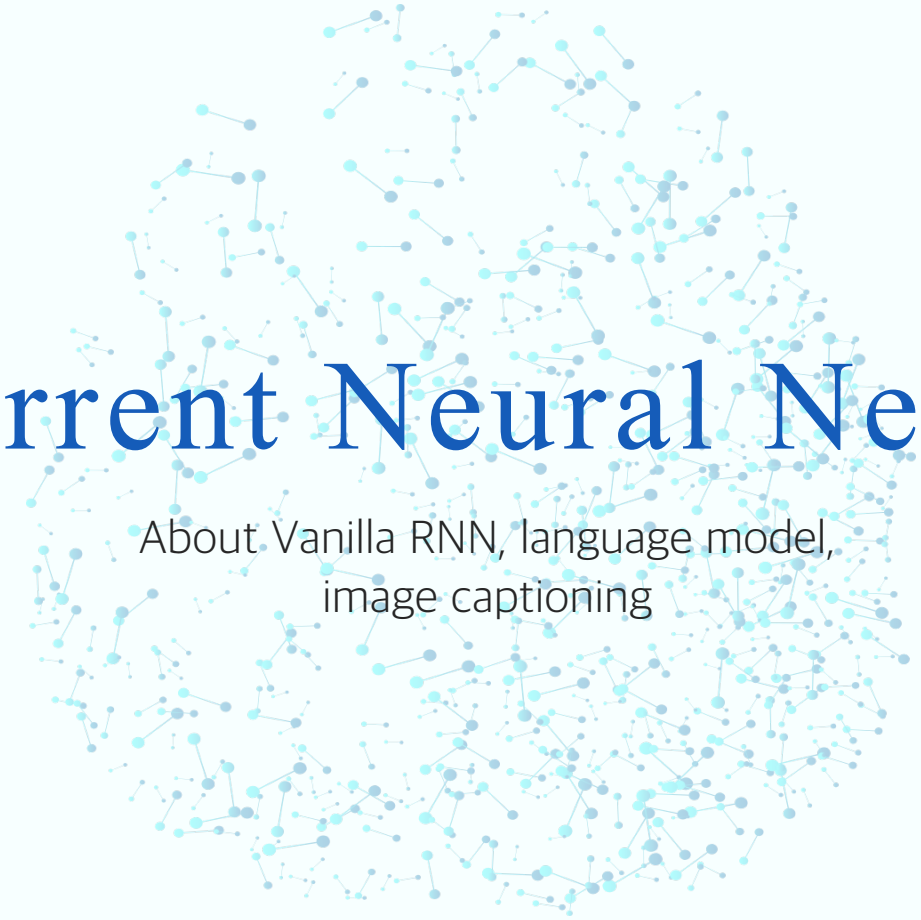


Recurrent Neural Network

About Vanilla RNN, language model,
image captioning

학부연구생 김지현

Overview

Recurrent Neural Networks

of Natural Language Processing

RNN 개요

- 기존의 신경망과 비교
- RNN Computation
- 언어 모델
  (Character level language model)

구현(순전파)

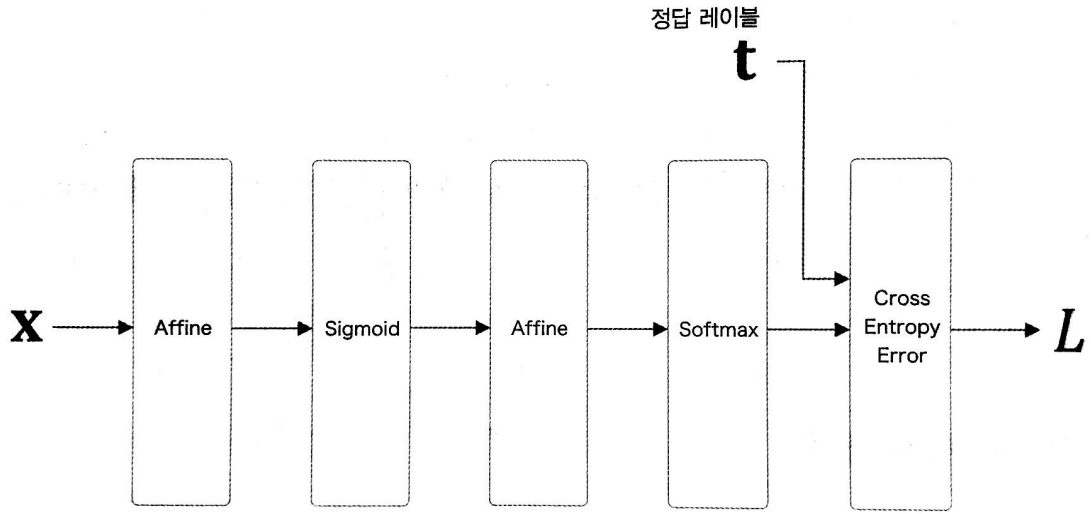
- 밑바닥 / API
- tanh 활성화 함수
- LSTM / GRU ?

image captioning

RNN 개요

기존 신경망?

- feed forward 신경망 (입력층 -> 출력층 한 방향)
- only fixed sized input and output
- 정해진 computational steps (고정된 layer수)



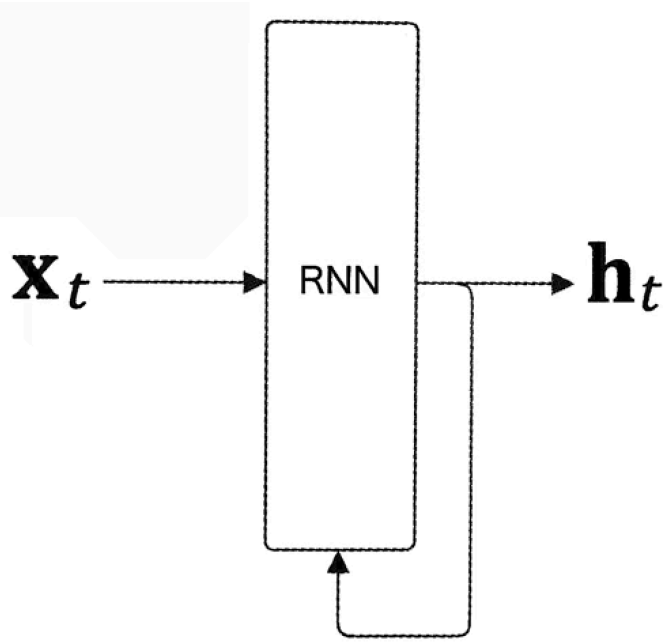
sequence data?

우리의 언어는 지속성(persistence)을 갖고 있다

지속성을 지닌 데이터, 즉 **시계열 데이터를 처리할 수 없다**는 큰 단점

RNN 개요

Recurrent : '순환하다'

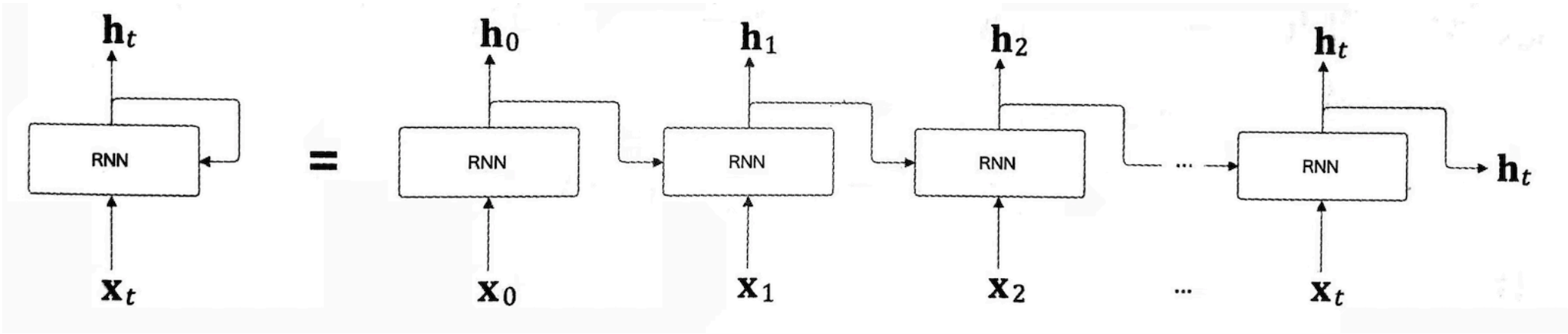


- 시계열 데이터(x0, x1, x2, ⋯, xt)가 RNN 계층에 입력
- 출력이 2개로 분기, 즉 같은 것이 복제되어 분기된 출력이 자기 자신에 입력
- 즉 데이터를 계층 안에서 '순환'시킬 수 있다

순환하는 경로(loop)를 포함하는 RNN 계층

xt: 단어의 분산 표현(word vector)
ht: 은닉 상태 벡터(hidden state vector)

순환 신경망을 펼쳐보면?

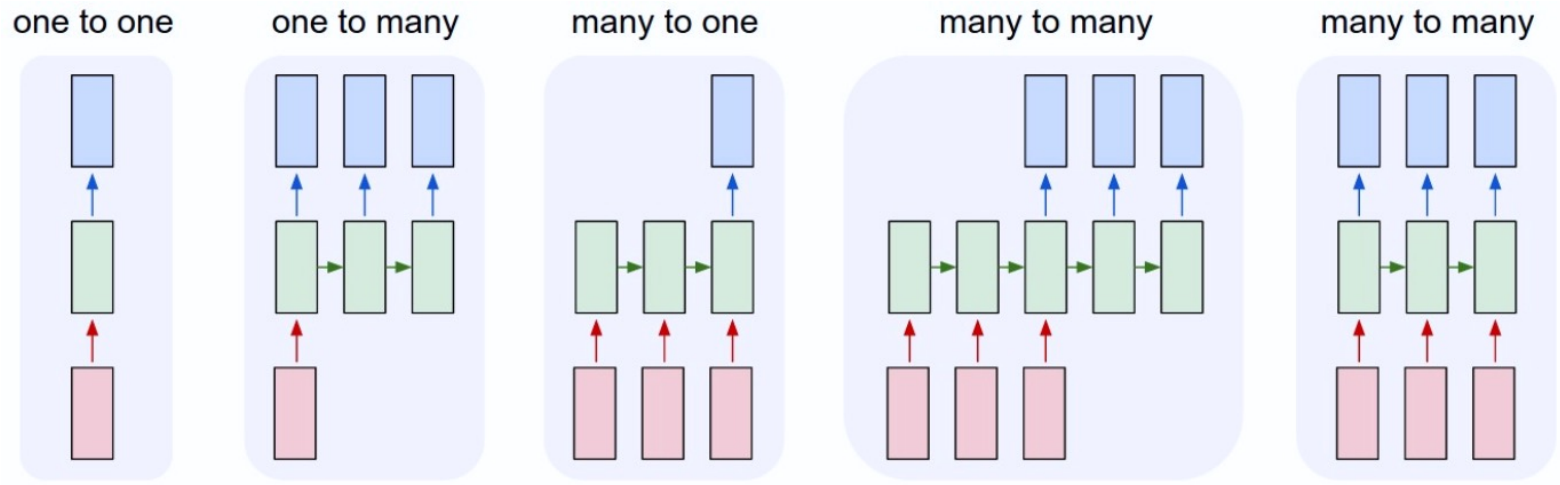


같은 네트워크가 여러 번 복제되어 successor에게 정보 전달하는 체인처럼 연결된 신경망

순환 신경망은 **연속적인**(sequential) **시계열 데이터**(time series) 데이터에 적합한 모델

RNN이 활용되는 분야들

- speech recognition
- language modeling
- translation
- image captioning
- etc



This sentence is sequence of words...

t = 1 t = 2 t = 3 t = 4 . . .

고정된 길이의 벡터가 아닌 연속적인 벡터들을 얼마든지 입력으로 받을 수 있다는 장점

RNN computation

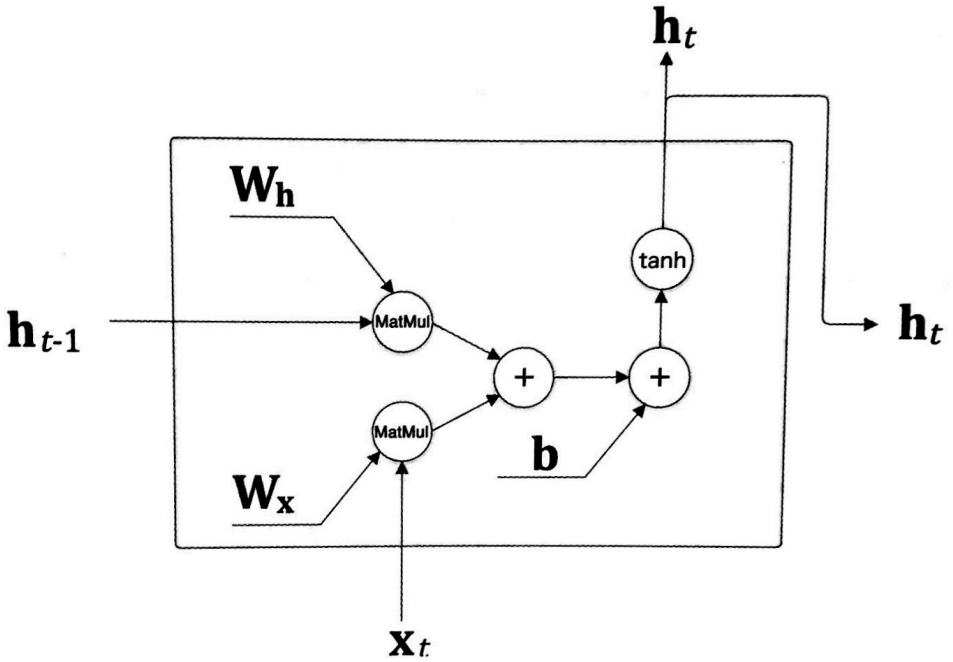
순전파 수식

$$\mathbf{h}_t = \tanh(\mathbf{h}_{t-1}\mathbf{W}_\mathbf{h} + \mathbf{x}_t\mathbf{W}_\mathbf{x} + \mathbf{b})$$

ht : 은닉 상태(hidden state)

가중치 2개

Wx : 입력 x과 행렬 곱
Wh : 이전 RNN 출력과 행렬 곱



언어 모델

언어를 '**확률(Probability)**'로 다루는 언어 모델

$$P(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(T)}) = P(\boldsymbol{x}^{(1)}) \times P(\boldsymbol{x}^{(2)} | \boldsymbol{x}^{(1)}) \times \cdots \times P(\boldsymbol{x}^{(T)} | \boldsymbol{x}^{(T-1)}, \ldots, \boldsymbol{x}^{(1)})$$
$$= \prod_{t=1}^{T} P(\boldsymbol{x}^{(t)} | \boldsymbol{x}^{(t-1)}, \ldots, \boldsymbol{x}^{(1)})$$

간단한 예시)

h e l l o

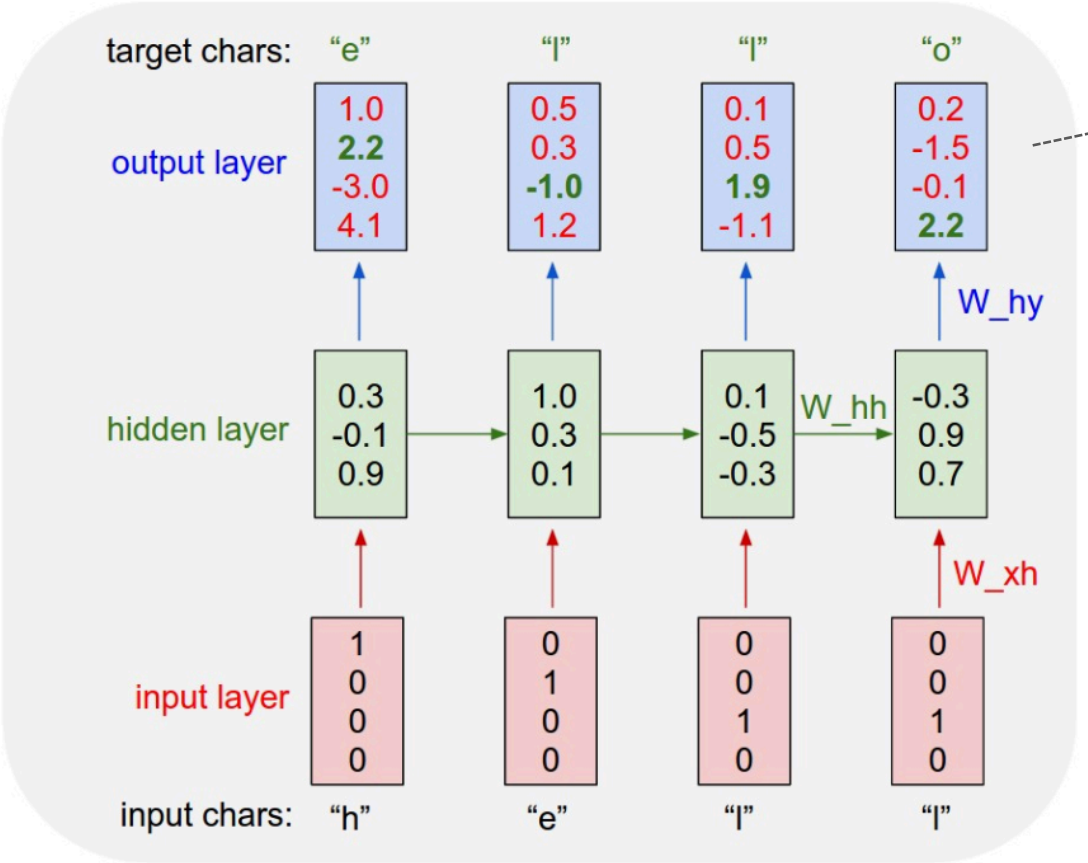
4가지 학습 예시

"h" 뒤에 - "e"
"he" 뒤에 - "l"
"hel" 뒤에 - "l"
"hell" 뒤에 - "o"

} 일 확률이 높도록 학습

언어 모델

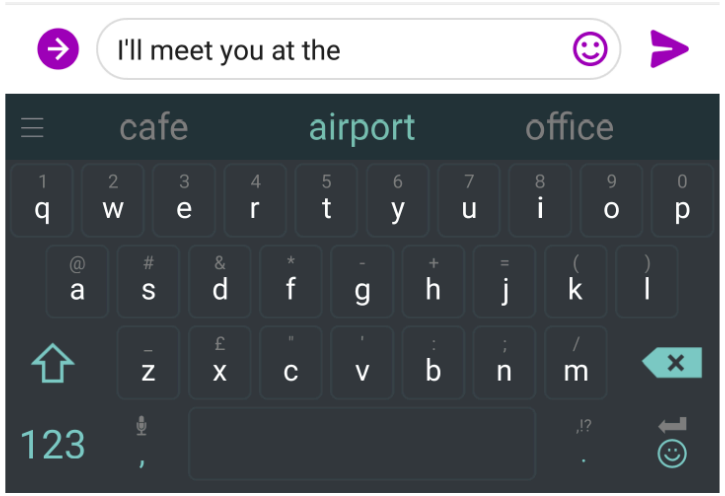
Character-Level Language Models

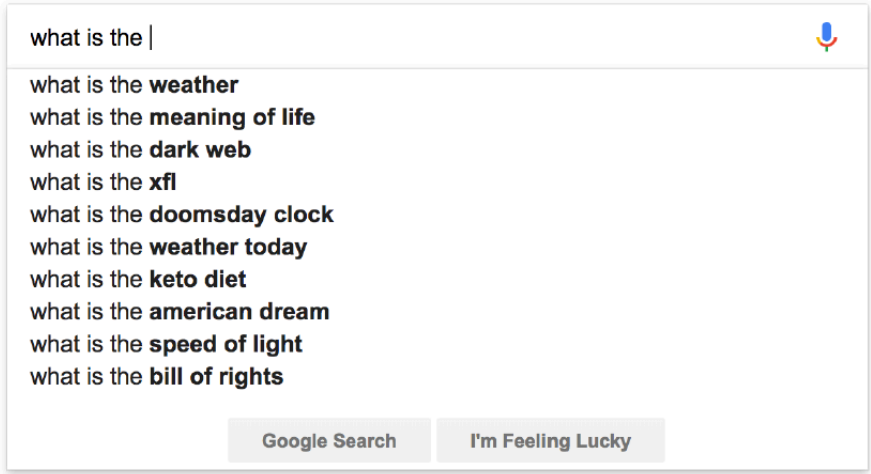


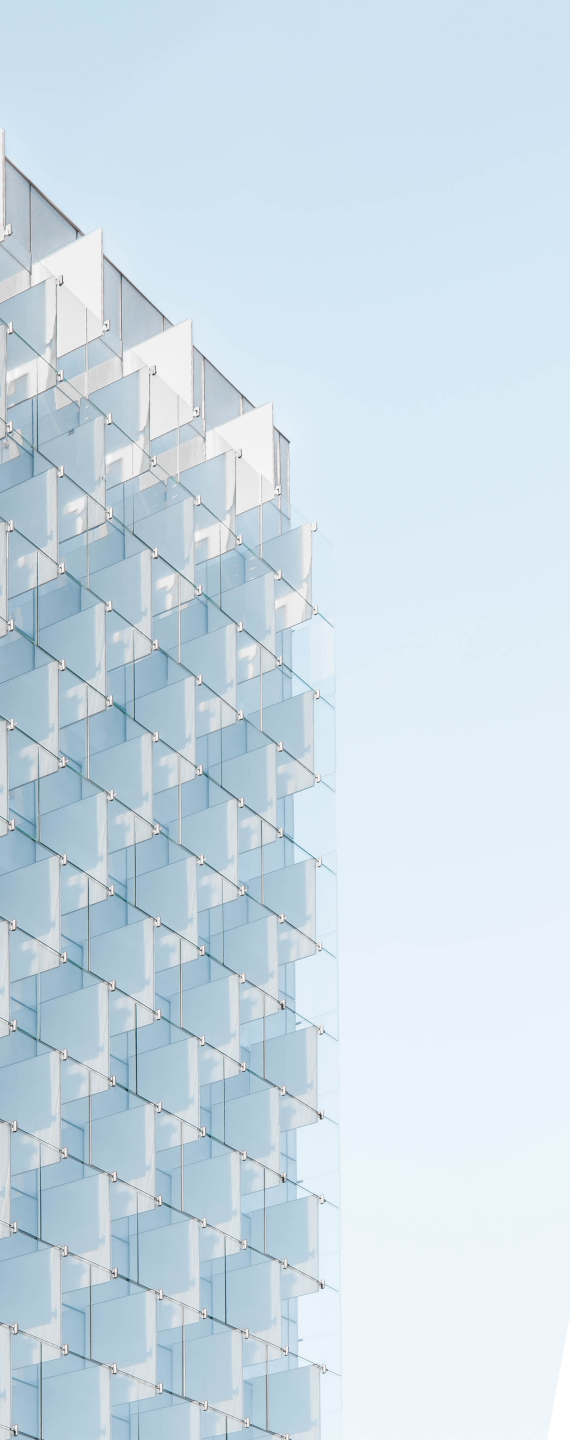
원하는 것
confidence 증가
confidence 감소

An example RNN with 4-dimensional input and output layers, and a hidden layer of 3 units (neurons). This diagram shows the activations in the forward pass when the RNN is fed the characters "hell" as input. The output layer contains confidences the RNN assigns for the next character (vocabulary is "h,e,l,o"); We want the green numbers to be high and red numbers to be low.

You use Language Models every day!

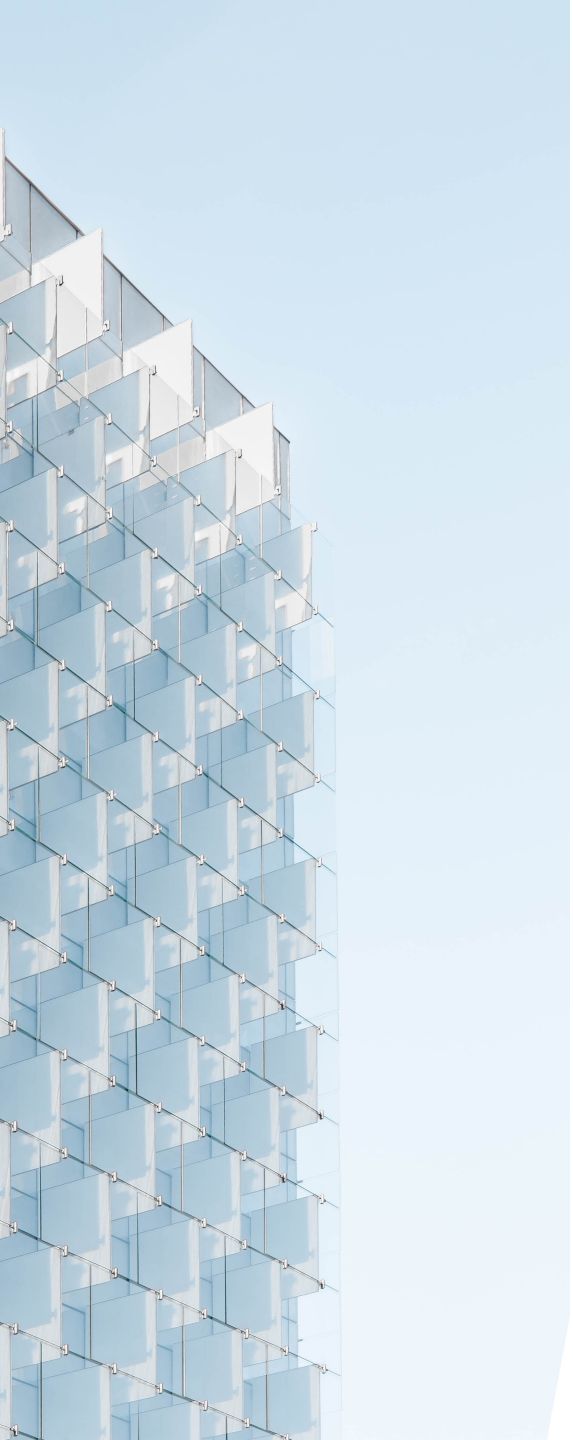






Vanilla RNN (순전파) 구현


```python
class RNN:
    def __init__(self, Wx, Wh, b):
        self.params = [Wx, Wh, b]
        self.grads = [np.zeros_like(Wx), np.zeros_like(Wh), np.zeros_like(b)]
        self.cache = None

    def forward(self, x, h_prev):
        Wx, Wh, b = self.params
        t = np.matmul(h_prev, Wh) + np.matmul(x, Wx) + b
        h_next = np.tanh(t)

        self.cache = (x, h_prev, h_next)
        return h_next

    def backward(self, dh_next):
        Wx, Wh, b = self.params
        x, h_prev, h_next = self.cache

        dt = dh_next * (1 - h_next ** 2)
        db = np.sum(dt, axis=0)
        dWh = np.matmul(h_prev.T, dt)
        dh_prev = np.matmul(dt, Wh.T)
        dWx = np.matmul(x.T, dt)
        dx = np.matmul(dt, Wx.T)

        self.grads[0][...] = dWx
        self.grads[1][...] = dWh
        self.grads[2][...] = db

        return dx, dh_prev
```



Vanilla RNN (순전파) 구현


```
rnn = RNN()
y = rnn.step(x) # x is an input vector, y is the RNN's output vector
```

```
class RNN:
  # ...
  def step(self, x):
    # update the hidden state
    self.h = np.tanh(np.dot(self.W_hh, self.h) + np.dot(self.W_xh, x))
    # compute the output vector
    y = np.dot(self.W_hy, self.h)
    return y
```

```
y1 = rnn1.step(x)
y = rnn2.step(y1)
```

참고)

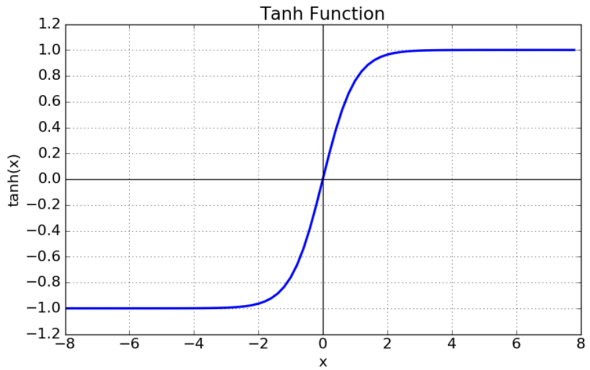
tanh 함수?

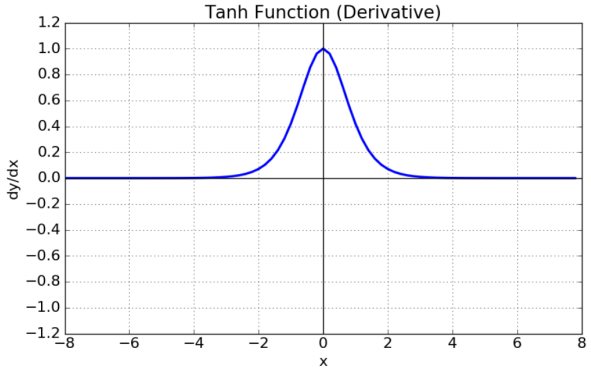
Hyperbolic tangent function

$$tanh(x) = 2\sigma(2x) - 1$$

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

$$tanh'(x) = 1 - tanh^2(x)$$





함수의 중심값을 0으로 옮겨 시그모이드의 최적화 과정이 느려지는 문제 해결

하지만 gradient vanishing 문제는 여전히 남아있음

그러나 성능?

Tom was watching TV in his room. Mary came into the room.
.....
.....
.....Mary said hi to ____ ?

Vanilla RNN은 **장기**(long term) **의존 관계**를 잘 학습할 수 없다

Long-Short Term Memory (LSTM) & Gated Recurrent Unit (GRU)

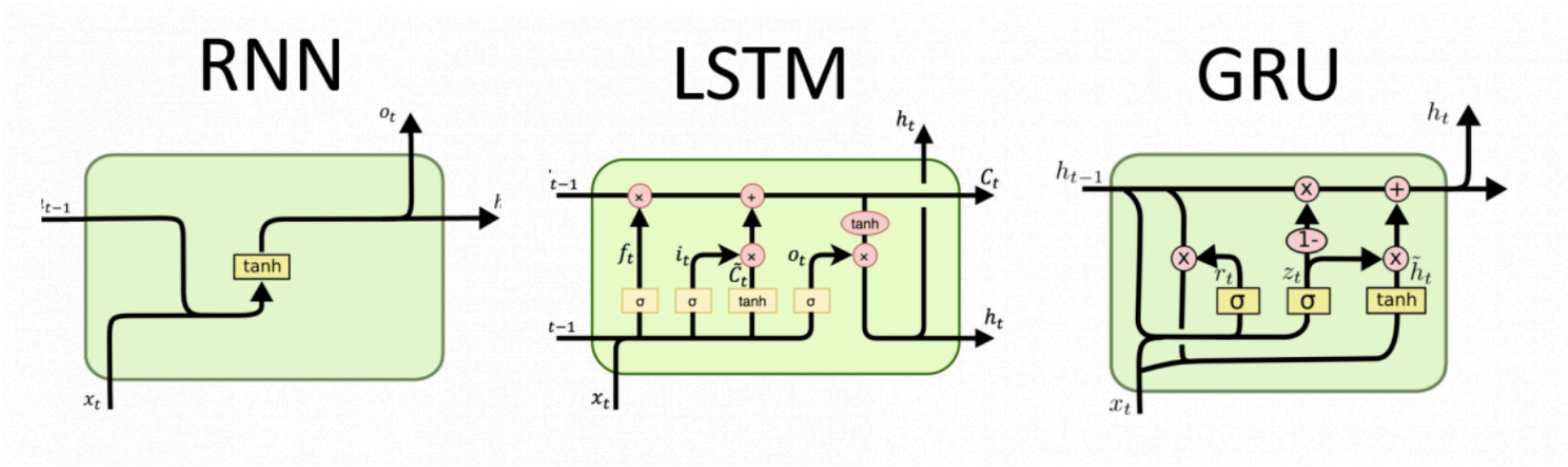


사진 출처: dProgrammer lopez님의 블로그

image captioning

"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."

https://github.com/tensorflow/tensorflow/blob/r1.13/tensorflow/contrib/eager/python/examples/generative_examples/image_captioning_with_attention.ipynb

Reference

밑바닥부터 시작하는 딥러닝2
Stanford CS224N lecture 6| Language Models and RNNs
Andrej Karpathy의 블로그 “The Unreasonable Effectiveness of RNN”
Google AI Blog “Show and ell: image captioning open sourced in Tensorflow”
Tensorflow github “image_captioning_with_attention”