Deep Learning
Deep Neural Network

Yoon Joong Kim,
Hanbat National University

Deep Learning

RNN(2)
Recurrent Neural Network

RNN(1) : RNN,RNN Applications, Language model, RNN models

RNN(2) : (1)정현파신호 샘플의 예측 모델(SimpleRNN) in keras
(2)문자 기반 신경 언어 모델(many-to-one RNN)-sixpence in keras

RNN(3) : (1)문자 기반 신경 언어 모델(many-to-many RNN)-sixpence in keras
(2) 주가예측 모델

Yoonjoong Kim

Department of Computer Engineering, Hanbat National University

yjkim@hanbat.ac.kr

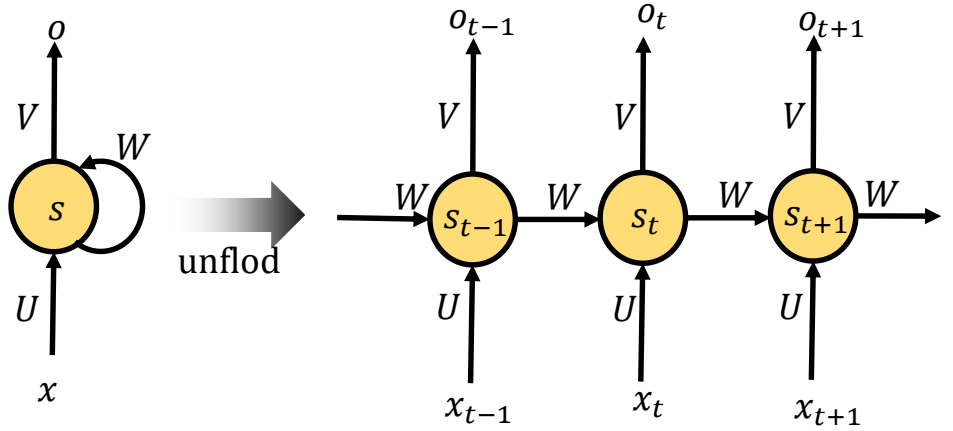
Agenda

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. RNN models
5. Examples in Keras
    1. 정현파신호 샘플의 예측 모델(SimpleRNN) in keras
    2. 문자 기반 신경 언어 모델(many-to-one RNN)-sixpence in keras
    3. 문자 기반 신경 언어 모델(many-to-many RNN)-sixpence in keras
    4. 주가예측 모델

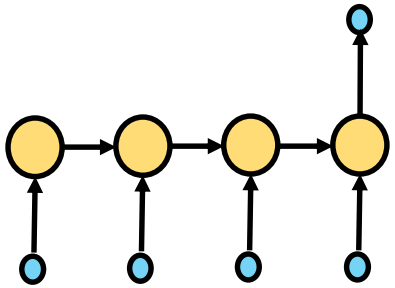
Example ‘1. 정현파신호 샘플의 예측 모델(SRNN) in keras

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

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



Many to One



감정분류
Sequence of words=>감정

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

- Data set 생성
  - 정현파신호로부터 20개 샘플을 구성
  - 정현파신호 샘플링

    x(t)=Asin($2\pi ft$)

    A:크기, f:주파수

  - 이산 샘플데이터 셋
    - 샘플링 주파수 f_s

      f_s: samples /sec, T_s

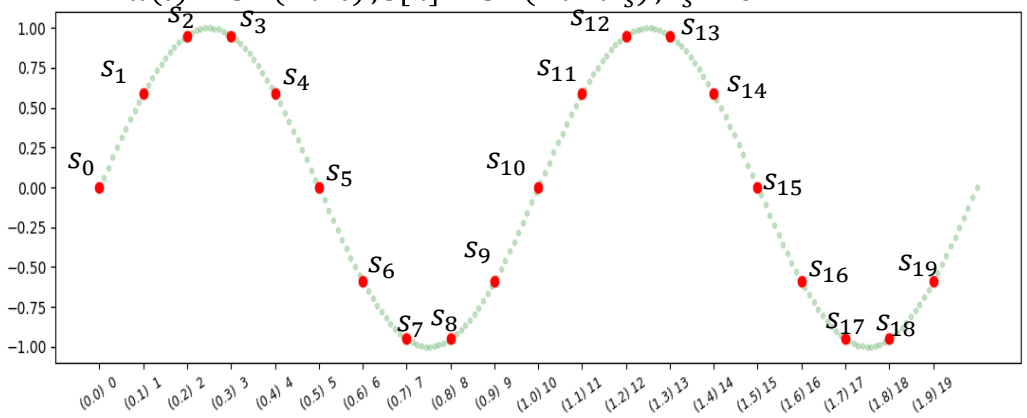
    - $s[n] = x(nT_s)$

      $= 10\sin(2\pi 1nT_s)$ $f_s: sampels/sec$

    - $A = 10, f = 1,\ f_s = 10, T_s = \frac{1}{10}$

      $s[n] = \sin(2\pi fnT_s)$

$x(t) = \sin(2\pi 1t), s[n] = \sin(2\pi 1nT_s), T_s = 0.1$



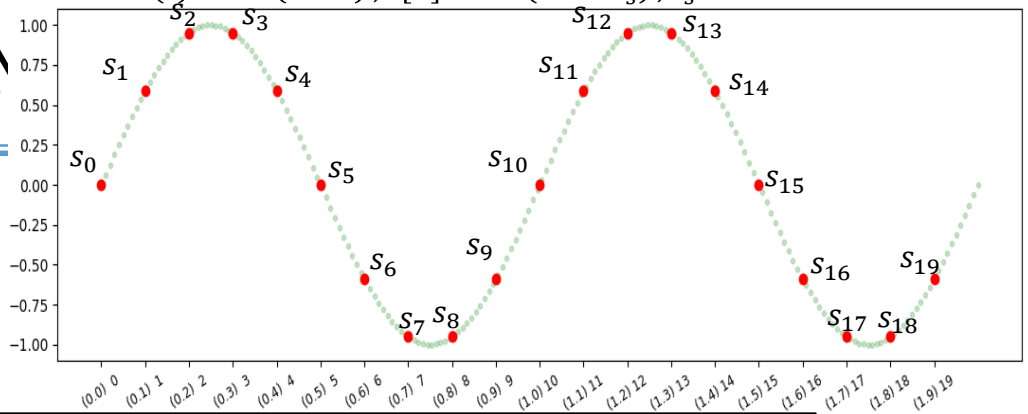

```
#샘플 리스트 생성
n=np.arange(0,2*10)          #2초간 20 vector,[0,1,2,3,4,…,19] fs=10
fs=10; Ts=1/fs
s=np.sin(2*np.pi*1* n*Ts)    # (20,)  samples
```

```
#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]
```

Example 1. 정현파신호 샘플 예측 모델(SRNN

- Data set 구성

$x(t) = \sin(2\pi 1t), s[n] = \sin(2\pi 1nT_s), T_s = 0.1$




```
# (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)          #2초간 20 sample,[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
```

$S = [s_0\ s_1\ s_2 \ldots s_{19}]$ (20,)

```
# 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]
```

$X = [[s_0\ s_1\ s_2]\ [s_1\ s_2\ s_3]\ \ldots\ [s_{16}\ s_{17}\ s_{18}]]$ (17,3)

$Y = [s_3\ s_4\ \ldots\ s_{19}]$ (17,)

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

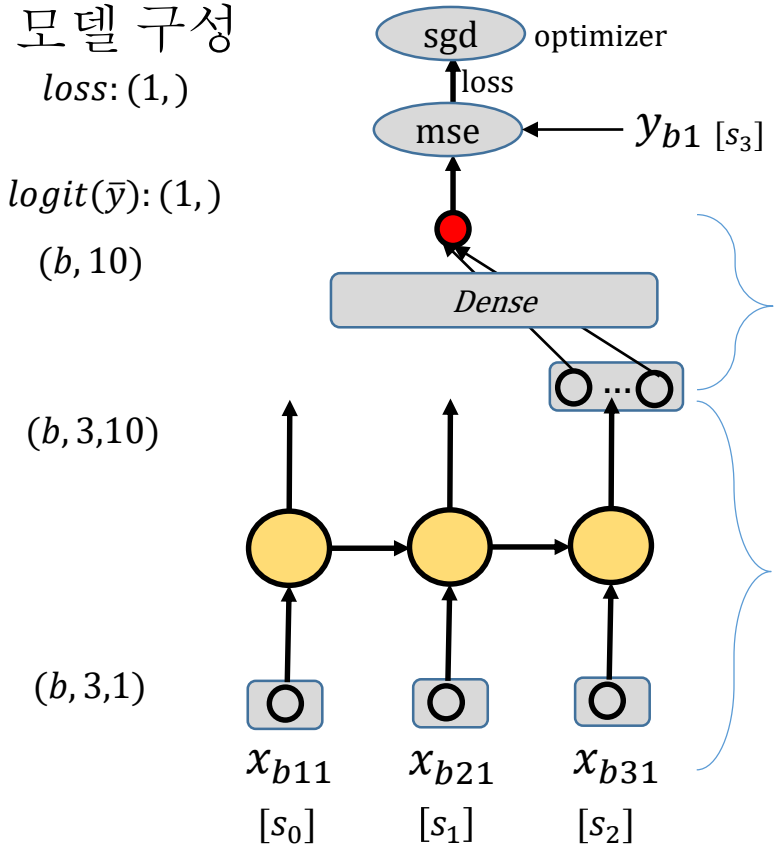
```
#학습용 shape으로 변환
X=np.array(X);Y=np.array(Y)      # shape X:(17,3) Y:(1
X=np.expand_dims(X,axis=2)       # (17,3,1)
Y=np.expand_dims(Y,axis=1)       # (17,1)

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

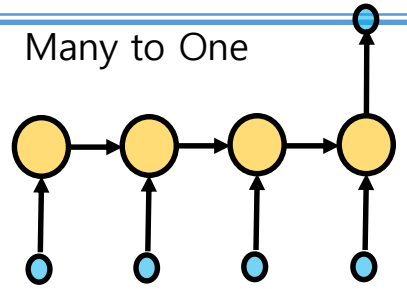
$X = [[[s_0], [s_1], [s_2]]\ [[s_1], [s_2], [s_3]]\ \ldots\ [[x_{16}], [s_{17}], [s_{18}]]]$ (17,3,1)

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

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)

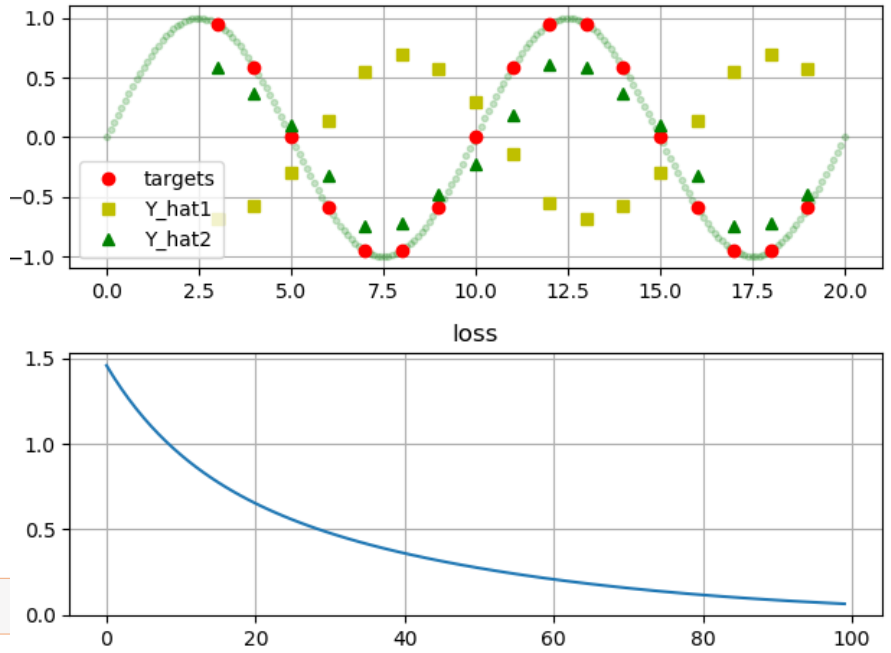
$Y = [[s_3],$ $[s_4],$. $[s_{19}]$] (17,1)

```
#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.)

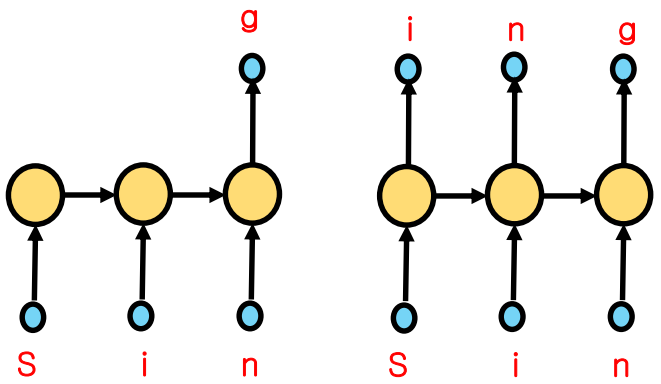
```
import numpy as np
import matplotlib.pyplot as plt
from keras.models import Sequential
from keras.layers import SimpleRNN, Dense

#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 모델은 문자 시퀀스로 다음에 나오는 문자를 학습하고 예측할 수 있다.
  - 개발 절차
    - 데이터 셋 생성
      - 문자 기반 언어 모델링을 위한 텍스트를 준비하는 방법.
    - 모델 생성 및 학습
      - LSTM을 사용하여 문자 기반 언어 모델을 개발하는 방법.
    - 모델 평가
      - 훈련 된 문자 기반 언어 모델을 사용하여 텍스트를 생성하는 방법.
  - Many-to-one RNN model
  - Many-to-many RNN model



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.

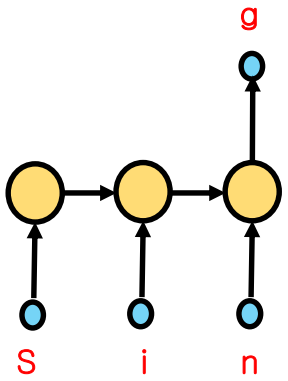
Sing a song of sixpence, a pocket full of rye,
Four and twenty blackbirds baked in a pie.
When the pie was opened the birds began to sing,
Oh wasn't that a dainty dish to set before the king?

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.

- 문제
  - 영국 동요 sixpence 로
    문자기반언어모델을 학습시켜서
    말하는 모델 만들기
  - Many to one rnn으로



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

- Dataset X_ohe, y_ohe 생성

text: Sing a song of sixpence, a pocket full of rye, Four and twen

Sequence text (399,10)

12345678901

```
Sing a song
ing a song
ng a song o
g a song of
 a song of
a song of s
 song of si
song of six
```

X y

```
sequence (399,11)
  =[[8, 19, 23, 17, 0, 11, 0, 28, 24, 23, 17 ]
    [19, 23, 17, 0, 11, 0, 28, 24, 23, 17, 0]
    …
   ]
```

```
X (399,10)
  =[[8, 19, 23, 17, 0, 11, 0, 28, 24,  23 ]
    [19, 23, 17, 0, 11, 0, 28, 24, 23, 17]
    …
   ]
```

```
y (399,)
  =[17
    0
    …
   ]
```

```
X_ohe (399,10,34)
  =[[[0 0 0 0 1 0..], [..],m…[…]]
    [[0 0 0 0 1 0..], [..],m…[…]]
    …
   ]
```

```
y_ohe (399,34)
  =[[0 0 0 .. 1 0 ..]
    [1 0 0 .. 0 0 ..]
    …
   ]
```

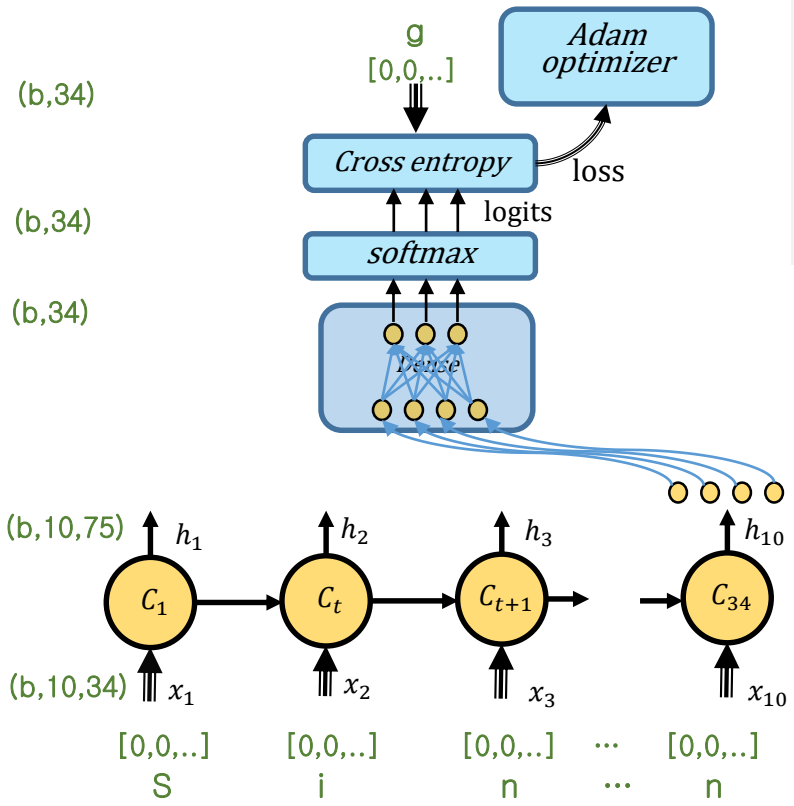
Sing a song of sixpence, a pocket full of rye,
Four and t
When the
Oh wasn't

The king w
The queen
The maid
When dow

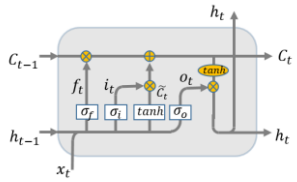
```
with  open('data/rnn_sixpence_data.txt','r') as f:
      raw_text= f.read()
      #Sing a song of sixpence, a pocket full of rye,
      #Four and twenty blackbirds baked in a pie.
      #...
text = raw_text.split()   #strip all of the new line characters , separated only by white space
text = ' '.join(text)
    #['Sing', 'a', 'song', 'of', 'sixpence,', 'a', 'pocket', 'full', 'of', 'rye,', 'Four', 'and', 'twenty'
    #Sing a song of sixpence, a pocket full of rye, Four and twenty

alphabet = sorted(list(set(text))) #extract alphabet
vocab_size = len(alphabet)        #34
    #[' ', '!', '"', ',', '.', '?', 'F', 'O', 'S', 'T', 'W', 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'k', 'l', 'm', 'n', 'o',
'p', 'q', 'r', 's', 't', 'u', 'w', 'x', 'y']
mapping_c2i = {c:i for i, c in enumerate(alphabet)}
    #{' ': 0, '!': 1, '"': 2, ',': 3, '.': 4, '?': 5, 'F': 6, 'O': 7, 'S': 8, 'T': 9, 'W': 10, 'a': 11, …,'y': 33}
text_encoded = [mapping_c2i[c] for c in text]          #(399,11)
    #[8, 19, 23, 17, 0, 11, 0, 28, 24, 23, 17, 0, 24, 16, 0, 28, 19, 32, 25, 15, 23, 13,

length = 10
sequence=np.array([text_encoded[i-length:i+1] for i in range(length, len(text_encoded))])
#(399,11)

#train test set
X,y=sequence[:,:-1],sequence[:,-1]                          #(399,10), (399,)
X_ohe =to_categorical(X, num_classes=vocab_size)#(399,10,34)
y_ohe = to_categorical(y, num_classes=vocab_size) #(399,34)
```

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

- Model 생성




```
12345678901
Sing a song
ing a song
ng a song o
g a song of
 a song of
a song of s
 song of si
song of six
X          y
```



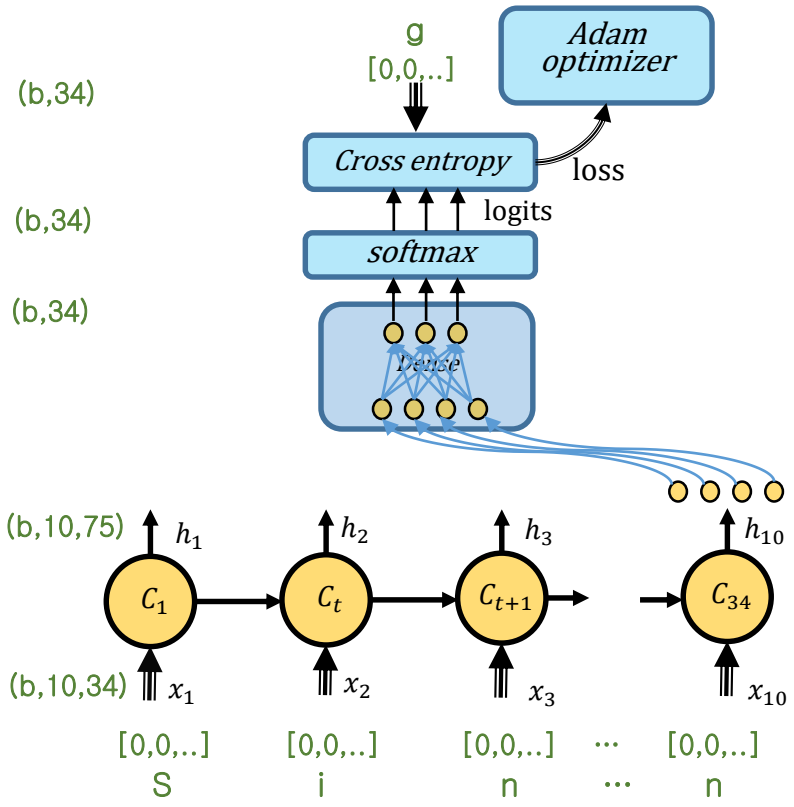
```
# define model
model = Sequential(name="many-to-one RNN")
model.add(LSTM(75, input_shape=(X_ohe.shape[1], X_ohe.shape[2])))       #(10,75)
model.add(Dense(vocab_size, activation='softmax'))                      #(34,) => (10,34)
print(model.summary())

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

# fit model
model.fit(X_ohe, y_ohe, epochs=100, verbose=2)

# save the model to file
model.save('model_many2one_sixpense.h5')

# save the mapping
dump(mapping_c2i, open('mapping.pkl', 'wb
```

```
Model: "many-to-one RNN"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 75)                33000
_________________________________________________________________
dense_1 (Dense)              (None, 34)                2584
=================================================================
Total params: 35,584
Trainable params: 35,584
Non-trainable params: 0
```

```
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 : [[0,0,0,0,0,0,0,1,0,0,…[…]]       #[8, 19, 23, 17, 0, 11, 0, 28, 24, 23],
    [ [[0,0,0,0,0,0,0,0,0,0,…[…]]     #[19, 23, 17, 0, 11, 0, 28, 24, 23, 17],
```

Example 2. 문자 기반 신경 언

- Model 생성




```
import numpy as np
from pickle import dump
from keras.utils import to_categorical
from keras.models import Sequential
from keras.layers import Dense,LSTM
# dataset 생성
with  open('data/rnn_sixpence_data.txt','r') as f:
    raw_text= f.read()
text = raw_text.split()          #strip all of the new line characters , separated only by white space
text = ' '.join(text)            #Sing a song of sixpence, a pocket full of rye, Four and twenty blackbirds ...\
alphabet = sorted(list(set(text)))
vocab_size = len(alphabet)     #34  [' ', '!', '"', ',', '.', '?', 'F', 'O', 'S', 'T', 'W', 'a', 'b', 'c', 'd', 'e', …}
mapping_c2i = {c:i for i, c in enumerate(alphabet)}   #mapping from char to code(inxdex)
text_encoded = [mapping_c2i[c] for c in text]
                                 #[8, 19, 23, 17, 0, 11, 0, 28, 24, 23, 17, 0, 24, 16, 0, 28, 19, 32, 25,
length = 10
sequence=np.array([text_encoded[i-length:i+1] for i in range(length, len(text_encoded))])   #(399,11)
X,y=sequence[:,:-1],sequence[:,-1]                          #(399,10),  (399,)
X_ohe =to_categorical(X, num_classes=vocab_size) #(399,10,34)
y_ohe = to_categorical(y, num_classes=vocab_size)  #(399,34)

# 모델 생성
model = Sequential(name="many-to-one RNN")
    model.add(LSTM(75,
            input_shape=(X_ohe.shape[1], X_ohe.shape[2])))#(10,34)(None, 75) 33000 (75x440)
model.add(Dense(vocab_size, activation='softmax'))          #(34,)(None, 34)    2584 (75x34(w)
+34(b))
print(model.summary())
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
model.fit(X_ohe, y_ohe, epochs=100, verbose=2)    # fit model
model.save('model_many2one_sixpence.h5')                # save the model to file
dump(mapping_c2i, open('mapping.pkl', 'wb'))       # save the mapping
```

Example 2. 문자 기반 신경 언어 모

- 모델 평가
  - 모델을 load하고
  - 모델에게 seedtext를 제시하고
  - 문자열(20개)를 생성하게 한다.

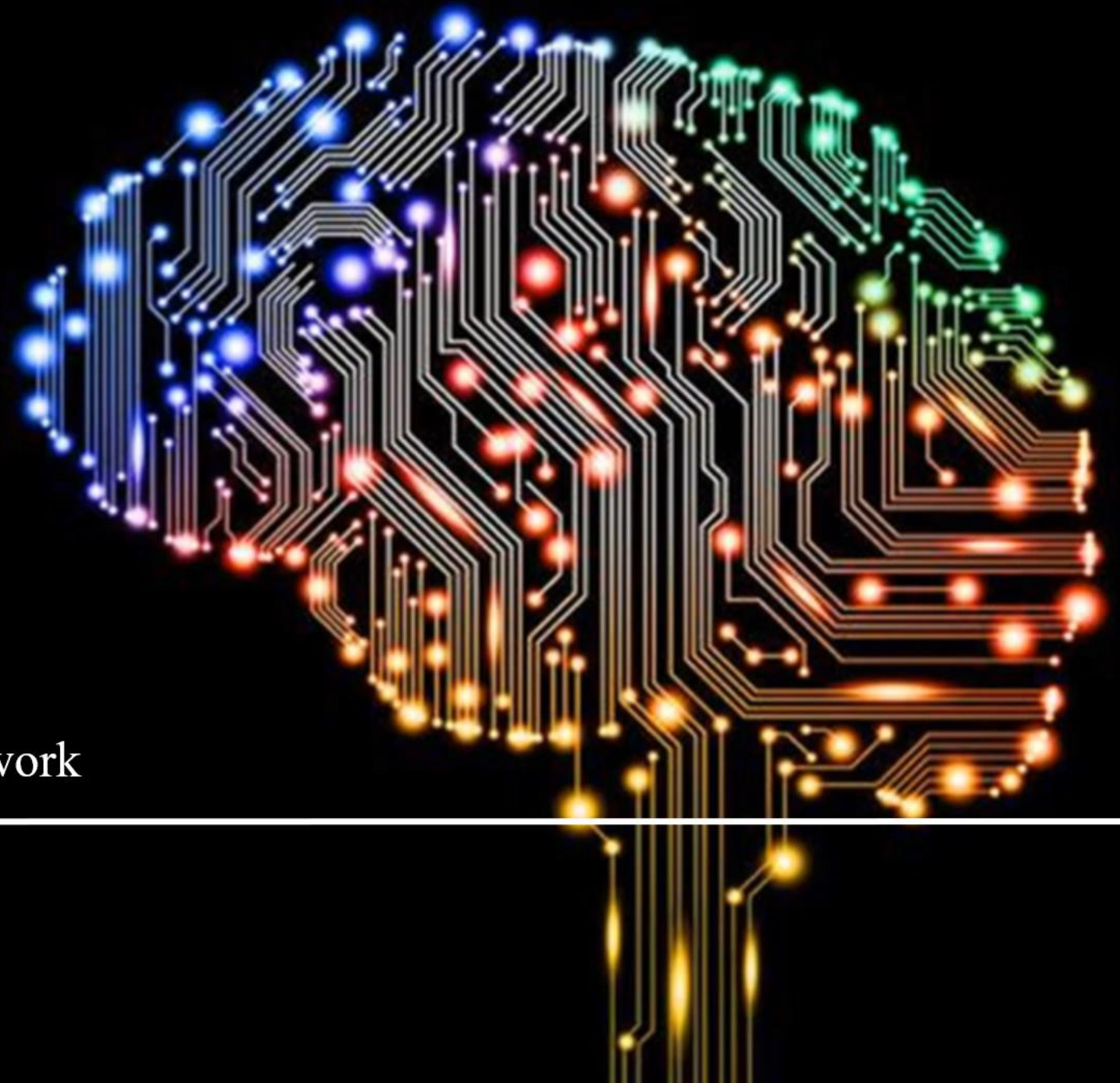
- 실험
  - 학습된 시작부분 텍스트로부터 20개의 문자의 텍스트(연속문자열) 생성시도
  - 학습된 중간 부분 텍스트로부터 20개문자의 텍스트 생성 시도
  - 학습된 않은 텍스트로부터 20개문자의 텍스트 생성시도

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

# generate a sequence of nb characters with a language model
def generate_seq(model,seq_length, seed_text='Sing a son', nb_chars=20):
    in_text = seed_text
    # generate a fixed number(nb_chars) of characters
    for _ in range(nb_chars):
        # encode the characters as integers
        encoded = [mapping_c2i[char] for char in in_text]#'Sing a son' (n,) =[8,19,...]
        # truncate sequences to a fixed length(seq_length)
        encoded = pad_sequences([encoded], maxlen=seq_length, truncating='pre')#(1,10)
        # one hot encode
        encoded_ohe = to_categorical(encoded, num_classes=len(mapping_c2i))#(1,10,34)
        # predict character
        yhat = model.predict_classes(encoded_ohe, verbose=0)        #[17] (1,)
        char=mapping_i2c[yhat[0]]  #'g'
        in_text += char
return in_text

# load the model and mapping
model = load_model('model_rnn_sixpence.h5')
mapping_c2i = load(open('mapping.pkl', 'rb'))        #{' ':0,...}
mapping_i2c={v:k for k,v in mapping_c2i.items()}  #{0:' ',...}

# test start of rhyme
print(generate_seq(model,10, 'Sing a son', 20))    #Sing a song of sixpence, a poc
# test mid-line
print(generate_seq(model, 10, 'king was i', 20))    #king was in his counting house
# test not in original
print(generate_seq(model,10, 'hello worl', 20))    #hello worl,, The aeen was in t
```

```
Epoch 98/100
 - 0s - loss: 0.3062 - accuracy: 0.9850
Epoch 99/100
 - 0s - loss: 0.2950 - accuracy: 0.9699
Epoch 100/100
 - 0s - loss: 0.2864 - accuracy: 0.9850
```

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. RNN models
5. Examples in Keras
    - (1)정현파신호 샘플의 예측 모델(SimpleRNN) in keras
    - (2)문자 기반 신경 언어 모델(many-to-one RNN)-sixpence in keras
    - (3)문자 기반 신경 언어 모델(many-to-many RNN)-sixpence in keras
    - (4) 주가예측 모델



Deep Learning
Deep Neural Network

Yoon Joong Kim,
Hanbat National University