

---

# Recurrent Neural Network (순환신경망)

---

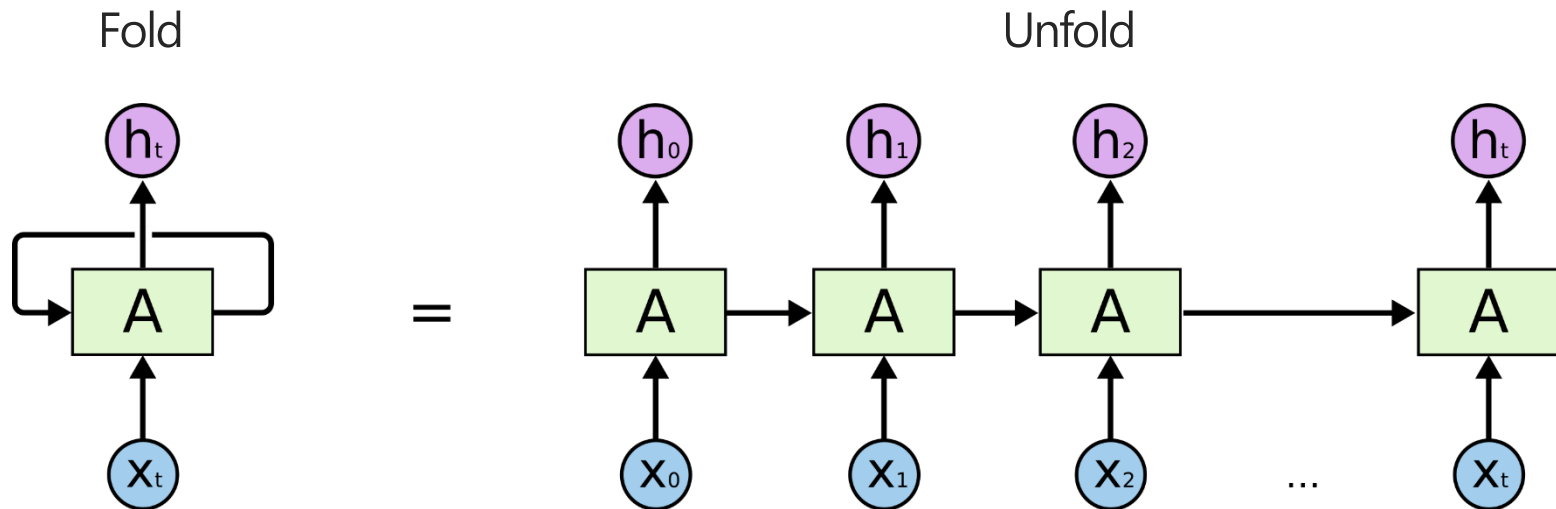
# 목차

---

- ❖ 순환 신경망 확장 모델 (LSTM)
- ❖ LSTM 변형 모델 (GRU)
- ❖ 모델링 가이드라인
- ❖ Sequence to Sequence & Attention mechanism

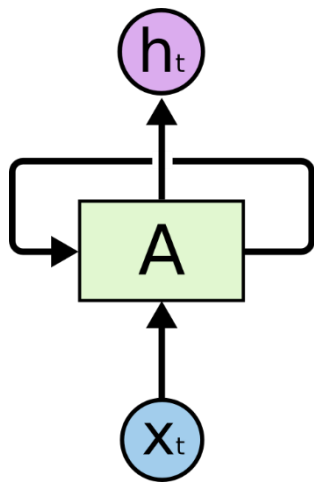
# 순환 신경망 확장 모델

- ❖ RNN은 긴 시퀀스를 모델링하기 어려움
  - 장기 의존성 문제



# 순환 신경망 확장 모델

- ❖ RNN은 긴 시퀀스를 모델링하기 어려움
  - 장기 의존성 문제



“안녕하세요, 오늘 순환 신경망 강의를 맡은 조운상입니다.

... RNN 설명 2시간 후 ...

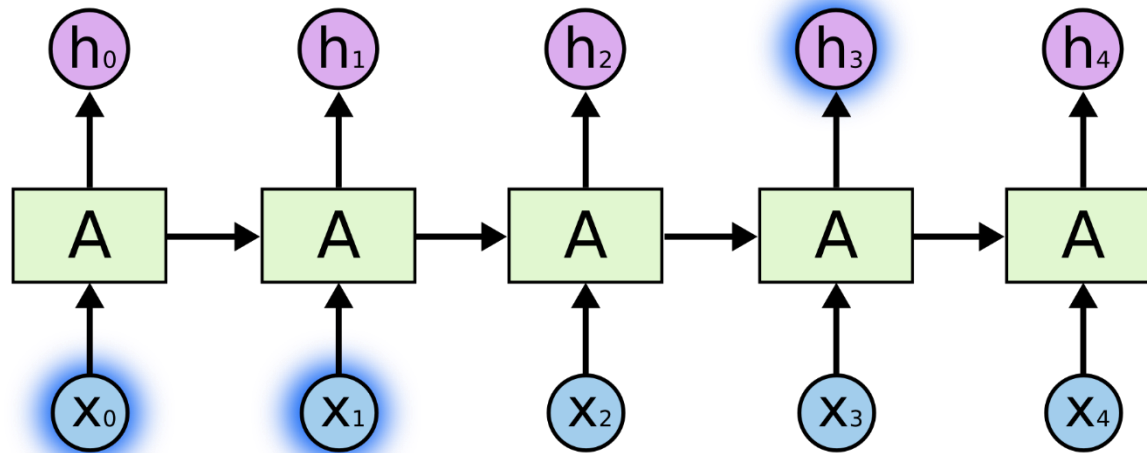
RNN 은 장기 의존성이라는 문제로 Vanishing gradient 라는 한 계점이 있었는데요.

그것을 해결한 것이 바로 **LSTM** 입니다.”

“**LSTM**” 을 예측하고 싶은데 “안녕하세요” 까지 고려하는 게 중요할까?

# 순환 신경망 확장 모델

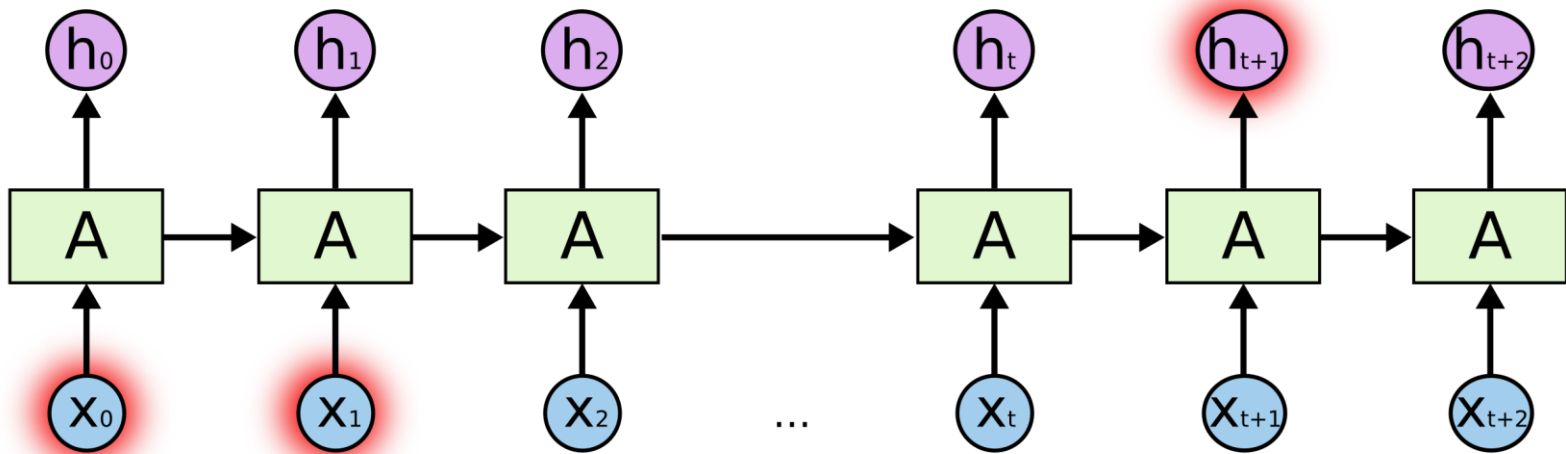
- ❖ RNN은 긴 시퀀스를 모델링하기 어려움
  - 장기 의존성 문제



<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

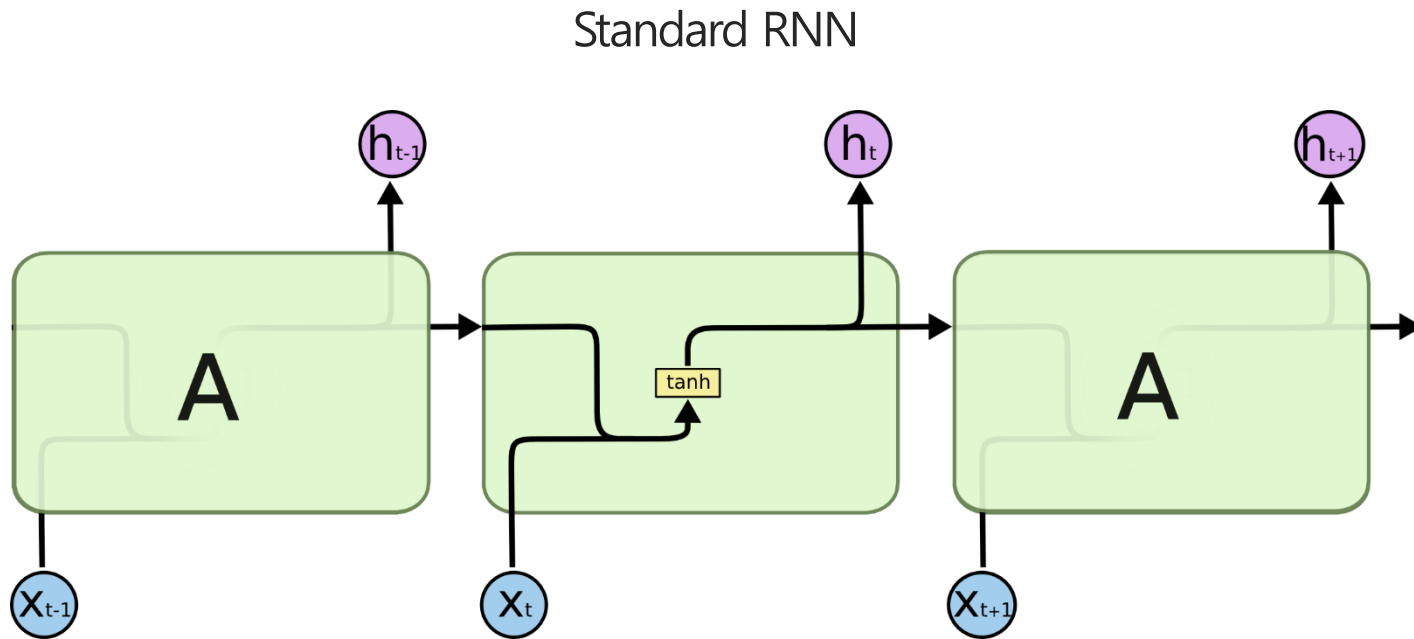
# 순환 신경망 확장 모델

- ❖ RNN은 긴 시퀀스를 모델링하기 어려움
  - 장기 의존성 문제



# 순환 신경망 확장 모델

## ❖ LSTM Networks

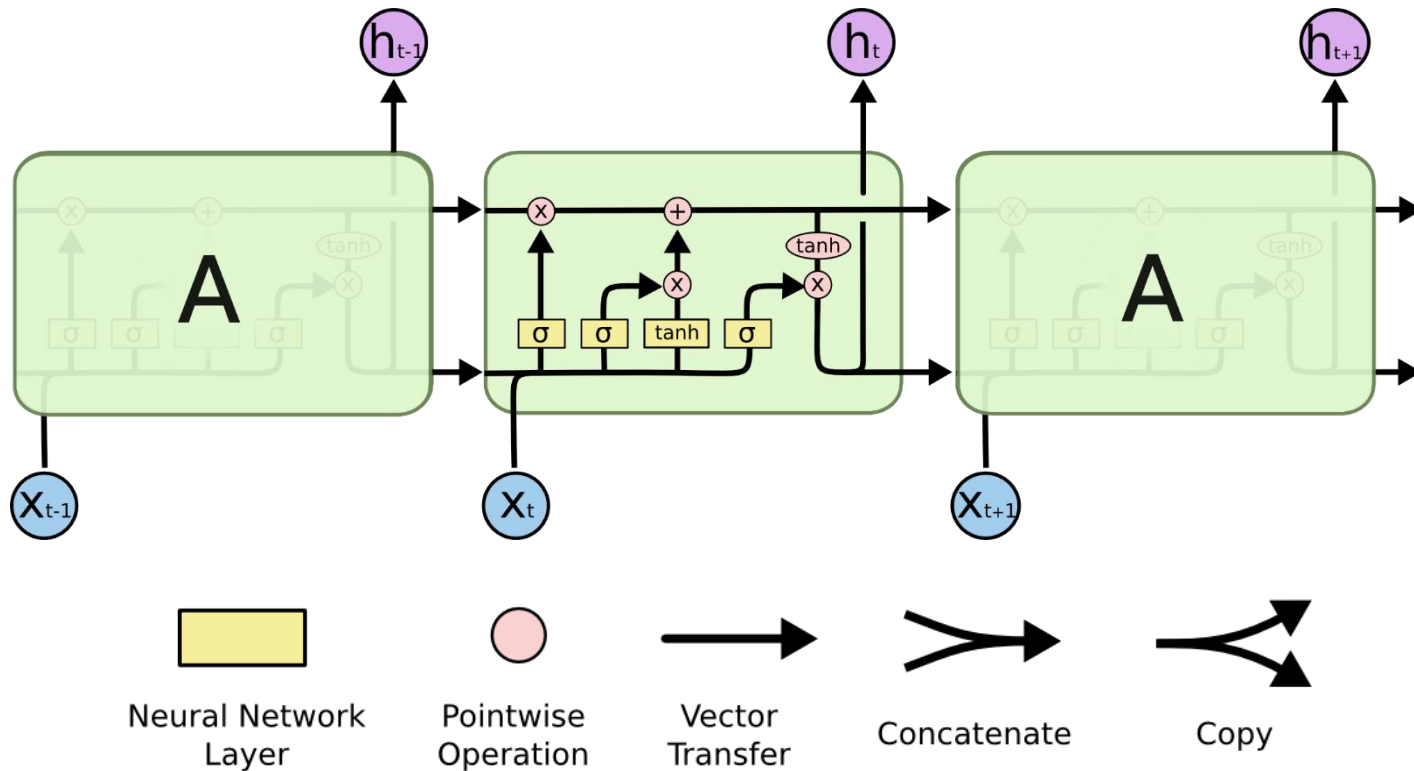


<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

# 순환 신경망 확장 모델

## ❖ LSTM Networks

- 잊을 것은 잊고, 가져갈 것은 가져가자. → Cell state 가 반영해주자!
- Cell state 에서 Hidden state 정보를 얼마나 가져갈 지 정해주자!



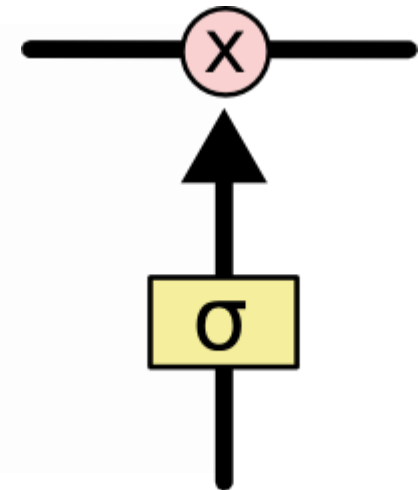
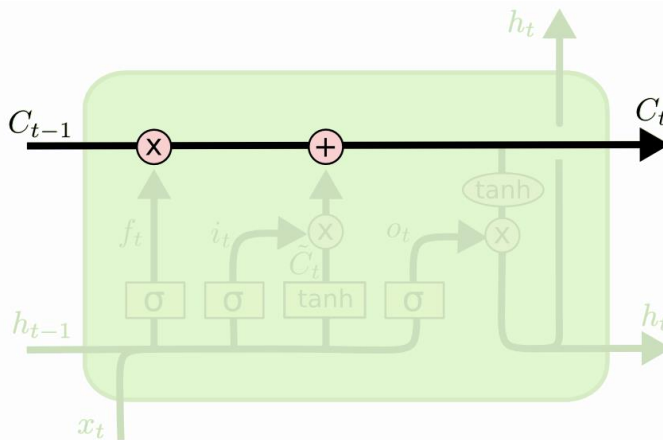
<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>



# 순환 신경망 확장 모델

## ❖ LSTM Networks

- 잊을 것은 잊고, 가져갈 것은 가져가자. → Cell state 가 반영해주자!
- Cell state 에서 Hidden state 정보를 얼마나 가져갈 지 정해주자!



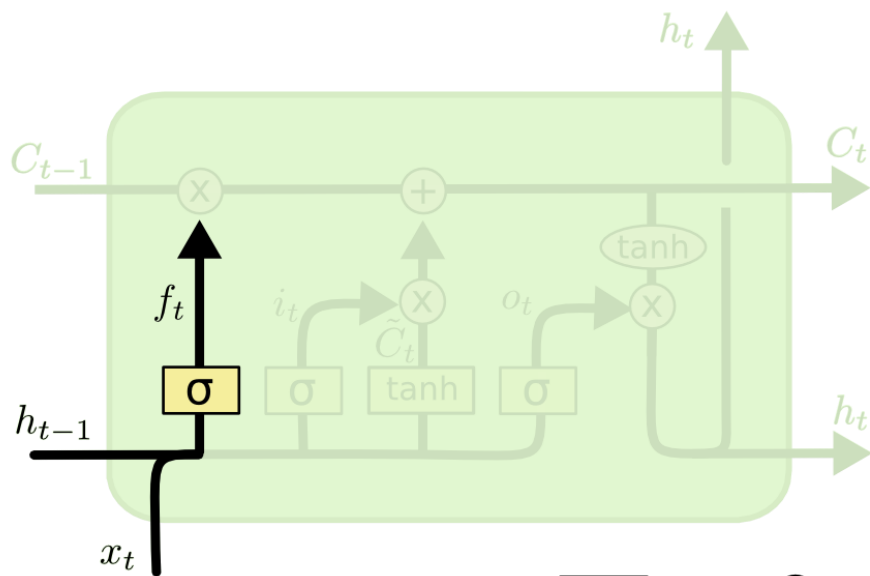
Step by Step으로 살펴보자

# 순환 신경망 확장 모델

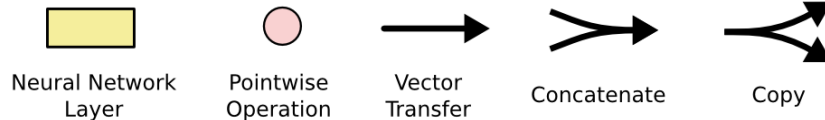
## ❖ LSTM Networks

- Forget gate( $f_t$ ): 잊을 것은 잊자

$[]$  → concatenate  
 $\cdot$  → point wise operation  
 $\sigma$  → sigmoid function



$$f_t = \sigma (W_f \cdot [h_{t-1}, x_t] + b_f)$$



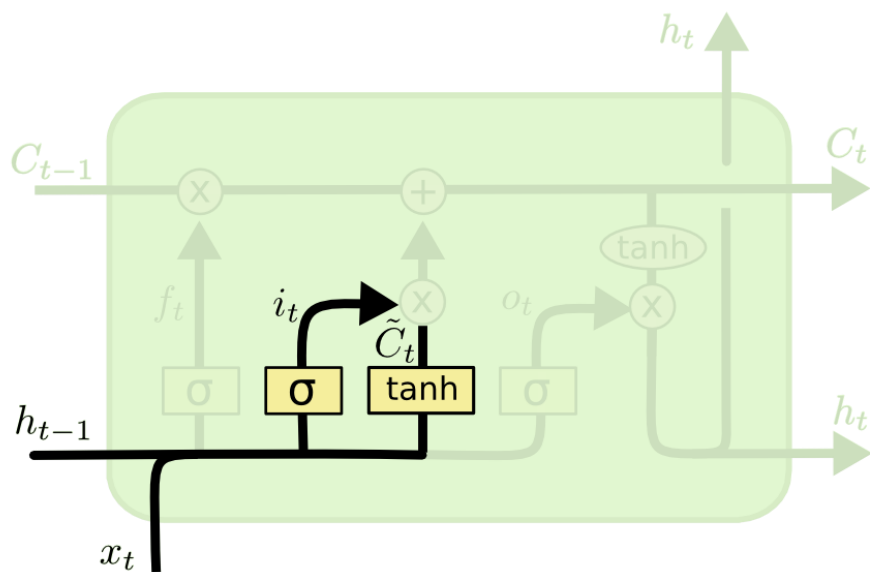
잊을 것은 잊자 STEP

<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

# 순환 신경망 확장 모델

## ❖ LSTM Networks

- Sigmoid layer: Input gate( $i_t$ ) layer: → 어느 값을 업데이트 할지 정해줘!
- Tanh layer: Candidate value( $\tilde{C}_t$ ) → 어느 값 후보 (cell state 후보들)



어떤 새로운 정보가 cell state 안에 들어갈래?

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

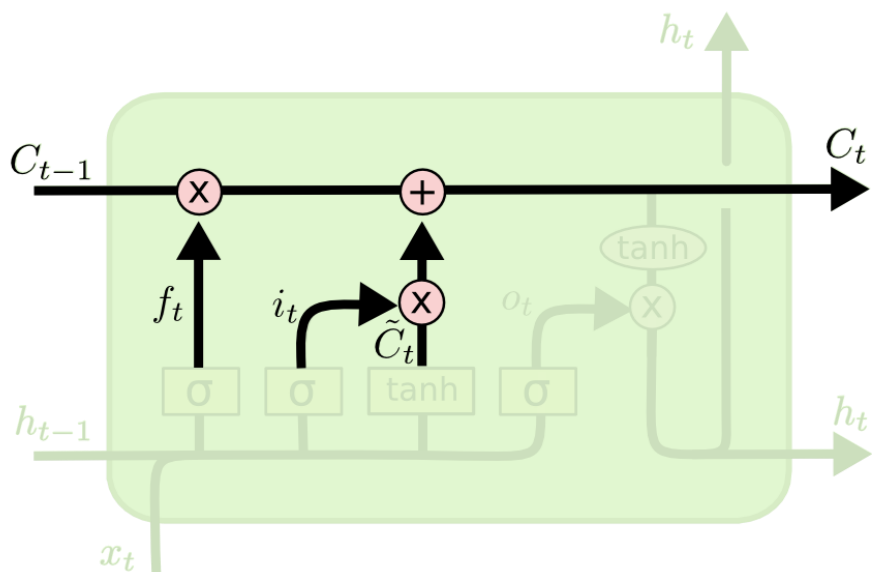
가져갈 것은 가져가자 STEP

<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

# 순환 신경망 확장 모델

## ❖ LSTM Networks 핵심:

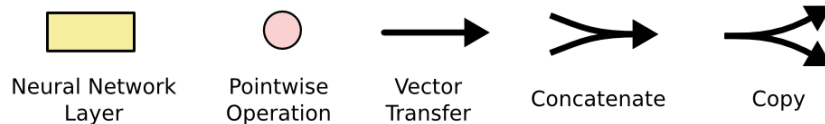
- 잊을 것은 잊고, 가져갈 것은 가져가자. → Cell state 가 반영해주자!



잊을 것은 잊고

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

가져갈 것은 가져가자



**이전 Cell state 업데이트 하자 STEP**

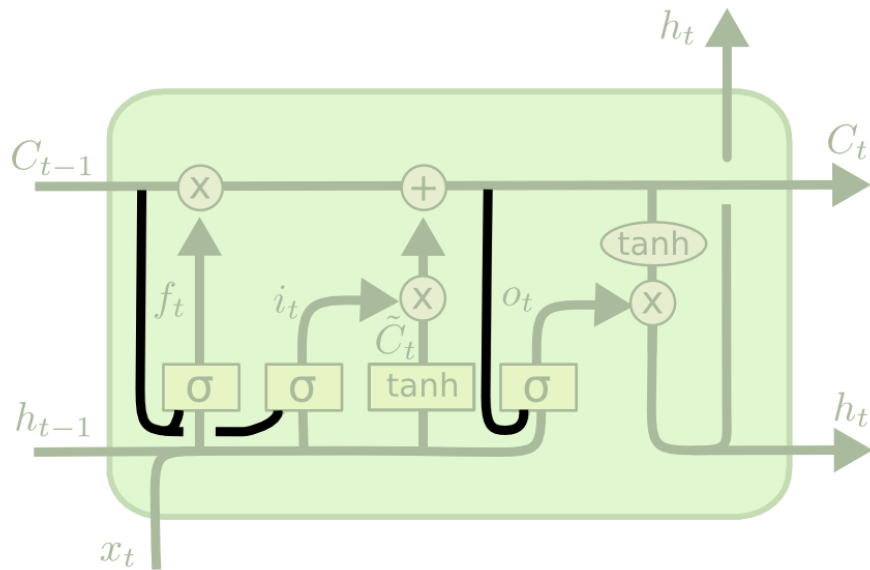
<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>



# LSTM 변형 모델

## ❖ Peephole connections

- $h_t : o_t$ 와  $C_t$ 로부터 잊을 건 잊은 Hidden state  $\rightarrow$  다음 시점으로 전달



$$f_t = \sigma(W_f \cdot [C_{t-1}, h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma(W_i \cdot [C_{t-1}, h_{t-1}, x_t] + b_i)$$

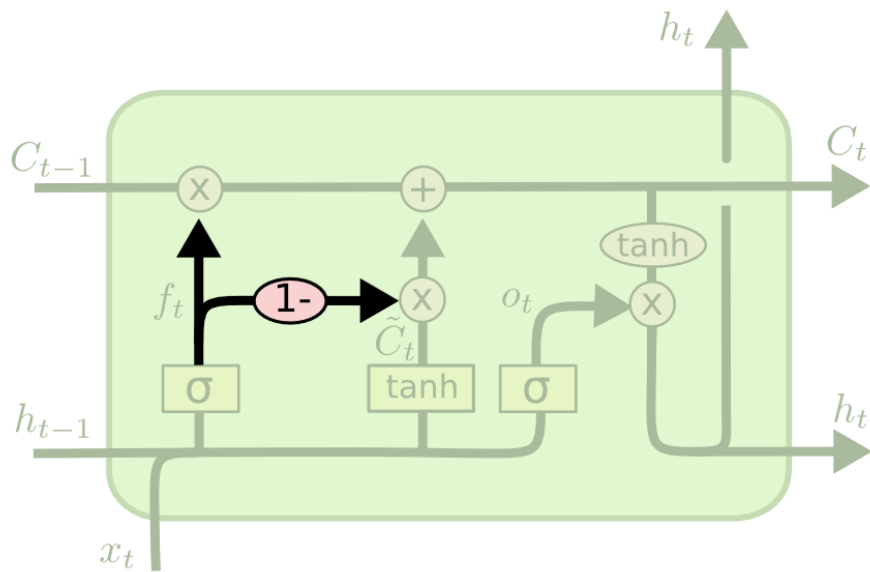
$$o_t = \sigma(W_o \cdot [C_t, h_{t-1}, x_t] + b_o)$$

Long-term dependency(장기의존성) 해결 방안 2

# LSTM 변형 모델

## ❖ Gated Recurrent Unit(GRU)

- Forget gate와 input gate 를 결합



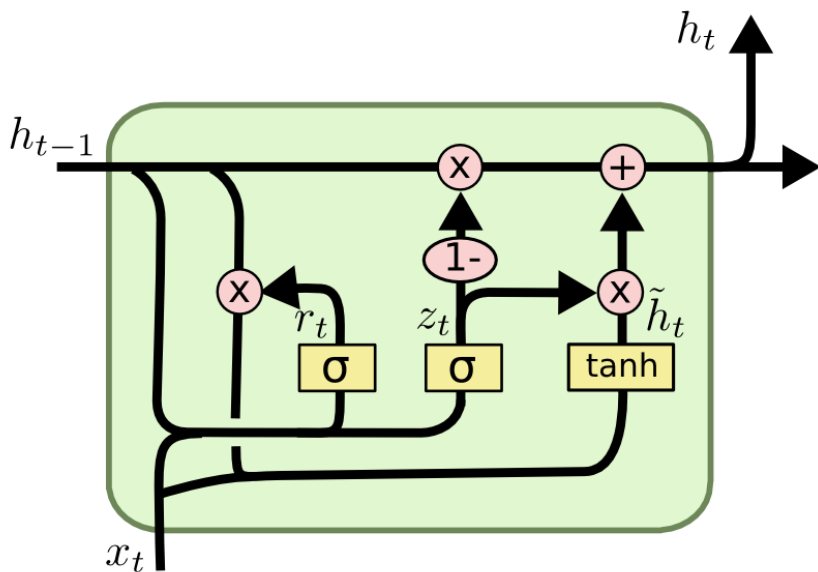
$$C_t = f_t * C_{t-1} + (1 - f_t) * \tilde{C}_t$$

Long-term dependency(장기의존성) 해결 방안 3

# LSTM 변형 모델

## ❖ Depth Gated RNN

- Depth Forget gate와 input gate 를 결합



$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Long-term dependency(장기의존성) 해결 방안 4



# 순환 신경망 모델링

---

## ❖ 모델링 가이드라인

그래서 뭐를 사용하는 것이 좋아?

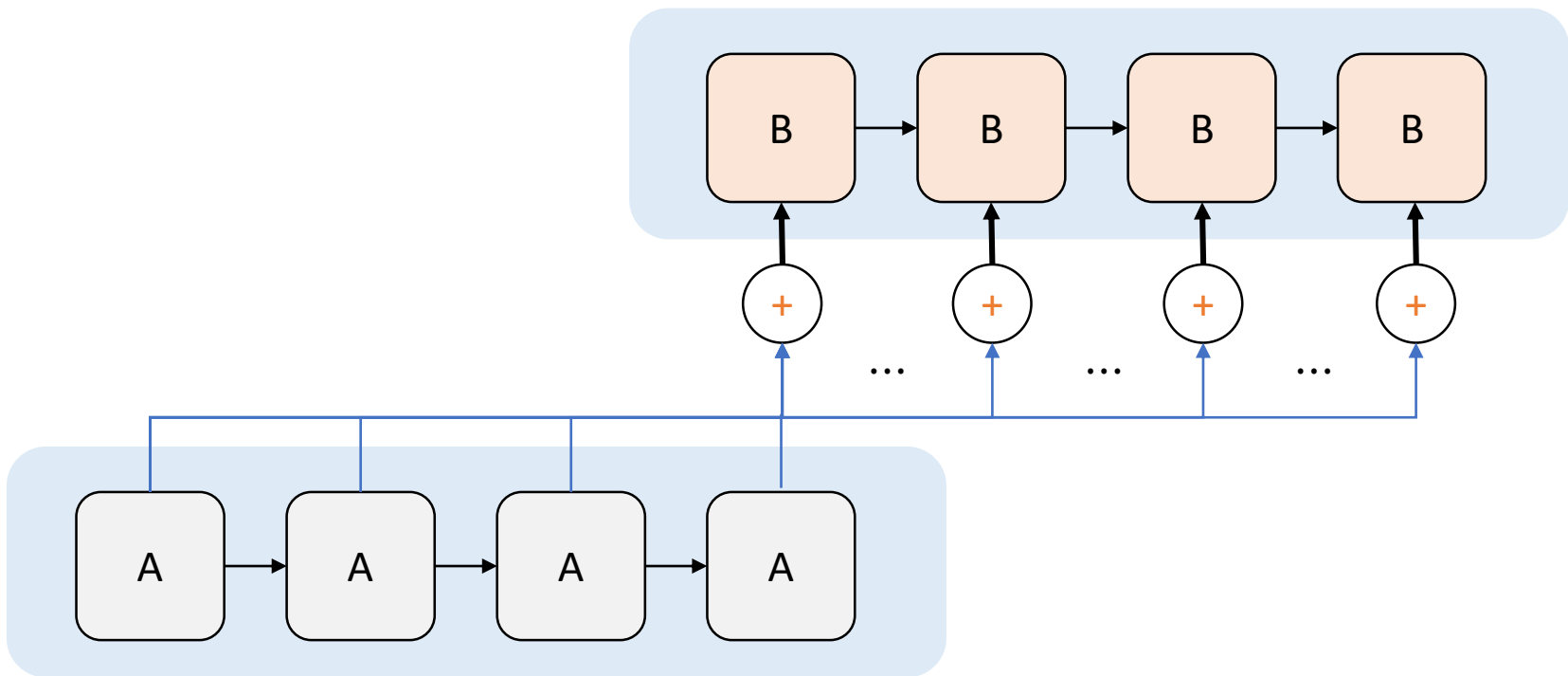
정답은 없는데 가이드라인을 줄게

1. 천리길도 한걸음 부터! 길이가 짧은 시퀀스에 첫번째 시도 → RNN 사용
2. 길이가 길어, 데이터가 많아? → LSTM 사용
3. 길이가 길어, 데이터가 많아, 시간이 없어? → GRU 사용
4. 나머지는 경험상 논외, 주목해 볼 테크닉은 **Attention!**

# 순환 신경망 모델링

## ❖ Seq2Seq (Many to Many RNN) with attention

- Network B는 Network A 매 시점 정보를 모두 참조

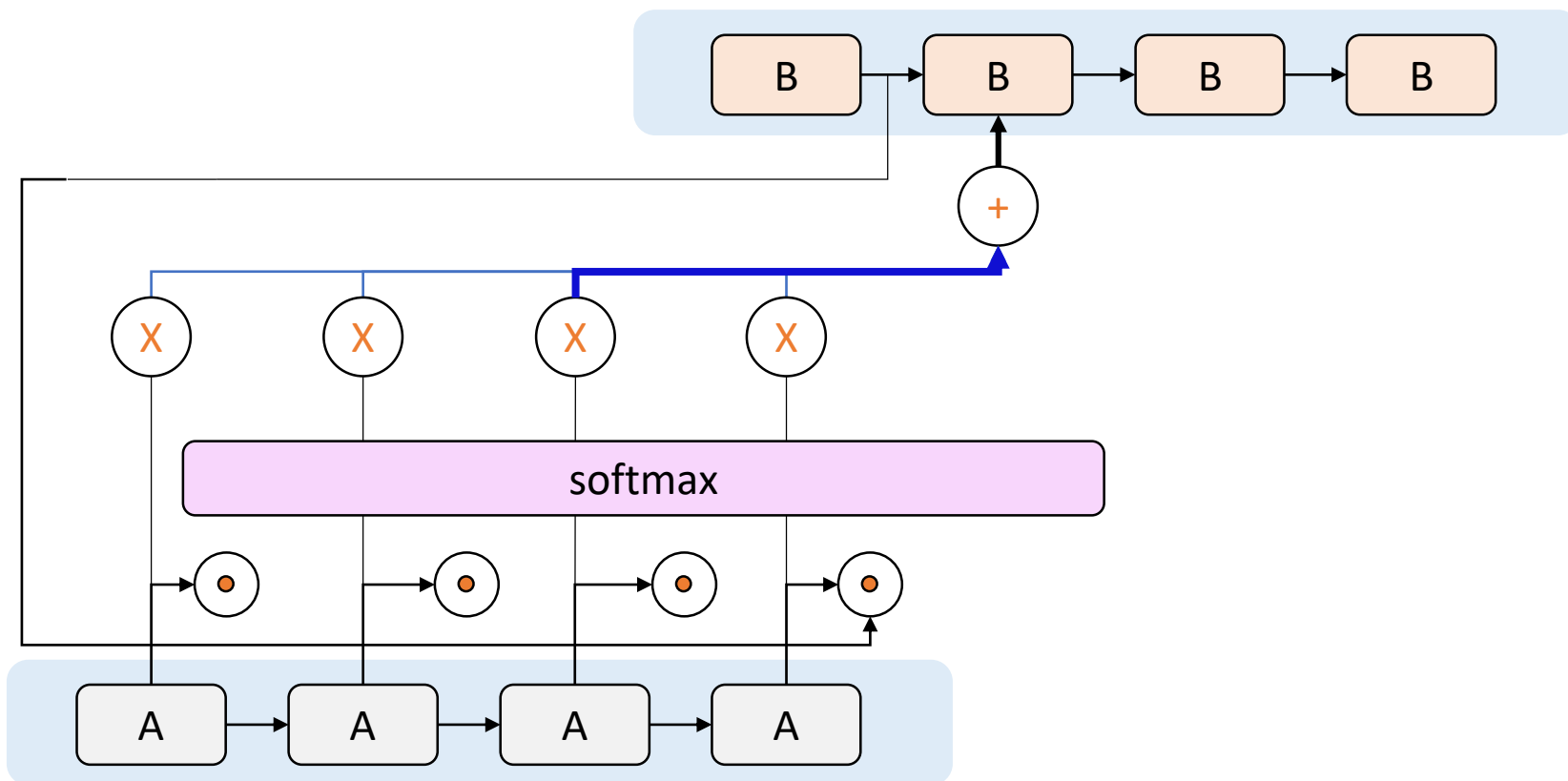


<https://distill.pub/2016/augmented-rnns/#attentional-interfaces>

# 순환 신경망 모델링

## ❖ Seq2Seq (Many to Many RNN) with attention

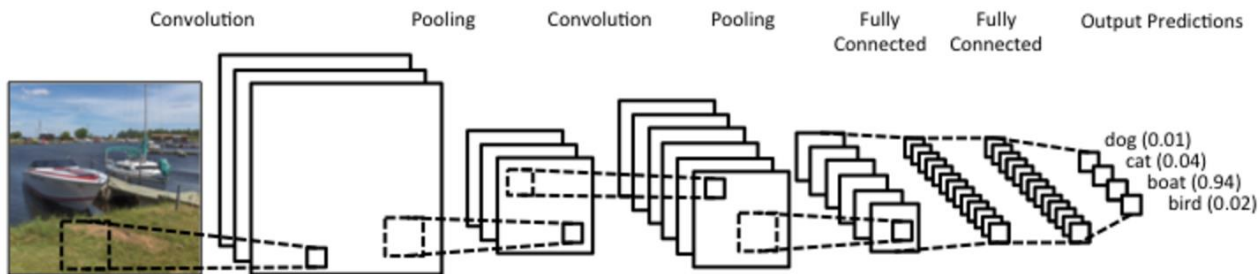
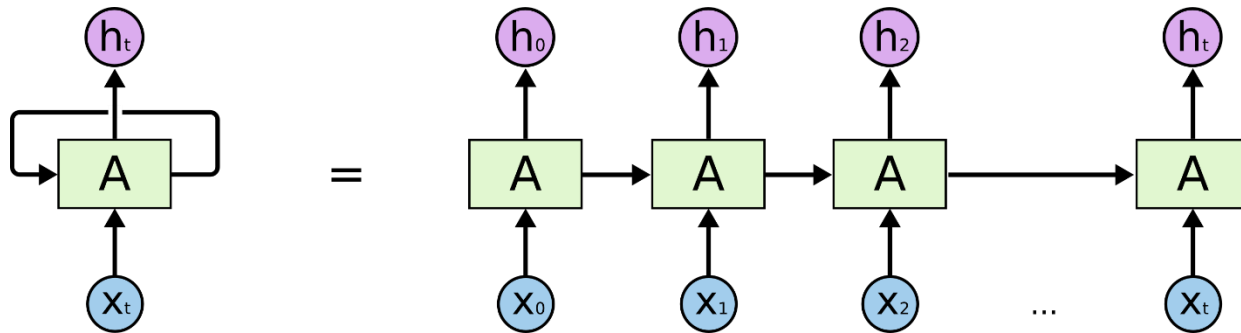
- Network B는 Network A 매 시점 정보를 모두 참조



<https://distill.pub/2016/augmented-rnns/#attentional-interfaces>

# 딥러닝 모델의 유연함

- ❖ Unit 간의 연결 구성에 따라 다양한 모델 구성이 가능
- ❖ RNN, CNN...



# 이형의 데이터를 이용한 모델링

- ❖ 현실의 많은 문제들은 이형의 데이터를 모델링 해야 함
- ❖ 딥러닝 이전에는 이형의 데이터로부터 개별적으로 변수를 추출하여 수행함 (feature extraction)
- ❖ 인공지능망 모델을 통해 단일 모델로 이형의 데이터를 모델링 수행

Task	X	Y
Image Captioning	이미지	텍스트
VQA	이미지 + 텍스트	텍스트
Video + Speech Recognition	Video + Sound	Class
⋮	⋮	⋮

# Image Captioning

- ❖ 주어진 사진을 자연어 (텍스트)로 묘사할 수 있는 모델



"man in black shirt is playing guitar."



"construction worker in orange safety vest is working on road."



"two young girls are playing with legos toy."



"boy is doing backflip on wakeboard."



"girl in pink dress is jumping in air."



"black and white dog jumps over bar."



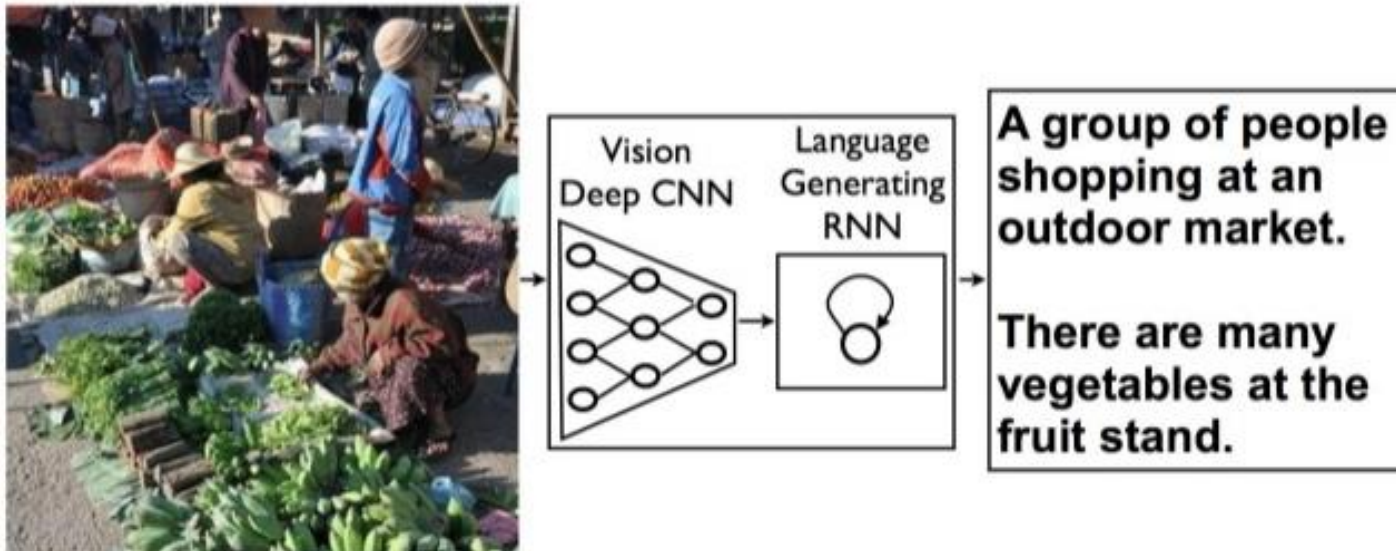
"young girl in pink shirt is swinging on swing."



"man in blue wetsuit is surfing on wave."

# Image Captioning

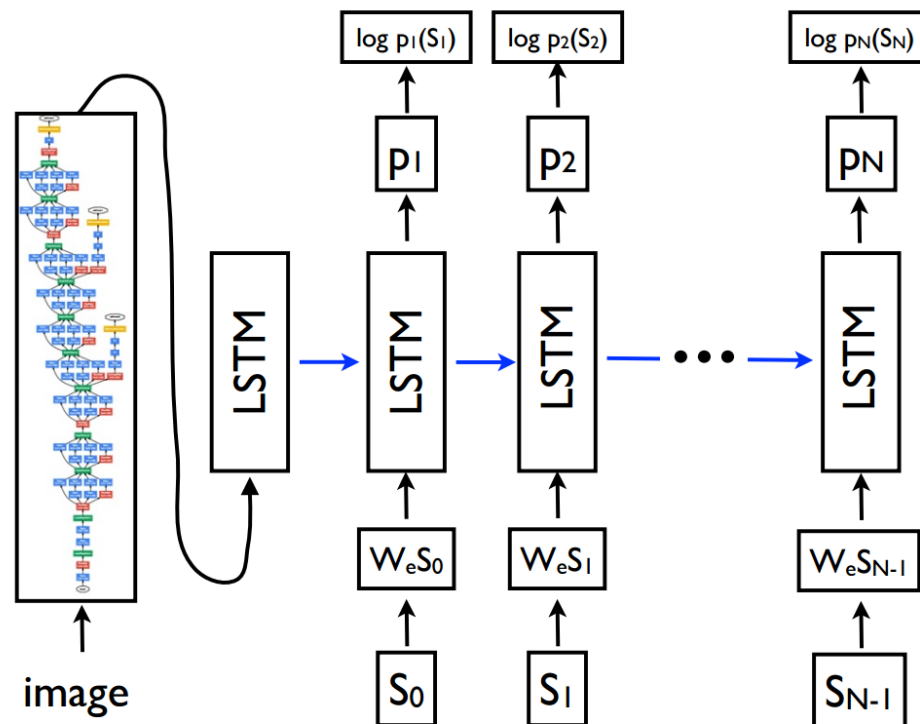
- ❖ 주어진 사진을 자연어 (텍스트)로 묘사할 수 있는 모델
- ❖ 이미지 학습을 위한 CNN과 텍스트 생성을 위한 RNN 이용



Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3156-3164).

# Image Captioning

- ❖ 주어진 사진을 자연어 (텍스트)로 묘사할 수 있는 모델
- ❖ 이미지 학습을 위한 CNN과 텍스트 생성을 위한 RNN 이용



Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2015). Show and tell: A neural image caption generator. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3156-3164).



# Image Captioning

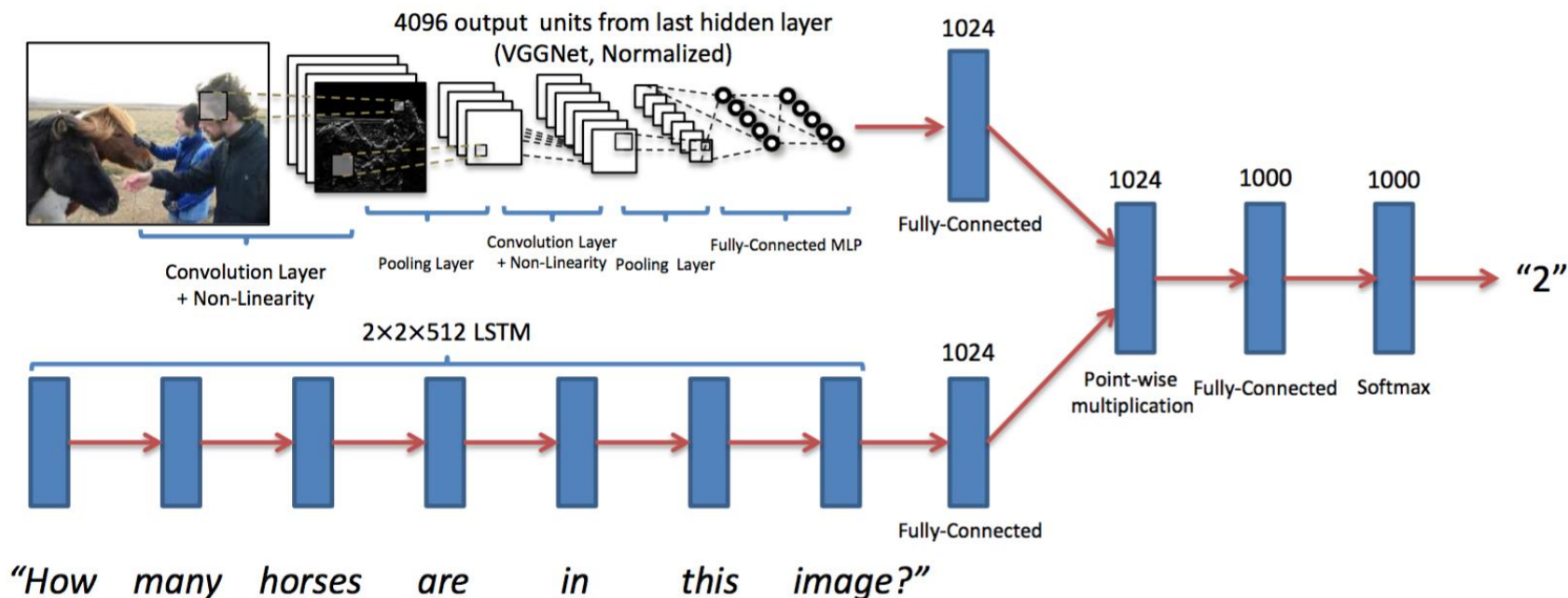
- ❖ 주어진 사진을 자연어 (텍스트)로 묘사할 수 있는 모델
- ❖ 이미지 학습을 위한 CNN과 텍스트 생성을 위한 RNN 이용



Figure 5. A selection of evaluation results, grouped by human rating.

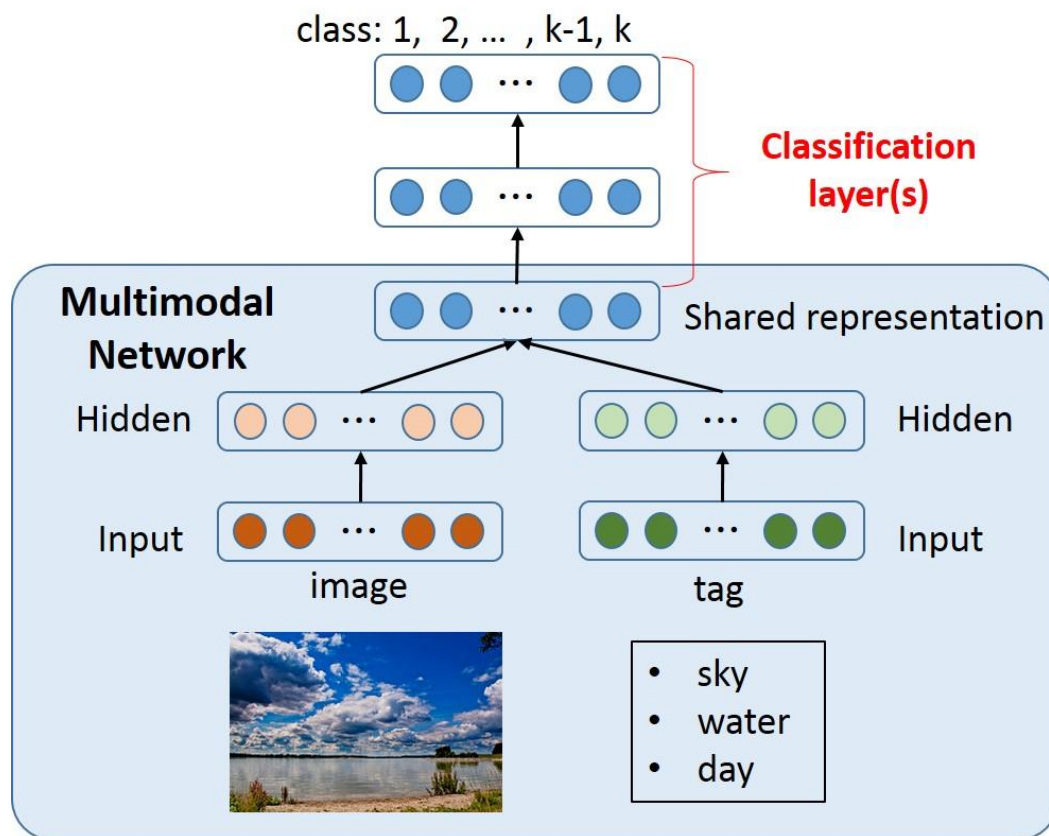
# Visual Question Answering (VQA)

- ❖ 이형의 데이터 (이미지, 텍스트)가 입력인 경우 multimodel 구조라 함
- ❖ 이미지 학습: CNN (VGGNet-19)
- ❖ 텍스트 학습: RNN (LSTM)



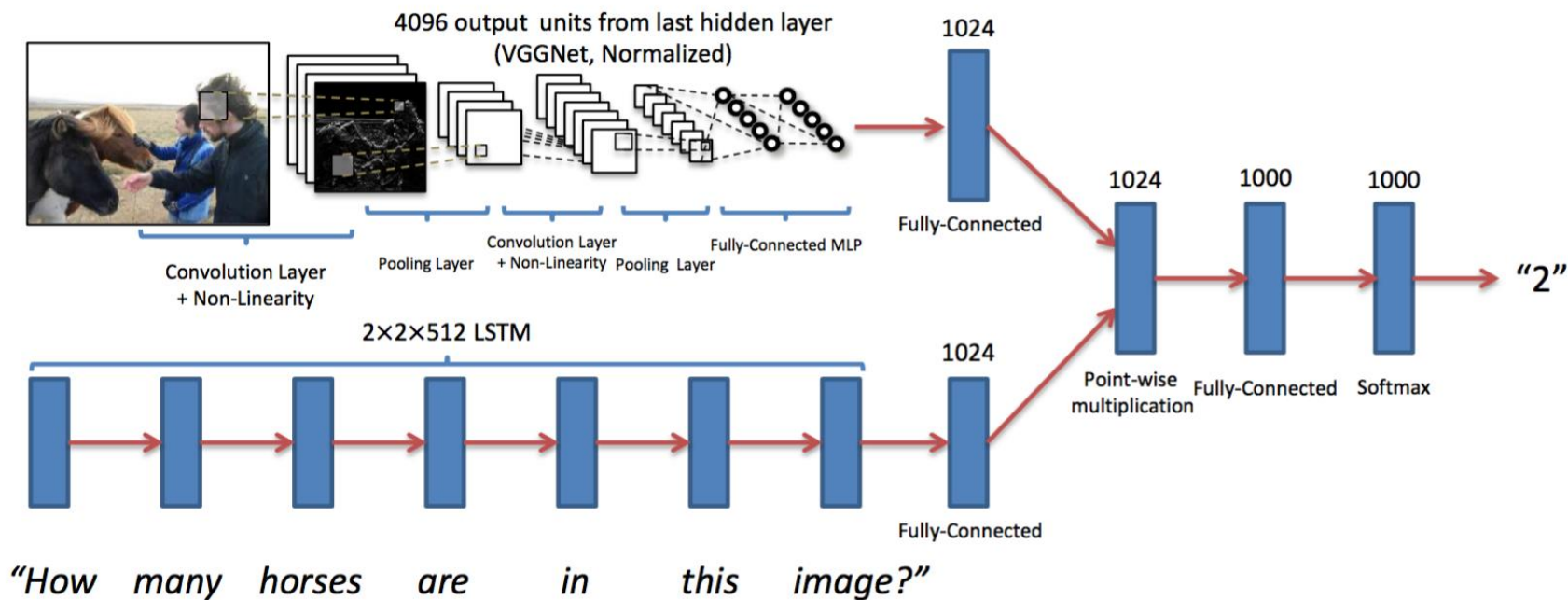
# Multimodal Neural Network

❖ 입력  $x$  가 이형의 데이터로 구성된 경우



# Multimodal Neural Network

- ❖ 이미지 학습: CNN (VGGNet-19)
- ❖ 텍스트 학습: RNN (LSTM) + **word2vec**



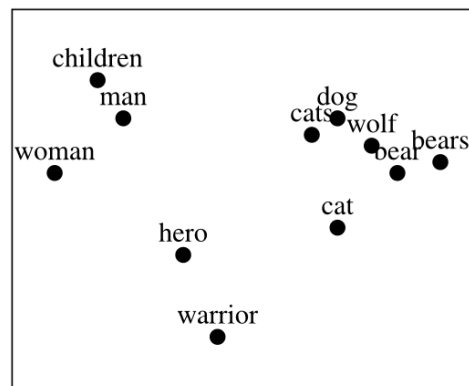
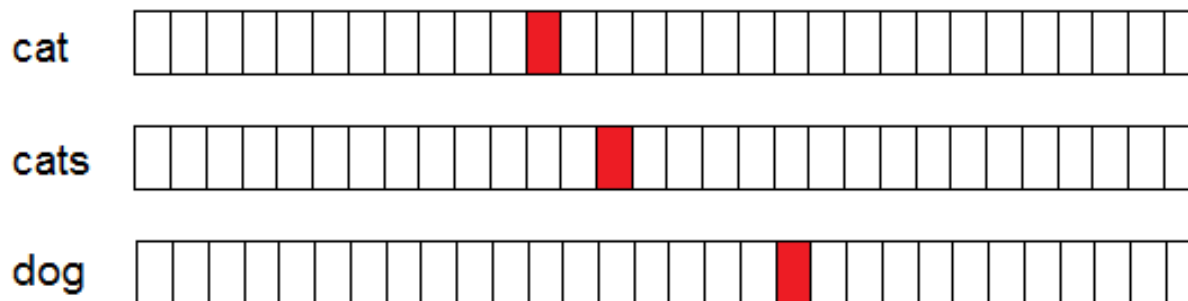
# Word Embedding (Word2Vec)

- ❖ 단어를 벡터로 바꾸는 방법
- ❖ 기존: One-hot vector
- ❖ 새로운 방법: Word2Vec

cat  $\rightarrow (1.4, 0)$

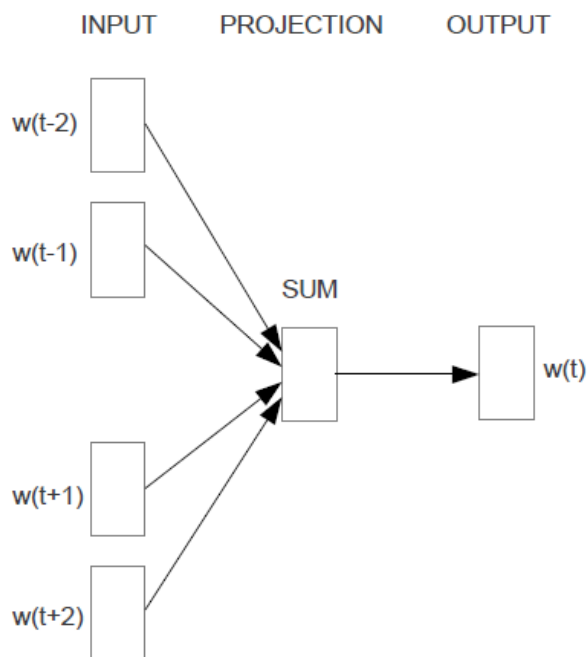
cats  $\rightarrow (1.4, 0.8)$

dog  $\rightarrow (1.4, 1)$

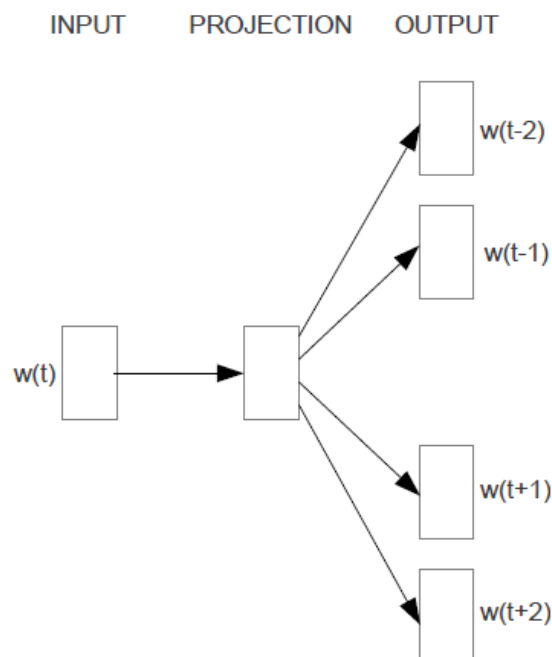


# Word Embedding (Word2Vec)

- ❖ 모델 구성 방식에 따라 두 가지 방식
- ❖ CBOW 선호



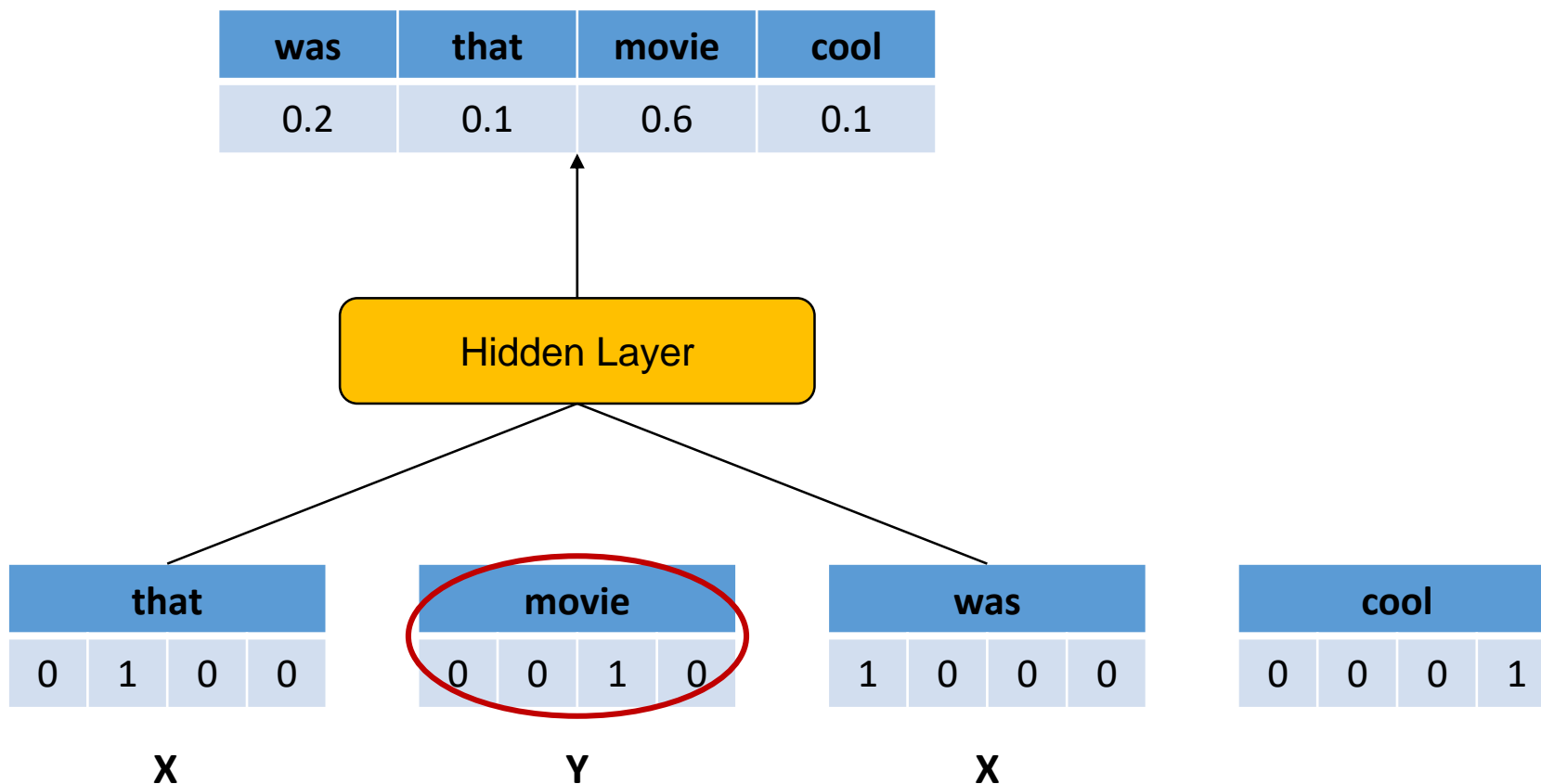
**CBOW**



**Skip-gram**

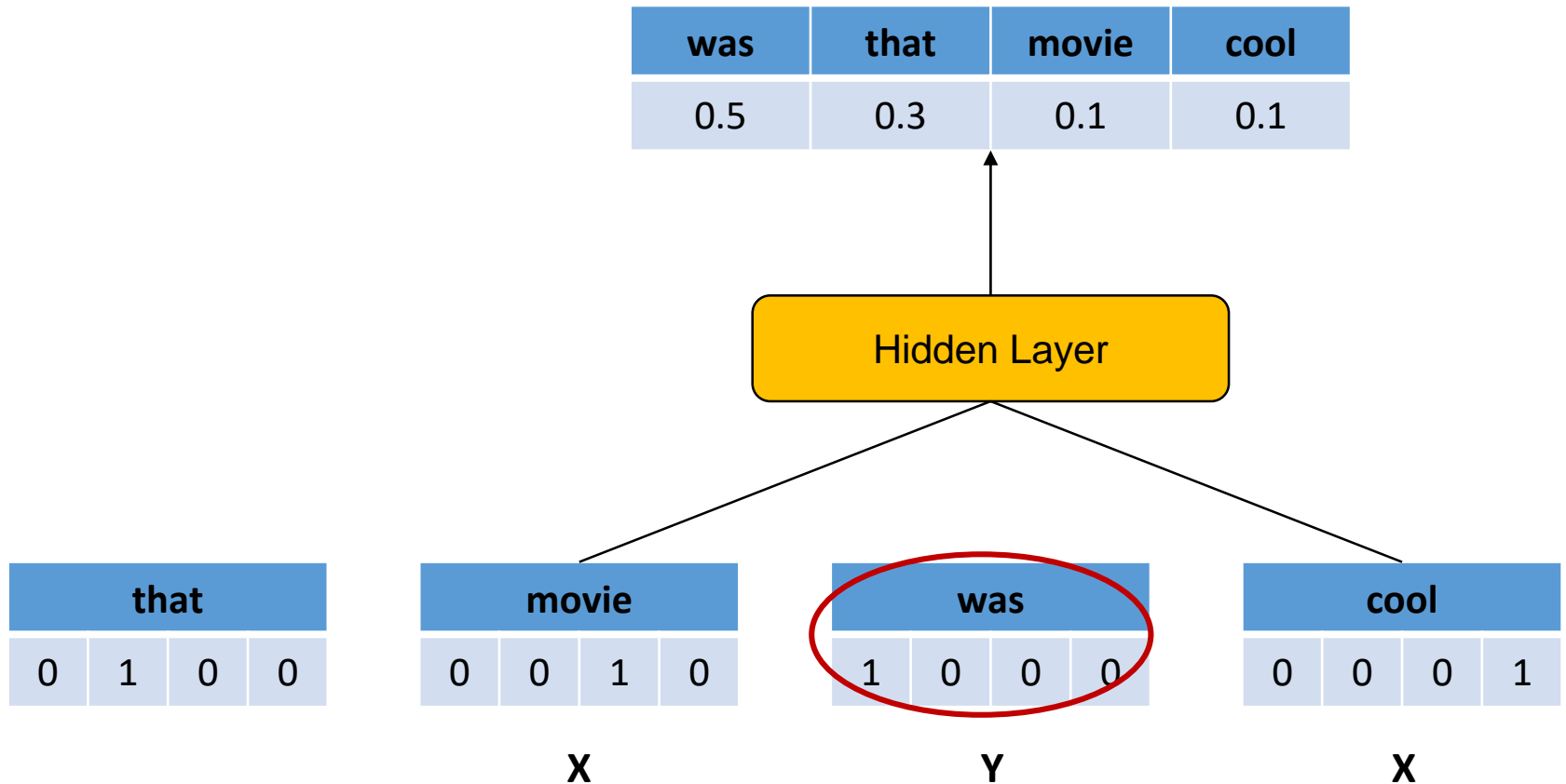
# Word Embedding (Word2Vec)

- ❖ 임의의 단어를 주변 단어로 예측하는 신경망 모델 구성



# Word Embedding (Word2Vec)

- ❖ 임의의 단어를 주변 단어로 예측하는 신경망 모델 구성

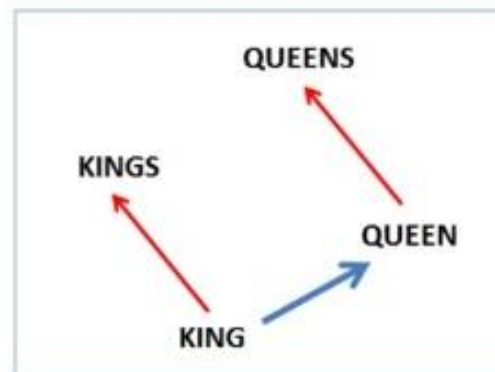
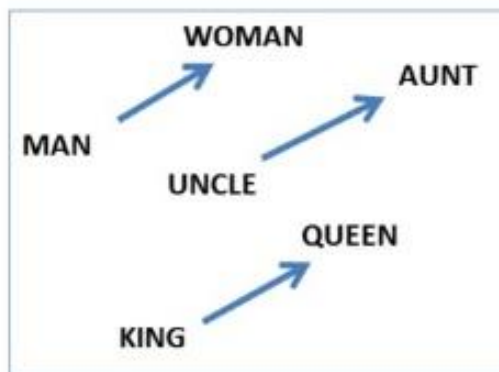




# Word Embedding (Word2Vec)

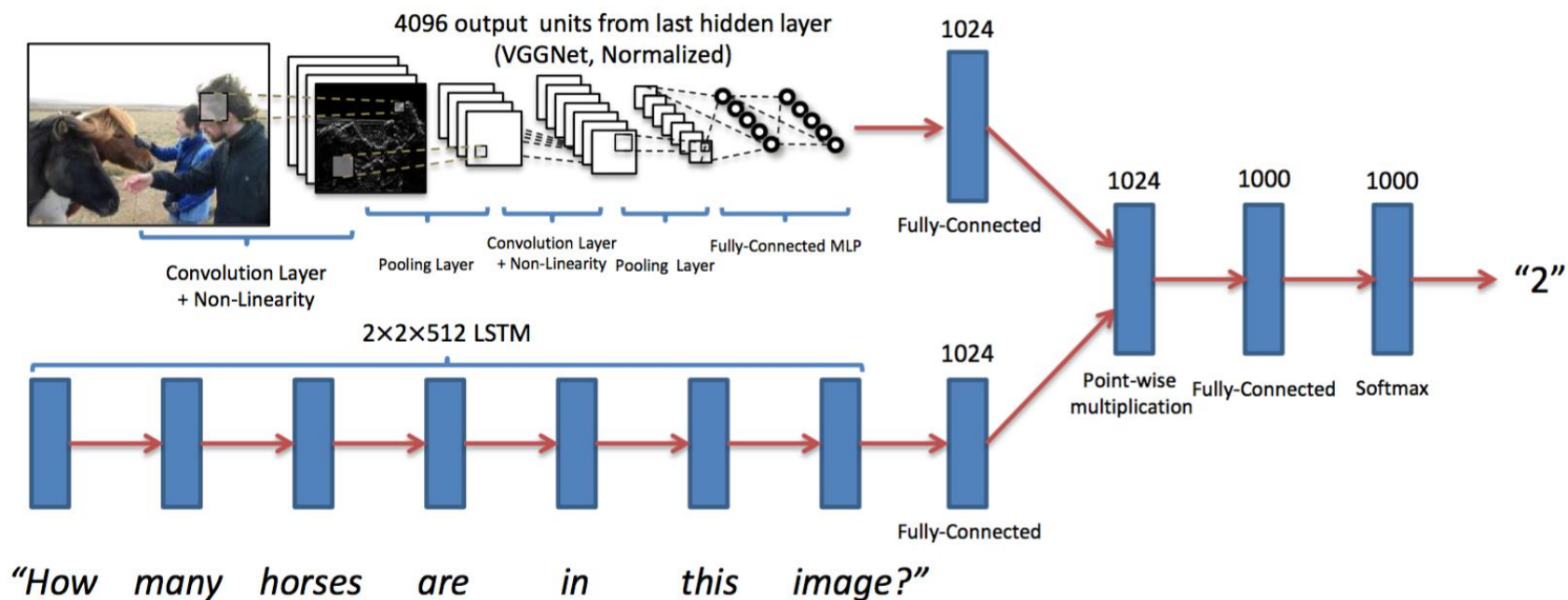
- ❖ Word2Vec으로 얻은 단어 벡터를 이용하여 단어간 연산 수행 가능
- ❖ 저차원 벡터가 단어의 의미를 반영하여 학습되었음을 의미

$$\text{vec}(\text{"man"}) - \text{vec}(\text{"king"}) + \text{vec}(\text{"woman"}) = \text{vec}(\text{"queen"})$$



# Word Embedding (Word2Vec)

- ❖ 이미지 학습: CNN (VGGNet-19)
- ❖ 텍스트 학습: RNN (LSTM) + **word2vec**
- ❖ 성능평가는 <http://cloudcv.org/vqa/> 에서



---

# EOD