Recurrent Neural Network (RNN)
Example Application

- Slot Filling

I would like to arrive **Taipei** on **November 2\(^{nd}\)**.
Example Application

Solving slot filling by Feedforward network?

Input: a word

(Each word is represented as a vector)
1-of-N encoding

How to represent each word as a vector?

1-of-N Encoding  \[ \text{lexicon} = \{\text{apple, bag, cat, dog, elephant}\} \]

The vector is lexicon size.

Each dimension corresponds to a word in the lexicon

The dimension for the word is 1, and others are 0

apple = [1 0 0 0 0 0]

bag = [0 1 0 0 0 0]

cat = [0 0 1 0 0 0]

dog = [0 0 0 1 0 0]

elephant = [0 0 0 0 0 1]
Beyond 1-of-N encoding

**Dimension for “Other”**

<table>
<thead>
<tr>
<th>Word</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>apple</td>
<td>0</td>
</tr>
<tr>
<td>bag</td>
<td>0</td>
</tr>
<tr>
<td>cat</td>
<td>0</td>
</tr>
<tr>
<td>dog</td>
<td>0</td>
</tr>
<tr>
<td>elephant</td>
<td>0</td>
</tr>
<tr>
<td>“other”</td>
<td>1</td>
</tr>
</tbody>
</table>

**Word hashing**

<table>
<thead>
<tr>
<th>Word</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>a-a-a</td>
<td>0</td>
</tr>
<tr>
<td>a-a-b</td>
<td>0</td>
</tr>
<tr>
<td>p-l-e</td>
<td>1</td>
</tr>
<tr>
<td>p-p-l</td>
<td>1</td>
</tr>
</tbody>
</table>

26 X 26 X 26

w = “apple”

w = “Gandalf”

w = “Sauron”
Example Application

Solving slot filling by Feedforward network?

Input: a word

(Each word is represented as a vector)

Output:

Probability distribution that the input word belonging to the slots

Taipei
Example Application

Problem?

Neural network needs memory!

Taipei

place of departure

arrive
Taipei on November 2nd

other dest other time time

leave
Taipei on November 2nd

dest
time of departure

y1

y2

X1

X2
Recurrent Neural Network (RNN)

The output of hidden layer are stored in the memory.

Memory can be considered as another input.
Example

Input sequence: $\begin{bmatrix} 1 \\ 1 \\ 1 \\ 2 \end{bmatrix}$ .......

output sequence: $\begin{bmatrix} 4 \\ 4 \end{bmatrix}$

All the weights are “1”, no bias
All activation functions are linear
Example

Input sequence: \[
\begin{bmatrix}
1 \\ 1 \\ 2
\end{bmatrix}
\cdots
\begin{bmatrix}
1 \\ 1 \\ 2
\end{bmatrix}
\]

Output sequence: \[
\begin{bmatrix}
4 \\ 12
\end{bmatrix}
\begin{bmatrix}
4 \\ 12
\end{bmatrix}
\]

All the weights are “1”, no bias
All activation functions are linear
Example

Input sequence: 
\[
\begin{bmatrix}
1 \\
1 \\
2 \\
\end{bmatrix}
\]
Output sequence: 
\[
\begin{bmatrix}
4 \\
12 \\
32 \\
\end{bmatrix}
\]

Changing the sequence order will change the output.

All the weights are “1”, no bias
All activation functions are linear
The same network is used again and again.

Probability of “arrive” in each slot

Probability of “Taipei” in each slot

Probability of “on” in each slot

arrive    Taipei    on    November 2

RNN

The values stored in the memory is different.

Prob of “leave” in each slot

Prob of “Taipei” in each slot

Prob of “arrive” in each slot

Prob of “Taipei” in each slot

Different
Of course it can be deep ...
Elman Network & Jordan Network

**Elman Network**

\[
y^t \\
\uparrow W^o \\
\cdots \\
\uparrow W^h \\
\cdots \\
x^t
\]

\[
y^{t+1} \\
\uparrow W^o \\
\cdots \\
\uparrow W^h \\
\cdots \\
x^{t+1}
\]

**Jordan Network**

\[
y^t \\
\uparrow W^o \\
\cdots \\
\uparrow W^h \\
\cdots \\
x^t
\]

\[
y^{t+1} \\
\uparrow W^o \\
\cdots \\
\uparrow W^h \\
\cdots \\
x^{t+1}
\]
Bidirectional RNN

\[ y_{t+1} \]

\[ y_{t} \]

\[ x_{t} \]

\[ x_{t+1} \]

\[ x_{t+2} \]
Long Short-term Memory (LSTM)

- Input Gate
- Output Gate
- Forget Gate
- Memory Cell

LSTM: Special Neuron
- 4 inputs, 1 output

Signal control:
- the input gate
- the output gate
- the forget gate
Activation function $f$ is usually a sigmoid function between 0 and 1. Mimic open and close gate.

$$c' = g(z)f(z_i) + cf(z_f)$$
When $x_2 = 1$, add the numbers of $x_1$ into the memory.

When $x_2 = -1$, reset the memory.

When $x_3 = 1$, output the number in the memory.
Original Network:

- Simply replace the neurons with LSTM

Diagram:

- $\vdots$
- $a_1$
- $z_1$
- $x_1$
- $x_2$
- $a_2$
- $z_2$
- Input
4 times of parameters
LSTM

\[ \mathbf{x}_t, \mathbf{c}_{t-1} \]

4 vectors
LSTM

\[ x_t \times z \]

\[ y_t \]

\[ c_{t-1} \]

\[ z_i \]

\[ z_f \]

\[ z_o \]

\[ z^o \]

Output Gate

Forget Gate

Input Gate

Cell

Block

\[ z^f \]
\[
\begin{align*}
LSTM & : \\
& \text{Extension: “peephole”}
\end{align*}
\]
Multiple-layer LSTM

Don’t worry if you cannot understand this. Keras can handle it.

Keras supports “LSTM”, “GRU”, “SimpleRNN” layers

This is quite standard now.

https://img.komicolle.org/2015-09-20/src/14426967627131.gif
Learning Target

Training Sentences:
- arrive
- Taipei
- on
- November 2\textsuperscript{nd}
- other
dest
- other
time
time
Learning

Backpropagation through time (BPTT)

\[ w \leftarrow w - \eta \frac{\partial L}{\partial w} \]
Unfortunately ......

• RNN-based network is not always easy to learn

Real experiments on Language modeling

Total Loss

Epoch

sometimes

Lucky
The error surface is rough.

The error surface is either very flat or very steep.

[Razvan Pascanu, ICML’13]
Why?

\[ w = 1 \quad \rightarrow \quad y^{1000} = 1 \]
\[ w = 1.01 \quad \rightarrow \quad y^{1000} \approx 20000 \]
\[ w = 0.99 \quad \rightarrow \quad y^{1000} \approx 0 \]
\[ w = 0.01 \quad \rightarrow \quad y^{1000} \approx 0 \]

Toy Example

\[
\begin{align*}
y^1 &= 1 \\
w &= \text{small learning rate?} \\
y^2 &= 1 \\
w &= \text{large learning rate?} \\
y^3 &= 1 \\
&\quad\cdots\cdots\quad\text{small learning rate?} \\
y^{1000} &= 0
\end{align*}
\]

= \text{small learning rate?}
Helpful Techniques

• Long Short-term Memory (LSTM)
  • Can deal with gradient vanishing (not gradient explode)
    ➢ Memory and input are added
    ➢ The influence never disappears unless forget gate is closed

No Gradient vanishing (If forget gate is opened.)

Gated Recurrent Unit (GRU): simpler than LSTM

[Cho, EMNLP’14]
Helpful Techniques

Vanilla RNN Initialized with Identity matrix + ReLU activation function [Quoc V. Le, arXiv’15]

➢ Outperform or be comparable with LSTM in 4 different tasks
More Applications ......

Input and output are both sequences with the same length

RNN can do more than that!

Probability of “arrive” in each slot

Probability of “Taipei” in each slot

Probability of “on” in each slot

arrive  Taipei  on  November  2\textsuperscript{nd}
Many to one

• Input is a vector sequence, but output is only one vector

**Sentiment Analysis**

<table>
<thead>
<tr>
<th>Positive (正雷)</th>
<th>Negative (負雷)</th>
<th>Positive (正雷)</th>
</tr>
</thead>
<tbody>
<tr>
<td>看了這部電影覺得很高興 ......</td>
<td>這部電影太糟了 ......</td>
<td>這部電影很棒 ......</td>
</tr>
</tbody>
</table>

超好雷
好雷
普雷
負雷
超負雷
Many to one

• Input is a vector sequence, but output is only one vector

Key Term Extraction

Key Terms: DNN, LSTN

Embedding Layer

Output Layer

Hidden Layer

\[ \sum \alpha_i V_i \]
Many to Many (Output is shorter)

• Both input and output are both sequences, **but the output is shorter.**
  • E.g. **Speech Recognition**

Input: (character sequence)
Output: “好棒” (character sequence)

Problem?

Why can’t it be “好棒棒”

Trimming

Input: (vector sequence)

Output: (vector sequence)
Many to Many (Output is shorter)

- Both input and output are both sequences, **but the output is shorter.**
- Connectionist Temporal Classification (CTC) [Alex Graves, ICML’06][Alex Graves, ICML’14][Haşim Sak, Interspeech’15][Jie Li, Interspeech’15][Andrew Senior, ASRU’15]

Add an extra symbol “φ” representing “null”
Many to Many (Output is shorter)

- CTC: Training

Acoustic Features:

Label: 好 棒

All possible alignments are considered as correct.
Many to Many (Output is shorter)

• CTC: example

Many to Many (No Limitation)

• Both input and output are both sequences with different lengths. → Sequence to sequence learning
  • E.g. Machine Translation (machine learning → 機器學習)

Containing all information about input sequence
Many to Many (No Limitation)

- Both input and output are both sequences with different lengths. → Sequence to sequence learning
- E.g. **Machine Translation** (machine learning → 機器學習)

Don’t know when to stop
Many to Many (No Limitation)

接龍推文是 ptt 在推文中的一種趣味玩法，與推齊有些類似但又有所不同，是指在推文中接續上一樓的字句，而推出連續的意思。該類玩法確切起源已不可知(鄉民百科)
Many to Many (No Limitation)

• Both input and output are both sequences with different lengths. → Sequence to sequence learning
  • E.g. **Machine Translation** (machine learning → 機器學習)

Add a symbol “===“ (斷)

[Ilya Sutskever, NIPS’14][Dzmitry Bahdanau, arXiv’15]
Many to Many (No Limitation)

- Both input and output are both sequences with different lengths. → **Sequence to sequence learning**
  - E.g. **Machine Translation** (machine learning → 機器學習)

![Machine translation alignment](https://arxiv.org/pdf/1612.01744v1.pdf)

Figure 1: Alignments performed by the attention model during training
Beyond Sequence

• Syntactic parsing

John has a dog .  →  S  
                  NP  VP
                NNP  VBZ
                john  has

John has a dog .  →  (S (NP NNP )_{NP} (VP VBZ (NP DT NN )_{NP} )_{VP} . )_{S}
To understand the meaning of a word sequence, the order of the words cannot be ignored.

- white blood cells destroying an infection
- exactly the same bag-of-word
- an infection destroying white blood cells
- positive
- different meaning
- negative

Sequence-to-sequence Auto-encoder - Text
Sequence-to-sequence Auto-encoder - Speech

- Dimension reduction for a sequence with variable length audio segments (word-level) → Fixed-length vector

Sequence-to-sequence Auto-encoder - Speech

Audio archive divided into variable-length audio segments

Audio Segment to Vector

On-line

Spoken Query

Audio Segment to Vector

Similarity

Search Result

Off-line
Sequence-to-sequence Auto-encoder - Speech

RNN Encoder

The values in the memory represent the whole audio segment

The vector we want

How to train RNN Encoder?

acoustic features

audio segment

vector
Sequence-to-sequence Auto-encoder

The RNN encoder and decoder are jointly trained.

RNN Encoder

RNN Decoder

Input acoustic features

acoustic features

audio segment
Sequence-to-sequence Auto-encoder - Speech

• Visualizing embedding vectors of the words

fear, name, fame
Demo: Chat-bot

電視影集 (~40,000 sentences)、美國總統大選辯論
Demo: Chat-bot

• Develop Team
  • Interface design: Prof. Lin-Lin Chen & Arron Lu
  • Web programming: Shi-Yun Huang
  • Data collection: Chao-Chuang Shih
  • System implementation: Kevin Wu, Derek Chuang, & Zhi-Wei Lee (李致緯), Roy Lu (盧柏儒)
  • System design: Richard Tsai & Hung-Yi Lee
Demo: Video Caption Generation

- Video
  - A group of people is knocked by a tree.
  - A group of people is walking in the forest.
  - A girl is running.
Demo: Video Caption Generation

• Can machine describe what it see from video?
• Demo: 台大語音處理實驗室 曾柏翔、吳柏瑜、盧宏宗
• Video: 莊舜博、楊棋宇、黃邦齊、萬家宏
Demo: Image Caption Generation

• Input an image, but output a sequence of words

[Kelvin Xu, arXiv’15][Li Yao, ICCV’15]
Demo: Image Caption Generation

• Can machine describe what it see from image?
• Demo: 台大電機系 大四 蘇子睿、林奕辰、徐翊祥、陳奕安

MTK 產學大聯盟
Attention-based Model

What is deep learning?

What you learned in these lectures

Breakfast today

summer vacation 10 years ago

Answer

Organize

http://henrylo1605.blogspot.tw/2015/05/blog-post_56.html
Attention-based Model

Ref:
Attention-based Model v2

Input → DNN/RNN → output

- **Reading Head Controller**
- **Writing Head Controller**

... → **Machine’s Memory** → ...
Reading Comprehension

Query → DNN/RNN → answer

Reading Head Controller

Semantic Analysis

......

Each sentence becomes a vector.
Reading Comprehension


The position of reading head:

<table>
<thead>
<tr>
<th>Story (16: basic induction)</th>
<th>Support</th>
<th>Hop 1</th>
<th>Hop 2</th>
<th>Hop 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brian is a frog.</td>
<td>yes</td>
<td>0.00</td>
<td>0.98</td>
<td>0.00</td>
</tr>
<tr>
<td>Lily is gray.</td>
<td></td>
<td>0.07</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Brian is yellow.</td>
<td>yes</td>
<td>0.07</td>
<td>0.00</td>
<td>1.00</td>
</tr>
<tr>
<td>Julius is green.</td>
<td></td>
<td>0.06</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Greg is a frog.</td>
<td></td>
<td>0.76</td>
<td>0.02</td>
<td>0.00</td>
</tr>
</tbody>
</table>

What color is Greg? Answer: yellow Prediction: yellow

Keras has example: https://github.com/fchollet/keras/blob/master/examples/babi_memnn.py
Visual Question Answering

What is the mustache made of?

source: http://visualqa.org/
Visual Question Answering

Query -> DNN/RNN -> answer

Reading Head Controller

CNN

A vector for each region
Speech Question Answering

• **TOEFL Listening Comprehension Test by Machine**
• Example:

  Audio Story:  
  (The original story is 5 min long.)

  Question: “What is a possible origin of Venus’ clouds?”

  Choices:
  
  (A) gases released as a result of volcanic activity
  (B) chemical reactions caused by high surface temperatures
  (C) bursts of radio energy from the plane's surface
  (D) strong winds that blow dust into the atmosphere
Question: “what is a possible origin of Venus’ clouds?”

...... It be quite possible that this be due to volcanic eruption because volcanic eruption often emit gas. If that be the case volcanism could very well be the root cause of Venus 's thick cloud cover. And also we have observe burst of radio energy from the planet 's surface. These burst be similar to what we see when volcano erupt on earth ......
Simple Baselines

Naive Approaches

- **(1)** random
- **(2)** select the **shortest** choice as answer
- **(3)**
- **(4)** the choice with semantic most similar to others
- **(5)**

Experimental setup: 717 for training, 124 for validation, 122 for testing
Memory Network

Memory Network: 39.2%
(proposed by FB AI group)

Accuracy (%)
Proposed Approach

Accuracy (%)

Memory Network: 39.2%
(proposed by FB AI group)

Proposed Approach: 48.8%

[Tseng & Lee, Interspeech 16]
[Fang & Hsu & Lee, SLT 16]
To Learn More ......

- The Unreasonable Effectiveness of Recurrent Neural Networks
  - http://karpathy.github.io/2015/05/21/rnn-effectiveness/
- Understanding LSTM Networks
  - http://colah.github.io/posts/2015-08-Understanding-LSTMs/
Deep & Structured
RNN v.s. Structured Learning

• RNN, LSTM
  • Unidirectional RNN does NOT consider the whole sequence
  • Cost and error not always related
  • Deep

• HMM, CRF, Structured Perceptron/SVM
  • Using Viterbi, so consider the whole sequence
  • How about Bidirectional RNN?
  • Can explicitly consider the label dependency
  • Cost is the upper bound of error
Integrated Together

- Explicitly model the dependency
- Cost is the upper bound of error

HMM, CRF, Structured Perceptron/SVM

Deep

RNN, LSTM

http://photo30.bababian.com/upload1/20100415/42E9331A6580A46A5F89E98638B8FD76.jpg
Integrated together

• Speech Recognition: CNN/LSTM/DNN + HMM

\[ P(x, y) = P(y_1|\text{start}) \prod_{l=1}^{L-1} P(y_{l+1}|y_l) P(\text{end}|y_L) \prod_{l=1}^{L} P(x_l|y_l) \]

\[ P(x_l|y_l) = \frac{P(x_l, y_l)}{P(y_l)} \]

\[ = \frac{P(y_l|x_l)P(x_l)}{P(y_l)} \]
Integrated together

• Semantic Tagging: Bi-directional LSTM + CRF/Structured SVM

\[ \tilde{y} = \arg \max_{y \in \mathcal{Y}} w \cdot \phi(x, y) \]
Is structured learning practical?

• Considering GAN

Inference:

\[ x = \arg \max_x F(x) \]

Problem 2

Noise From Gaussian

Generator

Real x

Generated

Discriminator

Evaluation Function F(x)

Problem 3: you know how to learn F(x)

Problem 1

1/0
Is structured learning practical?

- Conditional GAN

$$x \xrightarrow{} \text{Generator} \xrightarrow{} y$$

Real $(x,y)$ pair

$$1/0$$

This flower has small, round violet petals with a dark purple center

$$\varphi \xrightarrow{} \varphi(t)$$

$$z \sim \mathcal{N}(0, 1)$$

$$\hat{x} := G(z, \varphi(t))$$

This flower has small, round violet petals with a dark purple center

$$\varphi \xrightarrow{} \varphi(t)$$

$$D(\hat{x}, \varphi(t))$$
Sounds crazy?
People do think in this way ...

- Connect Energy-based model with GAN:
  - A Connection Between Generative Adversarial Networks, Inverse Reinforcement Learning, and Energy-Based Models
  - Deep Directed Generative Models with Energy-Based Probability Estimation
  - ENERGY-BASED GENERATIVE ADVERSARIAL NETWORKS
- Deep learning model for inference
  - Deep Unfolding: Model-Based Inspiration of Novel Deep Architectures
  - Conditional Random Fields as Recurrent Neural Networks
Machine learning and having it deep and structured (MLDS)

• 和 ML 的不同
  • 在這學期 ML 中有提過的內容 (DNN, CNN ...), 在 MLDS 中不再重複，只做必要的復習

• 教科書： “Deep Learning” (http://www.deeplearningbook.org/)
  • Part II 是講 deep learning 、Part III 就是講 structured learning

• Part II: Modern Practical Deep Networks
  ○ 6 Deep Feedforward Networks
  ○ 7 Regularization for Deep Learning
  ○ 8 Optimization for Training Deep Models
  ○ 9 Convolutional Networks
  ○ 10 Sequence Modeling: Recurrent and Recu
  ○ 11 Practical Methodology
  ○ 12 Applications

• Part III: Deep Learning Research
  ○ 13 Linear Factor Models
  ○ 14 Autoencoders
  ○ 15 Representation Learning
  ○ 16 Structured Probabilistic Models for Deep Learning
  ○ 17 Monte Carlo Methods
  ○ 18 Confronting the Partition Function
  ○ 19 Approximate Inference
  ○ 20 Deep Generative Models
Machine learning and having it deep and structured (MLDS)

• 所有作業都 2 ~ 4 人一組，可以先組好隊後一起來修
• MLDS 的作業和之前不同
  • RNN (把之前 MLDS 的三個作業合為一個) 、Attention-based model 、Deep Reinforcement Learning 、Deep Generative Model 、Sequence-to-sequence learning
• MLDS 初選不開放加簽，以組為單位加簽，作業 0 的內容是做一個 DNN （可用現成套件）