Multiple Data Types (2주차)

Young-Gon Kim
DLI Instructor

DEEP LEARNING INSTITUTE

DLI Mission

Helping people solve challenging problems using AI and deep learning.

- Developers, data scientists and engineers
- Self-driving cars, healthcare and robotics
- Training, optimizing, and deploying deep neural networks

TOPICS

- Week 1 Review
- Recurrent Neural Network

WEEK 1 REVIEW

What’s the problem in Week 1?

RECURRENT NEURAL NETWORK (RNN)

RECURRENT NEURAL NETWORK (RNN)

- Architecture

We can process a sequence of vectors **x** by applying a **recurrence formula** at every time step:

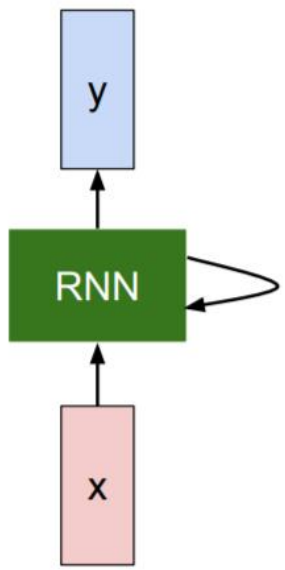
$$h_t = f_W(h_{t-1}, x_t)$$

new state

old state

input vector at some time step

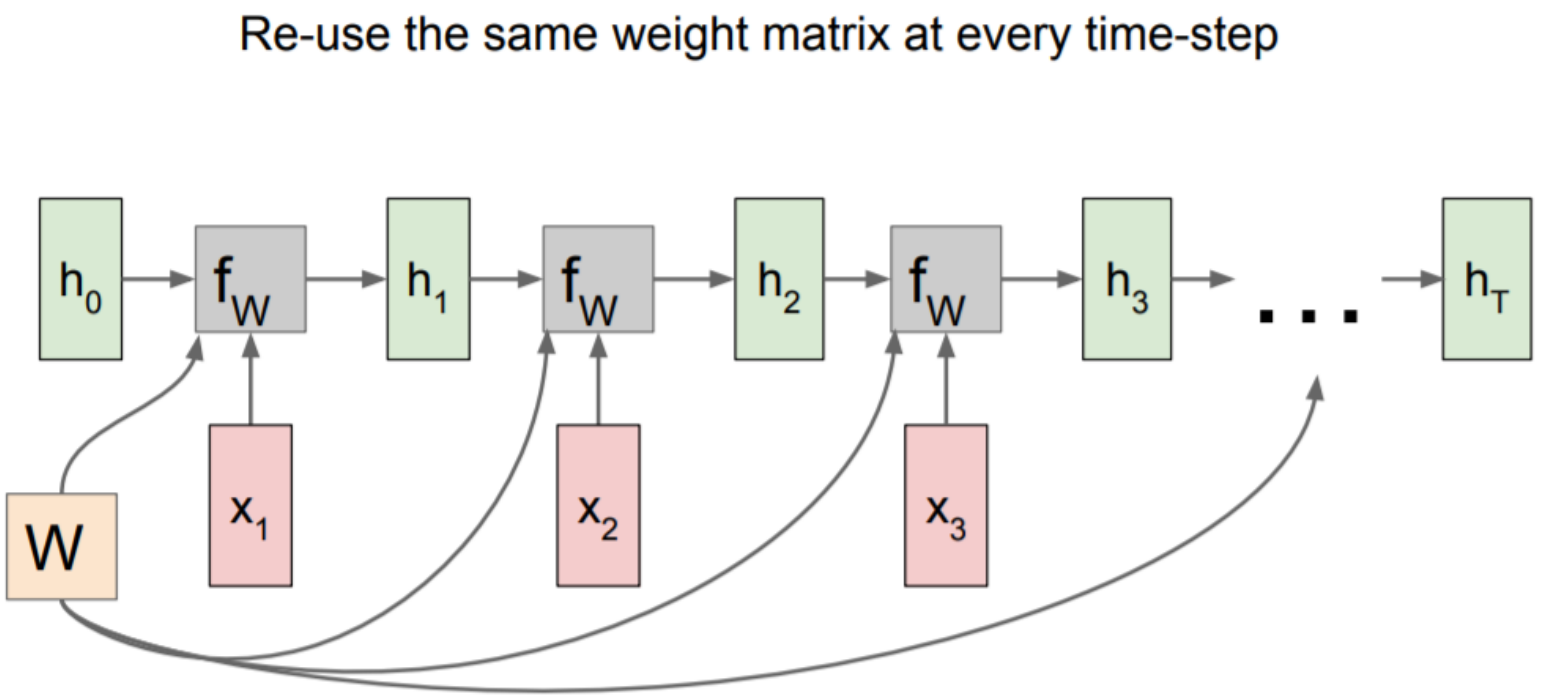
some function with parameters W



http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture10.pdf

RECURRENT NEURAL NETWORK (RNN)

- Architecture

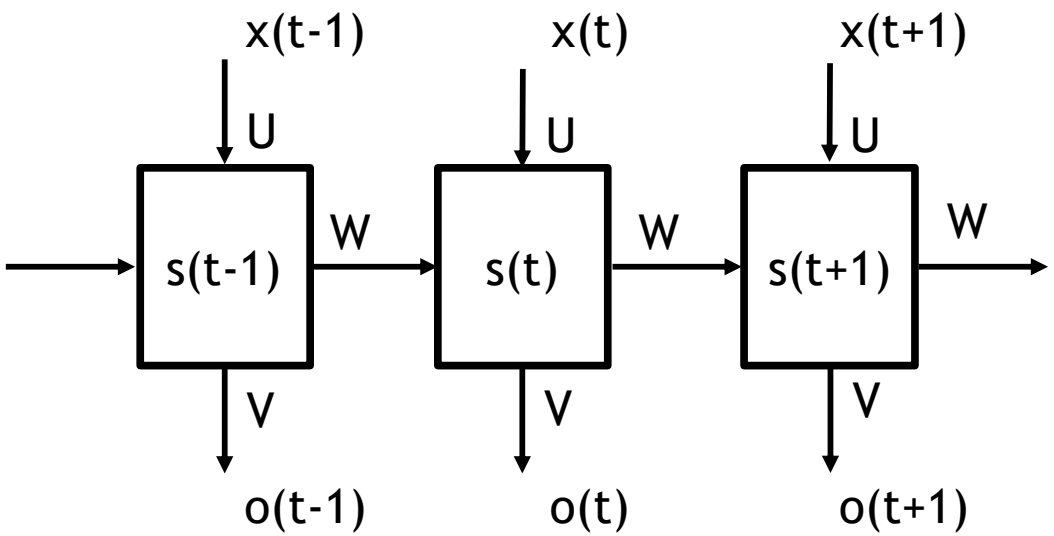


http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture10.pdf

RECURRENT NEURAL NETWORK (RNN)

- Architecture

a	the	on	is	cat	park	play	swing	grass	sitting
0	1	2	3	4	5	6	7	8	9



Unrolled Recurrent Layer

[0 , 4 , 3 , 2 , 1 , 8]

A cat is on the grass.

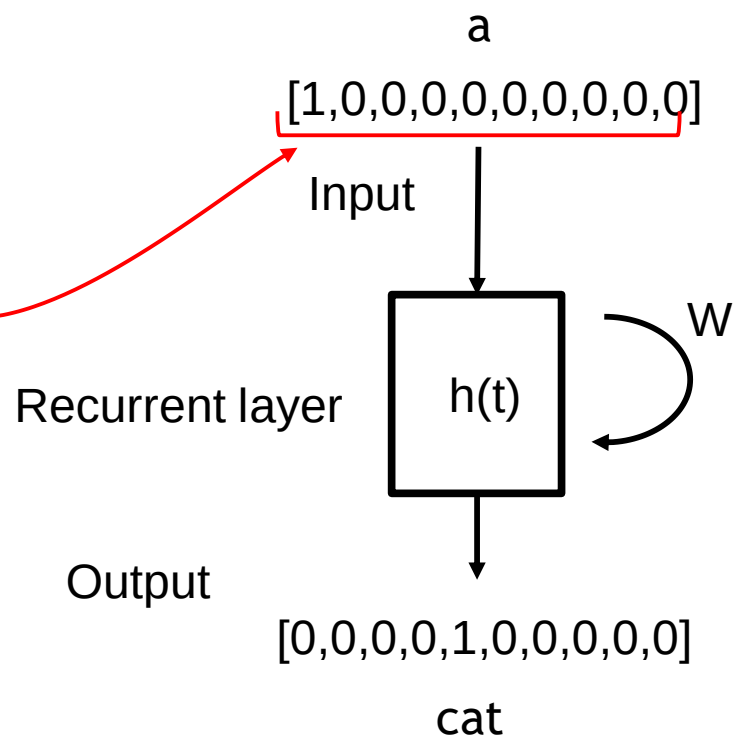
RNNs learn by reducing the error between their predicted next word and the actual next word in a corpus. RNNs are structured to "remember" the words that led to their prediction.

RECURRENT NEURAL NETWORK (RNN)

- Architecture
  - **One-hot encoding**

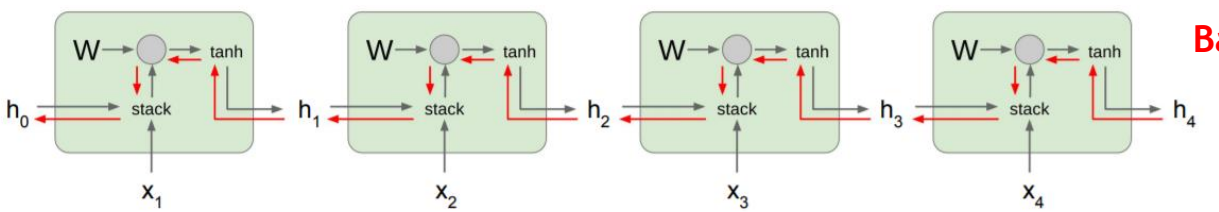
Word prediction example

a	1	0	0	0	0	0	0	0	0	0
cat	0	0	0	0	1	0	0	0	0	0
is	0	0	0	1	0	0	0	0	0	0
on	0	0	1	0	0	0	0	0	0	0
the	0	1	0	0	0	0	0	0	0	0
grass	0	0	0	0	0	0	0	0	1	0



RECURRENT NEURAL NETWORK (RNN)

- Architecture
  - **Vanilla RNN**
    - Input과 hidden state에 matrix연산 진행
    - **Vanishing gradient**



Computing gradient of h_0 involves many factors of W (and repeated tanh)

Largest singular value > 1:
Exploding gradients

Largest singular value < 1:
Vanishing gradients → Change RNN architecture

Backpropagation from h_t to h_{t-1} multiplies by W (actually W_{hh}^T)

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$
$$= \tanh\left((W_{hh} \quad W_{hx})\begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right)$$
$$= \tanh\left(W\begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right)$$

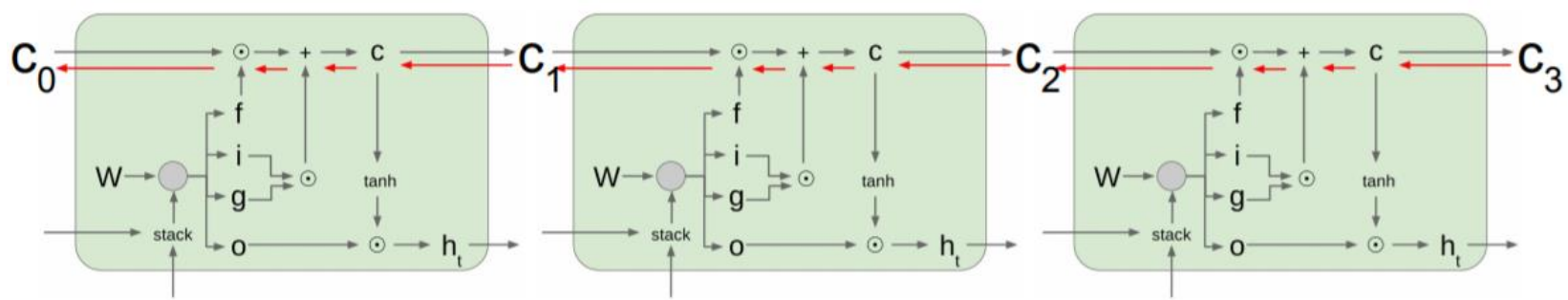
http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture10.pdf

RECURRENT NEURAL NETWORK (RNN)

- Architecture
  - **Long Short Term Memory (LSTM)**
    - **Cell state**를 통해 RNN의 vanishing gradient 해결
    - forget gate, input gate로 성능 향상

Uninterrupted gradient flow!

Backpropagation from c_t to c_{t-1} only elementwise multiplication by f (forget gate), no matrix multiply by W



$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

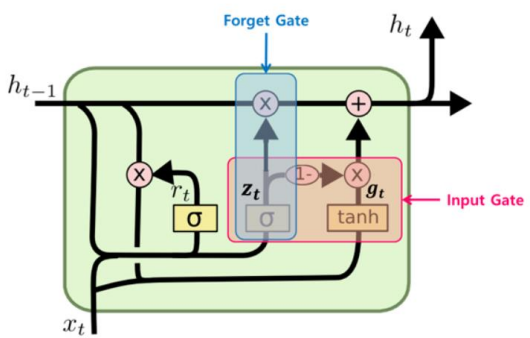
$$c_t = f \odot c_{t-1} + i \odot g$$

$$h_t = o \odot \tanh(c_t)$$

http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture10.pdf
https://ratsgo.github.io/natural%20language%20processing/2017/03/09/rnnlstm/

RECURRENT NEURAL NETWORK (RNN)

- Architecture
  - **Gated Recurrent Units (GRU)**
    - LSTM의 간소화 버전
    - c_t와 h_t를 하나의 벡터 h_t로 합침
    - 하나의 gate controller z_t가 forget, input gate 모두 제어
    - Output gate가 없기 때문에 h_t가 타임 스텝마다 출력

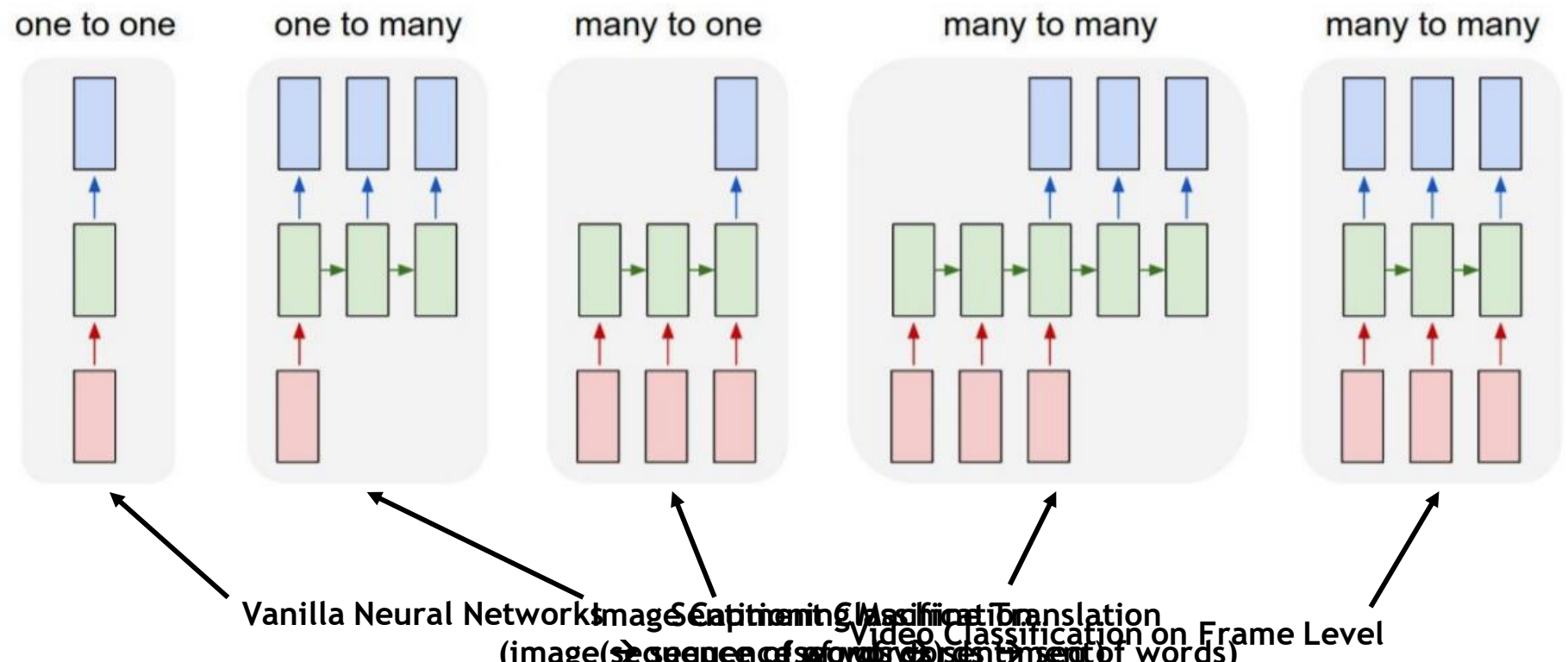


$$\mathbf{r}_t = \sigma\left(\mathbf{W}_{xr}^T \cdot \mathbf{x}_t + \mathbf{W}_{hr}^T \cdot \mathbf{h}_{t-1} + \mathbf{b}_r\right)$$
$$\mathbf{z}_t = \sigma\left(\mathbf{W}_{xz}^T \cdot \mathbf{x}_t + \mathbf{W}_{hz}^T \cdot \mathbf{h}_{t-1} + \mathbf{b}_z\right)$$
$$\mathbf{g}_t = \tanh\left(\mathbf{W}_{xg}^T \cdot \mathbf{x}_t + \mathbf{W}_{hg}^T \cdot (\mathbf{r}_t \otimes \mathbf{h}_{t-1}) + \mathbf{b}_g\right)$$
$$\mathbf{h}_t = \mathbf{z}_t \otimes \mathbf{h}_{t-1} + (1 - \mathbf{z}_t) \otimes \mathbf{g}_t$$

https://excelsior-cjh.tistory.com/185

RECURRENT NEURAL NETWORK (RNN)

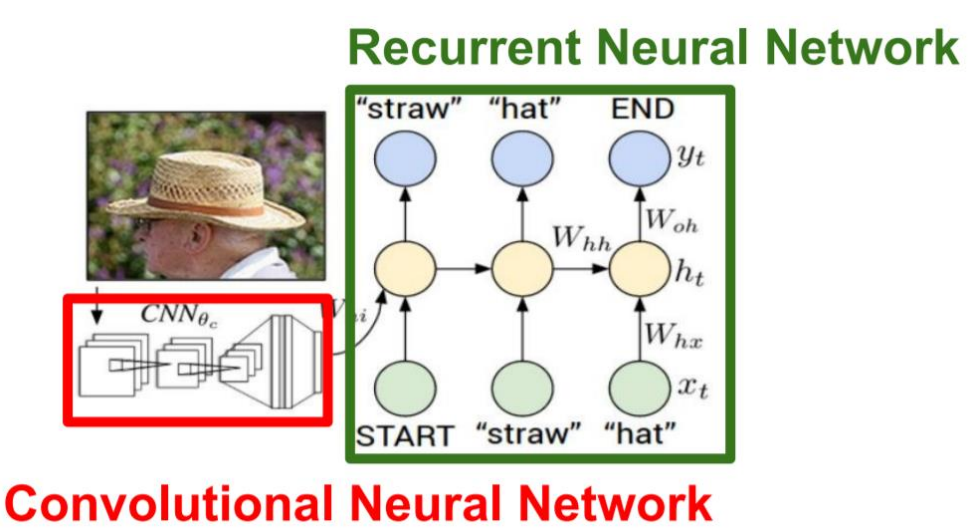
- Application



http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture10.pdf

RECURRENT NEURAL NETWORK (RNN)

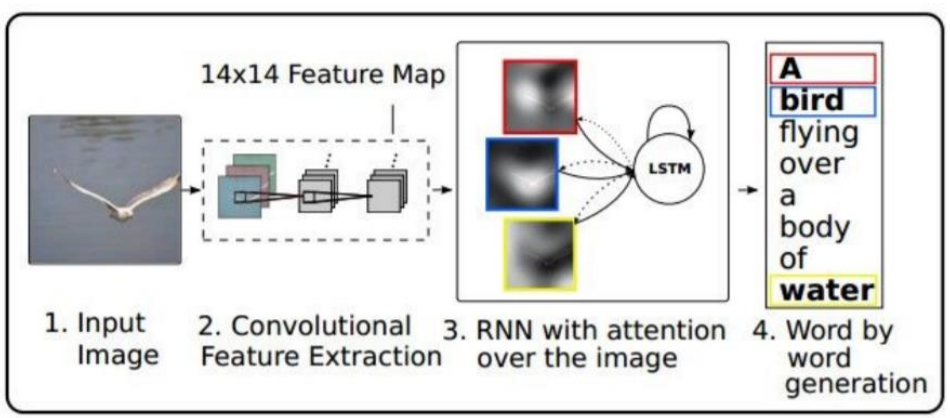
- Application
  - **Image Captioning**
    - 해당 Image를 설명하는 언어로 변환

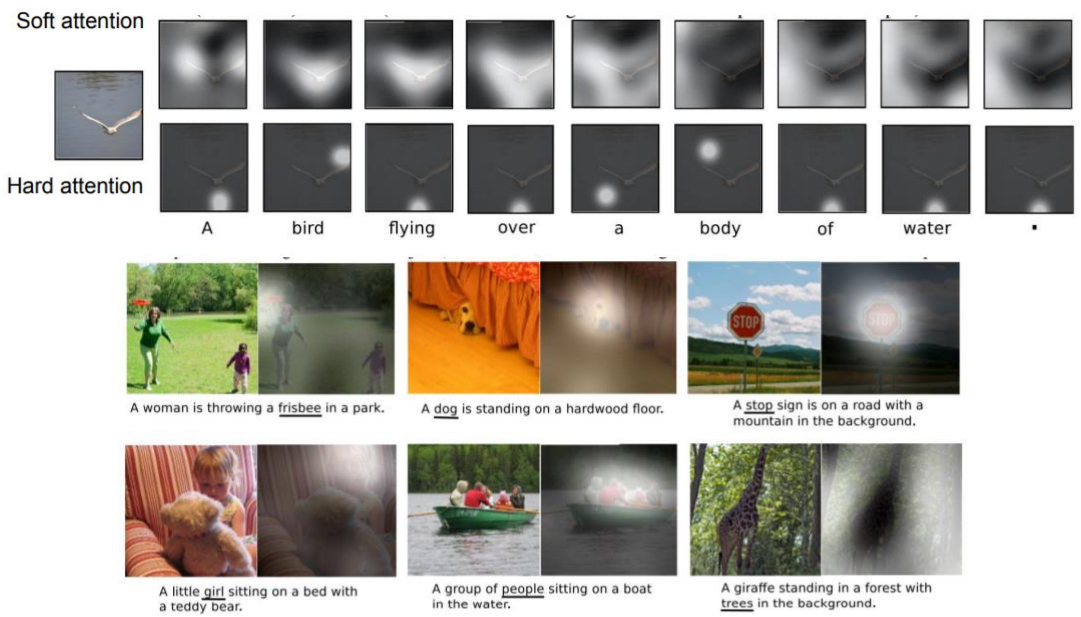


http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture10.pdf

RECURRENT NEURAL NETWORK (RNN)

- Application
  - **Image Captioning**
    - with Attention





http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture10.pdf

RECURRENT NEURAL NETWORK (RNN)

- Application
  - **Visual Question Answering**

Q: What endangered animal is featured on the truck?

A: A bald eagle.
A: A sparrow.
A: A humming bird.
A: A raven.

Q: Where will the driver go if turning right?

A: Onto 24 ¾ Rd.
A: Onto 25 ¾ Rd.
A: Onto 23 ¾ Rd.
A: Onto Main Street.

Q: When was the picture taken?

A: During a wedding.
A: During a bar mitzvah.
A: During a funeral.
A: During a Sunday church

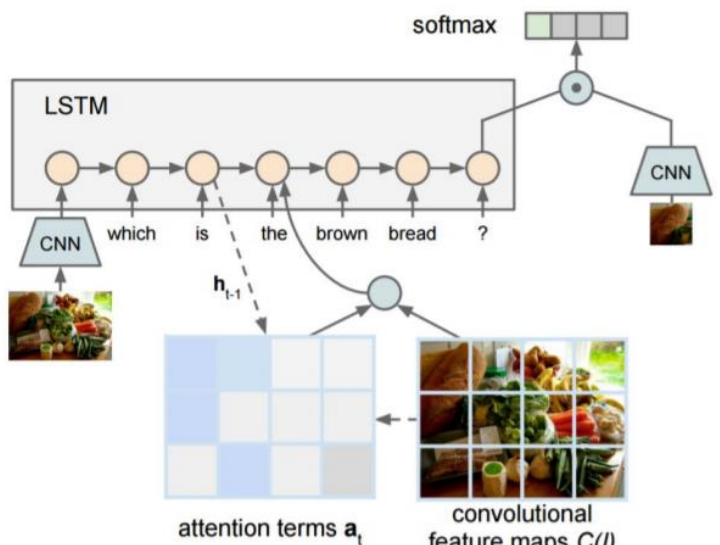
Q: Who is under the umbrella?

A: Two women.
A: A child.
A: An old man.
A: A husband and a wife.

http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture10.pdf

RECURRENT NEURAL NETWORK (RNN)

- Application
  - **Visual Question Answering**
    - RNNs with Attention



Zhu et al, "Visual 7W: Grounded Question Answering in Images", CVPR 2016
Figures from Zhu et al, copyright IEEE 2016. Reproduced for educational purposes.

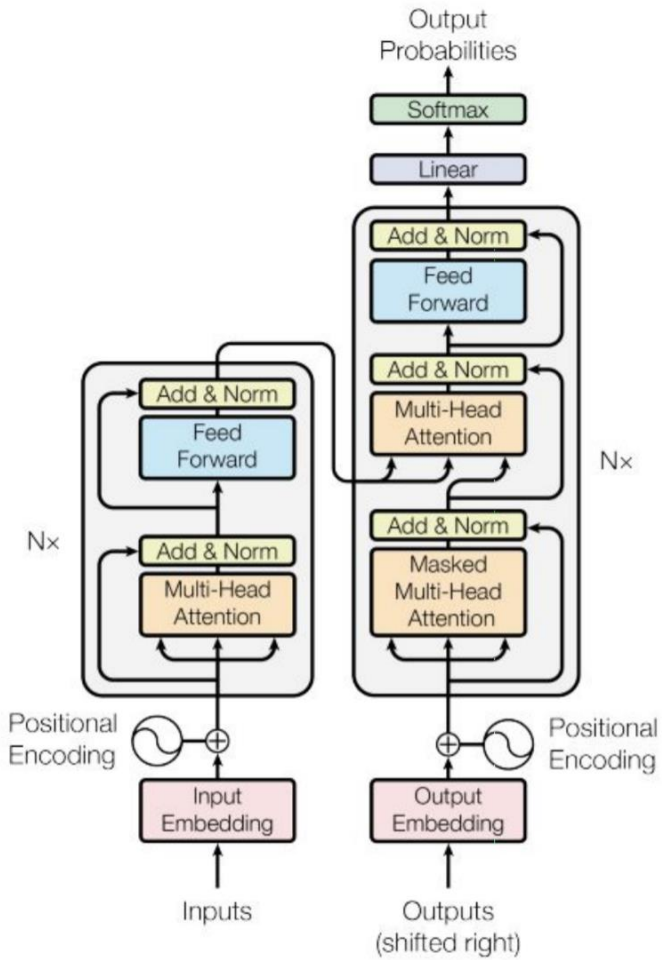
What kind of animal is in the photo?
A cat.

Why is the person holding a knife?
To cut the cake with.

http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture10.pdf

RECURRENT NEURAL NETWORK (RNN)

- Application (Extra)
  - **Transformer**
    - In NLP, Input을 더 이상 sequential로 처리 X
    - **Attention mechanism**
    - e.g.
      - BERT (J. Devlin et al., NAACL 2019.)
      - OpenAI GPT-2 (A. Radford et al., 2019.)



http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture10.pdf
Attention is all you need : https://arxiv.org/pdf/1706.03762.pdf
BERT : https://arxiv.org/pdf/1810.04805.pdf
GPT-2 : https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

Reference

- http://cs231n.stanford.edu/slides/2019/cs231n_2019_lecture10.pdf
- https://ratsgo.github.io/natural%20language%20processing/2017/03/09/rnnlstm/
- https://excelsior-cjh.tistory.com/185
- Attention is all you need : https://arxiv.org/pdf/1706.03762.pdf
- BERT : https://arxiv.org/pdf/1810.04805.pdf
- GPT-2 : https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

www.nvidia.com/dli