Deep Learning
Deep Neural Network

Yoon Joong Kim,
Hanbat National University

Deep Learning

RNN(1)
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

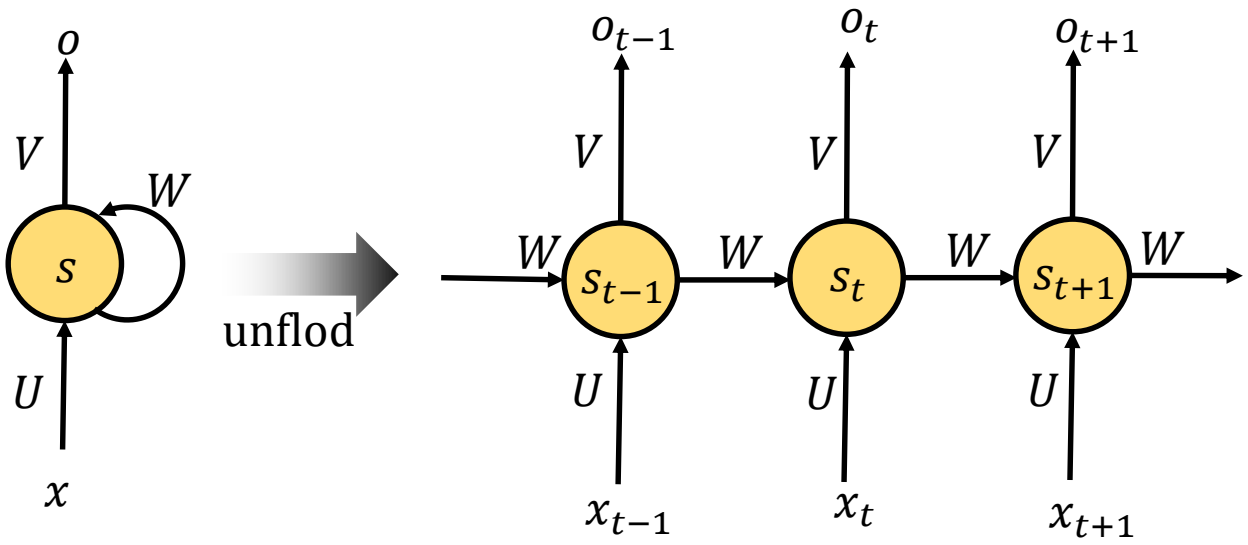
Agenda

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

1. RNN

- 1.1 From feed-forward to RNNs

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

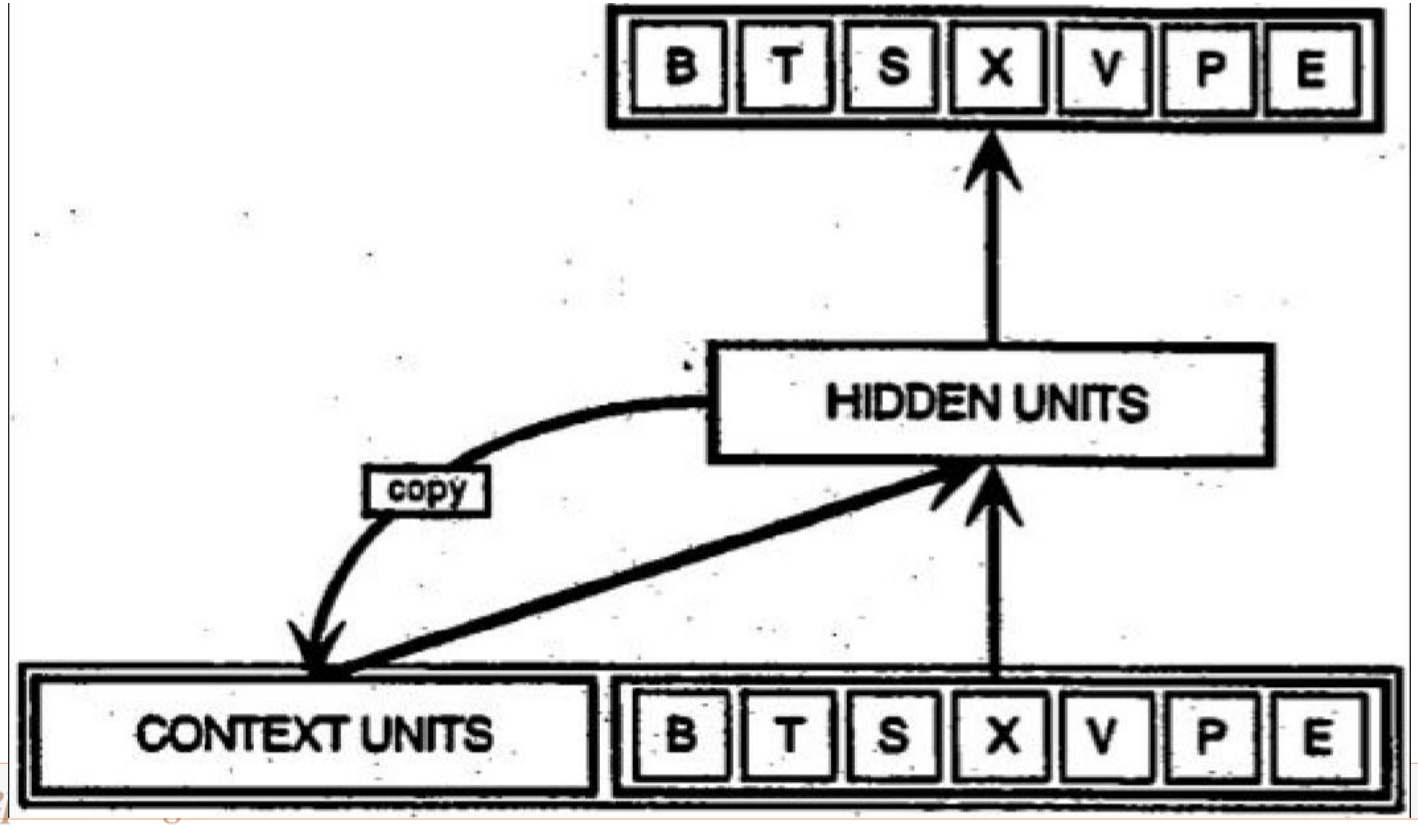


1. RNN(cont.)

- 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

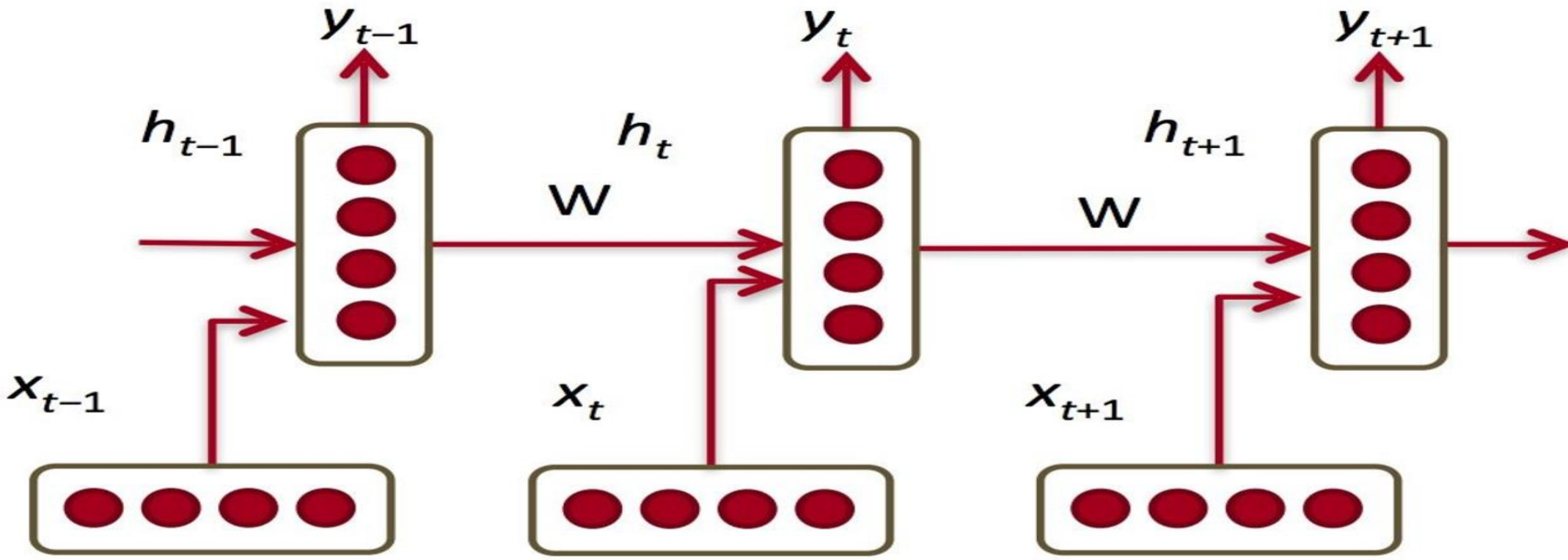


Diagram from CS 224D Slides

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)

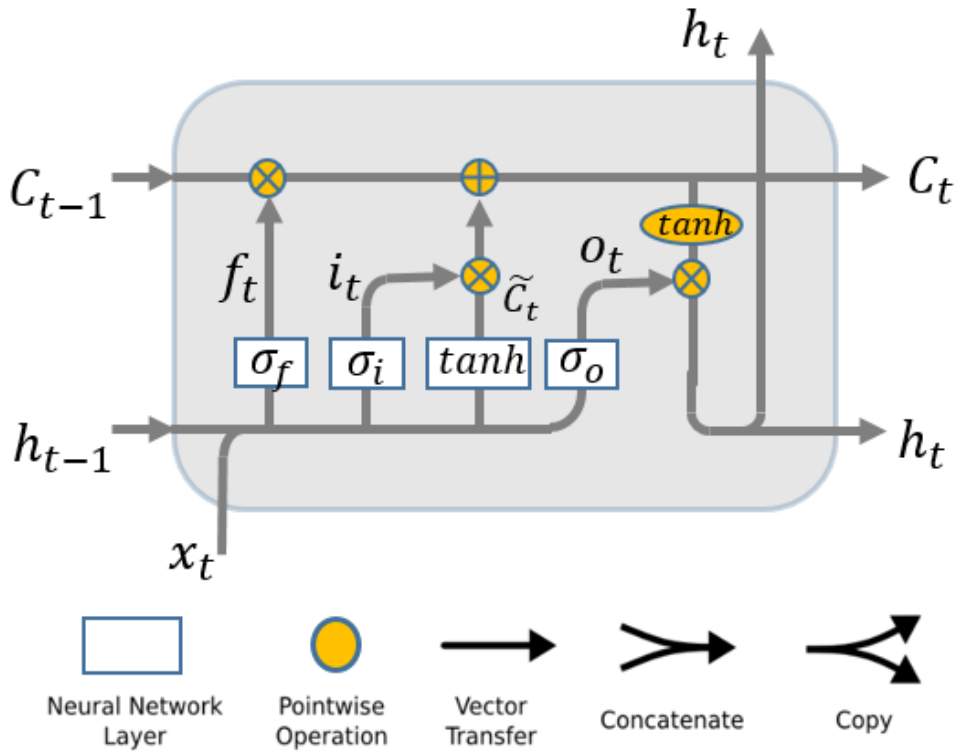
- 1.5 LSTM(Long Short Term Memory)
  - 새로운 입력 중 얼마의 량을 기억하고, 이전 기억된 숨겨진 상태(hidden state)중 얼마를 잊어야 하는지를 제어할 수이다.
  - 인간이 정보를 처리하는 방법과 매우 흡사하다.

  - 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, Jürgen 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
$\widetilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$ the new state(memory)
$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t$ the state(memory)
$h_t = o_t * \tanh(C_t)$ the output

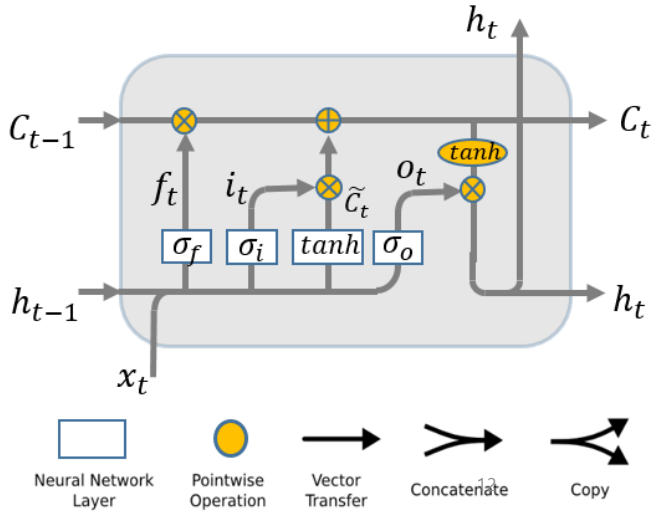
1.5 LSTM(cont.)

- LSTM(Long Short Term Memory)
  - LSTM [20]은 게이팅 메커니즘을 사용하여 별도의 메모리 셀을 사용하지 않고 시퀀스 상태를 추적합니다. 있다
  - 세 가지 유형의 게이트 망각 게이트 f_t, 입력 게이트 i_t 및 출력 o_t을 이용하여 정보가 상태로 업데이트되는 방법을 제어합니다. 시간 t에서 LSTM은 출력을 다음과 같이 계산합니다.

$$h_t = o_t * \tanh(C_t)$$
$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t$$
$$\widetilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

  - 출력 게이트 o_t는 현재의 정보C_t가 출력 h_t로 변화되는 방법을 제어합니다. 현재 상태C_t는 이전 상태C_{t-1}와 현재 새 상태 $\widetilde{C}_t$사이의 일종의 보간 입니다. 현상태 $\widetilde{C}_t$ 새로운 시퀀스 정보 x_t와 이전 출력h_{t-1}으로 계산됩니다.
  - 망각 게이트 f_t및 입력 게이트 i_t 는 과거 상태 및 현재 상태가 새로운 상태 C_t에 얼마나 기여하는지 제어합니다.
    출력 게이트o_t는 상태 C_t가 출력으로 변환되는 량을 제어합니다.
    각 게이트는 다음과 같이 계산됩니다.

$$f_t = \sigma_f(W_f \cdot [h_{t-1}, x_t] + b_f)$$
$$i_t = \sigma_i[W_I \cdot [h_{t-1}, x_t] + b_i)$$
$$o_t = \sigma_o(W_o \cdot [h_{t-1}, x_t] + b_o)$$



Y. Yu and Y. Kim, "A Voice Activity Detection Model Composed of Bidirectional LSTM and Attention Mechanism," 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM), Baguio City, Philippines, 2018, pp. 1-5, doi: 10.1109/HNICEM.2018.8666342.

1.5 LSTM(cont.)

- 최신 음성인식 모델들은 LSTM RNN을 기반으로 개발되고 있다. LSTM은 FNN(Feedforward Neural Network)에 비하여 재귀적으로 처리하는 기능이 포함되어 있으므로 음성과 같이 시계열 형태의 데이터 처리에 있어서 괄목할 만한 성능을 가지고 있다. LSTM의 처리 과정은 다음과 같다. 순차적으로 입력되는 데이터에 대하여 상태 값을 동적으로 업데이트 한다. 현 상태의 크기는 이전 상태의 크기와 현 입력의 크기가 비선형으로 인터폴레이션(Interpolation)되어 임시의 현재 상태 값을 계산해 내고 그 값을 출력 게이트로 제어하여 최종적인 현 상태 값을 계산한다. 비선형 인터폴레이션은 망각 게이트와 입력 게이트에 의해서 제어되는 것을 의미한다. 모든 게이트는 신경망으로 구성된다. LSTM은 이와 같은 구조를 가짐으로써 이전 정보를 선택적으로 기억할 수 있어 음성신호에 산재되어 있는 감정의 정보를 기억하는데 유효하다.
- 따라서 LSTM은 음성 특징열의 의존성을 학습할 수 있고 의존성의 패턴은 감정에 따라 고유하게 대응될 수 있으므로 시계열 특징 패턴으로부터 감정을 학습하고 인식하는 모델에서 중요한 역할을 수행한다. BLSTM은 순방향 의존성 뿐만 아니라 역방향 의존성까지 기억할 수 있어 감정학습에 효율적이다.

고유정, 김윤중. (2019). GAN을 이용한 음성 감정 인식 모델의 성능 개선. 한국정보기술학회논문지, 17(11), 77-85.

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 응용 사례

- RNN의 응용 사례
  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 입력의 중요성을 측정할 수 있다.
  - 새로운 텍스트를 생성 할 수 있습니다

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

- Linux Source Code Generator
  - rej 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;
}
```

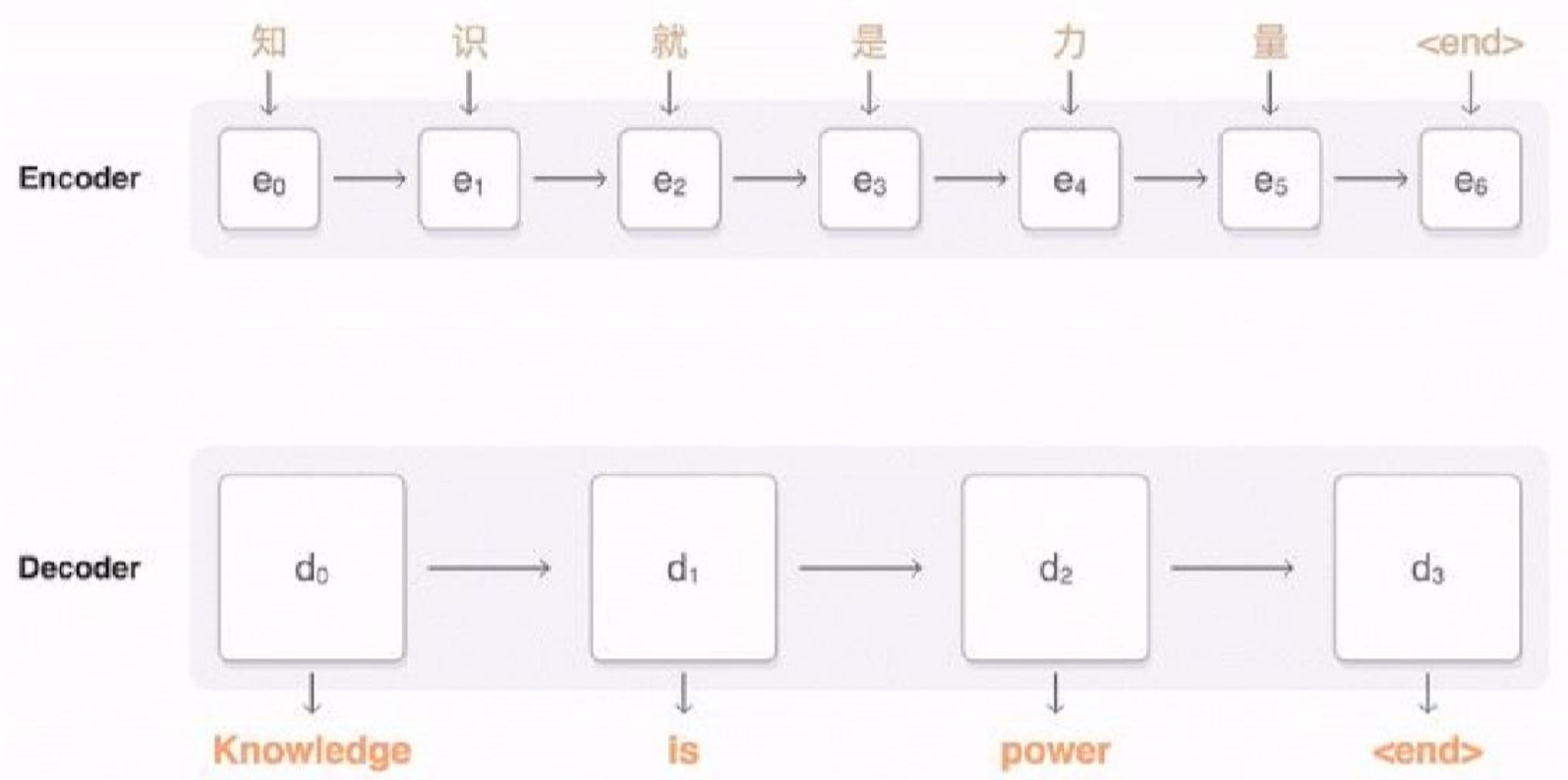
2.2 Character-level Language Modeling

- 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

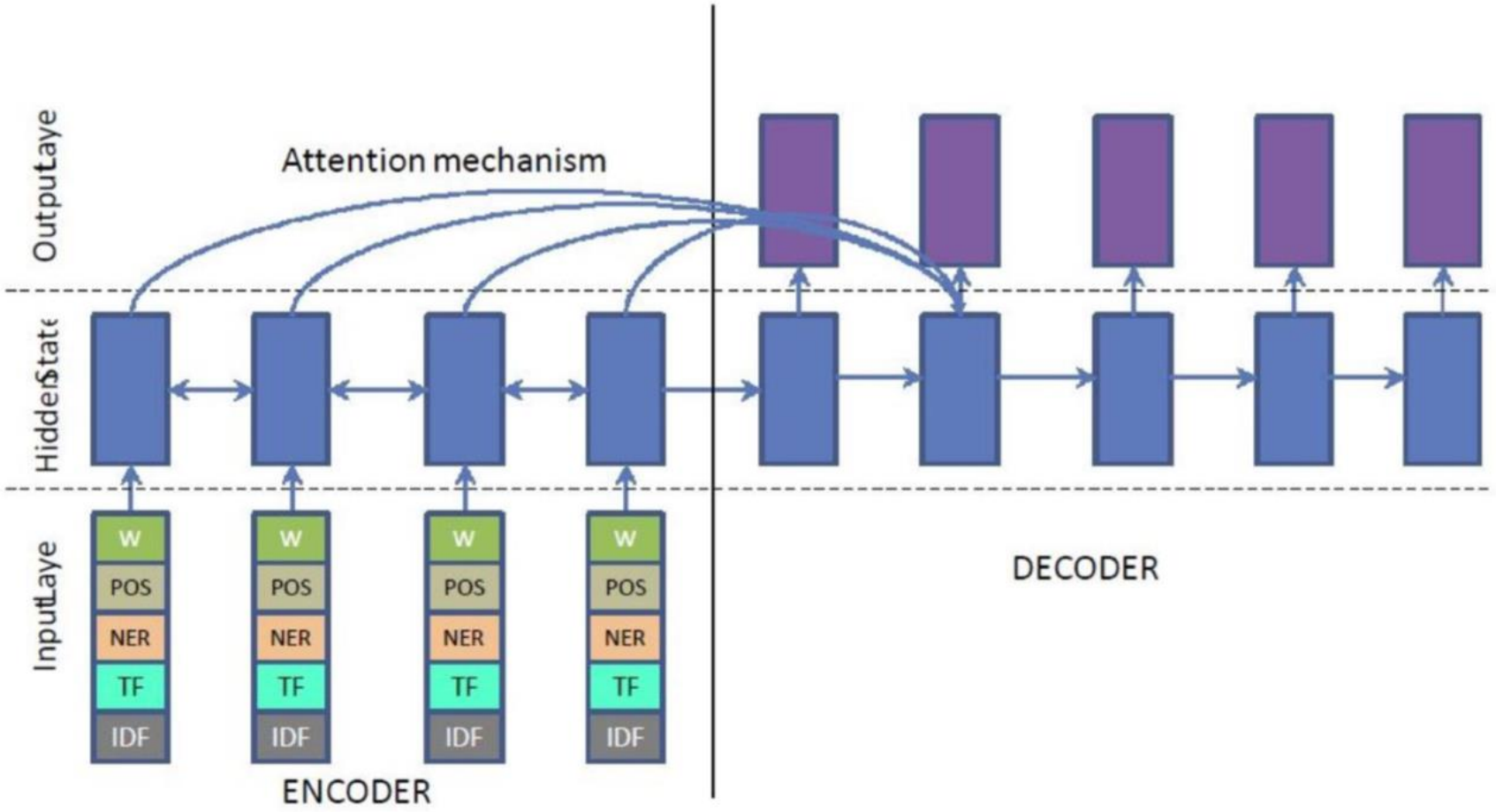
- Google Neural Machine Translation Google Research's blog)

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



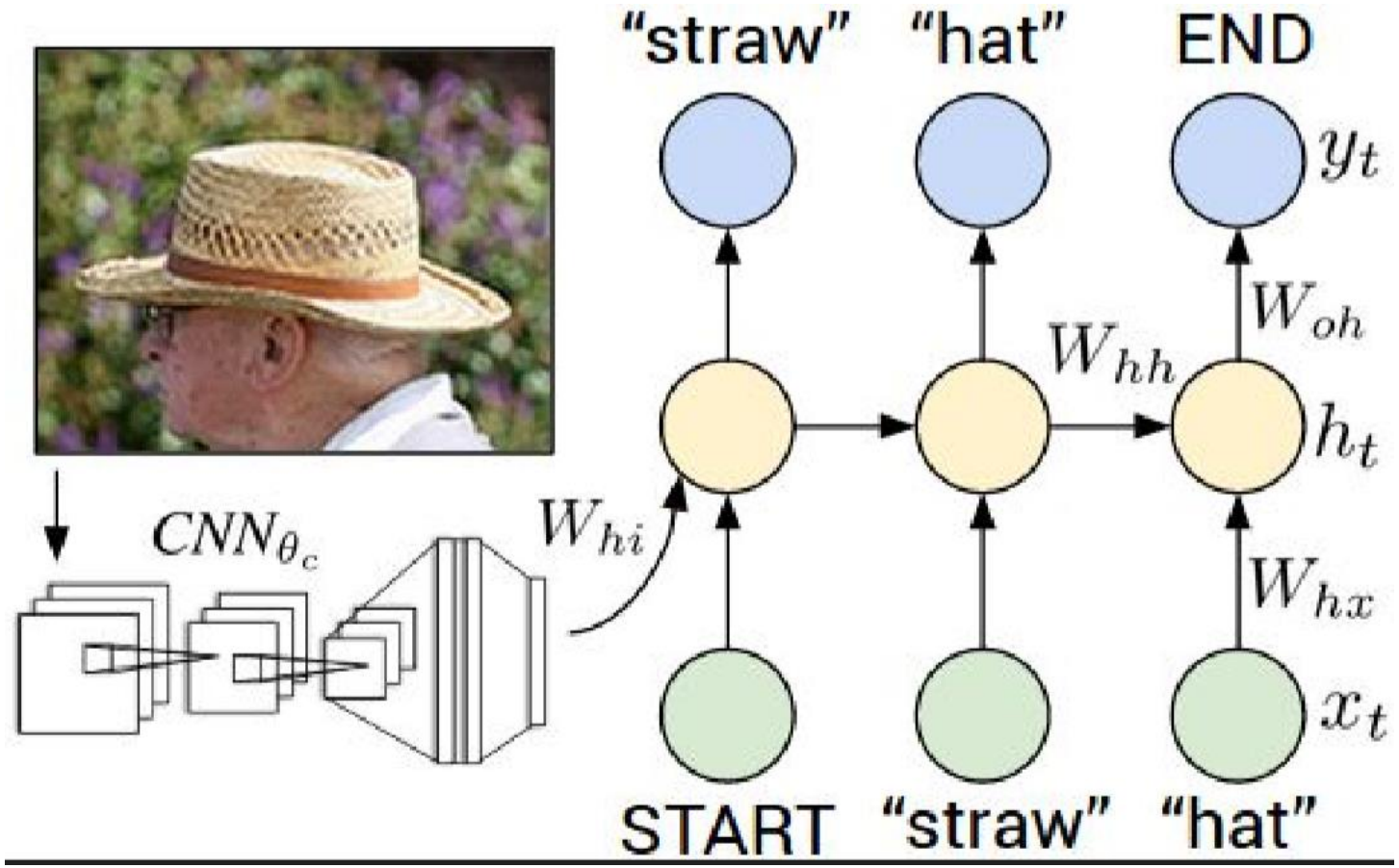
Nallapati, Ramesh, et al. "Abstractive text summarization using sequence-to-sequence rnns and beyond." arXiv preprint arXiv:1602.06023 (2016).

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

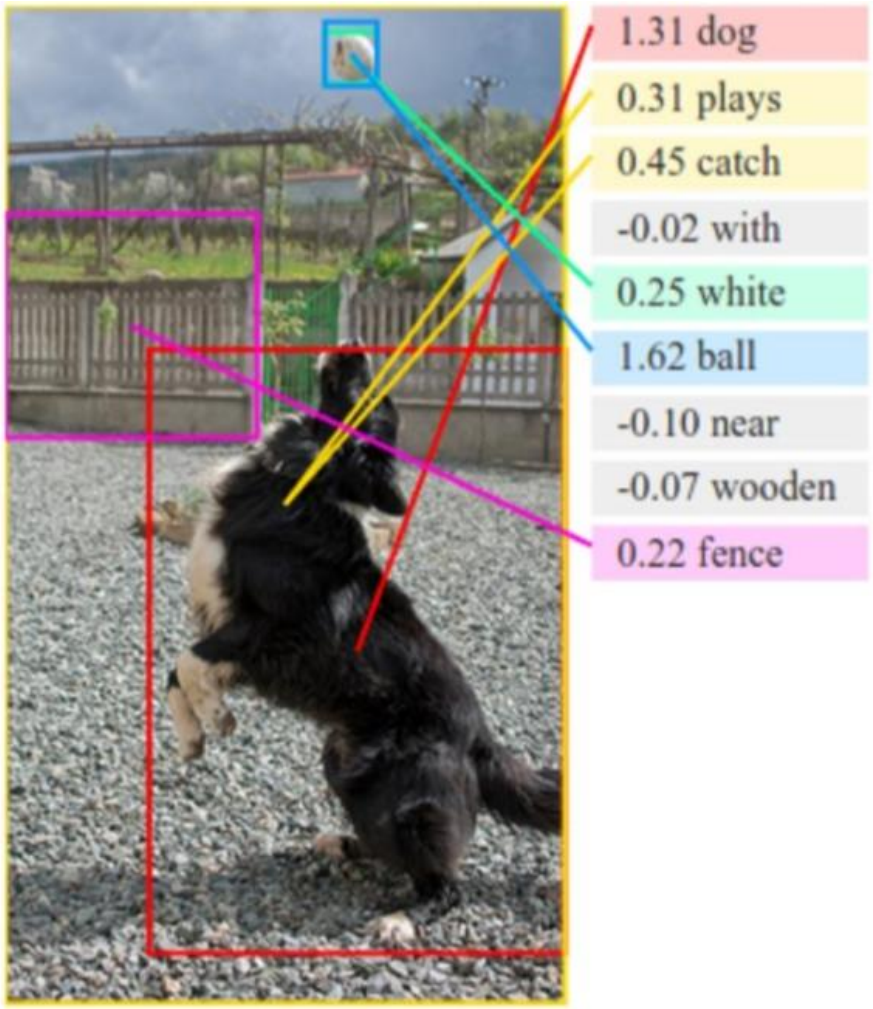


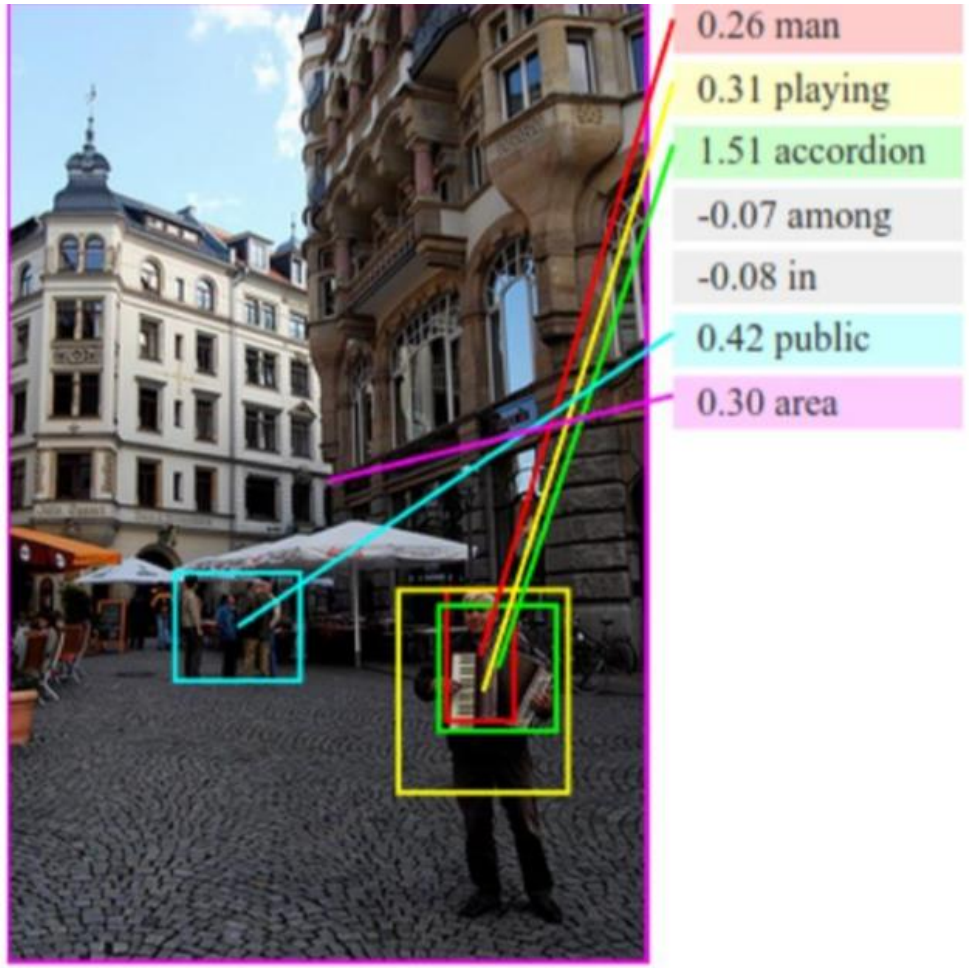
Karpathy, Andrej, and Li Fei-Fei. "Deep visual-semantic alignments for generating image descriptions." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

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.

3. Language Modeling

- Neural Language Modeling
  - 텍스트로 모델을 학습하고 모델로 텍스를 생성한다.
  - 문장이 기계 번역에 중요한 입력이 될 가능성을 측정 할 수 있습니다 (고 확률 문장은 일반적으로 정확하기 때문에)
  - 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

- 언어 모델링 : N-그램(language Model N-grams)
  - 아주 최근까지 전통적인 접근 방식
  - 이전 n- 그램을 기반으로 다음 단어를 예측하도록 모델을 훈련
  - 거대한 어휘
  - OOV(Out Of Vocabulary)로 일반화 할 수 없다.
  - 많은 메모리가 필요합니다
- 언어 모델링 : 문자 수준(Language Modeling: Character-level)
  - 2010 년대 초에 도입되다.
  - 입력과 출력 모두 문자
  - 장점(pros) :
    - 아주 작은 어휘
    - word embeddings이 필요하지 않습니다
    - 훈련이 빨리 된다.
  - 단점(Cons) :
    - 낮은 유창성 (많은 단어가 횡설수설 할 수 있음)

3. Language Modeling

- 언어 모델링 : 하이브리드(Language Modeling: Hybrid)
  - 기본은 단어수준으로
  - 알 수 없는 토큰에 대해서는 문자 수준으로 전환

- 언어 모델링 : 하위 단어 수준(Language Modeling: Subword-Level)
  - 입력 및 출력은 하위 단어입니다 ,
  - 가장 자주 쓰는 단어 W 유지
  - 가장 자주 발음 되는 음절 S을 유지
  - 나머지는 문자로 분할
  - 단어 수준 및 문자 수준 모델보다 성능이 더 좋은 것으로 보입니다
  - 예,
    - "conference" => 'con', 'f', 'er', 'ence' , 오늘은=>'오늘' '은'
    - new company dreamworks interactive
      =>new company dre+ am+ wo+ rks: in+ te+ ra+ cti+ ve interactive

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

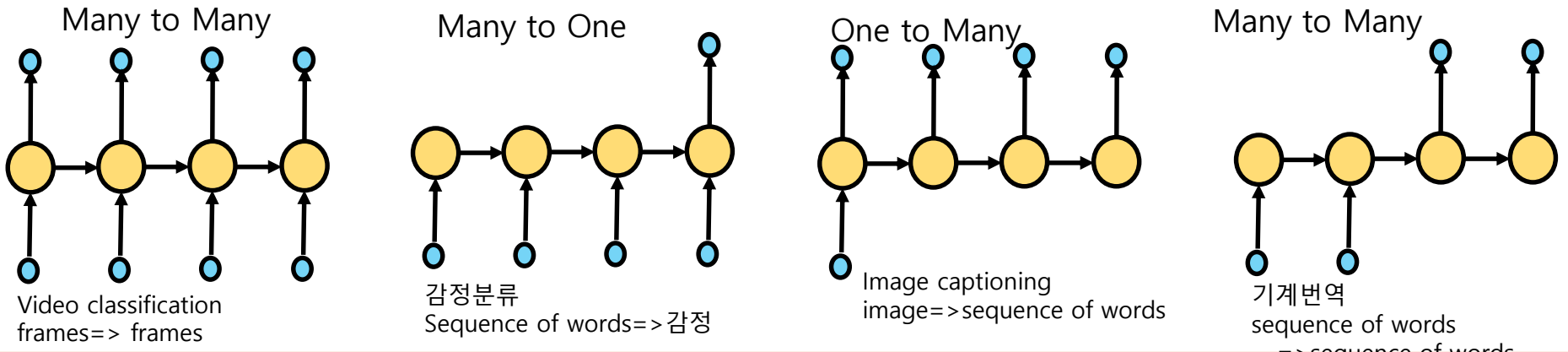
3.1 Language Modeling DEMO
Character-level, Language Modeling

- 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.”

4. RNN models

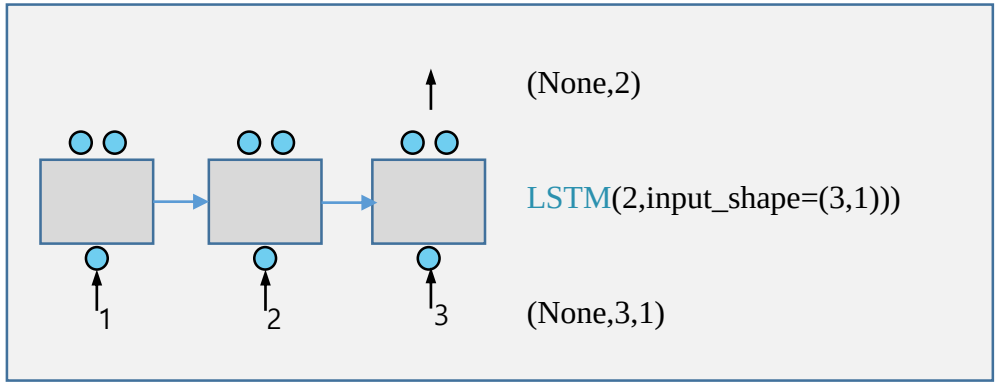
- **Back-Propagation Through Time (BPTT)**
  - RNN은 시간에 따라 펼쳐 놓으면 구조가 MLP와 유사하기 때문에 Back-Propagation 방법으로 gradient를 계산할 수 있다. 다만 실제로 여러 개의 은닉층이 있는 것이 아니라 시간 차원에서 존재하기 때문에 Back-Propagation Through Time (BPTT) 방법이라고 한다.

- **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/

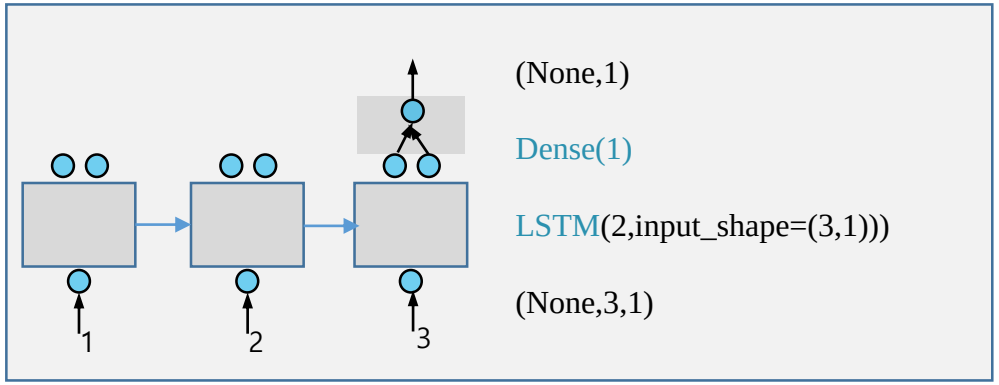


- Many-to-One model

```
from tensorflow.keras.layers import LSTM,TimeDistributed,Dense
from tensorflow.keras.models import Sequential
import numpy as np

d=np.array([[[1],[2],[3]]])
print(d.shape,d) # (1, 3, 1)  [[[1]  [2]  [3]]]

# 1) 3 to 1 model  (None, 3, 1)=>(None, 2)
model =Sequential()
model.add(LSTM(2, input_shape=(3,1),return_sequences=False))
print(model.input_shape,model.output_shape) #(None, 3, 1) (None, 2)
model.summary()
#lstm (LSTM)                 (None, 2)                 32
r=model.predict(d)# (1, 2) [[ 0.2092029  -0.68145883]]
```



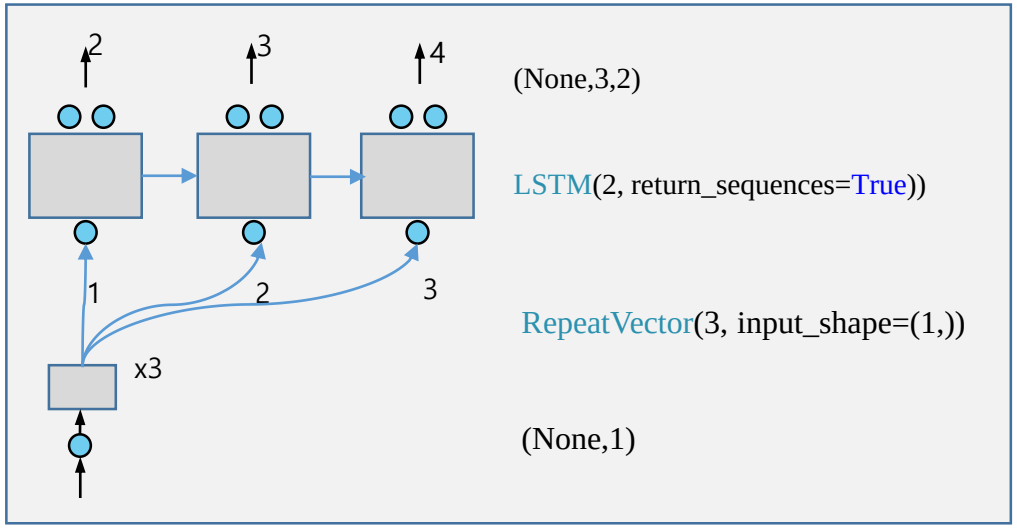

```
# 2) 3 to 1 model   (None, 3, 1)=>(None, 1)
model =Sequential()
model.add(LSTM(2, input_shape=(3,1),return_sequences=False))
model.add(Dense(1))
r=model.predict(d) # (1, 1) [[0.14355654]]

# lstm_2 (LSTM)               (None, 2)                 32
# dense_2 (Dense)             (None, 1)                 3
```



4. RNN models(cont.)

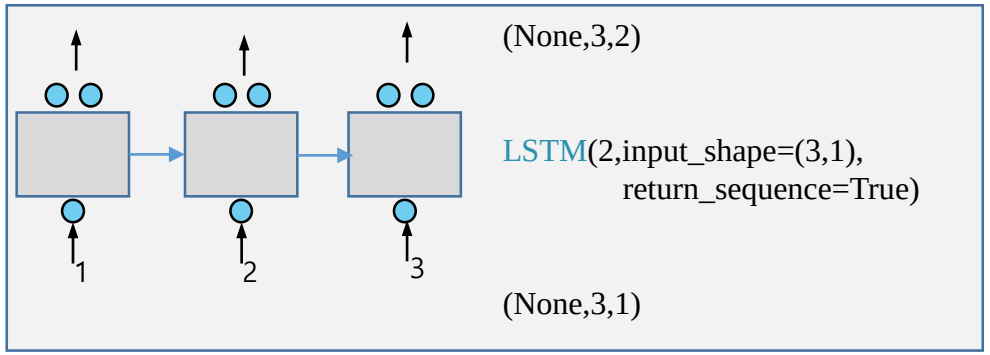
- One-to-many model

```
# one(1) to many(3)
model.add(RepeatVector(3, input_shape=(1,)))
model.add(LSTM(2, return_sequences=True))
print(model.summary())
#repeat_vector_1 (RepeatVecto (None, 3, 1)           0
#lstm_1 (LSTM)                (None, 3, 2)           32
#Total params: 32
```



- May-to-Many models

```
model =Sequential()
model.add(LSTM(2, input_shape=(3,1),return_sequences=True))

r=model.predict(d)  # (1, 3, 2)
[[[-0.07719133 -0.07411253]  [-0.14580584 -0.13367155]  [-0.16403513 -0.15489359]]]
#lstm (LSTM)                 (None, 2)                  32
```

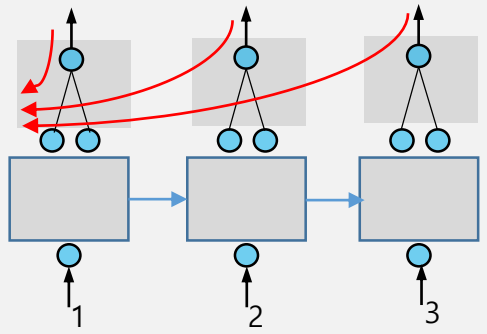


4. RNN models(cont.)

- Many-to-many , TimDistributed[link]

```
model =Sequential()
model.add(LSTM(2, input_shape=(3,1),return_sequences=True))
model.add(TimeDistributed(Dense(1)))

r=model.predict(d)
# (1, 3, 1)  [[[ 0.00479423]  [-0.00230395]  [-0.01651487]]]

model.summary()
#lstm (LSTM)                 (None, 3, 2)              32
#time_distributed (TimeDistr  (None, 3, 1)             3
# ibuted)
```

각 스텝에서 Cost가 계산되고 각 지점에서 오류가 전파된다.
(TimeDistributed)



(None,3,1)

TimeDistributed(Dense(1)))

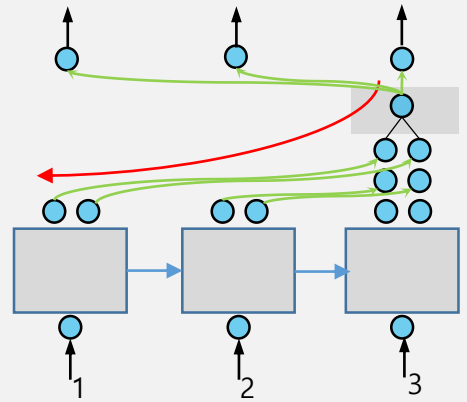
LSTM(2, input_shape=(3,1),
return_sequences=True))

(None,3,1)

- Many-to-many,

```
#3 to 3 model    (None, 3, 1)=>(None, 3, 1)
model =Sequential()
model.add(LSTM(2, input_shape=(3,1),return_sequences=True))
model.add(Dense(1))

r=model.predict(d)
# (1, 3, 1) [[[-0.06636765]  [-0.11807974]  [-
0.1118229 ]]]

#lstm_1 (LSTM)               (None, 3, 2)              32
#dense_1 (Dense)             (None, 3, 1)              3
```

Cost간 한번에 계산되고 끝까지 전파된다.



(None,3,1)

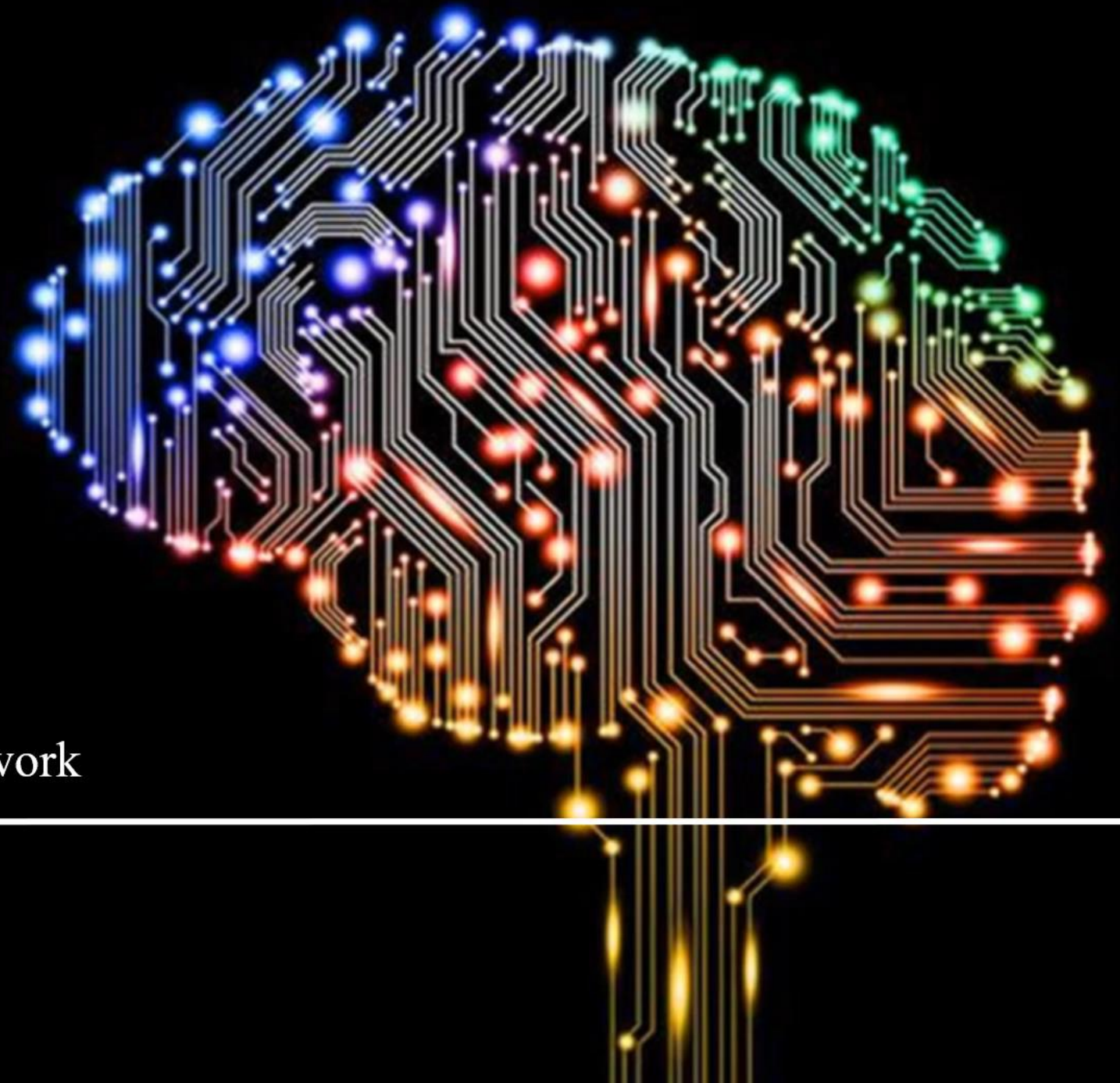
Dense(1)

LSTM(2, input_shape=(3,1),
return_sequences=True))

(None,3,1)

Recap

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. Example 1. 정현파신호 샘플예측 모델(SRNN)
    2. Example 2. 문자 기반 신경 언어 모델-sixpence
    3. Example 3. 주가예측 모델



Deep Learning
Deep Neural Network

Yoon Joong Kim,
Hanbat National University