Conditional Generation by RNN & Attention
Hung-yi Lee
李宏毅
Outline

• Generation
• Attention
• Tips for Generation
• Pointer Network
Generation

Generating a structured object component-by-component
Generation

- Sentences are composed of characters/words
  - Generating a character/word at each time by RNN

\[
P(w_1) \quad P(w_2 | w_1) \quad P(w_3 | w_1, w_2) \quad P(w_4 | w_1, w_2, w_3) \quad \ldots
\]

http://youtien.pixnet.net/blog/post/4604096-%E6%8E%A8%E6%96%87%E6%8E%A5%E9%BE%8D%E4%B9%8B%E5%B0%8D%E8%81%AF%E9%81%8A%E6%88%B2
Generation

• Images are composed of pixels
  • Generating a pixel at each time by RNN

Consider as a sentence
blue red yellow gray …...

Train a language model based on the “sentences”
Generation

• Images are composed of pixels

3 x 3 images
Generation

• Image
  • Aaron van den Oord, Nal Kalchbrenner, Koray Kavukcuoglu, Pixel Recurrent Neural Networks, arXiv preprint, 2016

• Video
  • Aaron van den Oord, Nal Kalchbrenner, Koray Kavukcuoglu, Pixel Recurrent Neural Networks, arXiv preprint, 2016

• Handwriting
  • Alex Graves, Generating Sequences With Recurrent Neural Networks, arXiv preprint, 2013

• Speech
Conditional Generation

• We don’t want to simply generate some random sentences.
• Generate sentences based on conditions:

*Caption Generation*

Given condition:

“A young girl is dancing.”

*Chat-bot*

Given condition:

“Hello”

“Hello. Nice to see you.”
Conditional Generation

- Represent the input condition as a vector, and consider the vector as the input of RNN generator

**Image Caption Generation**

![Diagram]

- CNN
- A vector
- <BOS>
- Input image
- A
- woman
- ....
- (period)
Conditional Generation

• Represent the input condition as a vector, and consider the vector as the input of RNN generator
• E.g. Machine translation / Chat-bot
Conditional Generation

M: Hello

U: Hi

M: Hi

Need to consider longer context during chatting

https://www.youtube.com/watch?v=e2MpOmyQJw4

Attention

Dynamic Conditional Generation
Dynamic Conditional Generation

Encoder

Information of the whole sentences

Decoder

machine

learning

. (period)

$c^1$

$c^2$

$c^3$
Machine Translation

- Attention-based model

\[ \alpha \]

Jointly learned with other part of the network

What is match?

Design by yourself

- Cosine similarity of \( z \) and \( h \)
- Small NN whose input is \( z \) and \( h \), output a scalar

\[ \alpha = h^T W z \]
Machine Translation

- Attention-based model

\[ c^0 = \sum \hat{\alpha}_0^i h^i \]
\[ = 0.5h^1 + 0.5h^2 \]
Machine Translation

• Attention-based model
Machine Translation

• Attention-based model

\[ c^1 = \sum \hat{\alpha}_i h^i \]
\[ = 0.5h^3 + 0.5h^4 \]
Machine Translation

• Attention-based model

The same process repeat until generating . (period)
Speech Recognition

Image Caption Generation

A vector for each region

CNN

z^0 → match → 0.7
Image Caption Generation

A vector for each region

CNN

weighted sum

Word 1

$z^0 \rightarrow z^1$

0.7
0.1
0.1
0.0
0.0

filter
filter
filter
filter

filter
filter
filter
filter
Image Caption Generation

A vector for each region

CNN

Word 1

Word 2

$z^0 \rightarrow z^1 \rightarrow z^2$

weighted sum

0.0 0.0 0.0 0.0

0.0 0.8 0.2

0.0 0.0 0.0 0.0
Image Caption Generation

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.

Image Caption Generation

A large white bird standing in a forest.

A woman holding a clock in her hand.

A man wearing a hat and a hat on a skateboard.

A person is standing on a beach with a surfboard.

A woman is sitting at a table with a large pizza.

A man is talking on his cell phone while another man watches.

Ref: A man and a woman ride a motorcycle
A man and a woman are talking on the road

Ref: A woman is frying food
Someone is frying a fish in a pot
Sentence to vector can be jointly trained.

\[ \sum_{n=1}^{N} \alpha_n x^n \]

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, Rob Fergus, “End-To-End Memory Networks”, NIPS, 2015
Jointly learned

Memory Network

Extracted Information

\[ \sum_{n=1}^{N} \alpha_n h^n \]

DNN

Hopping

Match

q

Document

Query

Answer
Wei Fang, Juei-Yang Hsu, Hung-yi Lee, Lin-Shan Lee, "Hierarchical Attention Model for Improved Machine Comprehension of Spoken Content", SLT, 2016
Neural Turing Machine

- von Neumann architecture

Neural Turing Machine not only read from memory
Also modify the memory through attention

https://www.quora.com/How-does-the-Von-Neumann-architecture-provide-flexibility-for-program-development
Neural Turing Machine

\[ r^0 = \sum \hat{\alpha}_0 m^i \]

Long term memory

Retrieval process
Neural Turing Machine

\[ r^0 = \sum \hat{\alpha}_0^i m_0^i \]

\[ \alpha_1^i = \cos(m_0^i, k^1) \]
Neural Turing Machine

• Real version

\[ \begin{align*}
  k^t &= \begin{pmatrix} 1 \\ 2 \end{pmatrix} \\
  \beta^t &= 50 \\
  g^t &= 0.5 \\
  s^t &= \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \\
  \gamma^t &= 50
\end{align*} \]
Neural Turing Machine

\[ m^i_1 = m^i_0 - \hat{\alpha}^i_1 e^1 \odot m^i_0 + \hat{\alpha}^i_1 a^1 \]

(element-wise)

\[ k^1 \]

\[ e^1 \]

\[ a^1 \]

\[ 0 \sim 1 \]
Neural Turing Machine
Tips for Generation
Attention

component

$\alpha_t^i$

time

Bad Attention

$\alpha_1^1 \alpha_2^1 \alpha_3^1 \alpha_4^1$

$\alpha_1^2 \alpha_2^2 \alpha_3^2 \alpha_4^2$

$\alpha_1^3 \alpha_2^3 \alpha_3^3 \alpha_4^3$

$\alpha_1^4 \alpha_2^4 \alpha_3^4 \alpha_4^4$

$w_1 \quad w_2 \, (\text{woman}) \quad w_3 \quad w_4 \, (\text{woman}) \quad \ldots \quad \text{no cooking}$

Good Attention: each input component has approximately the same attention weight

E.g. Regularization term: $\sum_i \left( \tau - \sum_t \alpha_t^i \right)^2$

For each component

Over the generation

Mismatch between Train and Test

• **Training**

\[ C = \sum_t C_t \]

Minimizing cross-entropy of each component

Reference:
Mismatch between Train and Test

• **Generation**

We do not know the reference

Testing: Output of model is the input of the next step.

Training: the inputs are reference.

*Exposure Bias*
One step wrong

May be totally wrong

Never explore ……

一步错，步步错
Modifying Training Process?

When we try to decrease the loss for both step 1 and 2 ..... 

Training is matched to testing.

In practice, it is hard to train in this way.
Scheduled Sampling

From model

From reference

Reference

Exponential decay
Inverse sigmoid decay
Linear decay

0 200 400 600 800 1000

0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
Scheduled Sampling

- Caption generation on MSCOCO

<table>
<thead>
<tr>
<th></th>
<th>BLEU-4</th>
<th>METEOR</th>
<th>CIDER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Always from reference</td>
<td>28.8</td>
<td>24.2</td>
<td>89.5</td>
</tr>
<tr>
<td>Always from model</td>
<td>11.2</td>
<td>15.7</td>
<td>49.7</td>
</tr>
<tr>
<td>Scheduled Sampling</td>
<td>30.6</td>
<td>24.3</td>
<td>92.1</td>
</tr>
</tbody>
</table>

Beam Search

The green path has higher score.
Not possible to check all the paths
Beam Search

Keep several best path at each step

Beam size = 2
Beam Search

The size of beam is 3 in this example.

https://github.com/tensorflow/tensorflow/issues/654#issuecomment-169009989
Better Idea?

U: 你覺得如何？
M: 高興想笑 or 難過想哭

高興

高興~難過

高興

想笑

想笑~想哭

高興～難過

High score
Object level v.s. Component level

- Minimizing the error defined on component level is not equivalent to improving the generated objects

Ref: The dog is running fast

\[ C = \sum_t C_t \]

Cross-entropy of each step

Optimize object-level criterion instead of component-level cross-entropy. Object-level criterion: \( R(y, \hat{y}) \)

\( y \): generated utterance, \( \hat{y} \): ground truth
Reinforcement learning?

Start with observation $s_1$

Action $a_1$: “right”

Obtain reward $r_1 = 0$

Observation $s_2$

Action $a_2$: “fire” (kill an alien)

Obtain reward $r_2 = 5$
Reinforcement learning?

The action we take influence the observation in the next step.

Scheduled sampling

reinforcement

<table>
<thead>
<tr>
<th>Task</th>
<th>DaD</th>
<th>E2E</th>
<th>MIXER</th>
</tr>
</thead>
<tbody>
<tr>
<td>summarization</td>
<td>0.95</td>
<td>0.95</td>
<td>1.25</td>
</tr>
<tr>
<td>machine translation</td>
<td>1.10</td>
<td>1.10</td>
<td>1.15</td>
</tr>
<tr>
<td>image captioning</td>
<td>1.05</td>
<td>1.05</td>
<td>1.05</td>
</tr>
</tbody>
</table>
DAD: Scheduled Sampling
MIXER: reinforcement
Pointer Network

Oriol Vinyals, Meire Fortunato, Navdeep Jaitly, Pointer Network, NIPS, 2015
<table>
<thead>
<tr>
<th>Method</th>
<th>Trained n</th>
<th>n</th>
<th>Accuracy</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM [1]</td>
<td>50</td>
<td>50</td>
<td>1.9%</td>
<td>FAIL</td>
</tr>
<tr>
<td>+Attention [5]</td>
<td>50</td>
<td>50</td>
<td>38.9%</td>
<td>99.7%</td>
</tr>
<tr>
<td>Ptr-Net</td>
<td>50</td>
<td>50</td>
<td>72.6%</td>
<td>99.9%</td>
</tr>
<tr>
<td>LSTM [1]</td>
<td>5</td>
<td>5</td>
<td>87.7%</td>
<td>99.6%</td>
</tr>
<tr>
<td>Ptr-Net</td>
<td>5-50</td>
<td>5</td>
<td>92.0%</td>
<td>99.6%</td>
</tr>
<tr>
<td>LSTM [1]</td>
<td>10</td>
<td>10</td>
<td>29.9%</td>
<td>FAIL</td>
</tr>
<tr>
<td>Ptr-Net</td>
<td>5-50</td>
<td>10</td>
<td>87.0%</td>
<td>99.8%</td>
</tr>
<tr>
<td>Ptr-Net</td>
<td>5-50</td>
<td>50</td>
<td>69.6%</td>
<td>99.9%</td>
</tr>
<tr>
<td>Ptr-Net</td>
<td>5-50</td>
<td>100</td>
<td>50.3%</td>
<td>99.9%</td>
</tr>
<tr>
<td>Ptr-Net</td>
<td>5-50</td>
<td>200</td>
<td>22.1%</td>
<td>99.9%</td>
</tr>
<tr>
<td>Ptr-Net</td>
<td>5-50</td>
<td>500</td>
<td>1.3%</td>
<td>99.2%</td>
</tr>
</tbody>
</table>
(a) LSTM, $m=50$, $n=50$  
(d) Ptr-Net, $m=5-50$, $n=500$
Applications


Machine Translation

French: Guillaume et Cesar ont une voiture bleue à Lausanne.

English: Guillaume and Cesar have a blue car in Lausanne.

Chat-bot

User: X你好，我是宏毅

Machine: 宏毅你好，很高興認識你