

11 장 순환 신경망

순차 데이터 (시계열 데이터)

- 순차데이터(sequence data), 시계열 데이터(time sequence data)
 - 순서가 있는 데이터. 시간적인 순서. 공간적인 순서
 - 예. 주식 가격. 텍스트 데이터, 오디오 데이터
- 순차 데이터를 처리하여 정확한 예측을 하려면 과거의 데이터를 어느 정도 기억하고 있어야.





순환 신경망의 응용 분야

- 텍스트, 오디오, 비디오 등의 시계열 데이터를 가지는 다양한 문제에 적용

분야	형태	최종 결과물
음성 인식		What are recurrent neural network?
감정 분석	"It is my favorite time travel sci-fi"	
자동 번역	"순환 신경망이란 무엇인가?"	"What is Recurrent Neural Network?"



- 문장 중의 빈칸을 예측하는 문제
 - 이전의 두 단어만을 가지고 다음 단어를 예측하는 것은 불가능.
 - 문장의 시작 부분에 있는 ‘영화’라는 단어를 기억해야 빈칸에 ‘극장’을 추측해 낼 수 있다.

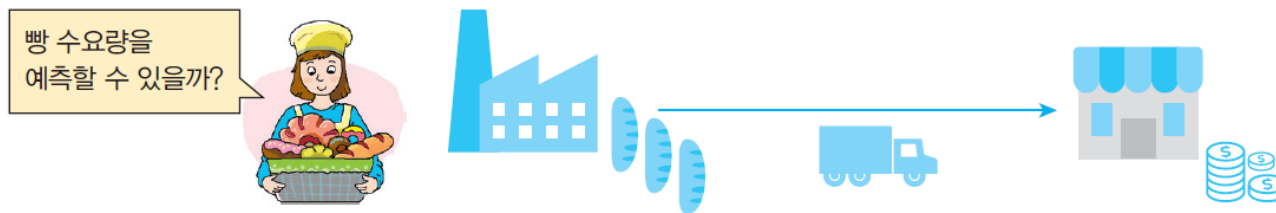
“난 주말이면 영화를 보기 위해 우리 동네의 _____에 간다.”

이전 단어로부터 새로운 단어를 예측한다.

- 순환 신경망의 기능
 - 가변 길이의 입력을 처리할 수 있어야 한다.
 - 장기 의존성을 추적할 수 있어야 한다.
 - 순서에 대한 정보를 유지해야 한다.
 - 시퀀스 전체의 파라미터를 공유할 수 있어야 한다.

순환 데이터의 이해

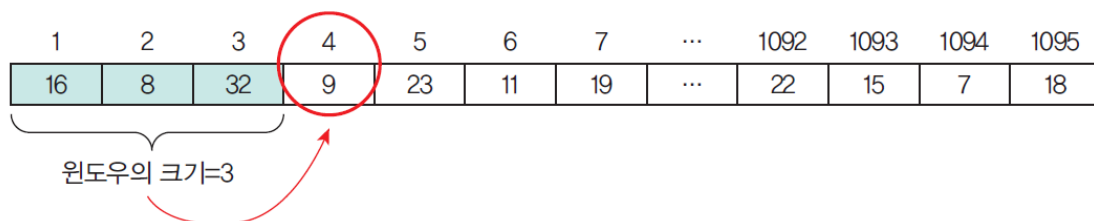
- 순환 데이터: 순환 신경망을 학습시키는 데 사용되는 데이터



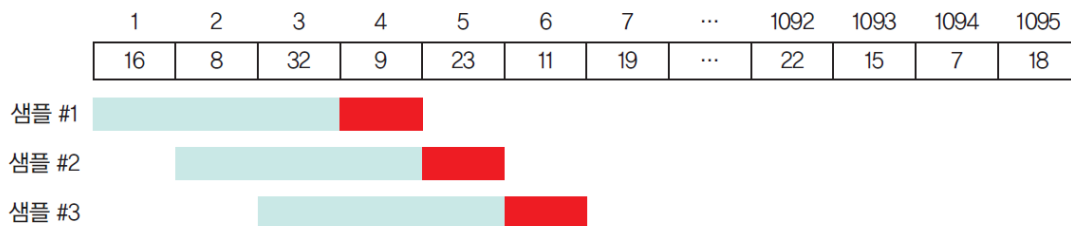
- 3년 x 365 = 1095 개의 데이터

1	2	3	4	5	6	7	...	1092	1093	1094	1095
16	8	32	9	23	11	19	...	22	15	7	18

- 순환 신경망을 학습시키려면 데이터를 일정한 길이(윈도우 크기=3)로 잘라서 여러 개의 훈련 샘플을 만든다.

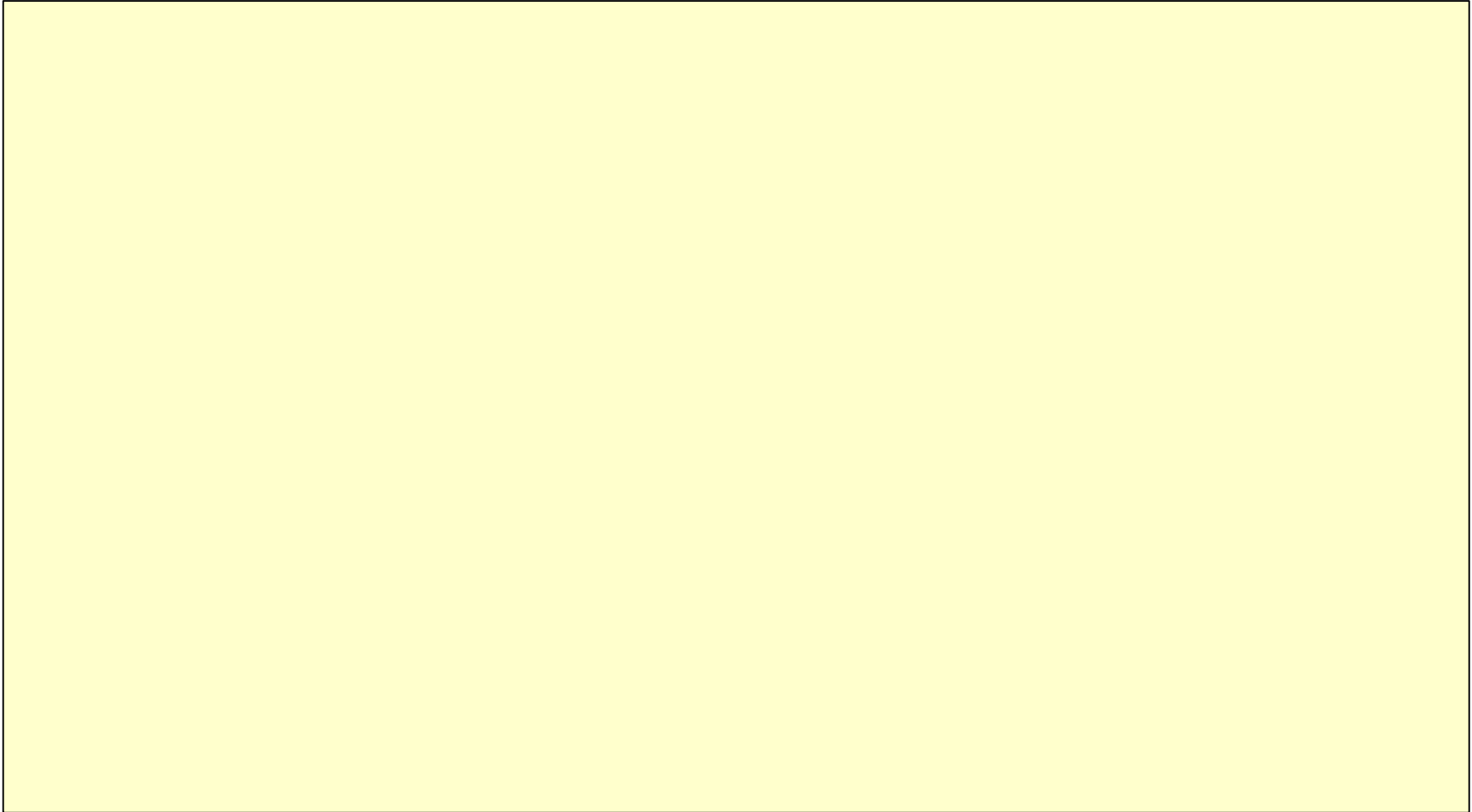


샘플 번호	x(입력)	y(정답)
1	[16, 8, 32]	[9]
2	[8, 32, 9]	[23]
3	[32, 9, 23]	[11]
...
1092	[22, 15, 7]	[18]



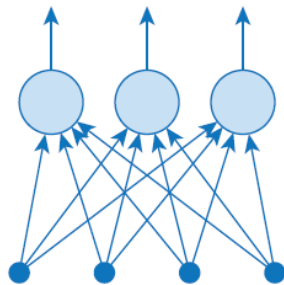


예제: 데이터를 다운로드하고 그래프로 그려보자.

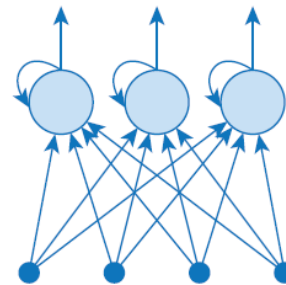


RNN의 구조

- 피드-포워드 신경망(feed-forward neural network)
 - 항상 현재 입력만 고려
- 순환 신경망(RNN)
 - 이전에 수신한 정보를 계속 보유. 즉 과거 정보 지속
 - 동일한 입력이더라도 이전 입력에 따라 서로 다른 출력을 생성



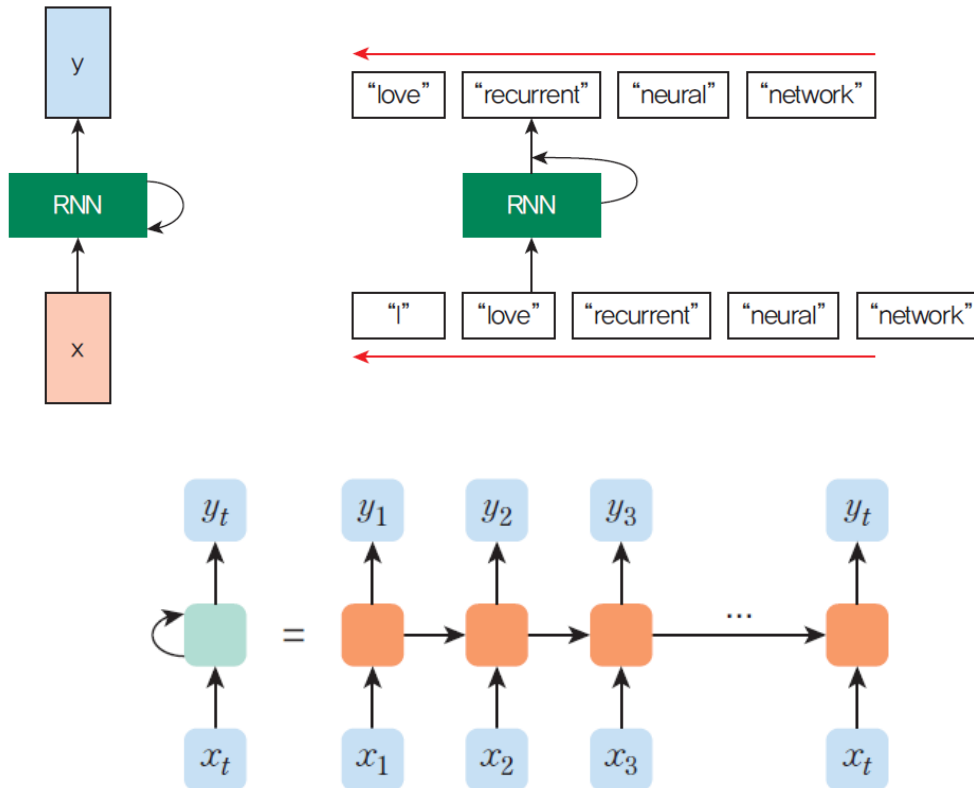
피드-포워드 신경망



순환 신경망



- 입력 벡터 x 가 처리되어서 출력 벡터 y 가 생성. 내부적으로 이전 상태가 다시 피드백. 벡터 x 는 시계열 데이터



RNN의 동작

- 새로운 은닉 상태 h_t 는 이전의 은닉 상태 h_{t-1} 과 현재의 입력 벡터 x_t 를 $f_w()$ 에 넣어서 얻는다.

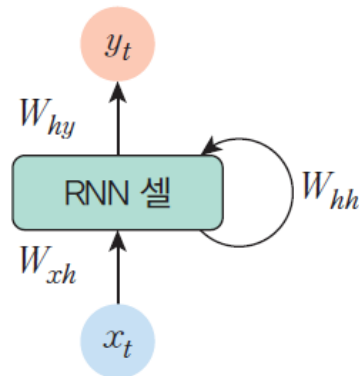
새로운 은닉 상태

이전 은닉 상태

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

가중치 W 를 가지는 함수

시간 t 에서의 입력 벡터



- 입력 벡터: x_t
- 출력 벡터: $y_t = f(W_{hy}h_t)$
- 은닉 상태: $h_t = \tanh(W_{xh}x_t + W_{hh}h_{t-1})$



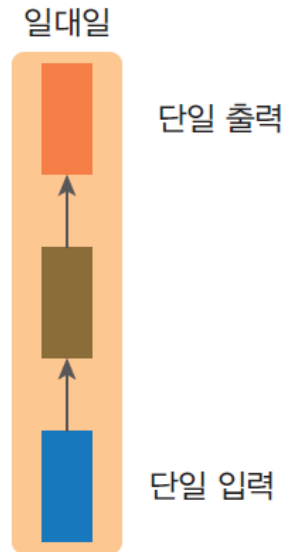
RNN의 유형

- 일대일 (One to One)
- 일대다 (One to Many)
- 다대일 (Many to One)
- 다대다 (Many to Many)



일대일(One to One)

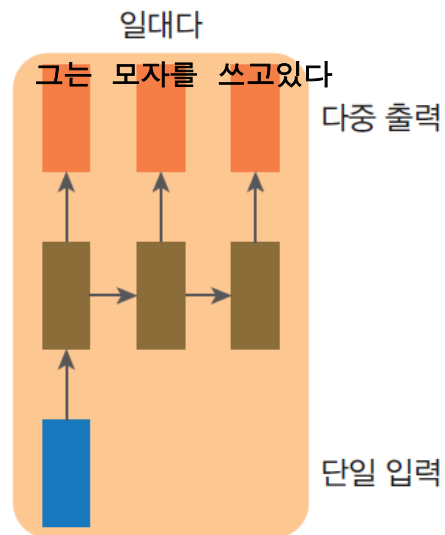
- 단일 입력과 단일 출력이 있는 가장 일반적인 신경망(Vanilla Neural Network)





일대다 (One to Many)

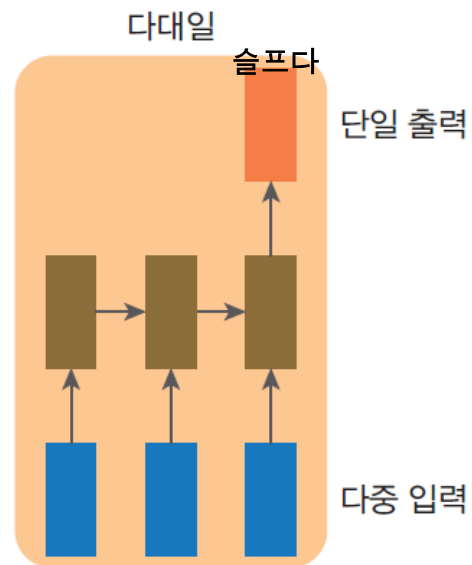
- 하나의 입력을 받아서 많은 수의 출력
- 예. 이미지 캡션 생성 **RNN** : 하나의 이미지가 입력되면 이미지를 가장 잘 설명하는 캡션들을 생성





다대일(Many to One)

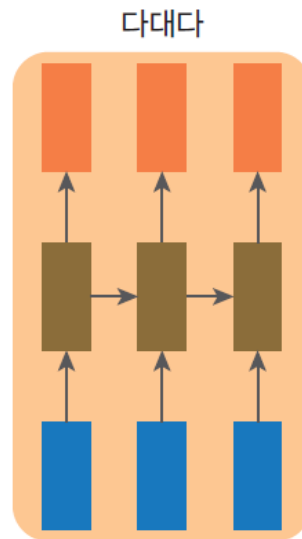
- 일련의 입력을 받아 단일 출력을 생성
- 예. 감정(Sentiment) 분석 신경망 : 주어진 문장들이 긍정적 또는 부정적 감정인지를 분류





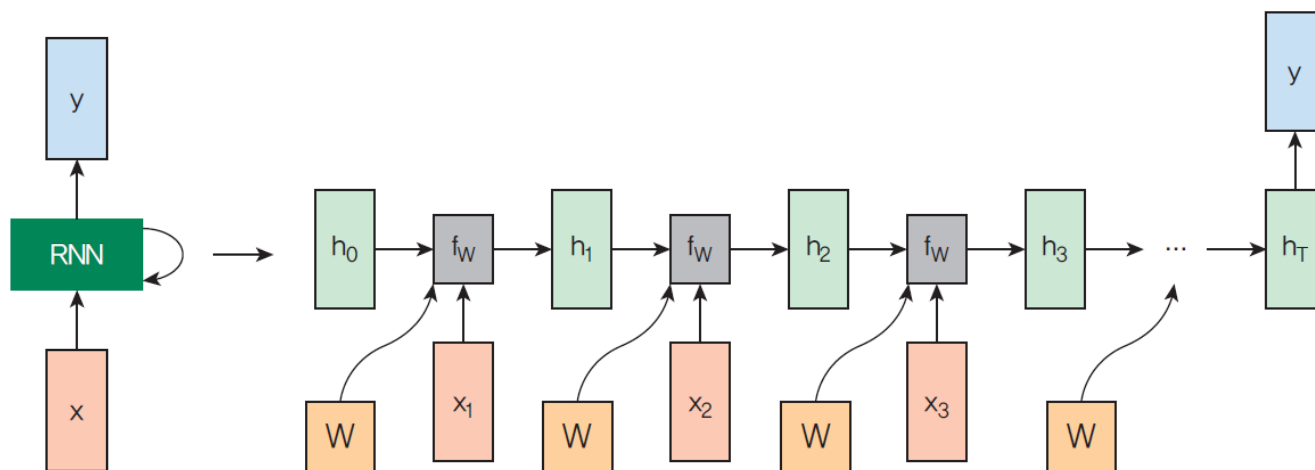
다대다 (Many to Many)

- 많은 수의 입력을 받아 많은 수의 출력을 생성
- 예. 기계 번역 : 단어들을 다른 단어들로 계속 출력



RNN의 순방향 패스

- 매시간 다른 가중치로 출력하는 다대일(many-to-one) 방식인 경우





케라스에서의 RNN

```
inputs = np.random.random([32, 10, 8]).astype(np.float32)
```

```
simple_rnn = tf.keras.layers.SimpleRNN(4)
```

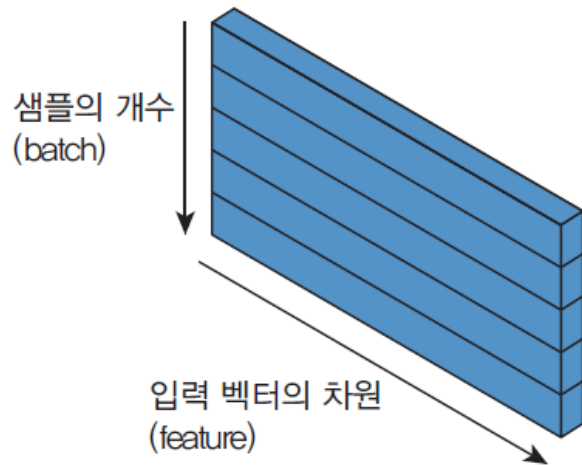
```
output = simple_rnn(inputs)
```

- SimpleRNN(4)와 같이 호출하면 셀이 4개인 RNN 레이어가 만들어진다.
- SimpleRNN의 입력 :
 - [batch, timesteps, feature] 3차원 텐서.
 - [32, 10, 8] : 32개의 샘플. 샘플당 10개의 시계열 데이터, 하나의 데이터는 8개의 실수로 구성

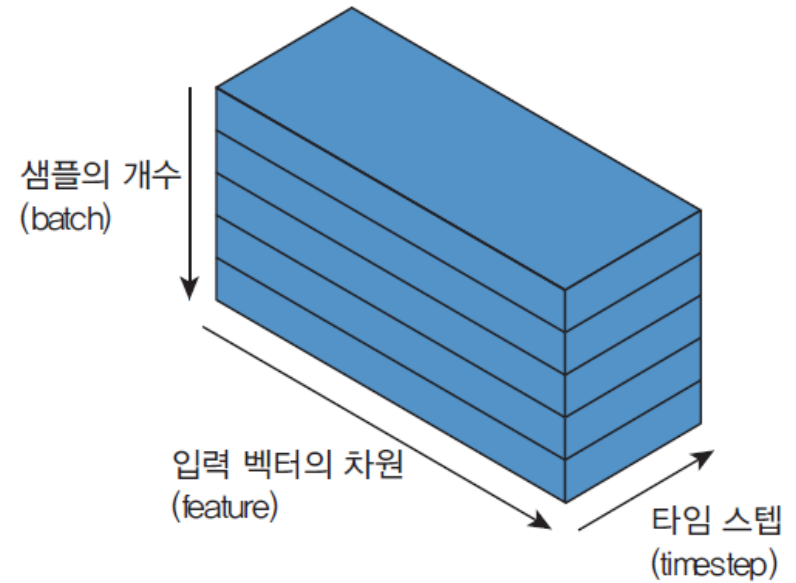


RNN에서의 입력 형상

피드-포워드 신경망



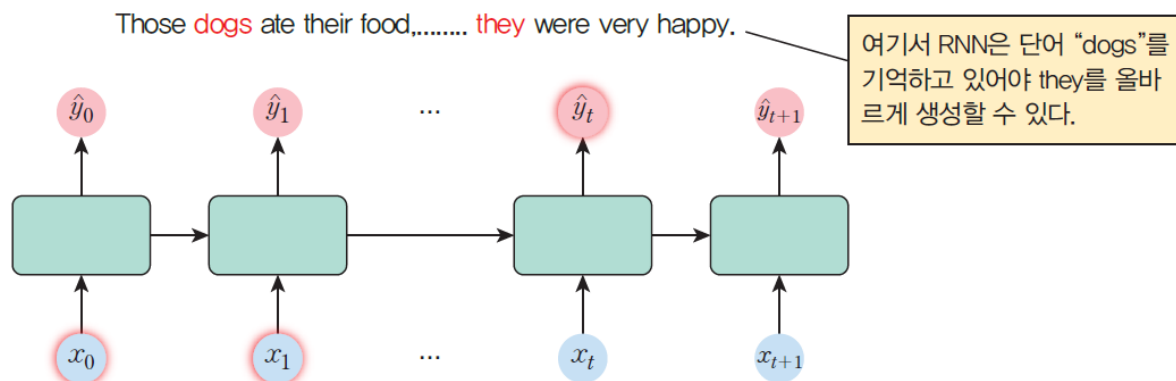
순환 신경망





그래디언트 소실 문제

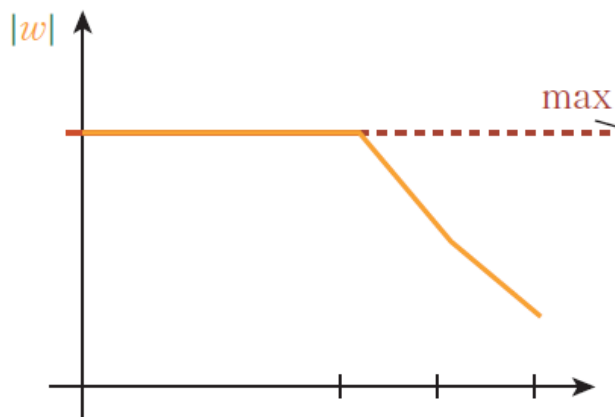
- 1보다 작은 값들이 여러 번 곱해지면 점점 작아지다 결국 없어지게 되는 현상
- 먼 거리의 의존 관계는 파악하지 못하고 근거리의 의존 관계만을 중시하게 된다.
- 해결책 : 보다 복잡한 순환 유닛인 LSTM, GRU 같은 Gated Cell 사용



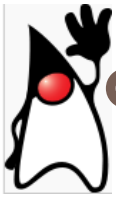


그래디언트 폭증 문제

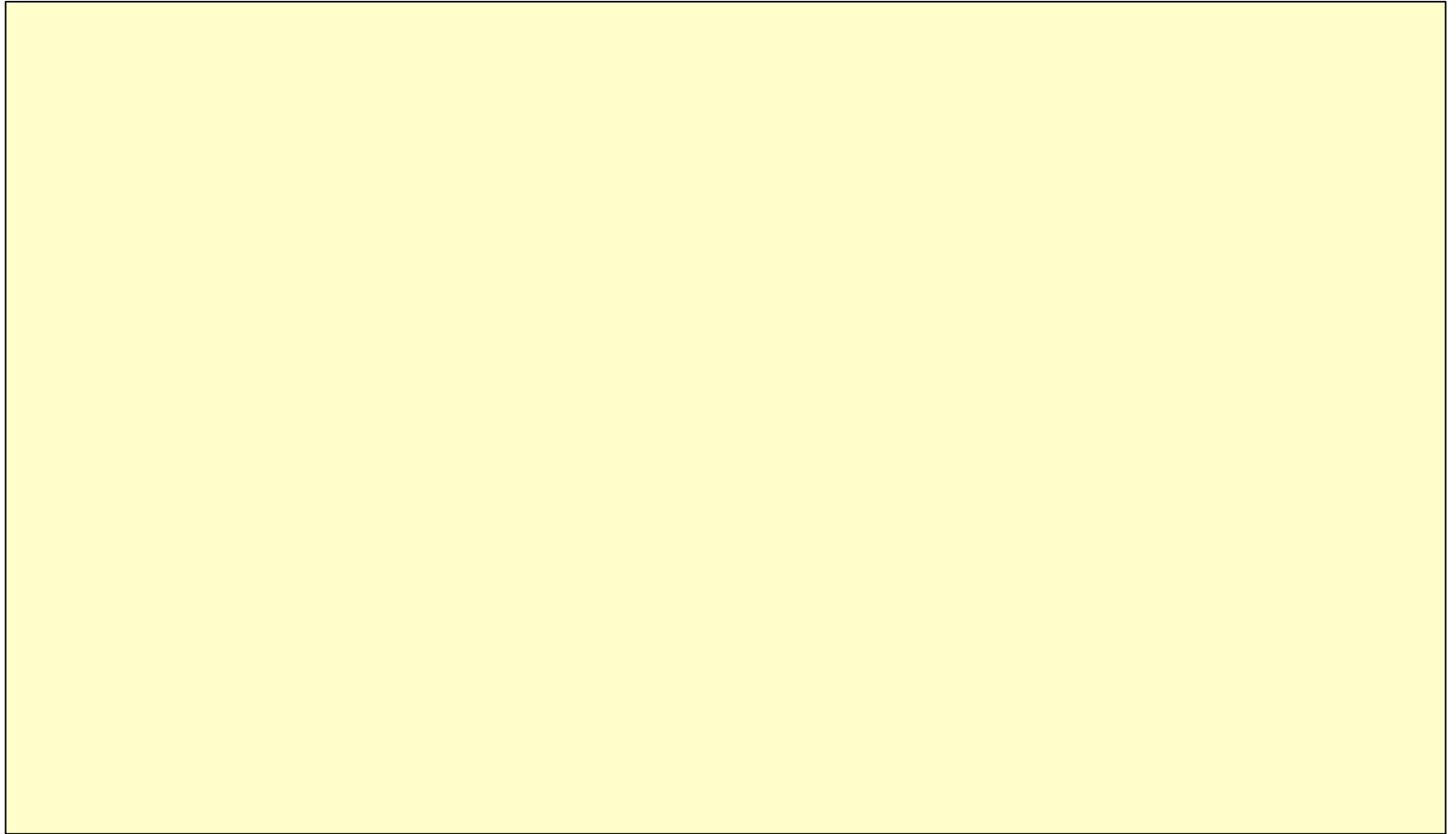
- 그래디언트가 너무 커지는 것



그래디언트의 절대 크기가 어느 정도 이상 커지지 못하게 하면 그래디언트 폭증 현상을 막을 수 있다.

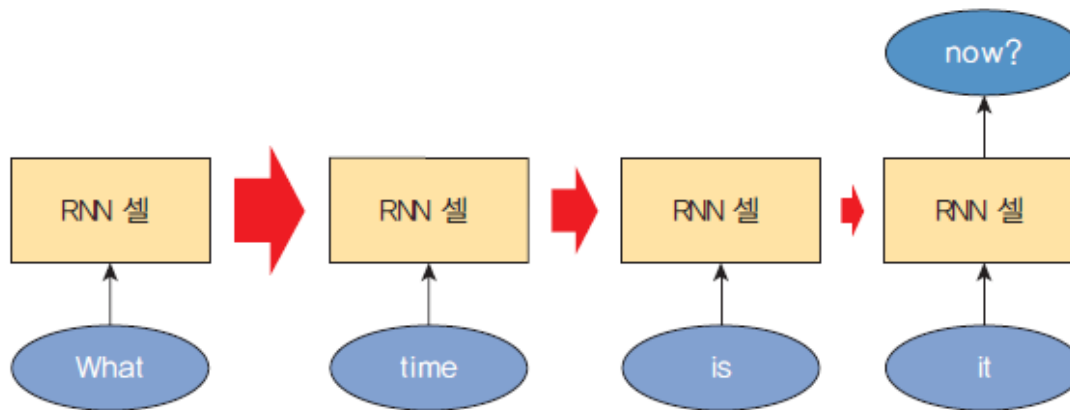


예제: 사인파 예측 프로그램



LSTM 신경망

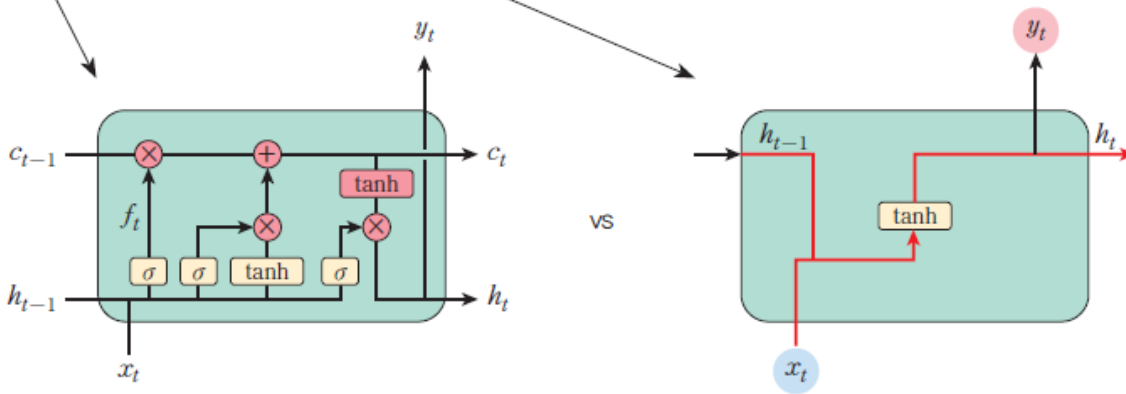
- RNN의 한계 - 시간이 길어질수록 이전에 입력되었던 정보가 뒤로 충분히 전달되지 못한다.
- 예. 첫번째와 두번째 단어인 “What”과 “time”을 기억하고 있어야 “now”를 올바르게 예측할 수 있다. 하지만 RNN에서는 그래디언트 소실 현상 때문에 초반의 입력은 뒤로 갈수록 점점 손실된다.





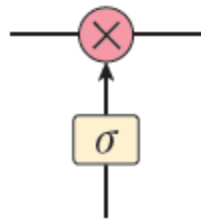
LSTM(Long short-term memory)

- 기존 RNN을 훈련할 때 발생할 수 있는 그래디언트 소실 문제를 해결하기 위해 개발
- LSTM 유닛 : 셀, 입력 게이트, 출력 게이트, 삭제 게이트로 구성
- 셀은 임의의 시점에 대한 값을 기억하고 세 개의 게이트(입력 게이트, 망각 게이트, 출력 게이트)는 셀로 들어오고 나가는 정보의 흐름을 조절한다.
- LSTM의 구조는 표준 RNN에 비하여 셀 상태를 나타내는 c_t 가 추가됨





게이트, 삭제, 저장, 업데이트, 출력 연산





그래디언트 소실 문제 해결





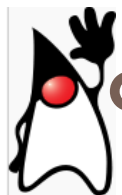
케라스에서의 LSTM





Keras가 제공하는 순환신경망 클래스들

클래스	구조	설명
<code>keras.layers.SimpleRNN</code>		기본 순환 신경망이다. 앞 절에서 다룬 바 있다.
<code>keras.layers.LSTM</code>		1997년 Hochreiter & Schmidhuber 이 처음 제안한 순환 신경망 모델로서 앞 절에서 다룬 바 있다.
<code>keras.layers.GRU</code>		Cho et al. 등이 2014년도에서 처음 제안한 순환 신경망 모델이다. 이 책에서는 다루지 않았다.



예제: Keras를 이용한 주가 예측(LSTM)

