7장_'120다산콜재단' 토픽 모델링과 RNN, LSTM`

오늘의 모델 : 토픽모델링 LDA

- 토픽 모델링 : 비지도학습

문서를 하나 또는 그 이상의 토픽(주제)으로 할당하는 작업을 통칭하는 말

- 잠재 디리클레 할당(Latent Dirichlet Allocation, LDA)

주어진 문서에 대해 각 문서에 어떤 토픽(주제)들이 있는지 서술하는 확률적 토픽 분류 기법 중 하나

오늘의 모델 : 토픽모델링 LDA

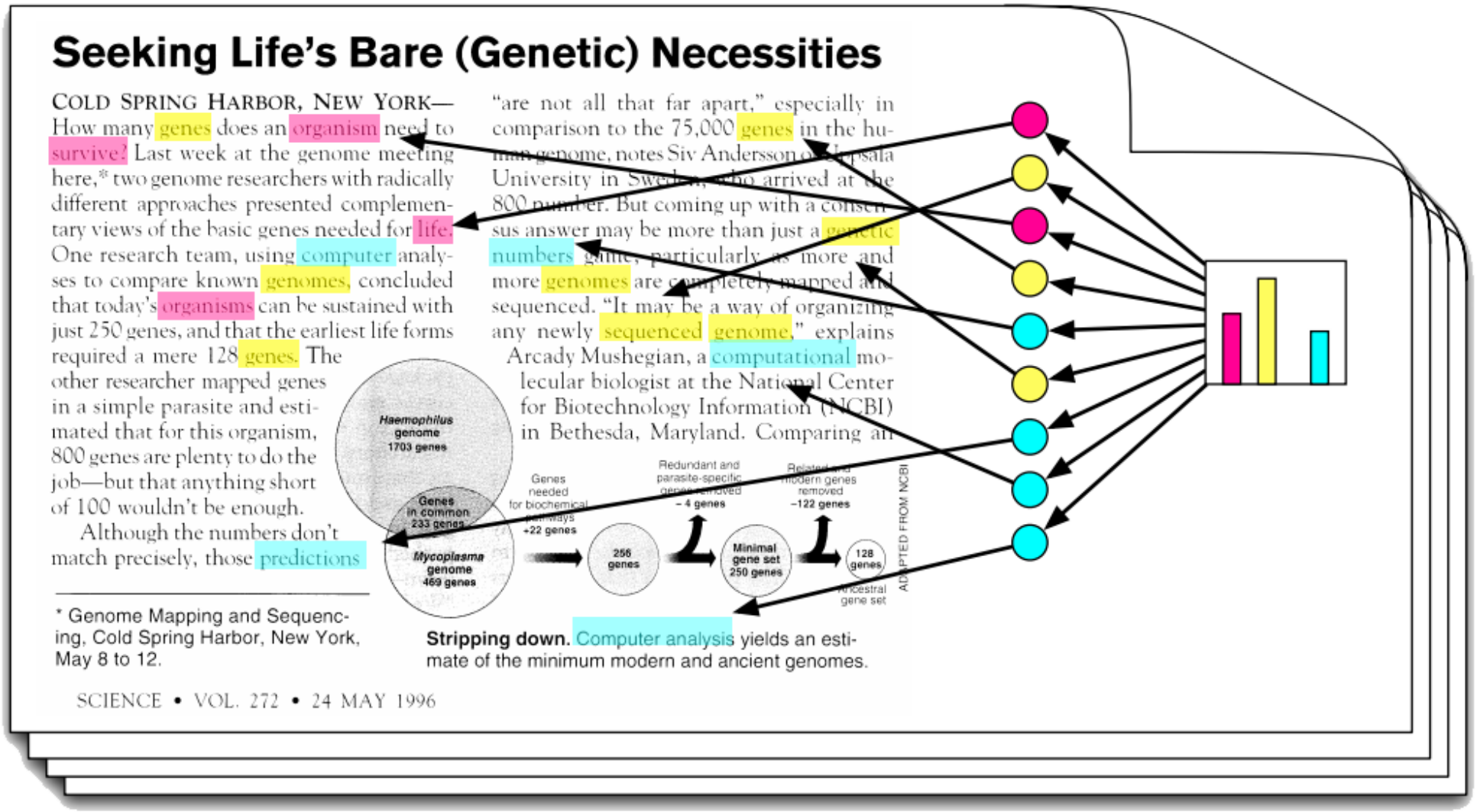
Topics *Documents* *Topic proportions & assignments*

gene	0.04
dna	0.02
genetic	0.01
...	

life	0.02
evolve	0.01
organism	0.01
...	

brain	0.04
neuron	0.02
nerve	0.01
...	

data	0.02
number	0.02
computer	0.01
...	



Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions "are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an

* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

SCIENCE • VOL. 272 • 24 MAY 1996

이미지 출처 : https://ratsgo.github.io/from%20frequency%20to%20semantics/2017/06/01/LDA/

우리의 코드 중 살펴볼 부분

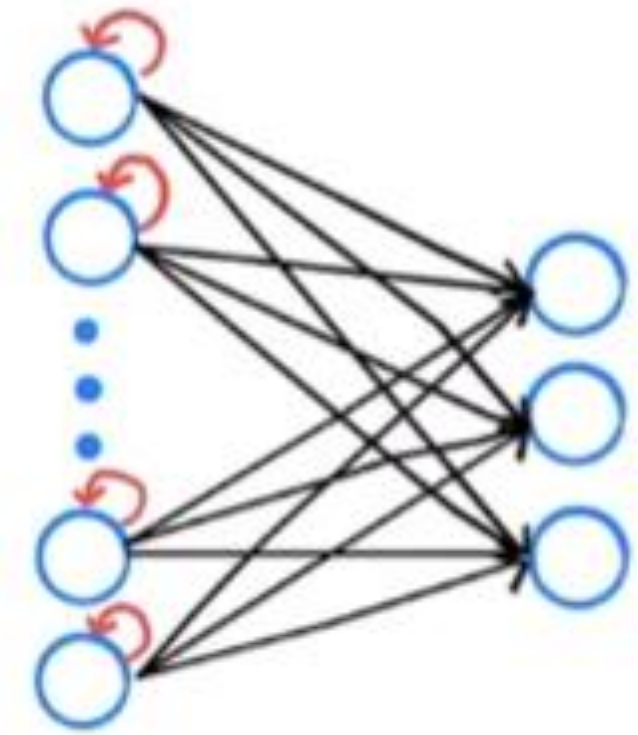
```
from sklearn.decomposition import LatentDirichletAllocation

NUM_TOPICS = 10
LDA_model = LatentDirichletAllocation(n_components=NUM_TOPICS, random_state=42)
```

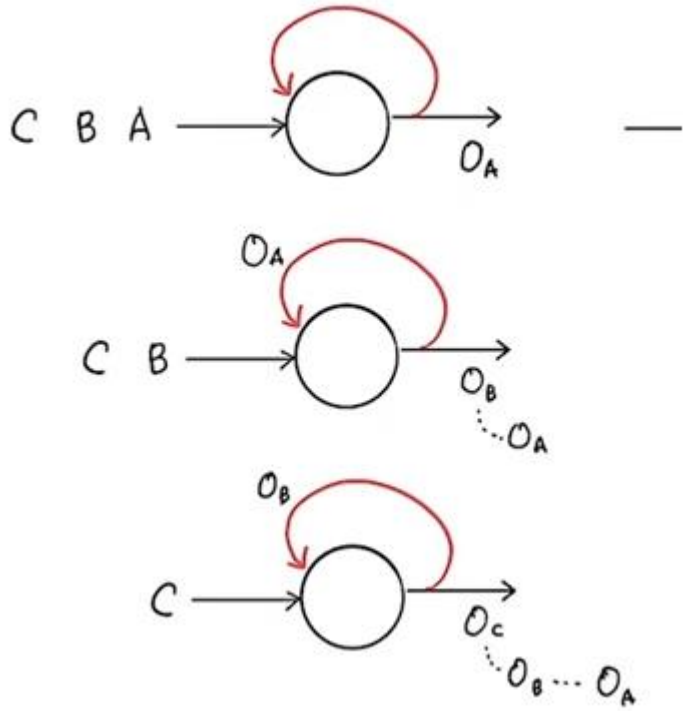
오늘의 모델 : RNN

- 순차 데이터 : 텍스트, 시계열 데이터와 같이 순서에 의미가 있는 데이터

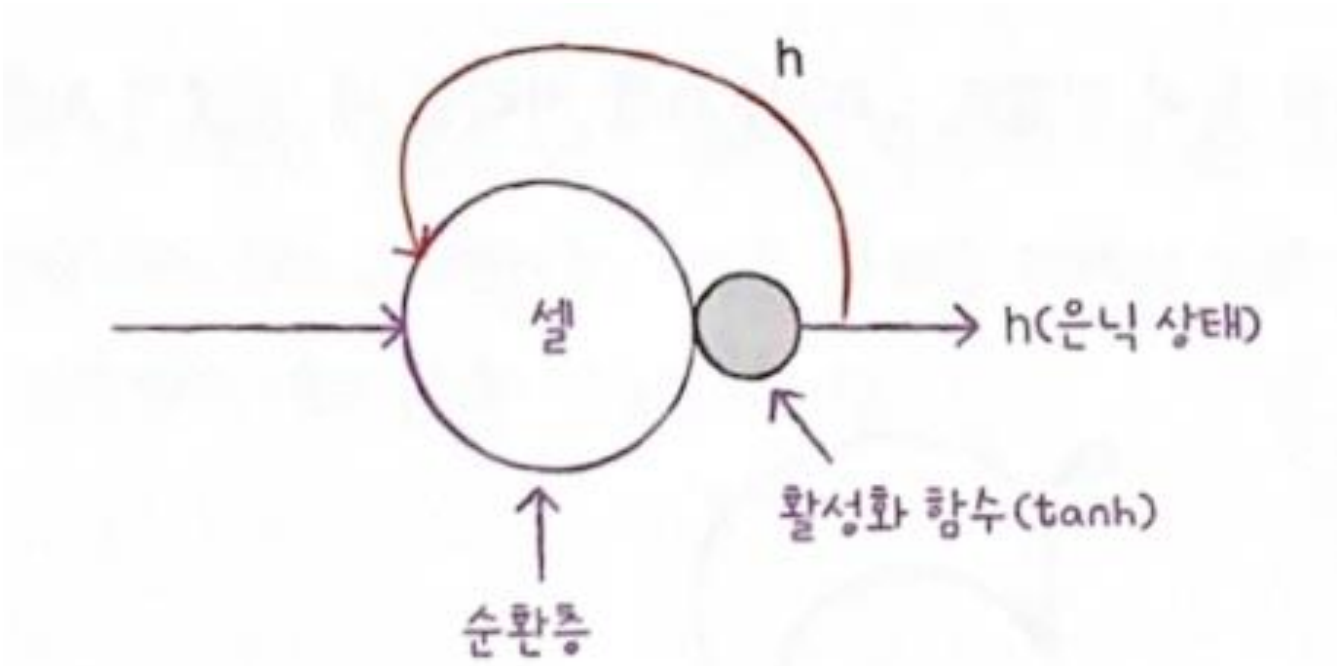
  Ex) “별로지만 추천해요 " , [1일 15°C, 2일 17°C, 3일 13°C…]

- RNN : 이전 데이터의 처리를 순환하는 고리를 추가





RNN



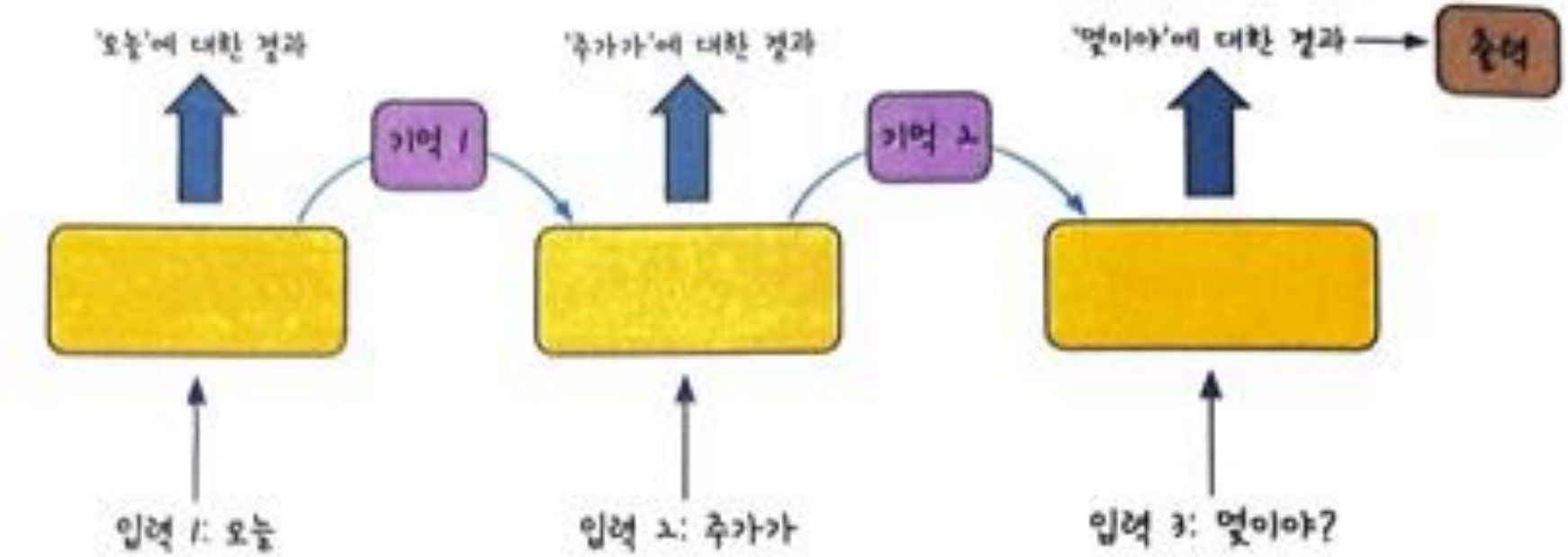
타임 스텝 : 샘플을 처리하는 한 단계

셀 : 층

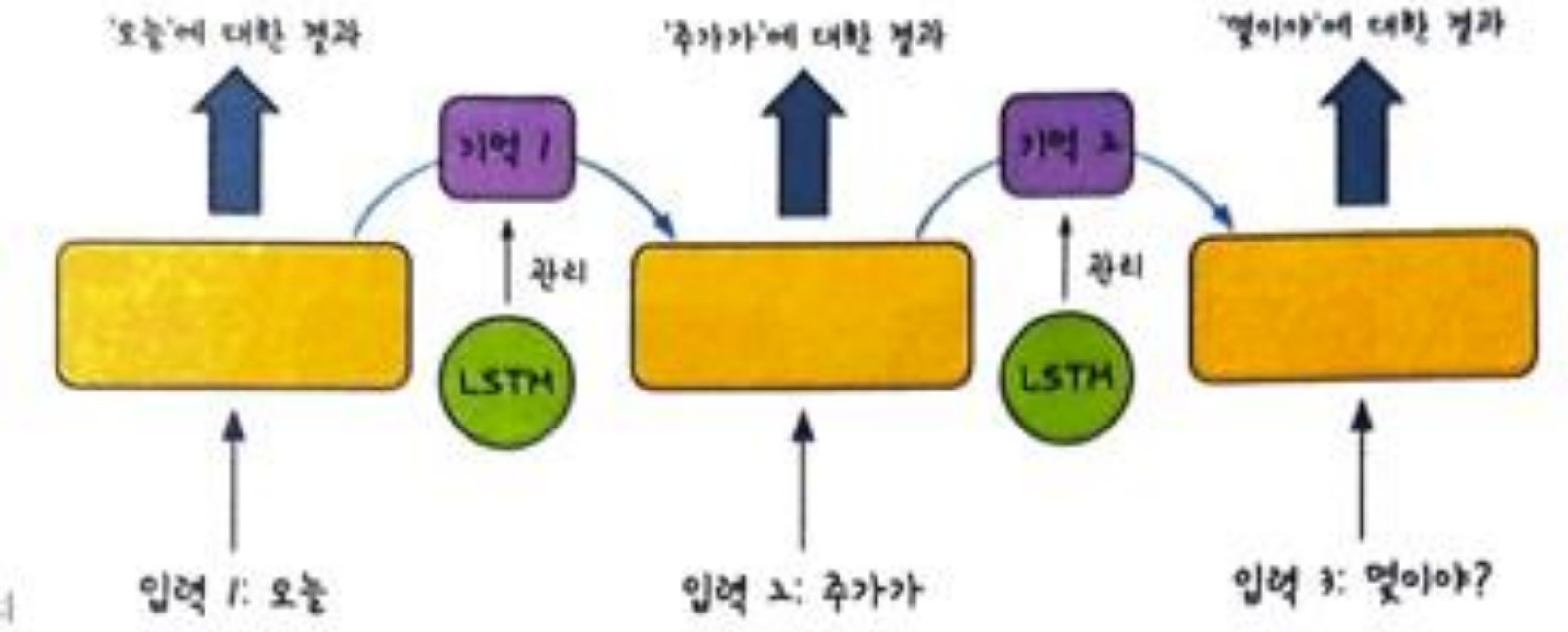
은닉 상태 : 셀의 출력

활성화 함수 : tanh(하이퍼볼릭 탄젠트 함수를 주로 사용)

RNN



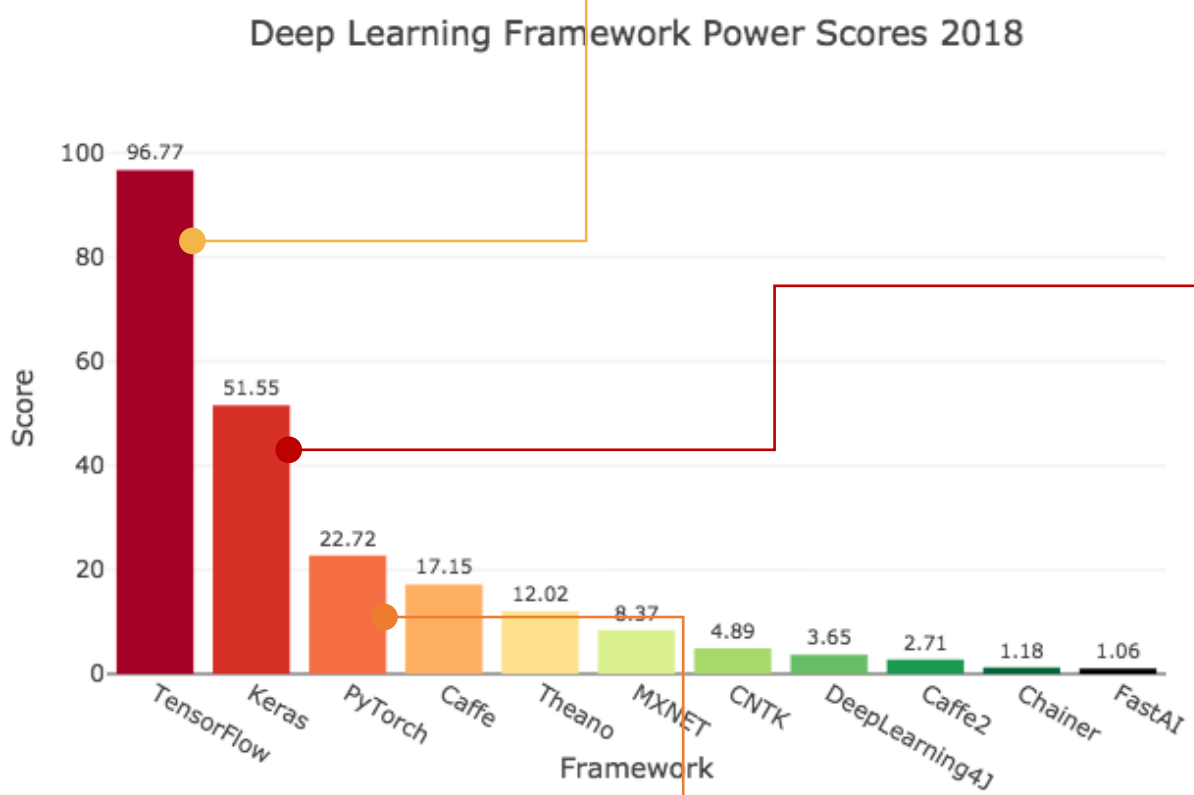
RNN, LSTM



RNN에 일반 신경망보다 더 빈번히 발생하는 기울기 소실 문제에 대한 단점을 보완한 방법.

반복되기 직전에 다음 층으로 기억된 값을 넘길지 넘기지 않을 지 관리하는 단계를 추가함

[참고] TensorFlow, Keras, PyTorch ?



TensorFlow™

- 구글(Google)에서 개발
- 독자적으로 사용 가능(stand-alone)

Keras

- 하이레벨 API (텐서플로의 Wrapper 라이브러리)
- 단순한 인터페이스로 손쉬운 개발을 도움

PYTORCH

- 페이스북(Facebook)에서 개발
- 독자적으로 사용 가능(stand-alone)

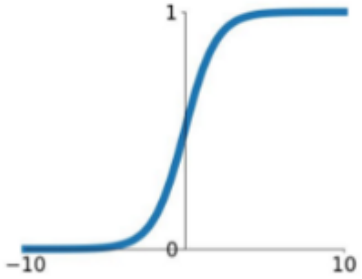
우리의 코드 중 살펴볼 부분

```
from tensorflow.keras import Sequential

model = Sequential([
    Embedding(input_dim=vocab_size, output_dim=embedding_dim, input_length=max_length),
    Bidirectional(LSTM(units=64, return_sequences=True)),
    BatchNormalization(),
    Bidirectional(LSTM(units=32)),
    Dropout(0.2),
    Dense(units=16, activation='relu'),
    Dense(units=n_class, activation='softmax')
])
```

우리의 코드 중 살펴볼 부분 : 다양한 활성함수

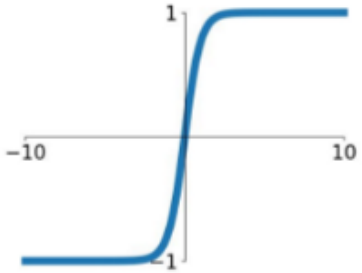
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



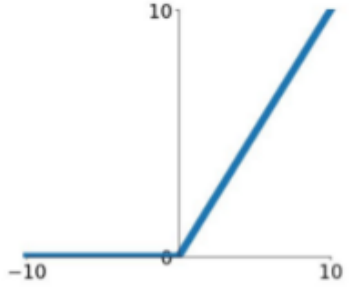
tanh

$$\tanh(x)$$



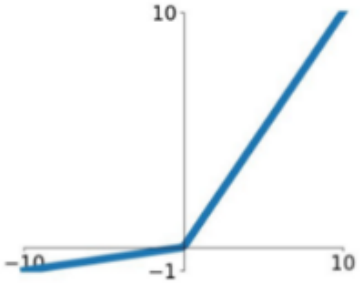
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

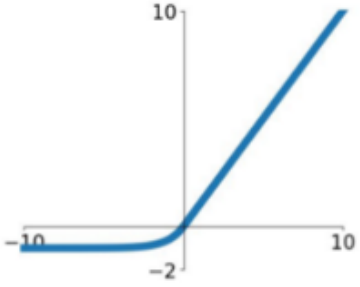


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



7장 끝..거의 끝나갑니다