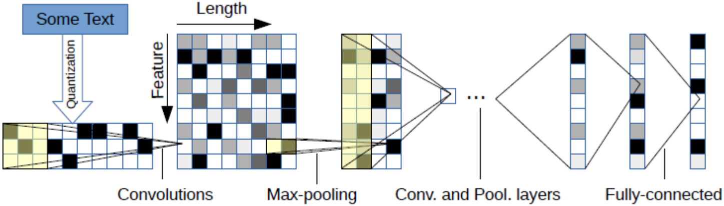
$$
\begin{aligned}
i_t &= \sigma(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} \circ C_{t-1} + b_i) \\
f_t &= \sigma(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} \circ C_{t-1} + b_f) \\
C_t &= f_t \circ C_{t-1} + i_t \circ \tanh(W_{hc} * X_t + W_{hc} * H_{t-1} + b_c) \\
o_t &= \sigma(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} \circ C_t + b_o) \\
H_t &= o_t \circ \tanh(C_t)
\end{aligned}
$$

where $*$ is the convolution operation and $\circ$ is the Hadamard product



Character CNN for text classification

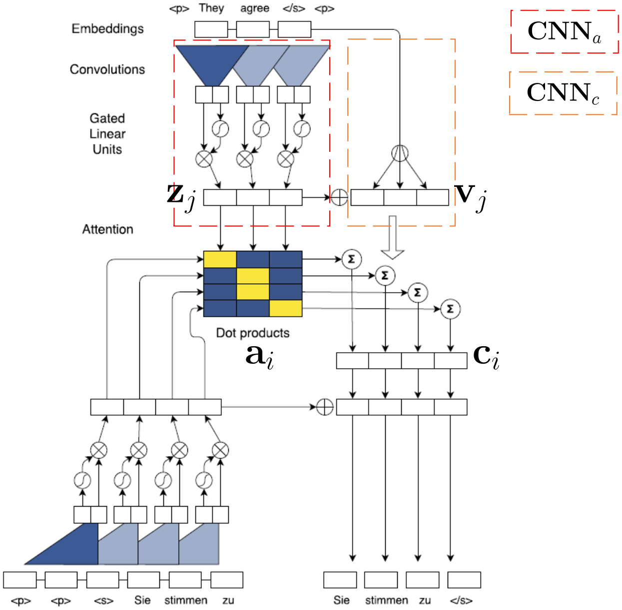
- Character-based convolutional neural network achieved better text classification than
  - word-based convolutional neural network
  - recurrent neural network



(Zhang et al., 2015)

Convolutional sequence to sequence learning

- Advantages of using convolutional neural network for sequence modeling
  - independence on the computations of the previous time step
  - computational parallelization
  - hierarchical representation over the input sequence
  - shorter path to capture long-range dependencies
    * CNN - $\mathcal{O}(\frac{n}{k})$ with a kernel of width k
    * RNN - $\mathcal{O}(n)$ for linear time
- An entirely convolutional sequence to sequence model (Gehring et al., 2017) was proposed for machine translation
  - GLU (Gated Linear Unit): a simplified gating mechanism that reduces the gradient vanishing problem
  - residual connections
  - attention mechanism


<p> They agree </s> <p>
Embeddings
Convolutions
Gated Linear Units
$\mathbf{z}_j$
$\mathbf{v}_j$
Attention
Dot products
$\mathbf{a}_i$
$\mathbf{c}_i$
<p> <p> <s> Sie stimmen zu
Sie stimmen zu </s>
CNN_a
CNN_c

Convolutional encoder

- Encoder consists of two stacked convolutional networks
  - $\mathrm{CNN_a}$ produces the key vector $\mathbf{z}_j$

$$\mathbf{z}_j = \mathrm{CNN_a}(\mathbf{e}_j)$$

  - $\mathrm{CNN_c}$ produces the value vector $\mathbf{v}_j$

$$\mathbf{v}_j = \mathrm{CNN_c}(\mathbf{e}_j)$$

- Conditional input $\mathbf{c}_i$ to the decoder is obtained by

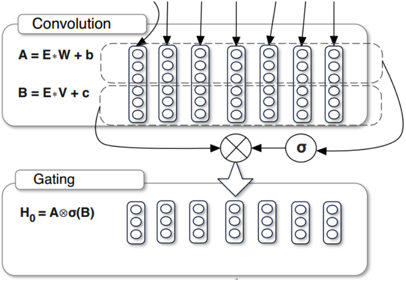
$$\mathbf{a}_i = \mathsf{Attention}(\mathbf{z}_j, \mathbf{s}_i)$$
$$\mathbf{c}_i = \sum_{j=1}^{T} a_{ij}\mathbf{v}_j$$

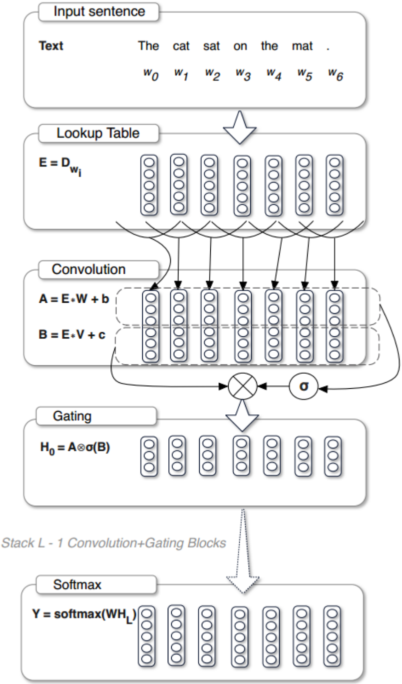
Convolutional encoder using gated CNN

- Gated linear unit (Dauphin et al., 2017) is calculated via convolution operation $*$ for hidden layers $h_0, \ldots, h_L$ as

$$h_l(\mathbf{E}) = (\mathbf{E} * \mathbf{W} + \mathbf{b}) \otimes \sigma(\mathbf{E} * \mathbf{V} + \mathbf{c})$$

  - LSTM style with no forget and input gates required
  - only possess output gate in which information to be propagated




Input sentence
Text
The cat sat on the mat .
w_0 w_1 w_2 w_3 w_4 w_5 w_6
Lookup Table
$E = D_{w_i}$
Convolution
$A = E * W + b$
$B = E * V + c$
σ
Gating
$H_0 = A \otimes \sigma(B)$
Stack L - 1 Convolution+Gating Blocks
Softmax
$Y = \text{softmax}(WH_L)$

Outline

Dilated convolutional neural network - WaveNet

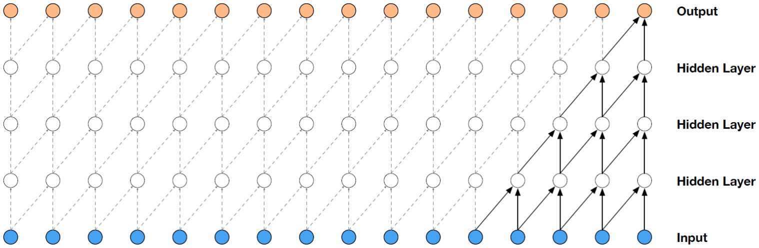
- Dilated CNN (Van Den Oord et al., 2016) was proposed to generate a raw audio waveform
  - probabilistic and autoregressive
  - dilated causal convolution
  - conditioned on speaker identity to generate different voices
  - generic and flexible framework

- Waveform $\mathbf{x} = \{x_1, \cdots, x_T\}$ is factorised as a product of conditional probabilities

$$p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t | x_1, \cdots, x_{t-1})$$

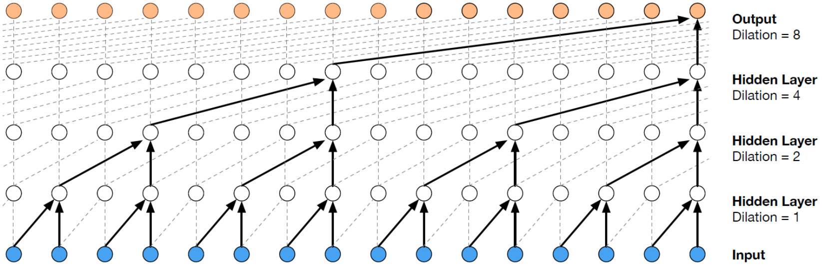
  - stack of convolutional layers
  - no pooling layers
  - optimize to maximize the log-likelihood

- Causal convolution
  - cannot depend on any of the future time steps
  - shifting the output of a normal convolution by a few time steps
  - CNN is faster than RNN



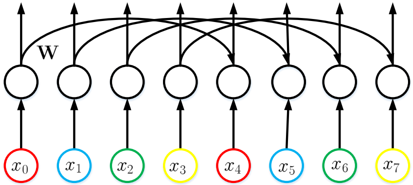
1-D convolution with kernel size 2

- Dilated convolution
  - filter is applied over an area larger than its length by skipping input values with a certain step
  - similar to pooling or strided convolutions, but the output has the same size as the input
  - dilation 1 yields the standard convolution
  - receptive field to grow exponentially with depth



Dilated recurrent neural network

- Challenges when learning on long sequences with RNNs
  - complex dependencies
  - vanishing and exploding gradients
  - efficient parallelization
- Multi-resolution with dilated recurrent skip connections (Chang et al., 2017)
  - neural connection architecture analogous to the dilated CNN
  - single-layer dilated RNN



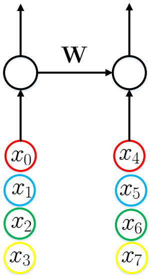
Dilated recurrent skip connection

- Denote $h_t^{(l)}$ as the cell in layer l and time t. Dilated recurrent skip connection is represented as

$$h_t^{(l)} = f(x_t^{(l)}, h_{t-d^{(l)}}^{(l)})$$

  - $d^{(l)}$ is the skip length or dilation of layer l
  - $x_t^{(l)}$ is the input to layer l at time t
  - $f(\cdot)$ denotes any output operation for a RNN cell

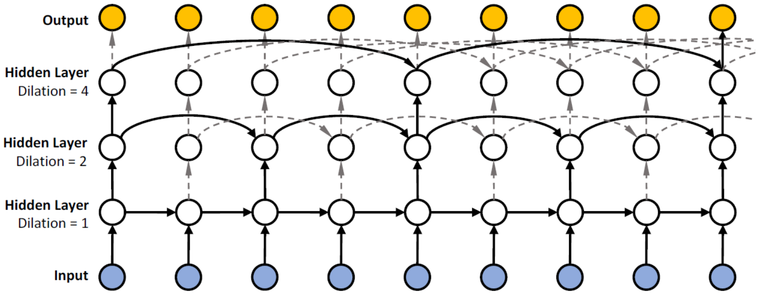
- Recurrent chains can be computed in parallel
- Degree of parallelization is increased by $d^{(l)}$



Multilayer dilated recurrent neural network

- Dilated RNN is constructed by stacking dilated recurrent layers
  - dilation increases exponentially across layers
  - dilated RNN with $L = 3$ and $M = 2$

$$d^{(l)} = M^{l-1}, \qquad l = 1, \cdots, L$$

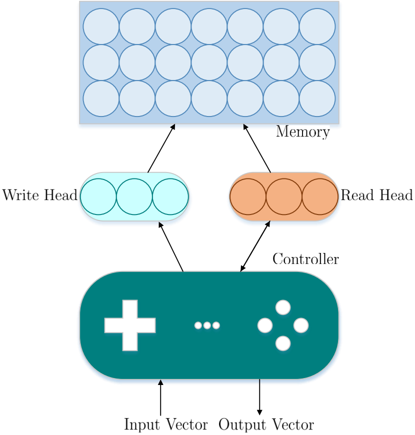


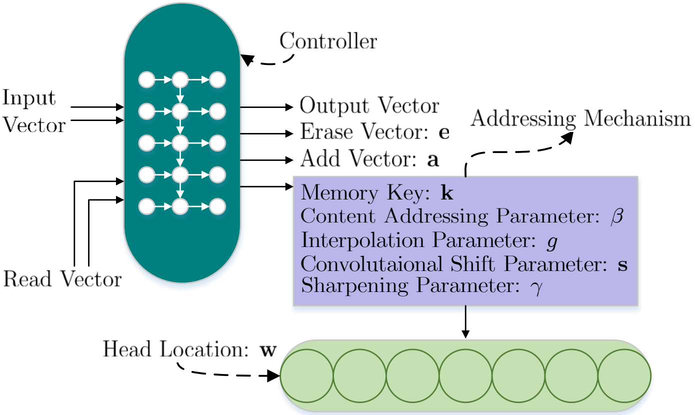
Outline

Neural Turing machine versus memory network

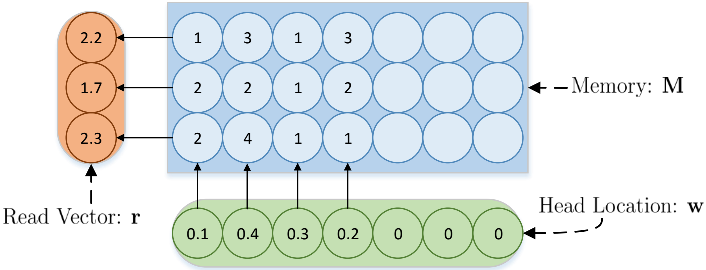
- Most machine learning models lack an easy way to
  - read and write to part of a long-term memory component
  - combine this seamlessly with inference
- Neural Turing machine (Graves et al., 2014)
  - learns to read from and write to memory cells without explicit supervision
  - allows end-to-end training via content-based soft attention
  - emulates algorithmic mechanism in a way that allows gradient-based optimization
- Memory network (Weston et al., 2015)
  - includes memory cells that can be accessed via an addressing mechanism
  - combines learning strategies for inference with a memory component that can be read and written to

- Neural Turing machine (Graves et al., 2014)
  - intelligence requires knowledge
  - acquiring knowledge can be done via large-scale deep learning
  - neural networks excel at storing implicit knowledge, but struggle to memorize facts
  - neural networks lack the working memory system that allows human beings to explicitly hold and manipulate pieces of information



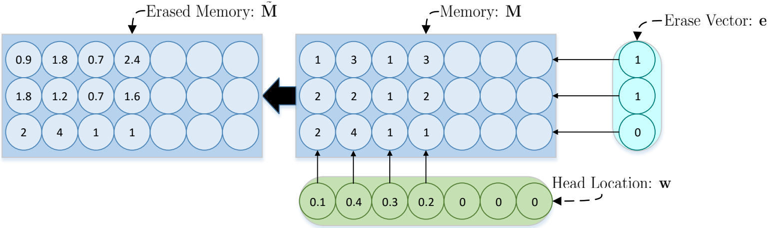

Controller
Input Vector
Read Vector
Output Vector
Erase Vector: **e**
Add Vector: **a**
Addressing Mechanism
Memory Key: **k**
Content Addressing Parameter: β
Interpolation Parameter: g
Convolutaional Shift Parameter: **s**
Sharpening Parameter: γ
Head Location: **w**

- Reading

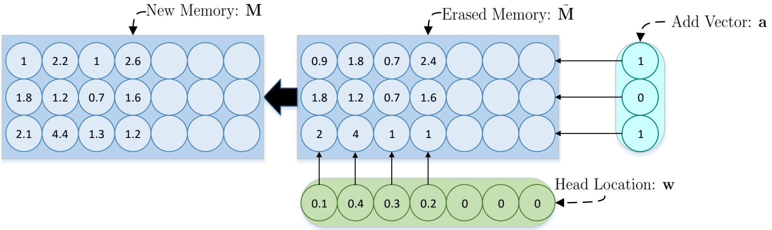


  - $\mathbf{M}_t$ is the $N \times M$ memory matrix at time t where N is the number of memory locations, and M is the vector size at each location
  - $\mathbf{w}_t = \{w_t(i)\}$ is a weight vector over N locations emitted by a read head at time t, and $\sum_i w_t(i) = 1$, $0 \leq w_t(i) \leq 1$
  - read vector $\boldsymbol{r}_t$ of length M, returned by the head, is defined as a $\boldsymbol{r}_t \leftarrow \sum_i w_t(i)\mathbf{M}_t(i)$

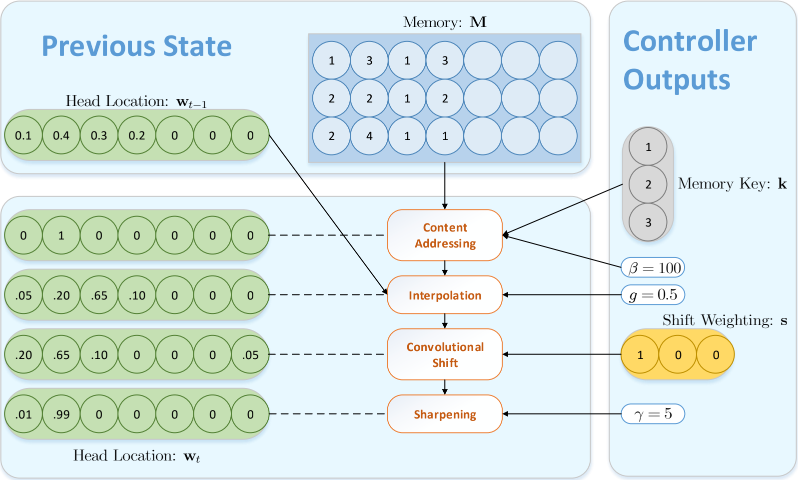
- Writing step 1 → Erasing $\tilde{\mathbf{M}}_t(i) \leftarrow \mathbf{M}_{t-1}(i)[1 - w_t(i)\boldsymbol{e}_t]$



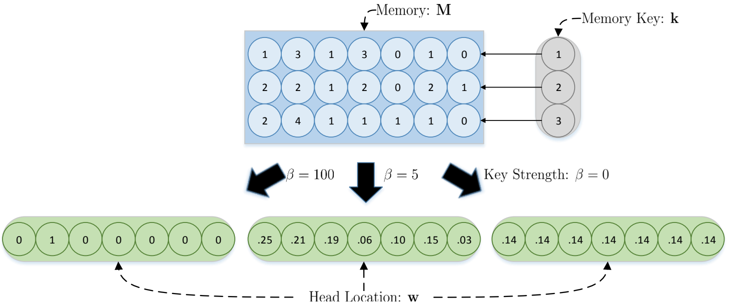
- Writing step 2 → Adding $\mathbf{M}_t(i) \leftarrow \tilde{\mathbf{M}}_t(i) + w_t(i)\boldsymbol{a}_t$



Addressing mechanism

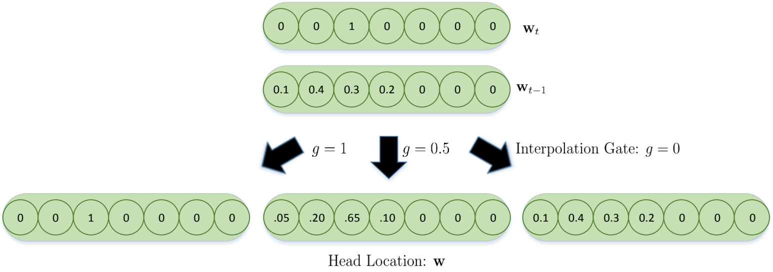


- Step 1: content addressing



$$w_t^c(i) \leftarrow \frac{\exp\Big(\beta_t K[\mathbf{k}_t, \mathbf{M}_t(i)]\Big)}{\sum_j \exp\Big(\beta_t K[\mathbf{k}_t, \mathbf{M}_t(j)]\Big)} \quad \text{where} \quad K[\mathbf{u}, \mathbf{v}] = \frac{\mathbf{u} \cdot \mathbf{v}}{||\mathbf{u}|| \cdot ||\mathbf{v}||}$$

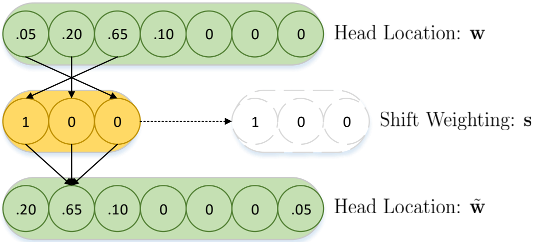
- Step 2: interpolation



  - facilitate both simple iteration across the locations of the memory and random-access jumps
  - prior to rotation, each head emits a scalar interpolation gate g_t

$$\mathbf{w}_t^g \leftarrow g_t \mathbf{w}_t^c + (1 - g_t)\mathbf{w}_{t-1}$$

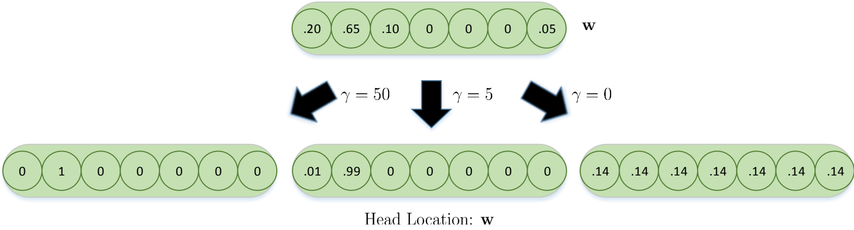
- Step 3: convolutional shift



  - each head emits a shift weighting $\mathbf{s}_t$ that defines a normalised distribution over the allowed integer shifts
  - memory locations from 0 to $N-1$
  - rotation is performed via the circular convolution

$$\tilde{w}_t(i) \leftarrow \sum_{j=0}^{N-1} w_t^g(j) s_t(i-j)$$

- Step 4: sharpening

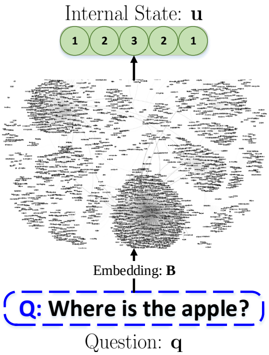


  - rotation will transform a weighting focused at a single point into one slightly blurred over three points
  - each head accordingly emits one further scalar γ_t to sharpen weight

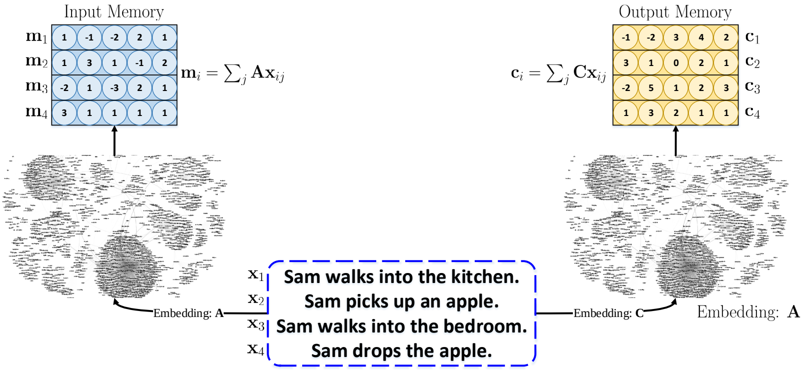
$$w_t(i) \leftarrow \frac{\tilde{w}_t(i)^{\gamma_t}}{\sum_j \tilde{w}_t(j)^{\gamma_t}}$$

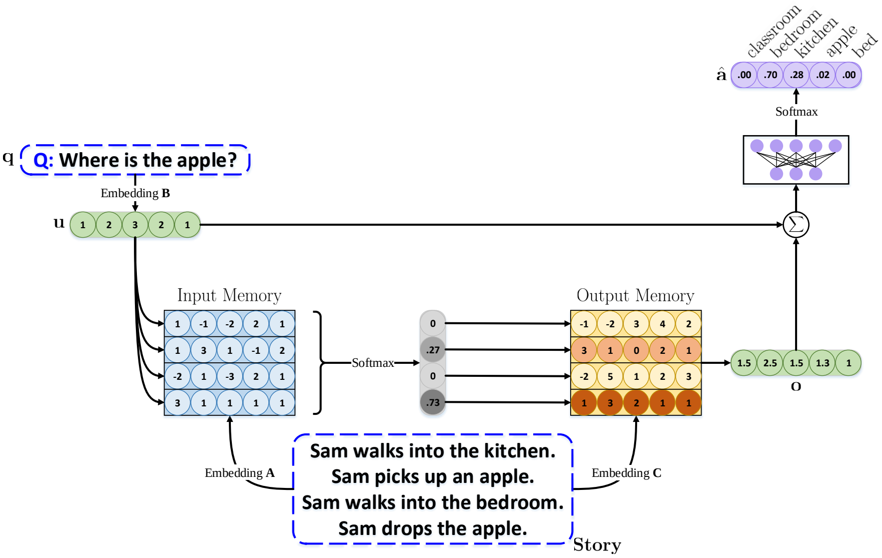
- End-to-end memory network (Sukhbaatar et al., 2015)
  - memory network (Weston et al., 2015) was not easy to train via error backpropagation
  - continuous form of memory network
  - it can be trained end-to-end from input-output pairs
  - supportive attention was introduced (Chien and Lin, 2018)

$$\mathbf{u} = \sum_j \mathbf{B}\mathbf{q}_j$$



- End-to-end memory network (Sukhbaatar et al., 2015)
  - memory network (Weston et al., 2015) was not easy to train via error backpropagation
  - continuous form of memory network
  - it can be trained end-to-end from input-output pairs
  - supportive attention was introduced (Chien and Lin, 2018)

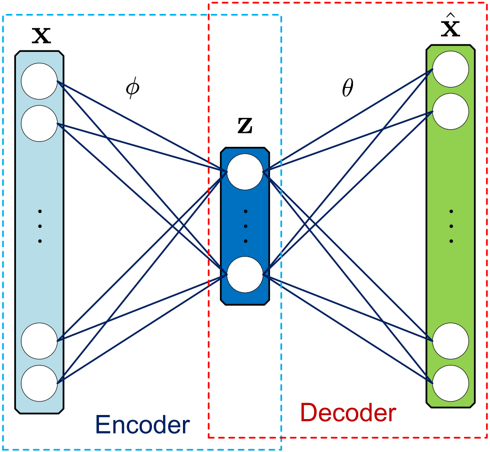



classroom
bedroom
kitchen
apple
bed
â
.00 .70 .28 .02 .00
Softmax
q
Q: Where is the apple?
Embedding B
u
1 2 3 2 1
Input Memory
1 -1 -2 2 1
1 3 1 -1 2
-2 1 -3 2 1
3 1 1 1 1
Softmax
0
.27
0
.73
Output Memory
-1 -2 3 4 2
3 1 0 2 1
-2 5 1 2 3
1 3 2 1 1
1.5 2.5 1.5 1.3 1
o
Embedding A
Embedding C
Sam walks into the kitchen.
Sam picks up an apple.
Sam walks into the bedroom.
Sam drops the apple.
Story

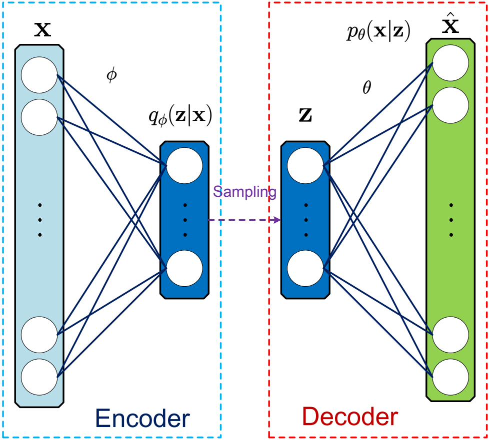
Outline

Outline

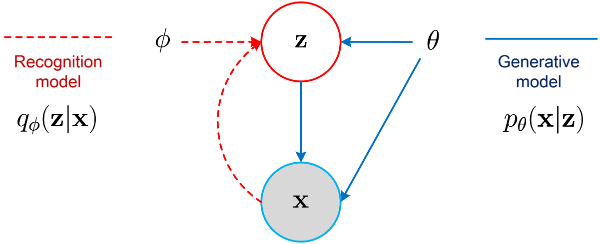
Auto-encoder



Variational auto-encoder



Variational auto-encoder



(Kingma and Welling, 2014)

- Mean-field approach requires analytical solution to maximum likelihood problem, which is intractable in case of neural network
- Use neural network to sample the latent variables $\mathbf{z}$ from variational posterior
- VAE was a building block for speaker recognition (Chien and Hsu, 2017)

Stochastic gradient variational Bayes

Objective: $\mathcal{L}_{\boldsymbol{\Theta}} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[f_{\boldsymbol{\Theta}}(\mathbf{x}, \mathbf{z})]$

Gradient:

Step1: sample $\boldsymbol{\epsilon}^{(l)}$ from $\mathcal{N}(\mathbf{0}, \mathbf{I})$

Step2: $\mathbf{z}^{(l)} = \boldsymbol{\mu}_{\mathbf{z}} + \boldsymbol{\sigma}_{\mathbf{z}} \odot \boldsymbol{\epsilon}^{(l)}$

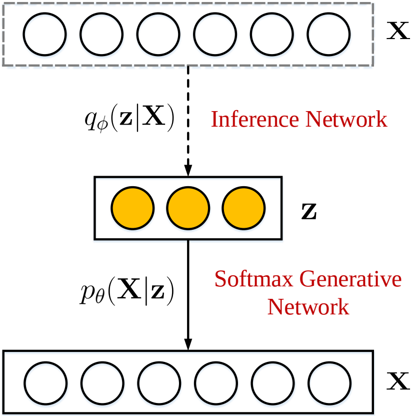
Step3: $\mathcal{L}_{\boldsymbol{\Theta}} \simeq f_{\boldsymbol{\Theta}}(\mathbf{x}|\mathbf{z}^{(l)})$

Step4: $\nabla_{\boldsymbol{\Theta}} \mathcal{L}_{\boldsymbol{\Theta}} \simeq \nabla_{\boldsymbol{\Theta}} f_{\boldsymbol{\Theta}}(\mathbf{x}, \mathbf{z}^{(l)})$

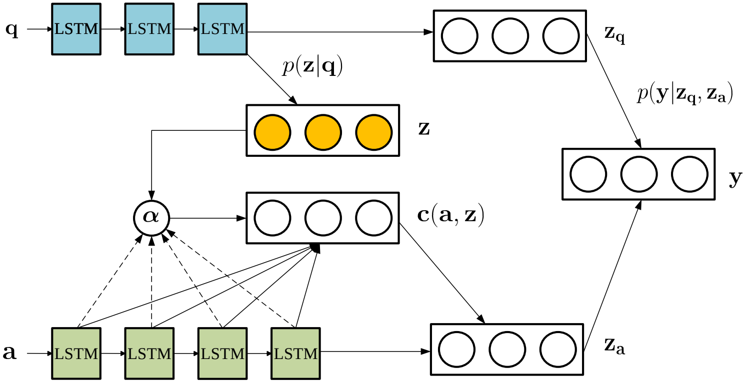
- Reduce the variance caused by directly sampling $\mathbf{z}$ (Rezende et al., 2014)

Neural variational document model

- Continuous semantic latent variable model for a document $\mathbf{X}$ (Miao et al., 2016)



Neural answer selection model



Generating sentences from a continuous space

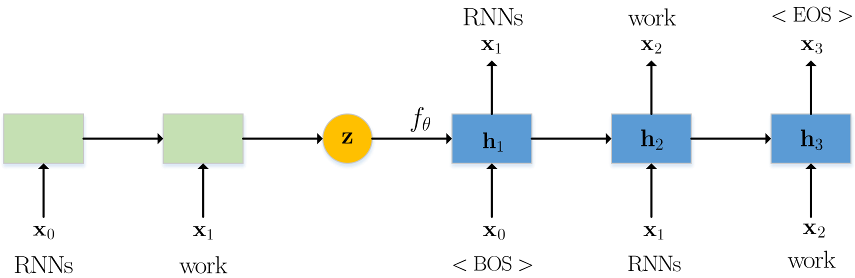
- Variational recurrent auto-encoder (VRAE) (**?**) is
  - composed of two RNNs for both encoder and decoder
  - developed for unsupervised learning for time series data
  - constructed to map data into latent representation
- Parameters of variational distribution over latent variable $\mathbf{z}$ are function of the last state of RNN $\mathbf{h}_T$

$$q_\phi(\mathbf{z}|\mathbf{X}) = \mathcal{N}(\boldsymbol{\mu}_z, \text{diag}(\boldsymbol{\sigma}_z^2)), \quad \text{where } \left[\boldsymbol{\mu}_z, \boldsymbol{\sigma}_z^2\right] = f_\phi^{(q)}(\mathbf{h}_T)$$

- Initial state of RNN decoder is computed by a sample $\mathbf{z}$

$$\mathbf{h}_0 = f_\theta^{(i)}(\mathbf{z})$$
$$\mathbf{h}_{t+1} = f_\theta^{\text{dec}}(\mathbf{h}_t, \mathbf{x}_t)$$
$$\mathbf{x}_t = f_\theta^{(o)}(\mathbf{h}_t)$$

Variational recurrent auto-encoder



Outline

Unsupervised variational recurrent neural network

- VAE and RNN are combined by
  - incorporating the hidden state $\mathbf{h}_t$ at time step t into VAE
- Stochastic or variational recurrent neural network was constructed for unsupervised learning (Chung et al., 2015)
- Hidden state is expressed for
  - RNN

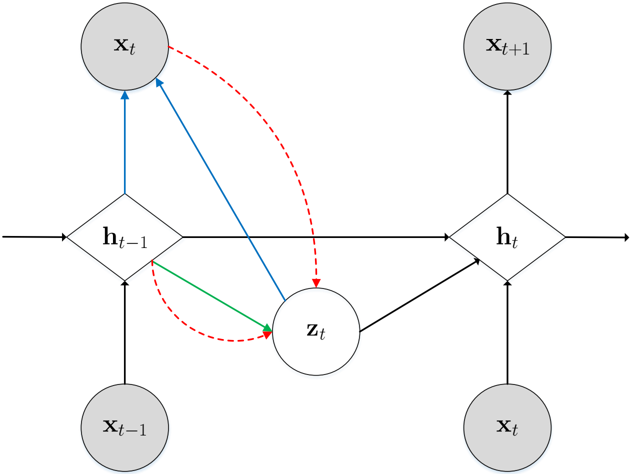
$$\mathbf{h}_t = \mathcal{F}_{\mathbf{w}}(\mathbf{x}'_t, \mathbf{h}_{t-1})$$

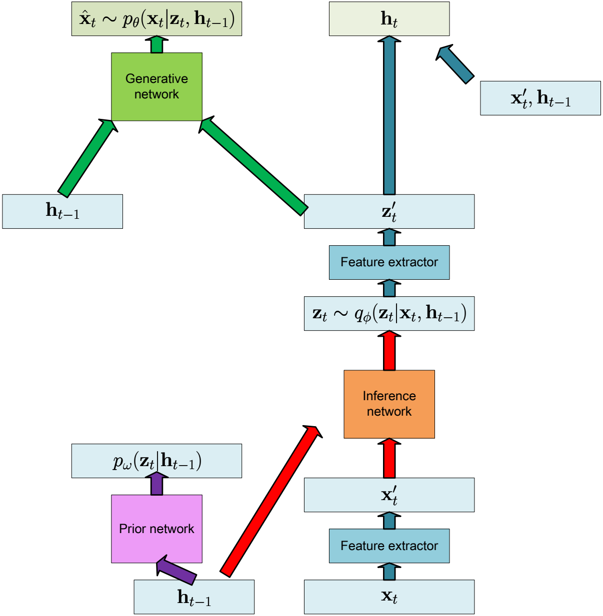
  - variational RNN (VRNN)

$$\mathbf{h}_t = \mathcal{F}_{\mathbf{\Theta}}(\mathbf{x}'_t, \mathbf{z}'_t, \mathbf{h}_{t-1})$$

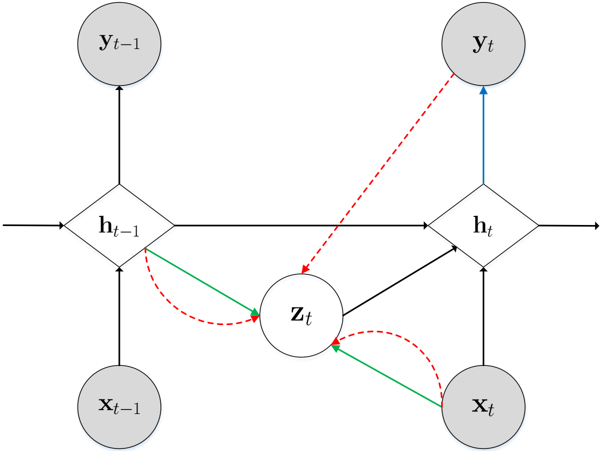
- Apply stochastic gradient variational Bayes for optimization
- Characterize the variability by using high-level latent random variable $\mathbf{z}'_t$

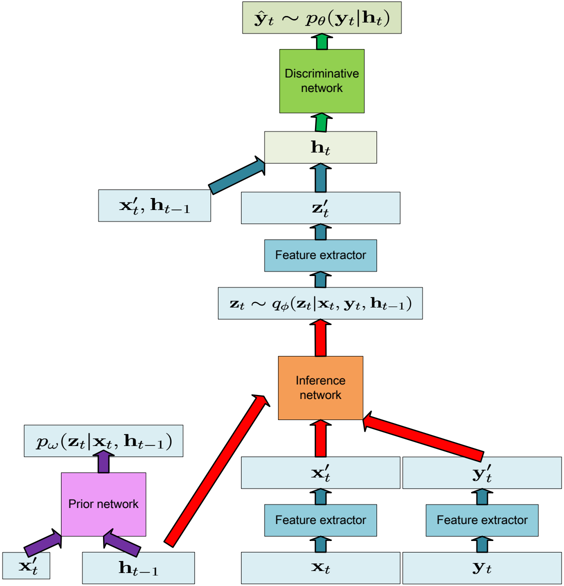
Graphical representation: *unsupervised VRNN*




$\hat{\mathbf{x}}_t \sim p_\theta(\mathbf{x}_t|\mathbf{z}_t, \mathbf{h}_{t-1})$
$\mathbf{h}_t$
Generative network
$\mathbf{x}'_t, \mathbf{h}_{t-1}$
$\mathbf{h}_{t-1}$
$\mathbf{z}'_t$
Feature extractor
$\mathbf{z}_t \sim q_\phi(\mathbf{z}_t|\mathbf{x}_t, \mathbf{h}_{t-1})$
Inference network
$p_\omega(\mathbf{z}_t|\mathbf{h}_{t-1})$
$\mathbf{x}'_t$
Prior network
Feature extractor
$\mathbf{h}_{t-1}$
$\mathbf{x}_t$

- Supervised VRNN was proposed for speech separation (Chien and Kuo, 2017) and speech recognition (Chien and Shen, 2017)
  - target variable $\mathbf{y}_t$ is introduced for supervised learning




$\hat{\mathbf{y}}_t \sim p_\theta(\mathbf{y}_t|\mathbf{h}_t)$
Discriminative network
$\mathbf{h}_t$
$\mathbf{x}'_t, \mathbf{h}_{t-1}$
$\mathbf{z}'_t$
Feature extractor
$\mathbf{z}_t \sim q_\phi(\mathbf{z}_t|\mathbf{x}_t, \mathbf{y}_t, \mathbf{h}_{t-1})$
Inference network
$p_\omega(\mathbf{z}_t|\mathbf{x}_t, \mathbf{h}_{t-1})$
Prior network
$\mathbf{x}'_t$
$\mathbf{y}'_t$
Feature extractor
Feature extractor
$\mathbf{x}'_t$
$\mathbf{h}_{t-1}$
$\mathbf{x}_t$
$\mathbf{y}_t$

Outline

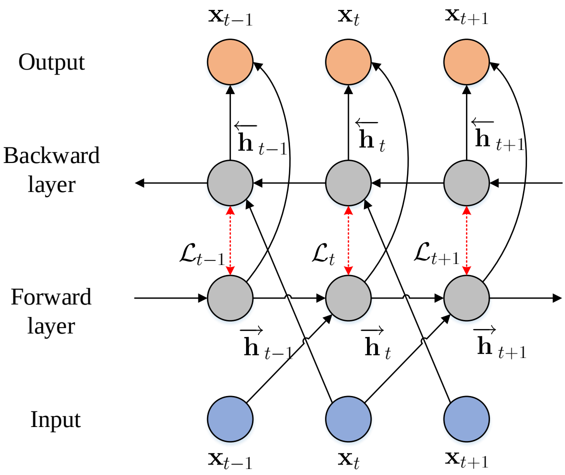
Planning long-term future

- RNN is usually trained with teacher forcing where
  - model is optimized to predict one-step ahead
  - local correlation dominates the long-term dependency
  - generated samples tend to exhibit local coherence but lack meaningful global structure
- Regularizing the recurrent neural network based on future information (Serdyuk et al., 2018)
  - run twin forward and backward RNNs with no parameter sharing
  - encourage hidden state of forward RNN to be close to that of backward RNN
  - allow forward RNN to catch past and future features that are useful in test time

Twin network

- Forward RNN
$$\overrightarrow{\mathbf{h}}_t = \overrightarrow{f}(\mathbf{x}_{t-1}, \overrightarrow{\mathbf{h}}_{t-1})$$
  - prediction of $\mathbf{x}_t$ using past information $p_f(\mathbf{x}_t|\mathbf{x}_{<t}) = \overrightarrow{\boldsymbol{\psi}}(\overrightarrow{\mathbf{h}}_t)$
- Backward RNN
$$\overleftarrow{\mathbf{h}}_t = \overleftarrow{f}(\mathbf{x}_{t+1}, \overleftarrow{\mathbf{h}}_{t+1})$$
  - prediction of $\mathbf{x}_t$ using future information $p_b(\mathbf{x}_t|\mathbf{x}_{>t}) = \overleftarrow{\boldsymbol{\psi}}(\overleftarrow{\mathbf{h}}_t)$
- $\overrightarrow{\mathbf{h}}_t$ and $\overleftarrow{\mathbf{h}}_t$ contain past and future features for predicting $\mathbf{x}_t$, respectively

Graphical representation



Learning objective

- Penalizing the distance between forward and backward hidden states leading to the same prediction

$$\mathcal{L}_t = \|g(\overrightarrow{\mathbf{h}}_t) - \overleftarrow{\mathbf{h}}_t\|$$

  - function $g(\cdot)$ is a parameterized affine transformation
  - affine transformation gives flexibility for equivalence between $\overrightarrow{\mathbf{h}}_t$ and $\overleftarrow{\mathbf{h}}_t$

- Training criterion

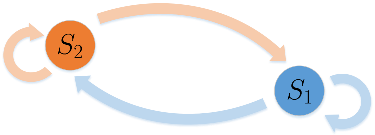
$$\mathcal{F}(\theta) = \sum_t \{\log p_f(\mathbf{x}_t|\mathbf{x}_{<t}) + \log p_b(\mathbf{x}_t|\mathbf{x}_{>t}) - \alpha\mathcal{L}_t\}$$

  - backward network is discarded during inference

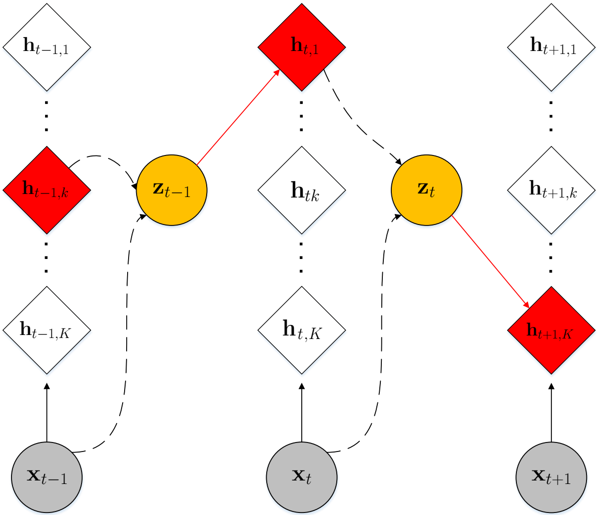
Outline

Markov recurrent neural network

- A large-scale RNN is hard to train and prone to be overfitting
- A single path of hidden states $\mathbf{h}_t$ is insufficient to capture temporal dependencies
- Deterministic hidden state $\mathbf{h}_t$ in RNN disregards the essence of stochastic process in sequential data
- Markov recurrent neural network (Kuo and Chien, 2018)
  - introduces the Markov property to build hidden state of RNN
  - incorporates the discrete latent variable into RNN
  - constructs the continuous hidden representation diversely
  - expresses the highly structured sequential data



Graphical representation



Markov recurrent neural network

- MRNN is developed to combine recurrent neural networks with probabilistic interpretation
  - introduces a Markov chain in latent representation
  - constructs multiple hidden state representation
  - conducts the stochastic state-to-state transitions
- Hidden state $\mathbf{h}_t$ is selected from $\{\mathbf{h}_{tk}\}_{k=1}^{K}$ according to $\mathbf{z}_t$

$$\mathbf{h}_t = \mathcal{S}_t^\top \mathbf{z}_t$$

- Transition of a stochastic state $\mathbf{z}_t$ complies with the property of Markov chain

$$p_{\boldsymbol{\phi}}(\mathbf{z}_t|\mathbf{z}_{1:t-1}, \mathbf{x}_{1:t}) = p(\mathbf{z}_t|\mathbf{z}_{t-1}, \mathbf{x}_t)$$

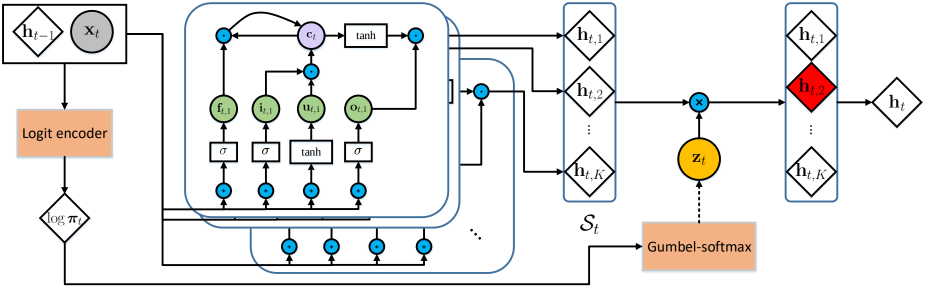
- State space
  - $\mathcal{S}_t \in \mathbb{R}^{K \times d}$ at each time t consists of all deterministic states $\{\mathbf{h}_{t1}, \ldots, \mathbf{h}_{tK}\}$ as basis vectors given by

$$\mathcal{S}_t \triangleq \begin{bmatrix} \mathbf{h}_{t1}^\top \\ \mathbf{h}_{t2}^\top \\ \vdots \\ \mathbf{h}_{tK}^\top \end{bmatrix} = \begin{bmatrix} \mathsf{LSTM}(\mathbf{h}_{t-1}, \mathbf{x}_t, \boldsymbol{\theta}_1) \\ \mathsf{LSTM}(\mathbf{h}_{t-1}, \mathbf{x}_t, \boldsymbol{\theta}_2) \\ \vdots \\ \mathsf{LSTM}(\mathbf{h}_{t-1}, \mathbf{x}_t, \boldsymbol{\theta}_K) \end{bmatrix}$$

- State encoder
  - each LSTM encoder k is calculated by

$$\begin{aligned} \mathbf{i}_{tk} &= \sigma(\mathbf{W}_{ik}[\mathbf{h}_{t-1}; \mathbf{x}_t] + \mathbf{b}_{ik}) \\ \mathbf{f}_{tk} &= \sigma(\mathbf{W}_{fk}[\mathbf{h}_{t-1}; \mathbf{x}_t] + \mathbf{b}_{fk}) \\ \mathbf{u}_{tk} &= \tanh(\mathbf{W}_{uk}[\mathbf{h}_{t-1}; \mathbf{x}_t] + \mathbf{b}_{gk}) \\ \mathbf{c}_{tk} &= \mathbf{f}_{tk} \odot \mathbf{c}_{t-1} + \mathbf{i}_{tk} \odot \mathbf{u}_{tk} \\ \mathbf{o}_{tk} &= \sigma(\mathbf{W}_{ok}[\mathbf{h}_{t-1}; \mathbf{x}_t] + \mathbf{b}_{ok}) \\ \mathbf{h}_{tk} &= \mathbf{o}_{tk} \odot \tanh(\mathbf{c}_{tk}) \end{aligned}$$

System implementation



Learning objective

- Parameters of state encoder and logit encoder $\{\theta, \phi\}$ are jointly trained by maximizing the likelihood of $\mathcal{D} = \{\mathbf{x}_t, \mathbf{y}_t\}_{t=1}^{T}$

$$p(\mathbf{y}_{1:T}|\mathbf{x}_{1:T}) = \prod_{t=1}^{T} \mathbb{E}_{p(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})}\left[p(\mathbf{y}_t|\mathbf{x}_{1:t}, \mathbf{z}_{1:t})p(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})\right]$$

- Monte Carlo method for log likelihood is calculated by

$$\sum_{t=1}^{T} \mathbb{E}_{p_{\phi}(\mathbf{z}_{1:t}|\mathbf{x}_{1:t})}\left[\log p_{\theta}(\mathbf{y}_t|\mathbf{x}_{1:t}, \mathbf{z}_{1:t})\right]$$
$$\approx \sum_{t=1}^{T}\left(\frac{1}{L}\sum_{l=1}^{L} \log p_{\theta}(\mathbf{y}_t|\mathbf{x}_{1:t}, \mathbf{z}_{1:t}^{(l)}) p_{\phi}(\mathbf{z}_{1:t}^{(l)}|\mathbf{x}_{1:t})\right)$$

References I

[1] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. of International Conference on Learning Representation*, 2015.

[2] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[3] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *arXiv preprint arXiv:1508.01211*, 2015.

[4] S. Chang, Y. Zhang, W. Han, M. Yu, X. Guo, W. Tan, X. Cui, M. Witbrock, M. A. Hasegawa-Johnson, and T. S. Huang, "Dilated recurrent neural networks," in *Advances in Neural Information Processing Systems 30*, 2017, pp. 77–87.

[5] J.-T. Chien and C.-W. Hsu, "Variational manifold learning for speaker recognition," in *Proc. of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2017.

[6] J.-T. Chien and K.-T. Kuo, "Variational recurrent neural networks for speech separation," in *Proc. Annual Conference of International Speech Communication Association*, 2017, pp. 1193–1197.

[7] J.-T. Chien and T.-A. Lin, "Supportive attention in end-to-end memory networks," in *Proc. of IEEE International Workshop on Machine Learning for Signal Processing*, 2018, pp. 1–6.

[8] J.-T. Chien and C. Shen, "Stochastic recurrent neural network for speech recognition," in *Proc. of Annual Conference of International Speech Communication Association*, 2017, pp. 1313–1317.

References II

[9] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," in *Advances in neural information processing systems*, 2015, pp. 2980–2988.

[10] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proc. of International Conference on Machine Learning*, 2017, pp. 933–941.

[11] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proc. of International Conference on Machine Learning*, 2017, pp. 1243–1252.

[12] A. Graves, G. Wayne, and I. Danihelka, "Neural Turing machines," *arXiv preprint arXiv:1410.5401*, 2014.

[13] F. Hill, A. Bordes, S. Chopra, and J. Weston, "The Goldilocks principle: Reading children's books with explicit memory representations," in *Proc. of International Conference on Learning Representation*, 2016.

[14] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," *arXiv preprint arXiv:1312.6114*, 2014.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, vol. 25, pp. 1097–1105.

References III

[16] C.-Y. Kuo and J.-T. Chien, "Markov recurrent neural networks," in *Proc. of IEEE International Workshop on Machine Learning for Signal Processing*, 2018, pp. 1–6.

[17] J. Li, A. H. Miller, S. Chopra, M. Ranzato, and J. Weston, "Learning through dialogue interactions," *arXiv preprint arXiv:1612.04936*, 2016.

[18] J. Li, W. Monroe, A. Ritter, and D. Jurafsky, "Deep reinforcement learning for dialogue generation," in *Proc. of Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 1192–1202.

[19] Y. Miao, L. Yu, and P. Blunsom, "Neural variational inference for text processing," in *Proc. of International Conference on Machine Learning*, 2016, pp. 1727–1736.

[20] K. Narasimhan, A. Yala, and R. Barzilay, "Improving information extraction by acquiring external evidence with reinforcement learning," in *Proc. of Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 2355–2365.

[21] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proc. of International Conference on Machine Learning*, 2014, pp. 1278–1286.

[22] D. Serdyuk, R. N. Ke, A. Sordoni, C. Pal, and Y. Bengio, "Twin networks: Using the future as a regularizer," in *Proc. of International Conference on Learning Representations*, 2018.

[23] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus, "End-to-end memory networks," in *Advances in Neural Information Processing Systems 28*, 2015, pp. 2440–2448.

[24] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27*. ., 2014, pp. 3104–3112.

[25] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[26] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3156–3164.

[27] J. Weston, A. Bordes, S. Chopra, A. M. Rush, B. van Merriënboer, A. Joulin, and T. Mikolov, "Towards AI-complete question answering: A set of prerequisite toy tasks," in *Proc. of International Conference on Learning Representation*, 2015.

[28] J. Weston, S. Chopra, and A. Bordes, "Memory networks," in *Proc. of International Conference on Learning Representation*, 2015.

[29] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Advances in Neural Information Processing Systems*, 2015, pp. 802–810.

[30] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems 28*, 2015, pp. 649–657.