https://datascienceschool.net/view-notebook/1d93b9dc6c624fbaa6af2ce9290e2479/

Deep Learning

RNNs In Keras

Yoon Joong Kim
yjkim@hanbat.ac.kr

Yoon Joong Kim
Department of Computer Engineering, Hanbat National University
yjkim@hanbat.ac.kr

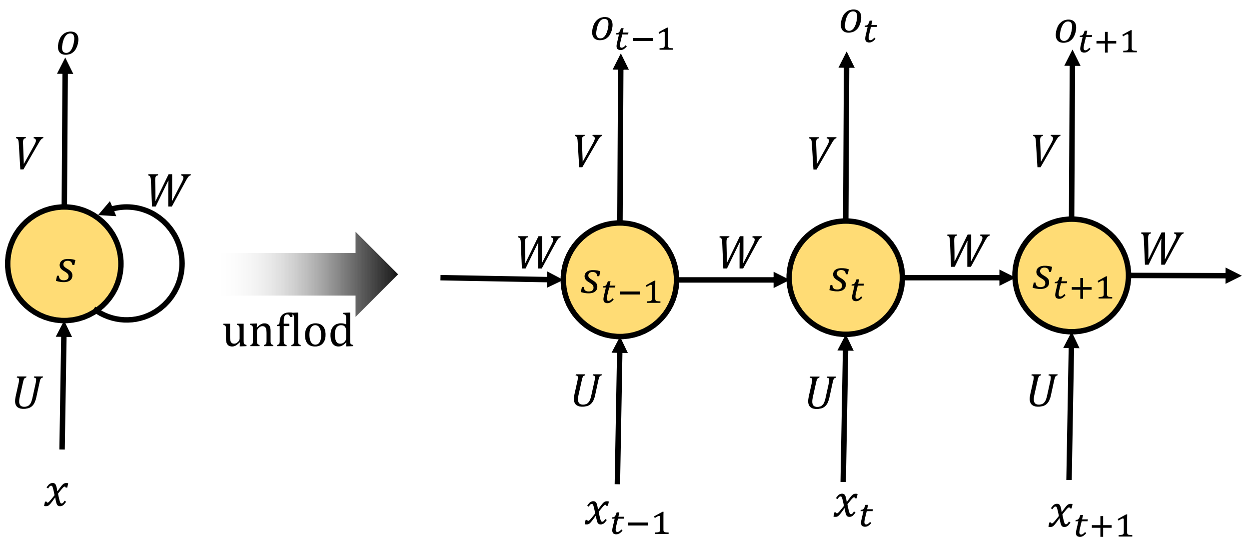
Agenda

1. RNNs
   1. From feed-forward to RNNs
   2. Simple Recurrent Neural Network (SRNN)
   3. RNNs in the context of NLP
   4. The problem with RNNs
   5. LSTM
2. RNN Applications
   1. Language Modeling
   2. Character-level Language Modeling
   3. Neural Machine Translation(Google Research’s blog)
   4. Text Summarization
   5. Image Captioning
3. Language Modeling
   1. Language Modeling DEMO Character-level, Language Modeling
4. Examples in Keras
   1. Example 1. 정현파신호 샘플예측 모델(SRNN)
   2. Example 2. 문자 기반 신경 언어 모델-sixpence
   3. Example 3. 주가예측 모델

1. RNN

- 1.1 From feed-forward to RNNs

$$s_t = \sigma(Ux_t + Ws_{t-1})$$
$$o = \sigma(Vs_t)$$

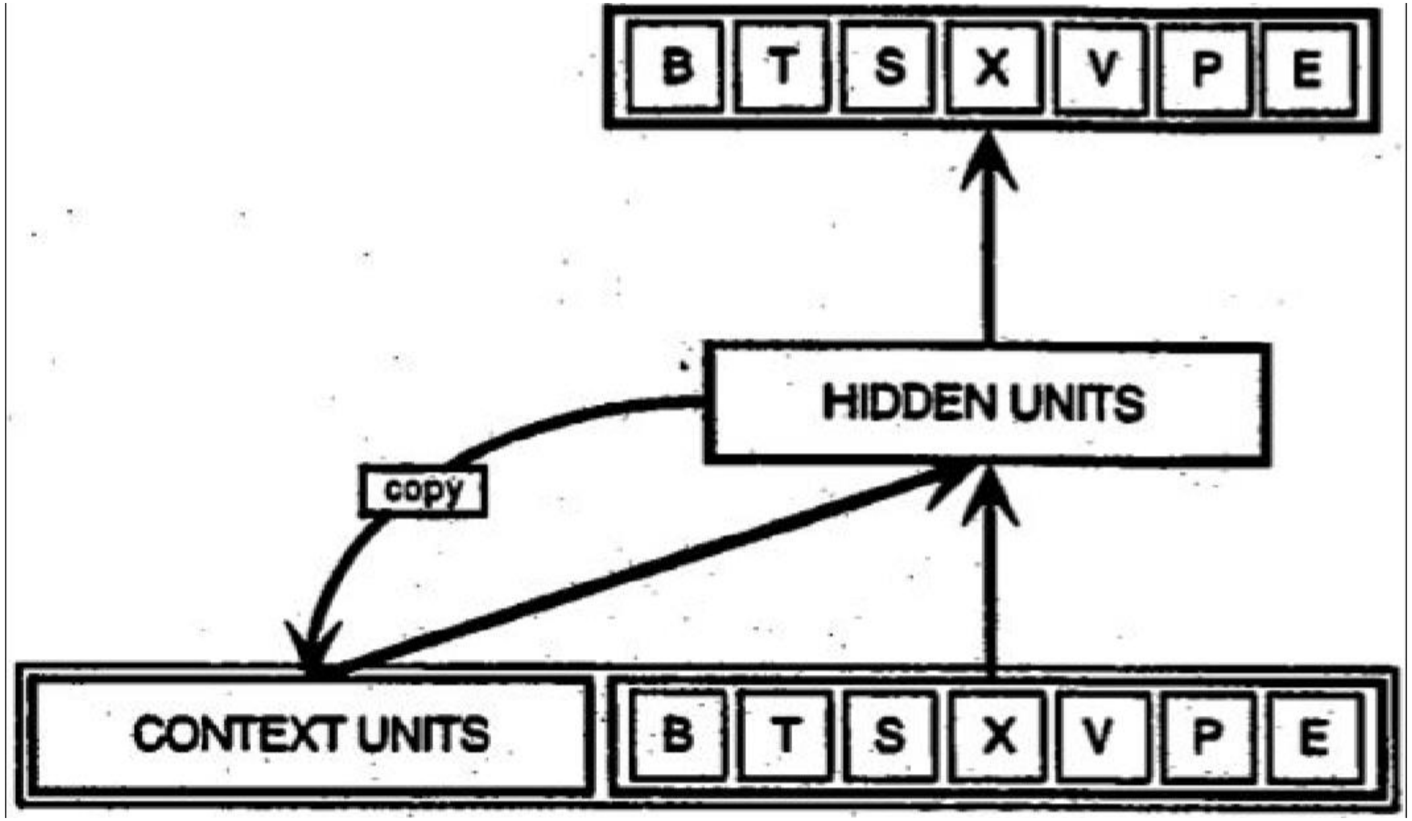


1. RNN(cont.)

- 1.1 RNN(Recurrent Neural Network)
  - RNN은 순차적 데이터 (텍스트, 게놈, 음성 단어 등)의 정보를 활용하는 신경회로망이다.
  - Directed cycles
  - 모든 단계(step)는 가중치를 공유한다. 따라서 하여 총 매개 변수 수가 줄어든다.
  - NLP(Natural Language Processing)의 중추를 형성한다.
  - 이미지처리에도 사용될 수 있습니다

1. RNN(cont.)

- 1.2 Simple Recurrent Neural Network (SRNN)
  - Introduced by Jeffrey Elman in 1990. Also known as Elman Network
  - Elman, Jeffrey L. "Finding structure in time." Cognitive science 14.2 (1990): 179-211



1. RNN(cont.)

- SRNNs(Simple RNN) are Simple

Elman and Jordan networks are also known as "simple recurrent networks" (SRN).

Elman network[10]

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h)$$
$$y_t = \sigma_y(W_y h_t + b_y)$$

Jordan network[11]

$$h_t = \sigma_h(W_h x_t + U_h y_{t-1} + b_h)$$
$$y_t = \sigma_y(W_y h_t + b_y)$$

Variables and functions

- x_t : input vector
- h_t : hidden layer vector
- y_t : output vector
- W, U and b: parameter matrices and vector
- σ_h and σ_y: Activation functions

1. RNN(cont.)

- 1.3 RNNs in the context of NLP

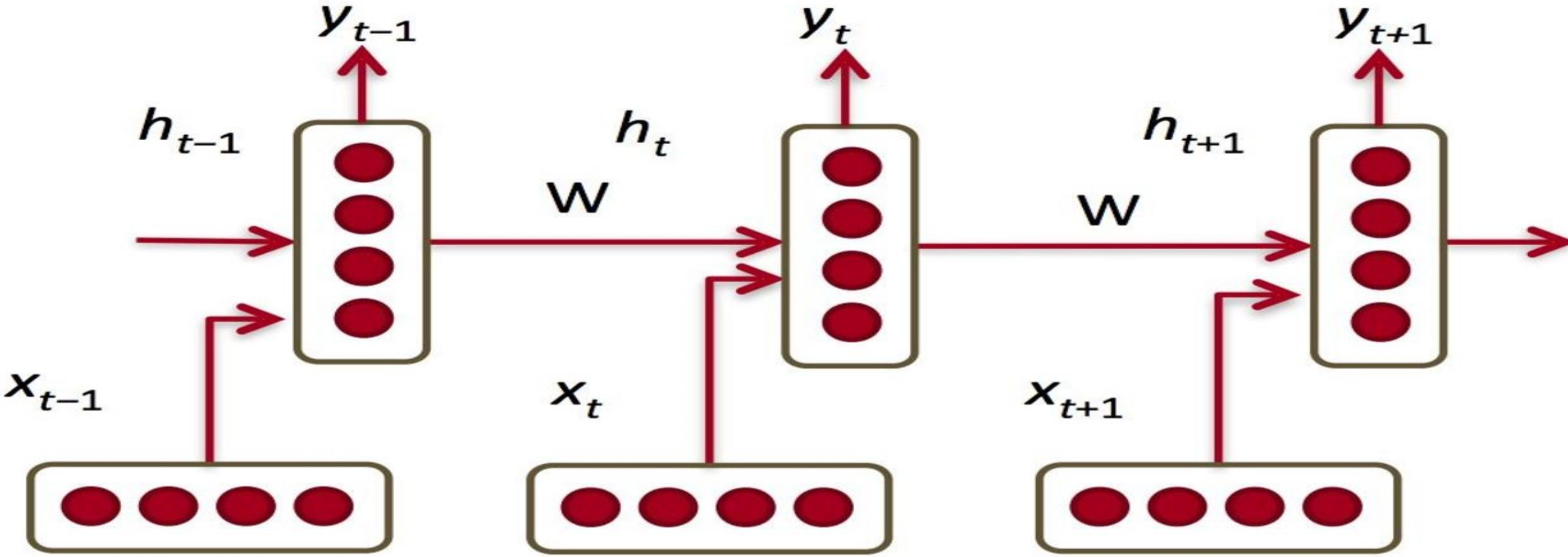


Diagram from CS 224D Slides

1. RNN(cont.)

- 1.4 The problem with RNNs
  - RNN은 단기종속성(short-term dependencies)는 잘 파악하지만 장기종속성(long-term dependencies)은 잘 파악하지 못합니다.

  - 예
    - “I grew up in France… I speak fluent ?”
      -> Needs information from way back

1.5 LSTM(Long Short Term Memory)

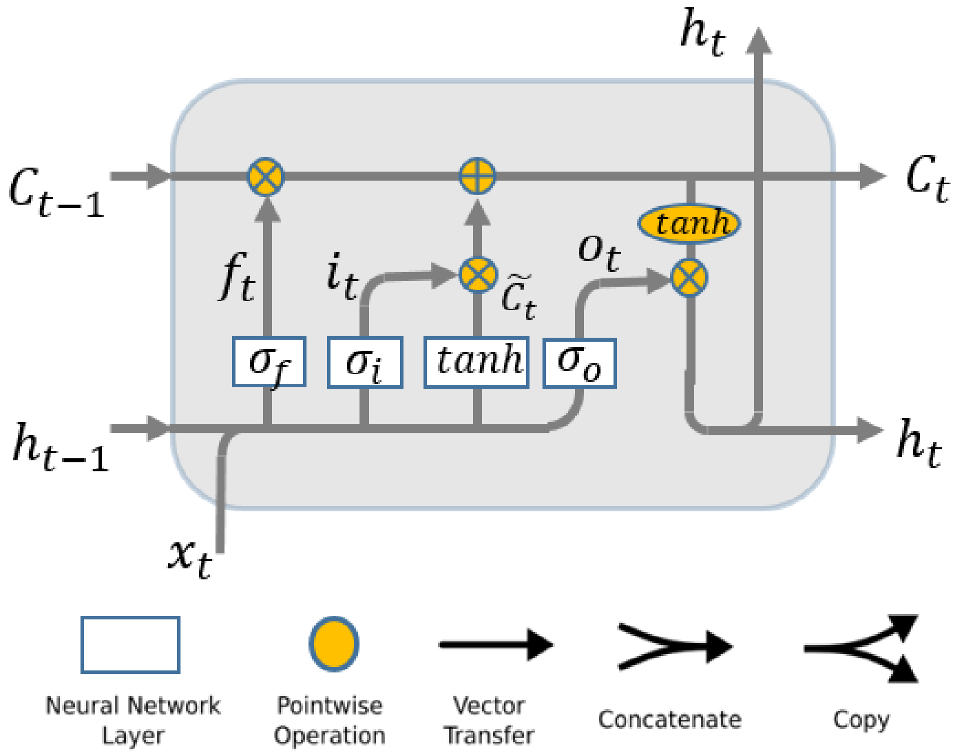
- 1.5 LSTM(Long Short Term Memory)
  - 새로운 입력 중 얼마의 량을 기억하고, 이전 기억된 숨겨진 상태(hidden state)중 얼마를 잊어야 하는 지를 제어할 수이다.
  - 인간이 정보를 처리하는 방법과 매우 흡사하다.

  - LSTM
    - a input gate and a output gate
    - Hochreiter and Schmidhuber published the paper in 1997*
  - LSTM (updated)
    - A forget gate is introduced to the LSTM
    - Felix A. Gers, J�rgen Schmidhuber and Fred Cummins 2000**

*Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.
Felix A. Gers; Jürgen Schmidhuber; Fred Cummins (2000). "Learning to Forget: Continual Prediction with LSTM". Neural Computation.* ***12 *(10): 2451–2471*

1.5 LSTM(cont.)

- LSTM(Long Short Term Memory)



$f_t = \sigma_f(W_f \cdot [h_{t-1}, x_t] + b_f)$	the forget gate
$i_t = \sigma_i[W_I \cdot [h_{t-1}, x_t] + b_i)$	the input gate
$o_t = \sigma_o(W_o \cdot [h_{t-1}, x_t] + b_o)$	the output gate
$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$	the new state(memory)
$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$	the state(memory)
$h_t = o_t * \tanh(C_t)$	the output

1.5 LSTM(cont.)

- LSTMs vs GRUs
  - LSTM이 잘 작동하지만 불필요하게 복잡하므로 GRU가 등장하게 되었다.
  - Computationally less expensive
  - Performance on par with LSTMs*

Two most widely used gated recurrent units

Gated Recurrent Unit
[Cho et al., EMNLP2014; Chung, Gulcehre, Cho, Bengio, DLUFL2014]

$$h_t = u_t \odot \tilde{h}_t + (1 - u_t) \odot h_{t-1}$$
$$\tilde{h} = \tanh(W\,[x_t] + U(r_t \odot h_{t-1}) + b)$$
$$u_t = \sigma(W_u\,[x_t] + U_u h_{t-1} + b_u)$$
$$r_t = \sigma(W_r\,[x_t] + U_r h_{t-1} + b_r)$$

Long Short-Term Memory
[Hochreiter & Schmidhuber, NC1999; Gers, Thesis2001]

$$h_t = o_t \odot \tanh(c_t)$$
$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$
$$\tilde{c}_t = \tanh(W_c\,[x_t] + U_c h_{t-1} + b_c)$$
$$o_t = \sigma(W_o\,[x_t] + U_o h_{t-1} + b_o)$$
$$i_t = \sigma(W_i\,[x_t] + U_i h_{t-1} + b_i)$$
$$f_t = \sigma(W_f\,[x_t] + U_f h_{t-1} + b_f)$$

*Chung, Junyoung, et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling." arXiv preprint arXiv:1412.3555 (2014).

2 . RNN Applications

- What can RNNs do?
  1. Language Modeling
  2. Character-level Language Modeling
  3. Google Neural Machine Translation(Google Research’s blog)
  4. Text Summarization
  5. Image Captioning

2.1 언어모델(Language Modeling)

- 2.1 언어모델(Language Modeling)
  - 언어의 문장들을 출현빈도에 비례한 확률로 기술된 모델이다.
  - 주어지는 문장이 얼마나 정확한 문장인지를 측정할 수 있다.
  - 기계번역에서 입력이 중요한 입력 (고 확률 문장은 일반적으로 정확하므로)인지를 측정할 수 있다.
  - 새로운 텍스트를 생성 할 수 있습니다

2.2 Character-level Language Modeling

- 2.2 Character-level Language Modeling
  - Shakespeare Generator, Andrej Karpathy's blog
    - PANDARUS:
      Alas, I think he shall be come approached and the day When little srain would be attain'd into being never fed, And who is but a chain and subjects of his death, I should not sleep.
    - Second Senator:
      They are away this miseries, produced upon my soul, Breaking and strongly should be buried, when I perish The earth and thoughts of many states.
    - DUKE VINCENTIO:
      Well, your wit is in the care of side and that.
    - Second Lord:
      They would be ruled after this chamber, and my fair nues begun out of the fact, to be conveyed, Whose noble souls I'll have the heart of the wars.
    - Clown:
      Come, sir, I will make did behold your worship.

2.2 Character-level Language Modeling

- Character-level Language Modeling
  - Linux Source Code Generator, Andrej Karpathy’ s blog

```
/*
* Increment the size file of the new incorrect UI_FILTER group information
* of the size generatively.
*/
static int indicate_policy(void)
{
    int error;
    if (fd == MARN_EPT) {
        /*
         * The kernel blank will coeld it to userspace.
        */
        if (ss->segment < mem_total)
            unblock_graph_and_set_blocked();
        else
            ret = 1;
         goto bail;
}
segaddr = in_SB(in.addr);
selector = seg / 16;
setup_works = true;
for (i = 0; i < blocks; i++) {
    seq = buf[i++];
   bpf = bd->bd.next + i * search;
    if (fd) {
      current = blocked;
    }
  }
  rw->name = "Getjbbregs";
  bprm_self_clearl(&iv->version);
  regs->new = blocks[(BPF_STATS << info->historidac)] | PFMR_CLOBATHINC_SECONDS << 12;
  return segtable;
}
```

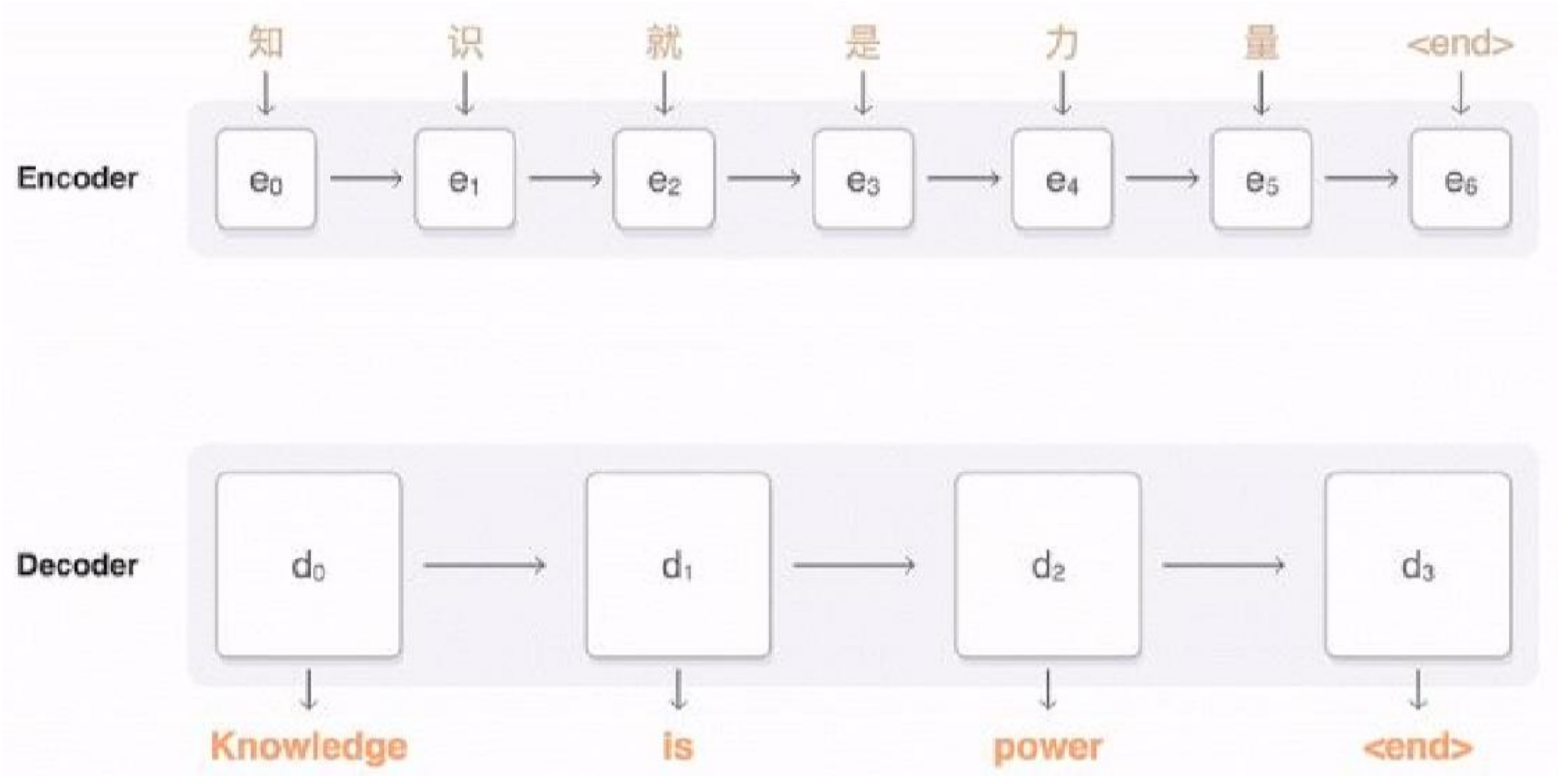
2.2 Character-level Language Modeling

- Character-level Language Modeling
  - Fake Arvix Abstracts Generator
    - Deep learning neural network architectures can be used to best developing a new architectures contros of the training and max model parametrinal Networks (RNNs) outperform deep learning algorithm is easy to out unclears and can be used to train samples on the state-of-the-art RNN more effective Lorred can be used to best developing a new architectures contros of the training and max model and state-of-the-art deep learning algorithms to a similar pooling relevants. The space of a parameter to optimized hierarchy the state-of-the-art deep learning algorithms to a simple analytical pooling relevants. The space of algorithm is easy to outions of the network are allowed at training and many dectional representations are allow develop a groppose a network by a simple model interact that training algorithms to be the activities to maximul setting, ..

We'll build this!!!!

2.3 Neural Machine Translation

- 2.3 Neural Machine Translation
  - Google Neural Machine Translation(Google Research’ s blog)



2.3 Neural Machine Translation

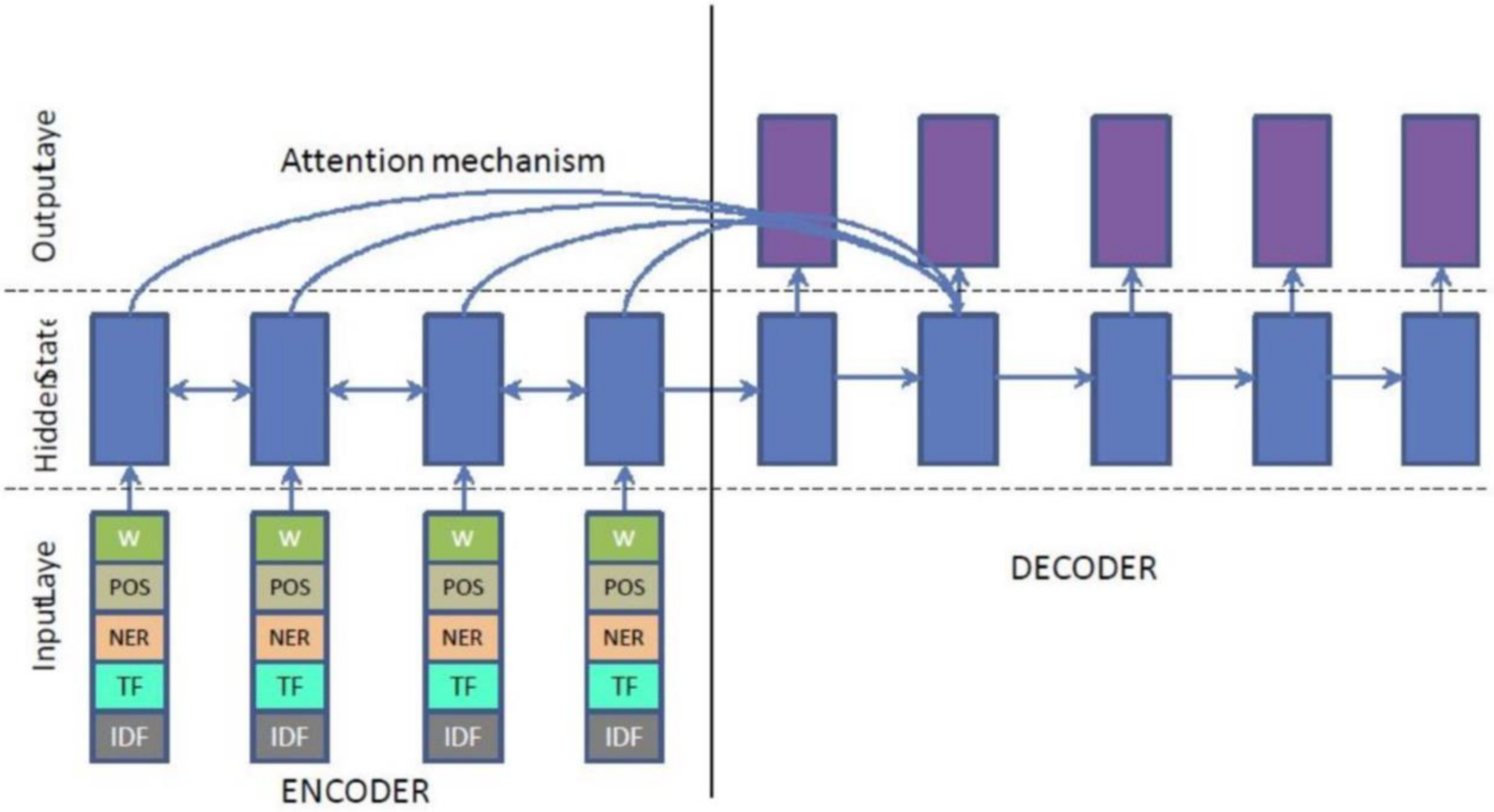
- Google Neural Machine Translation Google Research' s blog)

Input sentence:	Translation (PBMT):	Translation (GNMT):	Translation (human):
李克強此行將啟動中加總理年度對話機制，與加拿大總理杜魯多舉行兩國總理首次年度對話。	Li Keqiang premier added this line to start the annual dialogue mechanism with the Canadian Prime Minister Trudeau two prime ministers held its first annual session.	Li Keqiang will start the annual dialogue mechanism with Prime Minister Trudeau of Canada and hold the first annual dialogue between the two premiers.	Li Keqiang will initiate the annual dialogue mechanism between premiers of China and Canada during this visit, and hold the first annual dialogue with Premier Trudeau of Canada.

PBMT(Phrase Based Machine Translation)
GNMT(Google Neural Machine Translation)

2.4 Text Summarization

- 2.4 Text Summarization



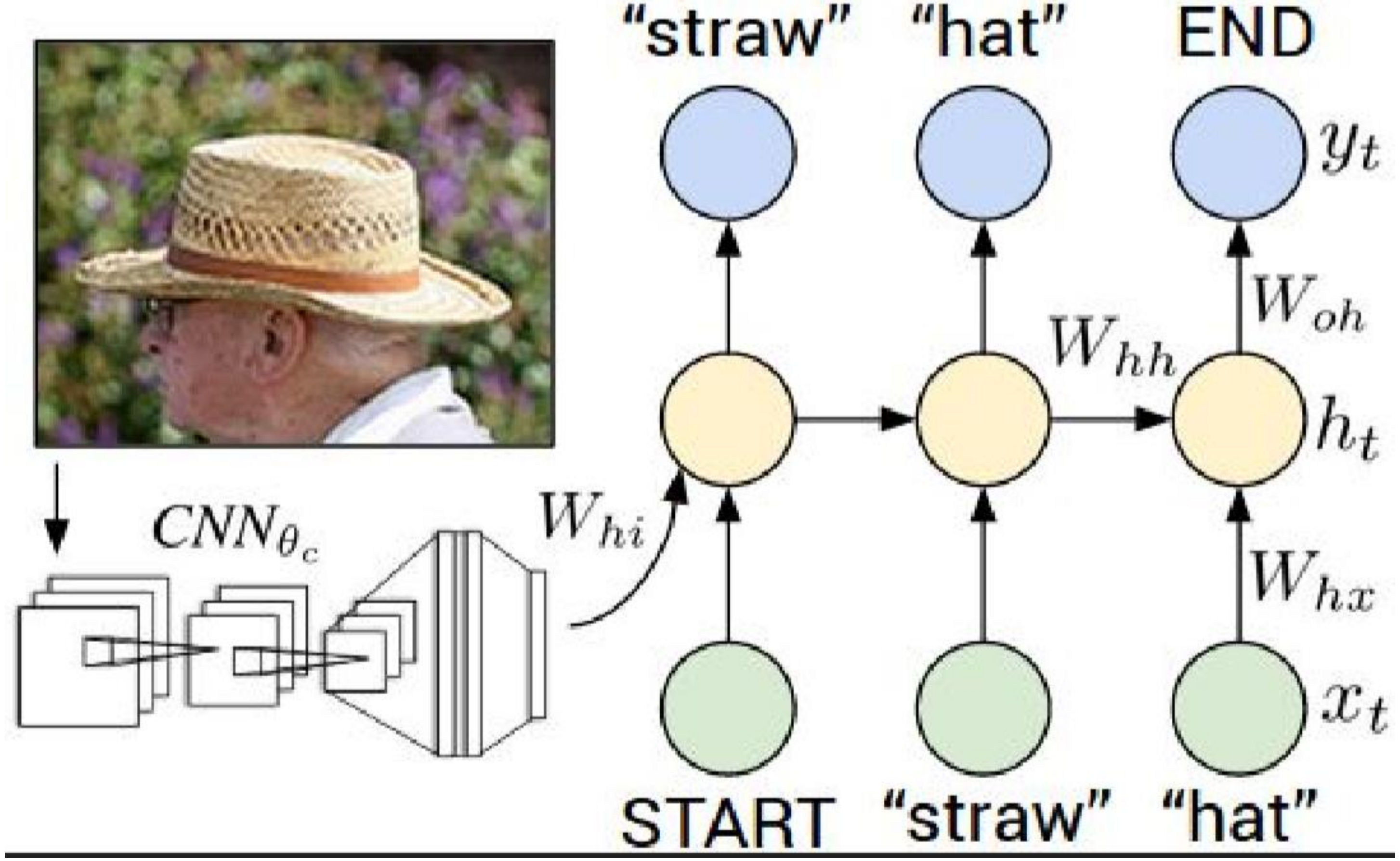
Nallapati, Ramesh, et al. "Abstractive text summarization using sequence-to-sequence rnns and beyond." arXiv preprint arXiv:1602.06023 (2016).
Text Summarization

2.4 Text Summarization

Source Document
(@entity0) wanted : film director , must be eager to shoot footage of golden lassos and invisible jets . <eos> @entity0 confirms that @entity5 is leaving the upcoming " @entity9 " movie (the hollywood reporter first broke the story) . <eos> @entity5 was announced as director of the movie in november . <eos> @entity0 obtained a statement from @entity13 that says , " given creative differences , @entity13 and @entity5 have decided not to move forward with plans to develop and direct ' @entity9 ' together . <eos> " (@entity0 and @entity13 are both owned by @entity16 . <eos>) the movie , starring @entity18 in the title role of the @entity21 princess , is still set for release on june 00 , 0000 . <eos> it 's the first theatrical movie centering around the most popular female superhero . <eos> @entity18 will appear beforehand in " @entity25 v. @entity26 : @entity27 , " due out march 00 , 0000 . <eos> in the meantime , @entity13 will need to find someone new for the director 's chair . <eos>
Ground truth Summary
@entity5 is no longer set to direct the first " @entity9 " theatrical movie <eos> @entity5 left the project over " creative differences " <eos> movie is currently set for 0000
words-lvt2k
@entity0 confirms that @entity5 is leaving the upcoming " @entity9 " movie <eos> @entity13 and @entity5 have decided not to move forward with plans to develop <eos> @entity0 confirms that @entity5 is leaving the upcoming " @entity9 " movie

Nallapati, Ramesh, et al. "Abstractive text summarization using sequence-to-sequence rnns and beyond." arXiv preprint arXiv:1602.06023 (2016).

2.5 Image Captioning



Karpathy, Andrej, and Li Fei-Fei. "Deep visual-semantic alignments for generating image descriptions." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

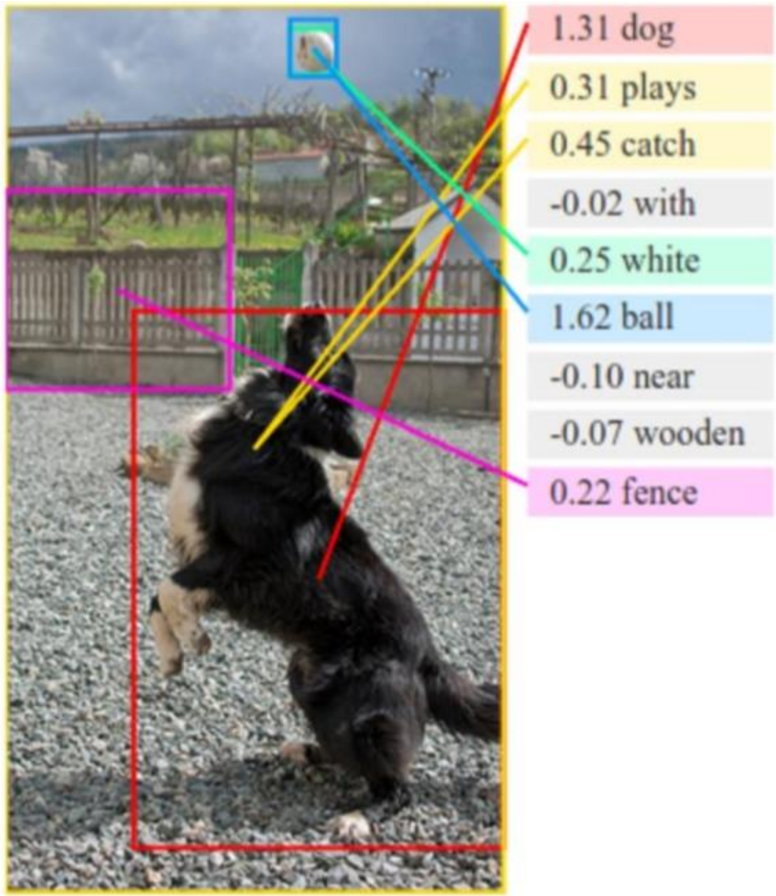
2.5 Image Captioning

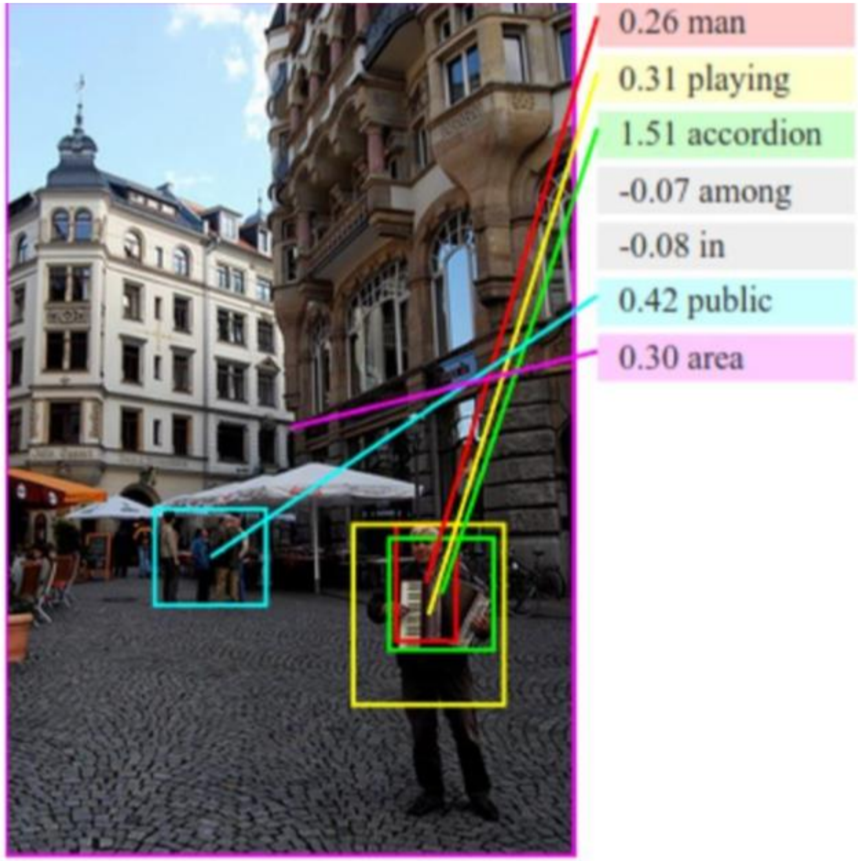
Karpathy, Andrej, and Li Fei-Fei. "Deep visual-semantic alignments for generating image descriptions." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

2.5 Image Captioning

Karpathy, Andrej, and Li Fei-Fei. "Deep visual-semantic alignments for generating image descriptions." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

2.5 Image Captioning





Karpathy, Andrej, and Li Fei-Fei. "Deep visual-semantic alignments for generating image descriptions." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

3. Language Modeling

- Neural Language Modeling
  - 텍스트로 모델을 학습하고 모델로 텍스를 생성한다.
  - Allows us to measure how likely a sentence is important input for Machine Translation (since high-probability sentences are typically correct)
  - 새로운 텍스트로 만들 수 있다.
- Language Modeling: 주요 방법
  - Word-level: n-grams
  - Character-level
  - Subword-level: somewhere in between the two above

What can be the problems?

3. Language Modeling

- Language Modeling: N-grams
  - The traditional approach up until very recently
  - Train a model to predict the next word based on previous n–grams
  - Huge vocabulary
  - Can' t generalize to OOV (out of vocabulary)
  - Requires a lot of memory
- Language Modeling: Character-level
  - Introduced in the early 2010s
  - Both input and output are characters
  - Pros:
    - Very small vocabulary
    - Doesn' t require word embeddings
    - Faster to train
  - Cons:
    - Low fluency (many words can be gibberish)

3. Language Modeling

- Language Modeling: Hybrid
  - Word-level by default, switching to character-level for unknown tokens
- Language Modeling: Subword-Level
  - Input and output are subwords
  - Keep W most frequent words
  - Keep S most frequent syllables
  - Split the rest into characters
  - Seem to perform better than both word-level and character-level models*

    new company dreamworks interactive
    new company dre+ am+ wo+ rks: in+ te+ ra+ cti+ ve:

Mikolov, Tomáš, et al. "Subword language modeling with neural networks." preprint (http://www. fit. vutbr. cz/imikolov/rnnlm/char. pdf) (2012).

3.1 Language Modeling DEMO
Character-level, Language Modeling

- Generate fake Arvix abstracts
  - Dataset: 7200 abstracts of Arvix papers about neural networks
  - "Heuristic optimisers which search for an optimal configuration of variables relative to an objective function often get stuck in local optima where the algorithm is unable to find further improvement. The standard approach to circumvent this problem involves periodically restarting the algorithm from random initial configurations when no further improvement can be found. We propose a method of partial reinitialization, whereby, in an attempt to find a better solution, only sub-sets of variables are re-initialised rather than the whole configuration. Much of the information gained from previous runs is hence retained. This leads to significant improvements in the quality of the solution found in a given time for a variety of optimisation problems in machine learning."

3.1 Language Modeling DEMO Character-level, Language Modeling

- Generate fake Arvix abstracts
  - Evaluation: no scientific way to evaluate
  - "Deep learning neural network architectures can be used to best developing a new architectures contros of the training and max model parametrinal Networks (RNNs) outperform deep learning algorithm is easy to out unclears and can be used to train samples on the state-of-the-art RNN more effective Lorred can be used to best developing a new architectures contros of the training and max model and state-of-the-art deep learning algorithms to a similar pooling relevants. The space of a parameter to optimized hierarchy the state-of-the-art deep learning algorithms to a simple analytical pooling relevants. The space of algorithm is easy to outions of the network are allowed at training and many dectional representations are allow develop a groppose a network by a simple model interact that training algorithms to be the activities to maximul setting, ..."

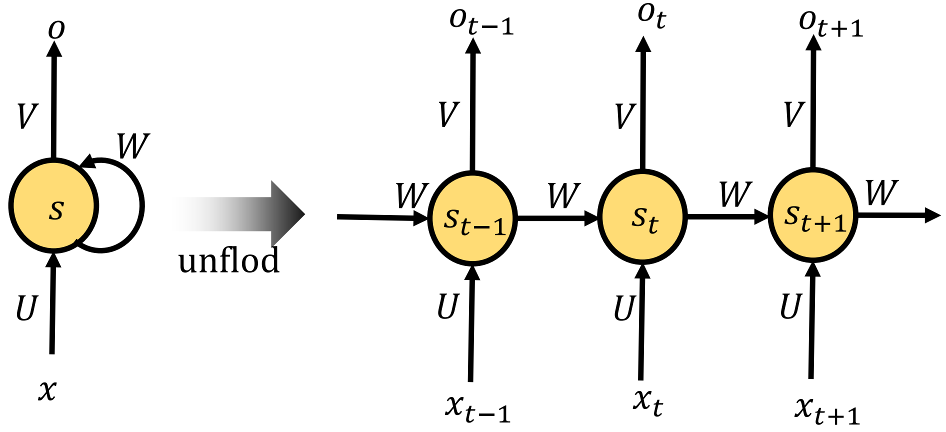
4. Examples

1. 정현파신호 샘플의 예측 모델(SimpleRNN) in keras
2. 문자 기반 신경 언어 모델(Character-Based Neural Language Model)-sixpence in keras
3. 주가예측 모델

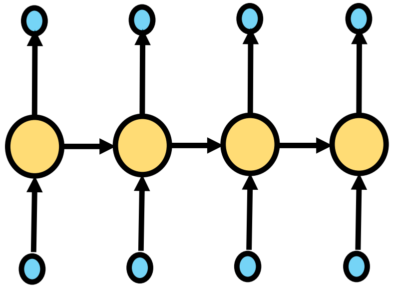
Example 1. 정현파신호 샘플의 예측 모델(SRNN) in keras

$s_t = \sigma(Ux_t + Ws_{t-1})$
$o = \sigma(Vs_t)$

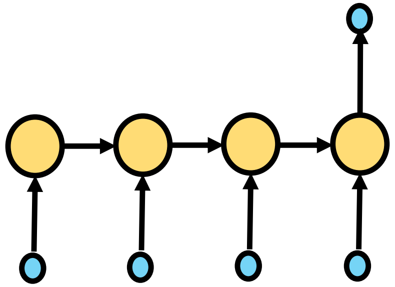
- SRNN의 순서 열 예측
  - 모델구조
    - 입력벡터 순서열 $x_0, x_1, \ldots, x_n$
    - 출력벡터 순서열 $o_0, o_1, \ldots, o_n$
    - 상태 순서열 $s_0, s_1, \ldots, s_n$
    - Target 열의 수에 따라 모델이 분류된다.



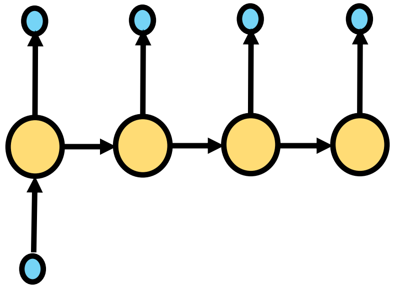
Many to Many



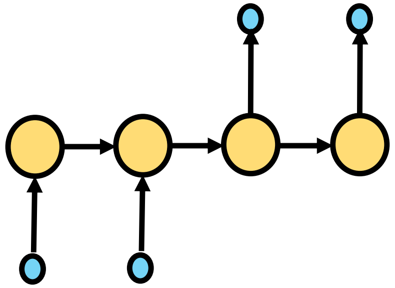
Many to One



One to Many



Many to Many

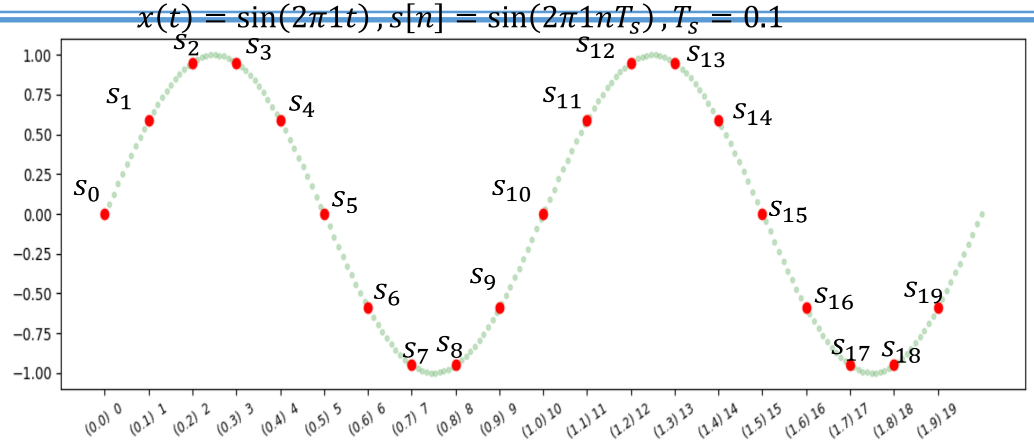


Example 1. 정현파신호 샘플 예측 모델(SRNN) (cont.)

- **Back-Propagation Through Time (BPTT)**
  - RNN은 시간에 따라 펼쳐 놓으면 구조가 MLP와 유사하기 때문에 Back-Propagation 방법으로 gradient를 계산할 수 있다. 다만 실제로 여러 개의 은닉층이 있는 것이 아니라 시간 차원에서 존재하기 때문에 Back-Propagation Through Time (BPTT) 방법이라고 한다.
- **Keras를 사용한 RNN 구현**
  - Keras 는 다양한 형태의 신경망 구조를 블럭 형태로 제공하고 있으며 SimpleRNN, LSTM, GRU 와 같은 RNN 구조도 제공한다. https://keras.io/layers/recurrent/

Example 1. 정현파신호 샘플 예측 모델(SRNN) (cont.)

- 정현파신호 샘플 값을 예측하는 모델
  - 샘플 리스트(3개)로 다음 샘플값 예측
  - 정현파신호 샘플링 $s[n] = \sin(2\pi 1 n T_s)$
  - seq_len=3으로 데이터 셋 X,Y생성




```
# (20,)
[ 0.00000000e+00  5.87785252e-01  9.51056516e-01  9.51056516e-01
5.87785252e-01  1.22464680e-16 -5.87785252e-01 -9.51056516e-01 …]
```

```
#샘플 리스트 생성
n=np.arange(0,2*10)          #[0,1,2,3,4,…,19] fs=10
fs=10; Ts=1/fs
s=np.sin(2*np.pi*1* n*Ts)    # (20,) sin signal
```

```
# X(17,3)                                       Y(17,)
[[0.        ,   0.58778525, 0.95105652]  [ 0.9510565162951536
[0.58778525, 0.95105652, 0.95105652]      0.5877852522924732
…]                                          …]
```

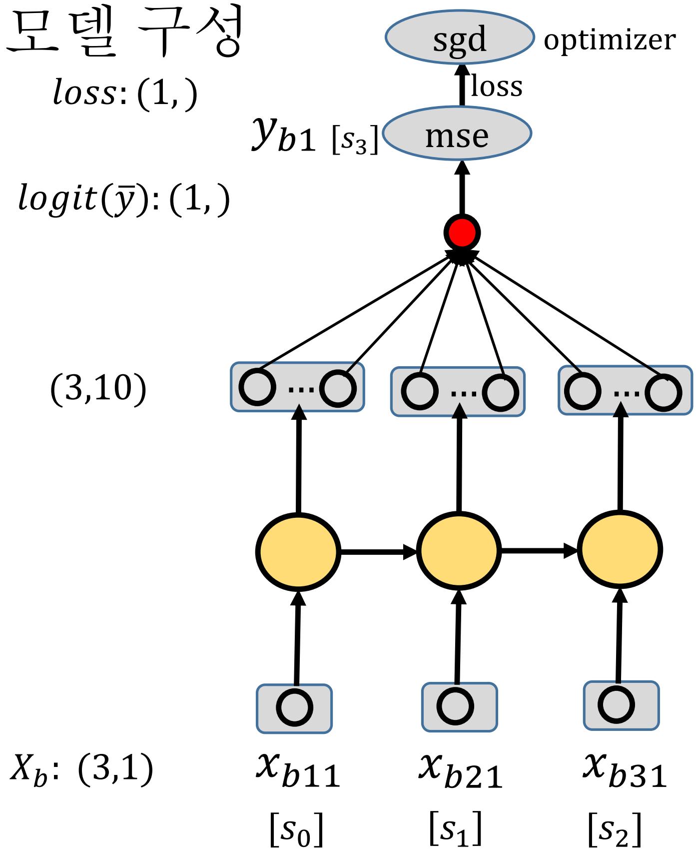
```
#학습용 X,Y 셋 생성
seq_len=3;
X=[];Y=[]
for i in range(seq_len,20,1) :
    X.append(s[i-seq_len:i])  #s[0:3]
    Y.append(s[i])            #s[3]
```

```
# X(17,3,1)                                     Y(17,1)
[[[0.] ,         [0.58778525],[0.95105652]]  [[ 0.9510565162951536]
[[0.58778525],[0.95105652],[0.95105652]]    [0.5877852522924732]
…]                                           …]
```

```
#학습용 shape으로 변환
X=np.array(X);Y=np.array(Y)      # shape X:(17,3) Y:(17,)
X=np.expand_dims(X,axis=2)       # (17,3,1)
Y=np.expand_dims(Y,axis=1)       # (17,1)

print(X.shape,Y.shape)           #(17,3,1),(17,)
```

Example 1. 정현파신호 샘플 예측 모델(SRNN) (cont.)

- 모델 구성



Linear regression layer

Dense(1, activation="linear"))

SRNN many-to-many model layer

SimpleRNN(10, input_shape=(3, 1))

$X = [[[s_0],[s_1],[s_2]]$ $[[s_1],[s_2],[s_3]]$ … $[[x_{16}],[s_{17}],[s_{18}]]$ $]$ (17,3,1)

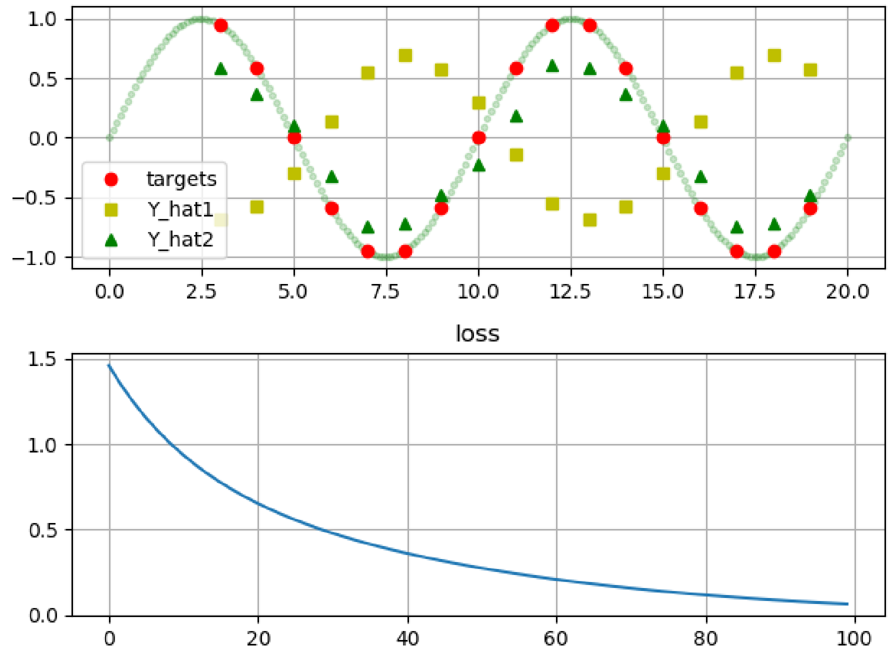
$Y = [[s_3],$ $[s_4],$. $[s_{19}]$ $]$ (17,1)

```
#SRNN 모델 설정
np.random.seed(0)
model = Sequential()
model.add(SimpleRNN(10, input_shape=(3, 1)))
model.add(Dense(1, activation="linear"))
model.compile(loss='mse', optimizer='sgd')
```

Example 1. 정현파신호 샘플 예측 모델(SRNN) (cont.)

```
import numpy as np
import matplotlib.pyplot as plt
from keras.models import Sequential
from keras.layers import SimpleRNN, Dense

#dataset X,Y 생성
n=np.arange(15)                         #[0,1,2,3,4,…,19]
s=np.sin(2*np.pi*0.125* n)              #sin signal
nb_timestep=3;
X=[];Y=[]
for i in range(0,s.shape[0]-1-nb_timestep,1) :
X.append(s[i:i+nb_timestep])
Y.append(s[i+nb_timestep])
X=np.array(X);Y=np.array(Y)             # X.shape:(17,3) Y.shape:(11,)
X=np.expand_dims(X,axis=2)              # X.shape:(17,3,1)
print(X.shape,Y.shape)                  #(17,3,1),(17,1)

#SRNN 모델 설정
np.random.seed(0)
model = Sequential()
model.add(SimpleRNN(10, input_shape=(3, 1)))
model.add(Dense(1, activation="linear"))
model.compile(loss='mse', optimizer='sgd')

Y_hat1=model.predict(X)     #randomly initialized Y

hist=model.fit(X,Y,epochs=100,verbose=0) #training

Y_hat2=model.predict(X)     #predicted after training
```

```
plt.subplot(211)                        #define subplot(211)
plt.plot(Y,     'ro-',label='targets’)  #target Y, s[nb_timestep:]
plt.plot(Y_hat1,'bs-',label='Y_hat1')#initialized Y
plt.plot(Y_hat2,'gx-',label='Y_hat2')#predicted Y
plt.grid()                              #grid on figure subplot
plt.legend()

plt.subplot(212)                        #define subplot(212)
plt.plot(hist.history['loss’])          #plot loss
plt.title('loss')
plt.grid()
plt.show()
```

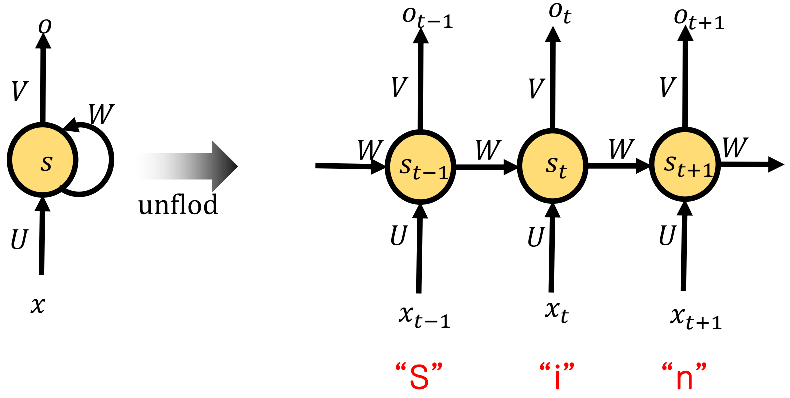


Example 2. 문자 기반 신경 언어 모델-sixpence in keras

- 언어 모델(Language Model)은
  - 시퀀스에서 앞에 오는 특정 단어를 기반으로 시퀀스에서 다음 단어를 예측합니다.
- 문자 기반 신경 언어 모델(Character-Based Neural Language Model)
    - 문자 기반 언어 모델의 장점은 단어, 문장 부호 및 기타 문서 구조를 처리 할 때 **작은 어휘와 유연성**입니다. 훈련 속도가 느린 대형 모델이 필요합니다.
  - 절차
    - 데이터 셋 생성
      - 문자 기반 언어 모델링을 위한 텍스트를 준비하는 방법.
    - 모델 생성
      - LSTM을 사용하여 문자 기반 언어 모델을 개발하는 방법.
    - 모델 평가
      - 훈련 된 문자 기반 언어 모델을 사용하여 텍스트를 생성하는 방법.

Example 2. 문자 기반 신경 언어 모델-sixpence in keras(cont.)

How to Develop a Character-Based Neural Language Model in Keras
Photo by hedera.baltica, some rights reserved.



- 동요

Sing a song of sixpence, a pocket full of rye,
Four and twenty blackbirds baked in a pie.
When the pie was opened the birds began to sing,
Oh wasn't that a dainty dish to set before the king?

The king was in his counting house counting out his money,
The queen was in the parlour eating bread and honey
The maid was in the garden hanging out the clothes,
When down came a blackbird and pecked off her nose!

6펜스 노래를 부르자, 주머니 가득 호밀이 있지.
찌르레기 24마리는 파이 안에서 구워 졌네.
파이를 잘랐을 때 새들은 노래를 부르기 시작했지
오, 저건 정말 왕에게 드릴만한 진미가 아닌가?

왕은 금고에서 돈을 세고 있었고,
왕비는 거실에서 꿀을 바른 빵을 먹고 있었네.
하녀는 정원에서 빨래를 널고 있는데
찌르레기 한 마리가 날아와 선 하녀의 코를 쪼았지.

Example 2. 문자 기반 신경 언어 모델-sixpence in keras(cont.)

- 데이터 셋 X,y생성

```
from pickle import load
from keras.models import load_model
from keras.utils import to_categorical
from keras.preprocessing.sequence import pad_sequences
```

```
#text
Sing a song of sixpence, a pocket full of rye,
Four and twenty blackbirds baked in a pie.
...
```

```
with  open('data/rnn_sixpence_data.txt','r') as f:
        text= f.read()
```

```
#cleaned Text
Sing a song of sixpence, a pocket full of rye, Four and twenty blackbirds ...
```

```
tokens = raw_text.split()              #strip all of the new line characters
cleaned_text = '  '.join(tokens)        #separated only by white space
```

```
#sequences with length 11:(399,11)
'Sing a song',
'ing a song ',
...
```

```
length = 10;        sequences = list()
for i in range(length, len(cleaned_text)):
        seq = raw_text[i-length:i+1]  #[0:11]# select sequence of tokens
        sequences.append(seq)         # store
```

```
# vocabulary : sorted char set in cleaned text  (34,)
[' ', '!', '"', ',', '.', '?', 'F', 'O', 'S', 'T', 'W', 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'q', 'r', 's', 't', 'u', 'w', 'x', 'y']
{' ': 0, '!': 1, '"': 2, ',': 3, '.': 4, '?': 5, 'F': 6, 'O': 7, 'S': 8, 'T': 9, 'W': 10, 'a': 11,…, 'x': 32, 'y': 33}
```

```
chars = sorted(list(set(raw_text)))
mapping = dict((c, i) for i, c in enumerate(chars))
```

```
# encoded sequences (399,11)
[8, 19, 23, 17, 0, 11, 0, 28, 24, 23, 17],        #Sing a song
[19, 23, 17, 0, 11, 0, 28, 24, 23, 17, 0],        #ing a song
```

```
encoded_sequences = list()
for sequence in sequences:
        encoded_seq = [mapping[char] for char in sequence]
        encoded_sequences.append(encoded_seq)      #(399,11)
```

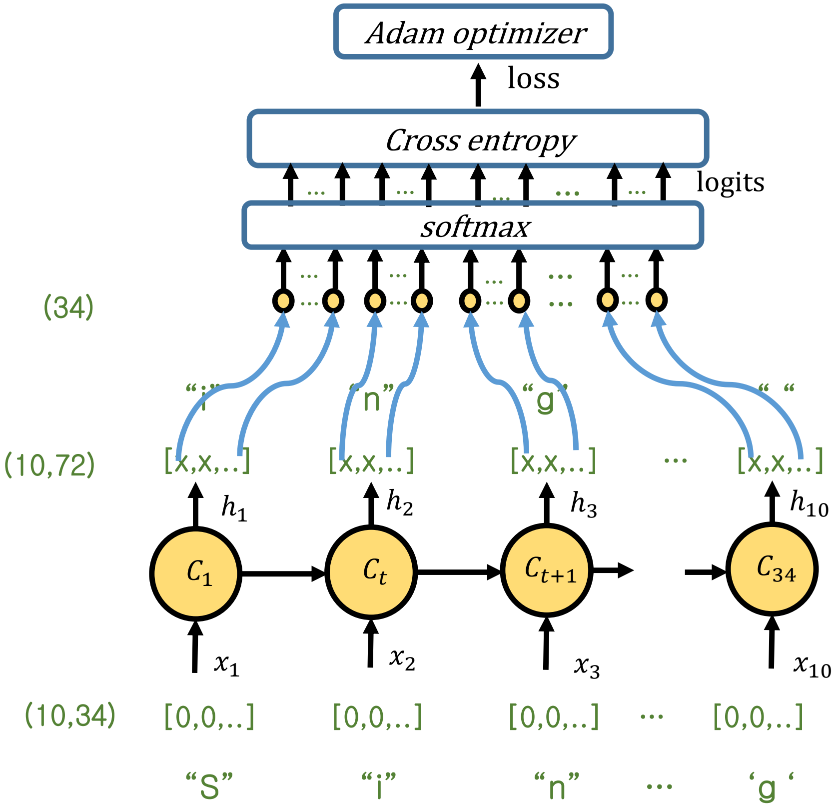
```
X: [8, 19, 23, 17, 0, 11, 0, 28, 24, 23],        y:[17,        #Sing a son   g
   [19, 23, 17, 0, 11, 0, 28, 24, 23, 17],          0,         #ing a song   ' '
X=ohe(X); y=ohe(y)
X : [[0,0,0,0,0,0,1,0,0,…],…
```

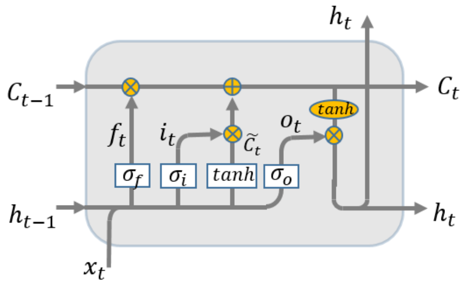
```
vocab_size=len(chars)                               #34
sequence=np.array(encoded_sequences)                #(399,11)
X,y=sequence[:,:-1],sequence[:,-1]                  #(399,10),(399,)
sequences = [to_categorical(x, num_classes=vocab_size) for x in X]
X = np.array(sequences)                             #(399,10,34)
y = to_categorical(y, num_classes=vocab_size)#(399,34)
```

Example 2. 문자 기반 신경 언어 모델-sixpence in keras(cont.)

- Model 생성

```
# define model
model = Sequential()
model.add(LSTM(75, input_shape=(X.shape[1], X.shape[2])))    #(10,34)
model.add(Dense(vocab_size, activation='softmax’))            #(34,)
print(model.summary())

# set compile method
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy’])

# fit model
model.fit(X, y, epochs=100, verbose=2)

# save the model to file
model.save('model.h5’)

# save the mapping
dump(mapping, open('mapping.pkl', 'wb'))
```






```
X: [8, 19, 23, 17, 0, 11, 0, 28, 24, 23],      y:[17,#Sing a son   g
   [19, 23, 17, 0, 11, 0, 28, 24, 23, 17],        0,   #ing a song   ‘ ‘
X=ohe(X); y=ohe(y)
X : [[0,0,0,0,0,0,1,0,0,…[…]]                  #[8, 19, 23, 17, 0, 11, 0, 28, 24, 23],
    [ [[0,0,0,0,0,0,0,0,0,…[…]]                #[19, 23, 17, 0, 11, 0, 28, 24, 23, 17],
```

Example 2. 문자 기반 신경 언어 모델-sixpence i

```
print(predict_one(''Sing a son'))    #g
print(predict_one('ing a song'))    #' '
print(predict_one('ing a song '))    #o
print(predict_one('ng a song o'))    #f
```

- 모델 평가
    - 모델을 load하고
    - 모델에게 seedtext를 제시하고
    - text를 생성하게 한다.
- 실험
    - 학습된 시작부분 텍스트를 제시하고 텍스트(연속문자열) 생성시도
    - 학습된 중간 부분 텍스트를 제시하고 텍스트 생성 시도
    - 학습된 않은 텍스트를 제시하고 텍스트를 생성시도

```
def generate_text(model, mapping, seq_length=10, seed_text='Sing a son', n_chars=20) :
        # mapping                                          {' ':0,...}
        mapping_i2c={v:k for k,v in mapping.items()}#{0:' ',...}
        nb_vocab=len(mapping)                              #34

        def predict_one(text='Sing a son') :
                #'Sing a son'=>'g'                                  S  i  n  g   a   s  o  n
                encoded_text = [mapping[char] for char in text]     #[8, 19, 23, 17, 0, 11, 0, 28, 24, 23]    (10,)
                encoded_text= [encoded_text]                        #[[8, 19, 23, 17, 0, 11, 0, 28, 24, 23]]  (1,10)
                encoded_ohe = to_categorical(encoded_text,          #[[[0,0,0,0,0,0,0,1,0,..],..[…]]          (1,10,34)
                            num_classes=nb_vocab)
                yhat = model.predict_classes(encoded_ohe, verbose=0)#[18]  (1,)
                ychar=''.join([mapping_i2c[c] for c in yhat])       #[18]=>[i2c[18]]=['g']=>'g'
                return ychar
        text=seed_text
        for _ in range(n_chars) :                          # 20개 문자 예측
                char=predict_one(text[-seq_length:])       #뒤 부터 10개의 문자열
                text = text+char                           #예측된 문자 추가
        return text

model = load_model('model.h5')                   # load the model
mapping = load(open('mapping.pkl', 'rb'))        # load the mapping {' ':0,…}
# test start of rhyme
print(generate_text(model, mapping, 10, 'Sing a son', 20))    #Sing a song of sixpence, a poc
# test mid-line
print(generate_seq(model, mapping, 10, 'king was i', 20))     #king was in his counting house
# test not in original
print(generate_seq(model, mapping, 10, 'hello worl', 20)      #hello worl,, The aeen was in t
```

Example 3. 주가예측 모델

- 내용
  - 데이터를 읽고 분석합니다. (Pandas)
    - df_ge = pandas.read_csv(os.path.join(os.getcwd(), "data\\us.ge.txt"), engine='python’)
      plot(df_ge)
  - 데이터 셋 생성
    - stock=df_ge.loc[:,["Open","High","Low","Close","Volume"]].values
      stock__rain,stock_val=sklearn.model_selection.train_test_split(stock, train_size=0.8, test_size=0.2, shuffle=False)
      min_max_scaler=sklearn.preprocessing.MinMaxScaler
      stock_train = min_max_scaler.fit_transform(stock_train) #(11246,5)
      stock_val == min_max_scaler.fit_transform(stock_val) #(2812,5)
  - 시계열 및 지도 학습 문제로 데이터 변환
    - X−train, y_train, #(11242,3,5),(11242,)
      X_val, y_val #(2808,3,5), (2808,)
  - 모델 만들기 (Keras)
    - model = {LSTM(32),Dense(64),Dense(32),Dense(1) }
      model.fit(…)
  - 결과 훈련, 예측 및 시각화.
    - y_hat=Model.predict(X_val)
    - score = model.evaluate(X_val, y_val)

https://towardsdatascience.com/predicting-stock-price-with-lstm-13af86a74944

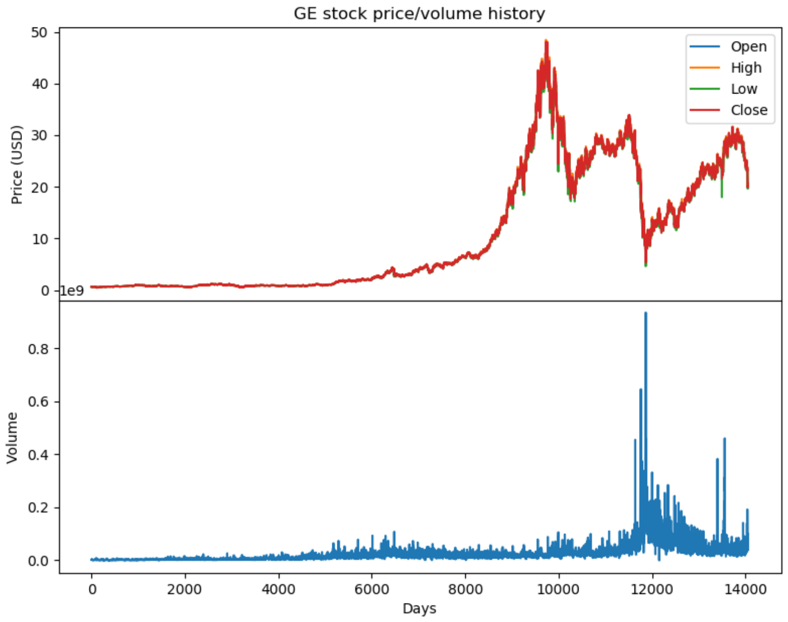
Example 3. 주가예측 모델 (cont.)

- 주가 로딩 및 분석
  - General Electronic ‘corp. us.ge.txt’

```
import pandas as pd
df_ge = pd.read_csv(os.path.join(os.getcwd(), "data\\us.ge.txt"), engine='python')
print(df_ge))   #(14058,7)
```

```
       Date    Open    High    Low    Close    Volume  OpenInt
0     1962-01-02  0.6277  0.6362  0.6201  0.6201   2575579       0
1     1962-01-03  0.6201  0.6201  0.6122  0.6201   1764749       0
...        ...      ...     ...     ...     ...        ...      ...
14056  2017-11-09  20.0400  20.0710  19.8500  19.9900   50831779       0
14057  2017-11-10  19.9800  20.6800  19.9000  20.4900  100698474       0

[14058 rows x 7 columns]
```

```
def plot_stock() :
        fig,axes=plt.subplots(2,1,sharex=True)
        axes[0].plot(df_ge["Open"],label='Open’)
        axes[0].plot(df_ge["High"],label='High’)
        axes[0].plot(df_ge["Low"],label='Low’)
        axes[0].plot(df_ge["Close"],label='Close’)
        axes[0].set_ylabel('Price (USD)’)
        axes[0].set_xlabel('Days’)
        axes[0].legend()
        axes[1].plot(df_ge["Volume"])
        axes[1].set_ylabel('Volume’)
        axes[1].set_xlabel('Days’)
        plt.subplots_adjust(hspace=0)
        axes[0].title.set_text('GE stock price/volume history’)
        plt.show()
plot_stock()
```




```
print("checking if any null values are present\n", df_ge.isna().sum())
```

```
checking if any null values are present
 Date       0
Open       0
High       0
Low        0
Close      0
Volume     0
OpenInt    0
dtype: int64
```

Example 3. 주가예측 모델 (cont.)

- 데이터 셋 생성

```
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split

train_cols = ["Open","High","Low","Close","Volume"]
stock=df_ge.loc[:,train_cols].values                    #np.array로 변환 (14058,7)=>(14058,5)

stock_train, stock_test = train_test_split(stock,       #훈련 및 검증용 셋으로 분리
        train_size=0.8, test_size=0.2, shuffle=False)   #(14058,5)=>(11246,5),(2812,5)

# scale the feature MinMax
min_max_scaler = MinMaxScaler()                         #(0,1)의 수로 변환
stock_train = min_max_scaler.fit_transform(stock_train) #(11246,5)
stock_val   = min_max_scaler.transform(stock_test)      #(2812,5)

#Converting data to time-series and supervised learning problem
TIME_STEPS=3
BATCH_SIZE=128
def split_xy(stock,TIME_STEPS,predict_price=3) :
        X=[];y=[]
        for i in range(0,stock.shape[0]-TIME_STEPS-1,1) :
        X.append(stock[i:i+TIME_STEPS])                 #["Open","High","Low","Close","Volume"]
        y.append(stock[i+TIME_STEPS,predict_price])     #"Close"
        return np.array(X),np.array(y)                  # nparray로 변환

X_train,y_train=split_xy(stock_train,TIME_STEPS)        #(11242,3,5),(11242,)
X_val,y_val=split_xy(stock_val,TIME_STEPS)              #(2808,3,5),  (2808,)
```

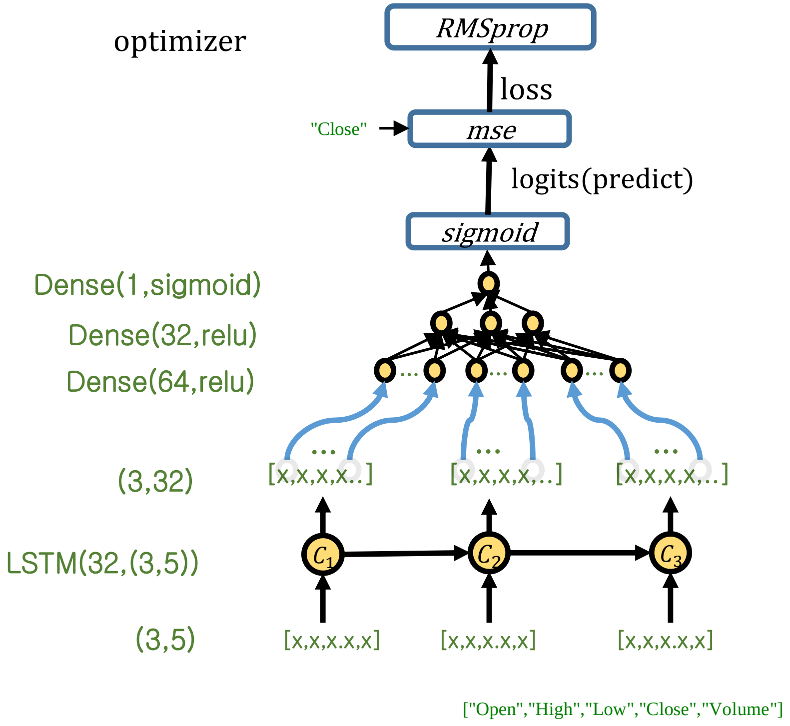
```
         Date    Open    High     Low   Close   Volume  OpenInt
0  1962-01-02  0.6277  0.6362  0.6201  0.6201  2575579        0
1  1962-01-03  0.6201  0.6201  0.6122  0.6201  1764749        0
2  1962-01-04  0.6201  0.6201  0.6037  0.6122  2194010        0
3  1962-01-05  0.6122  0.6122  0.5798  0.5957  3255244        0
```

```
[[0.00356678 0.00352766 0.00358385 0.00338425 0.02108267]
 [0.00340607 0.00319219 0.00341628 0.00338425 0.01365459]
 [0.00340607 0.00319219 0.00323598 0.00321827 0.01758709]]
```

Example 3. 주가예측 모델 (cont.)

- 모델 생성 및 학습

```python
#Creating model
model = Sequential()
model.add(LSTM(32, input_shape=(X_train.shape[1],X_train.shape[2])))
model.add(Dropout(0.2))
model.add(Dense(64,activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(32,activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(1),activation='sigmoid’))

#set train-methods
optimizer = keras.optimizers.RMSprop(lr=0.001)
model.compile(loss='mean_squared_error', optimizer=optimizer)

#set logging file(.log)
csv_logger = keras.callbacks.CSVLogger(
        os.path.join(os.getcwd(), 'logs/stock_model.log'), append=True)

# train model
history = model.fit(X_train,y_train,              #train data
            epochs=300, batch_size=1204,             #
            verbose=2,                            # set ouput amount
            shuffle=False,                        #no shuffle
            validation_data=(X_val,y_val),        #evaluation data
            callbacks=[csv_logger])               #set logging callback
```



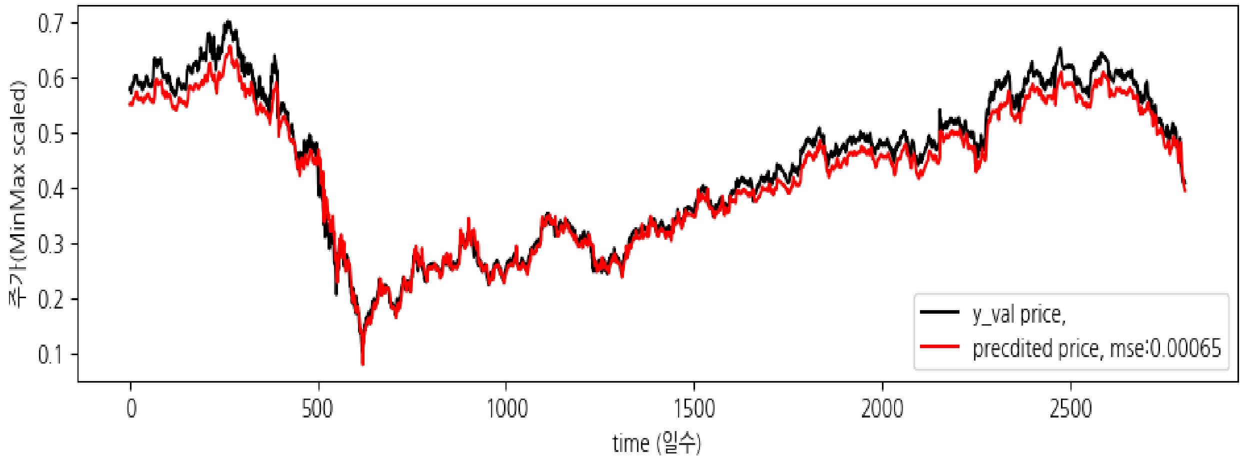

```
Epoch 293/300
 - 0s - loss: 5.1985e-04 - val_loss: 0.0010
Epoch 294/300
 - 0s - loss: 7.1853e-04 - val_loss: 0.0013
Epoch 295/300
 - 0s - loss: 5.7094e-04 - val_loss: 4.2705e-04
Epoch 296/300
 - 0s - loss: 5.1088e-04 - val_loss: 3.8838e-04
Epoch 297/300
 - 0s - loss: 5.0382e-04 - val_loss: 6.6422e-04
Epoch 298/300
 - 0s - loss: 5.9464e-04 - val_loss: 0.0014
Epoch 299/300
 - 0s - loss: 5.9211e-04 - val_loss: 7.3366e-04
Epoch 300/300
 - 0s - loss: 5.3272e-04 - val_loss: 6.5282e-04
```

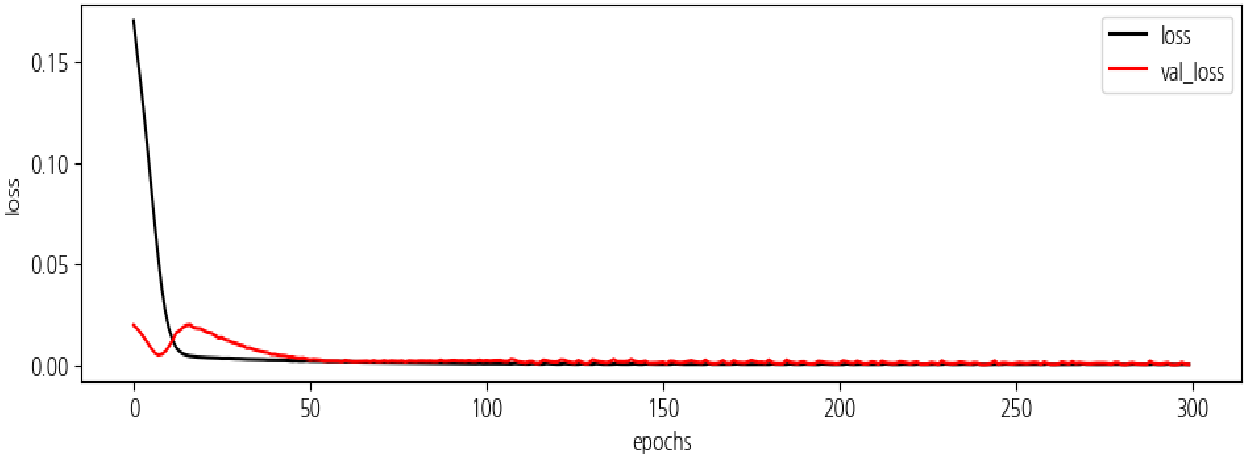
Example 4-3. 주가예측 모델 (cont.)

- 모델 검증

```
y_hat=model.predict(X_val)          #(2808,3,5)=>(2808,1)
y_hat1=np.reshape(y_hat,(-1))       #((2808,)

score_tr=model.evaluate(X_train,y_train) #(11242,3,5),(11242,)=>0.00047
score=model.evaluate(X_val,y_val)        #(2808,3,5), (2808,)     =>0.00105

plt.subplot(211)
plt.plot(y_val,  'k',label='real price’)
plt.plot(y_hat1,’r',label='precdited price, mse:{}'.format(score))
plt.legend()

plt.subplot(212)
plt.plot(history.history['loss’],     'k’, label='loss')
plt.plot(history.history['val_loss'],'r’, label='val_loss')
plt.legend()
plt.show()
```

```
evaluation score (x_train,y_train ) :  0.00030123115643539520
evaluation score (x_val,  y_val)    :  0.0006528214721985564
```





The End