

인공지능 (Artificial Intelligence)

시계열 딥러닝

Hyuntae Cho

Dept. of Digital Media Engineering

Tongmyong University

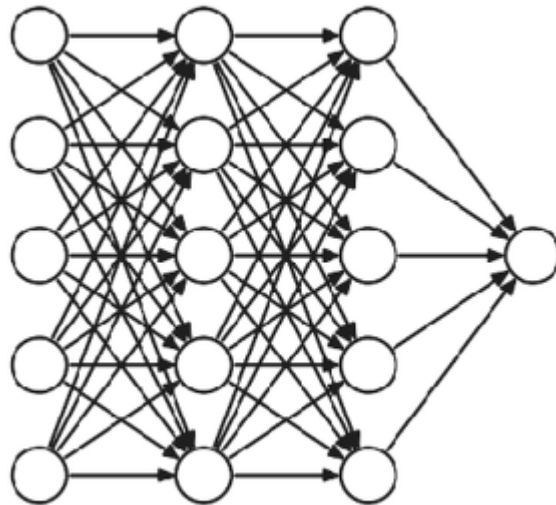
순차 데이터

- **순차 데이터(sequential data)**

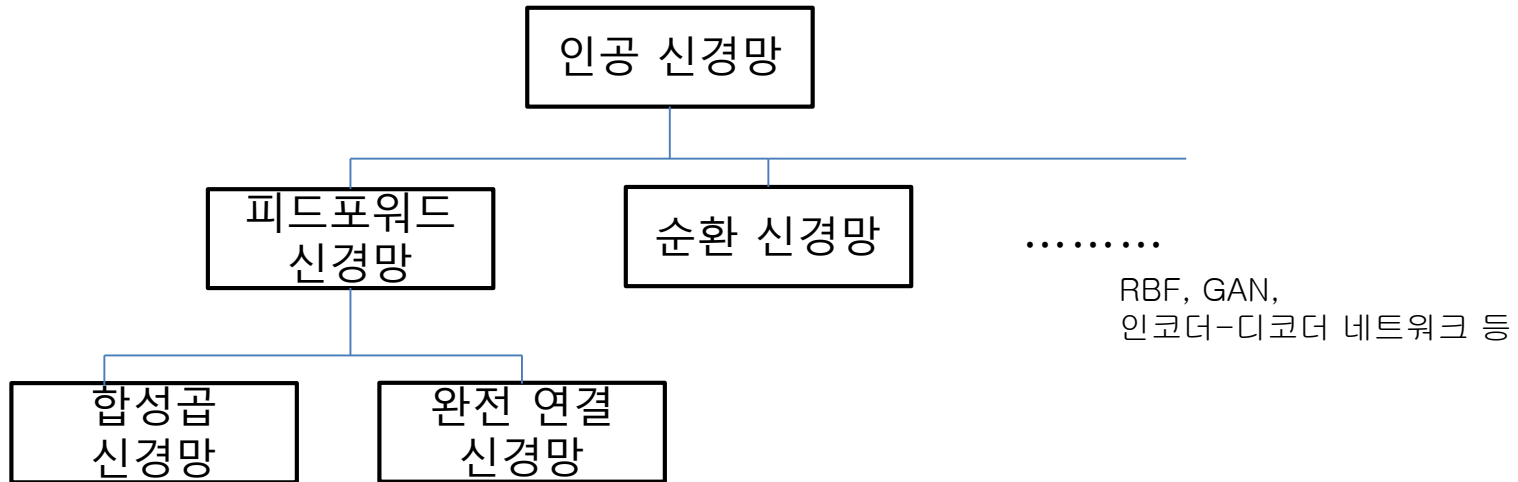
- 텍스트나 시계열(time series data)와 같이 순서에 의미가 있는 데이터
- e.g.)
 - 텍스트: "I am a boy" → boy a am I는 의미 없음
 - 시계열: 1일 15도, 2일 17도, 3일 16도, 4일 20도.....
- **순차 데이터는 순서가 중요한 데이터**
- 순차 데이터를 다룰 때는 이전에 입력한 데이터를 기억하는 기능이 필요
 - E.g.) "별로지만 추천해요"
 - "추천해요"를 인식할 때, "별로지만"을 기억하고 있어야 긍정적이라고 판단하지 않음
- 응용:
 - 언어 변환, 자연어 처리, 음성 인식, 이미지 캡션과 같은 순서 문제나 시간 문제에 흔히 사용

순환 신경망 (Recurrent Neural Network)

- 앞서 살펴본 완전 연결 신경망이나 합성곱 신경망은 기억을 하는 유닛이 없다.
- 입력 데이터의 흐름이 앞으로만 전달되는 신경망을 피드포워드 신경망 (feedforward neural network, FFNN)이라 함



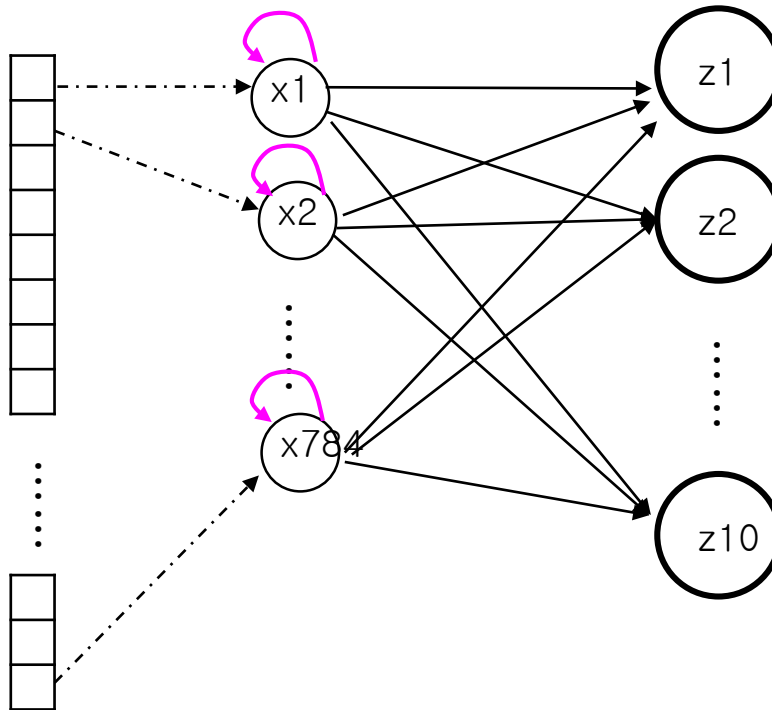
순환 신경망 (Recurrent Neural Network)



- 이전에 처리했던 샘플을 다음 샘플을 처리하는데 사용하기 위해서는 순환되는 층이 필요함 → 순환 신경망

순환 신경망 (Recurrent Neural Network)

- 기존의 신경망에 순환하는 고리를 추가 한 것
 - 즉, 샘플을 사용할 때 이전에 사용했던 데이터를 재사용하는 것



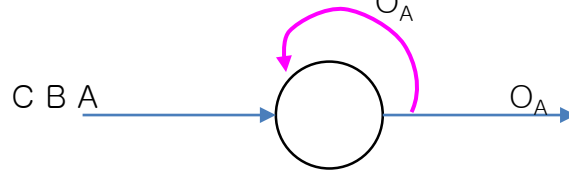
[순환 신경망]

순환 신경망 (Recurrent Neural Network)

- 예제:

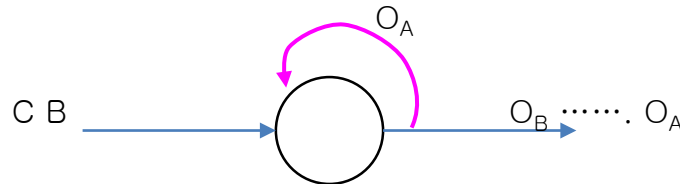
- 입력을 A,B,C순으로, 출력은 O라 하자

- 입력 A로부터 O_A 가 계산되고, O_A 는 다시 입력 뉴런으로 들어감



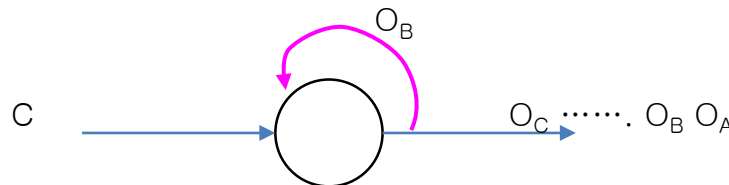
타임스텝
(timestep)

- O_B 가 계산될 때, 이전 O_A 와 함께 O_B 가 계산됨



타임스텝
(timestep)

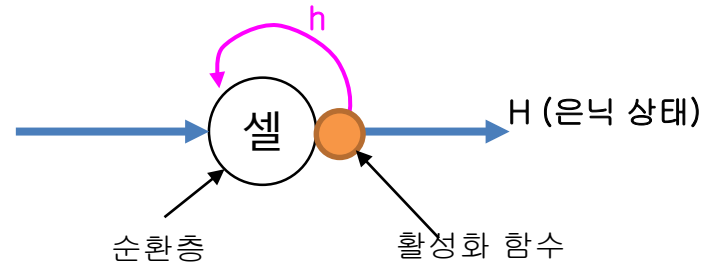
- O_C 가 계산될 때, O_B 도 함께 포함됨
 - 이때 O_B 는 O_A 를 포함하고 있음



타임스텝
(timestep)

순환 신경망 (Recurrent Neural Network)

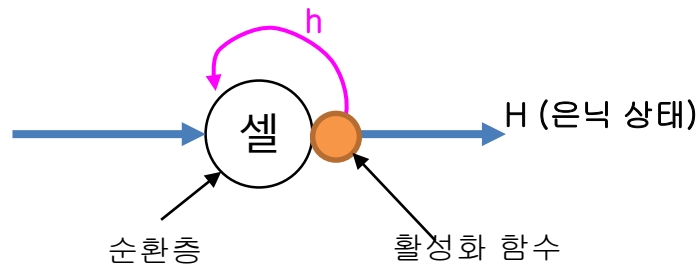
- 순환 신경망에서는 층을 셀(cell)이라고 부름
 - 한 셀에는 여러 개의 뉴런이 있지만, 모두 표시하지 않고 하나의 셀로 표현함
 - 셀의 출력 상태를 은닉 상태라 부름



- 그림이 설명하는 바:
 - 입력에 어떤 가중치를 곱하고, 활성화 함수를 통과시켜 다음 층으로 보냄
 - 기존과 다른 점은 층의 출력 (은닉 상태)를 다음 타임 스텝에 재사용하는 것

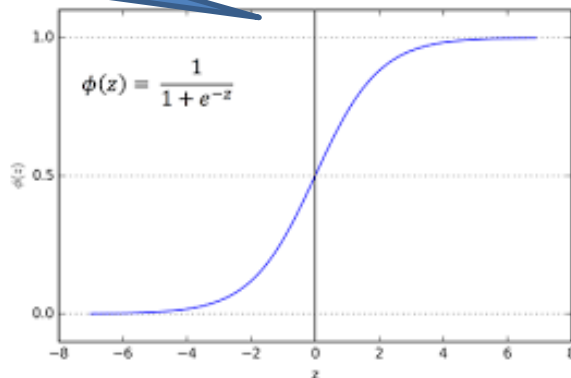
순환 신경망 (Recurrent Neural Network)

- 은닉층의 활성화 함수로는 하이퍼볼릭 탄젠트 (hyperbolic tangent, tanh) 함수가 주로 사용됨

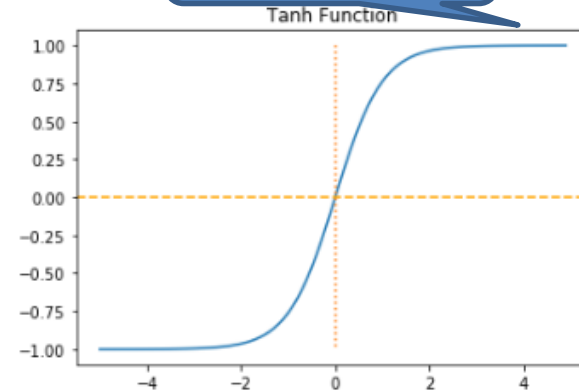


- Tanh는 -1 ~ 1의 범위를 가짐 (시그모이드는 0~1)

시그모이드



하이퍼볼릭
탄젠트

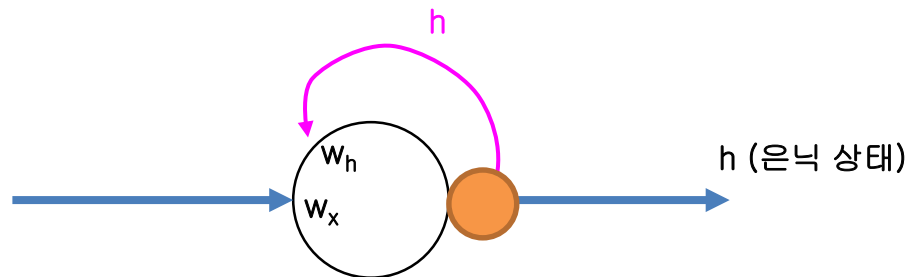


순환 신경망 (Recurrent Neural Network)

- 완전 연결 신경망과 합성곱에서는
 - 입력과 가중치를 곱했음

$$e.g.) z = 3*w1 + 1*w2 + 0*w3 + b \text{ (절편)}$$

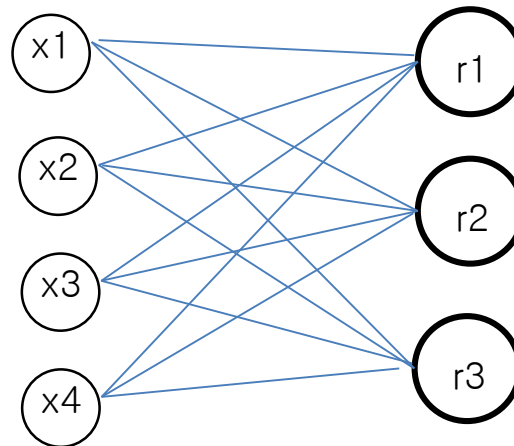
- 순환 신경망도 비슷
 - 대신 은닉 상태에 곱해지는 가중치 (w_h)가 하나 더 있음



$$e.g.) z = 3*w1*wh + b \text{ (절편)}$$

순환 신경망 (Recurrent Neural Network)

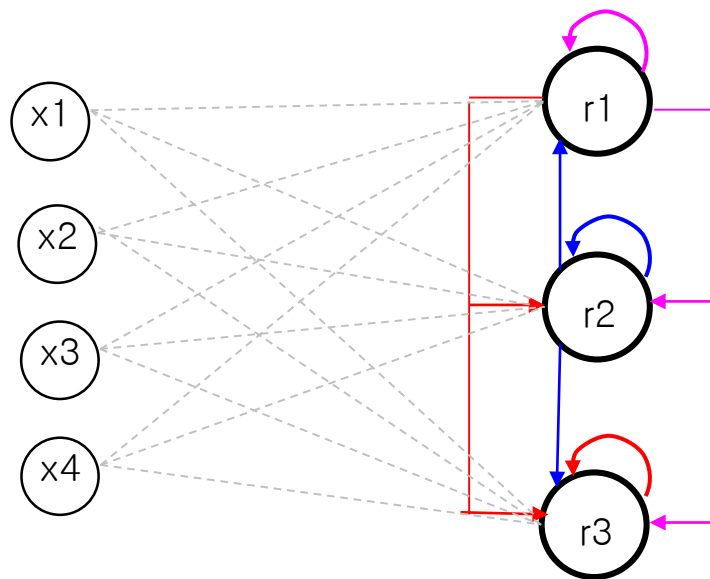
- 입력층 4개 뉴런과 순환층 3개일 경우
 - 가중치 w_x 의 크기는 $3 \times 4 = 12$ 개 임
 - 즉, 완전 연결 층만 본다면... (순환은 제외하고)



입력 x 가중치

순환 신경망 (Recurrent Neural Network)

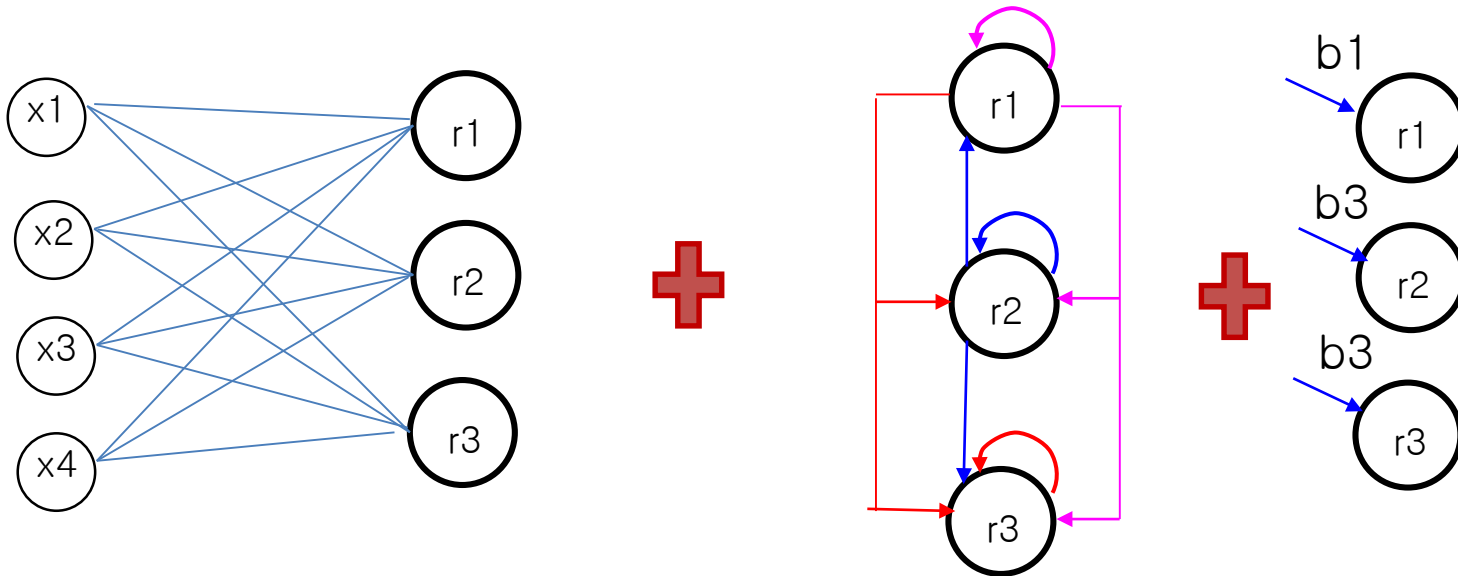
- 순환층에서의 은닉 상태 $w_h = 3 \times 3 = 9$
 - R1에 3개, R2에 3개, R3에 3개의 재사용 은닉 상태가 있음
 - 즉 순환 은닉 상태가 입력층에 모두 연결됨
 - 즉, 이웃한 뉴런에서 재사용 은닉 상태(w_h)가 입력 됨



층의 뉴런이 3개일 경우
재사용되는 w_h 가 3개 더
곱해짐

순환 신경망 (Recurrent Neural Network)

- 가중치 w_x 12 개 + 재사용 은닉상태 w_h 9개 + 절편 3개
→ 24

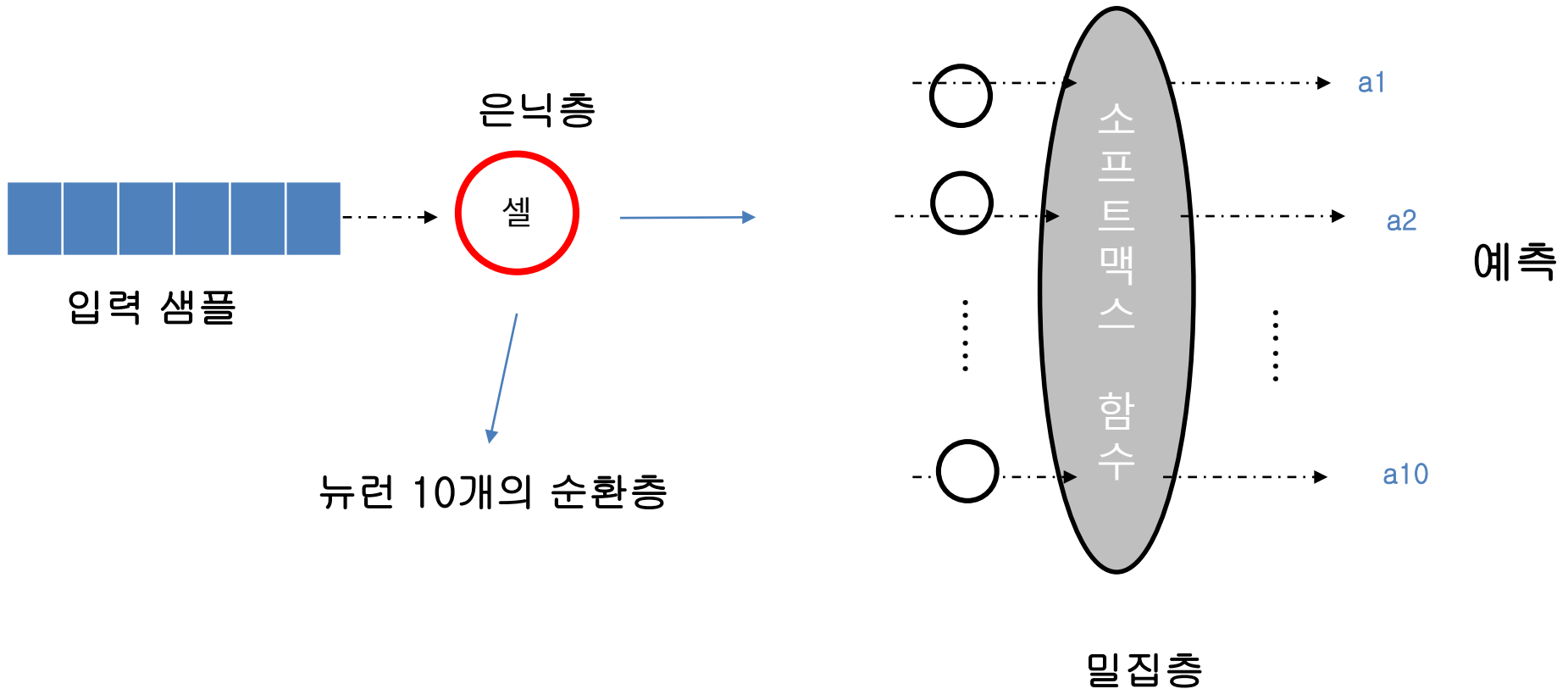


- 신경망이 복잡할 수록 표현이 어려워 진다

e.g.) $z = 3 * w1 * wh + b$ (절편)

순환 신경망 (Recurrent Neural Network)

- 간략화된 표현 방법 → 은닉층을 '셀' 로 표현



요약

- 순차 데이터

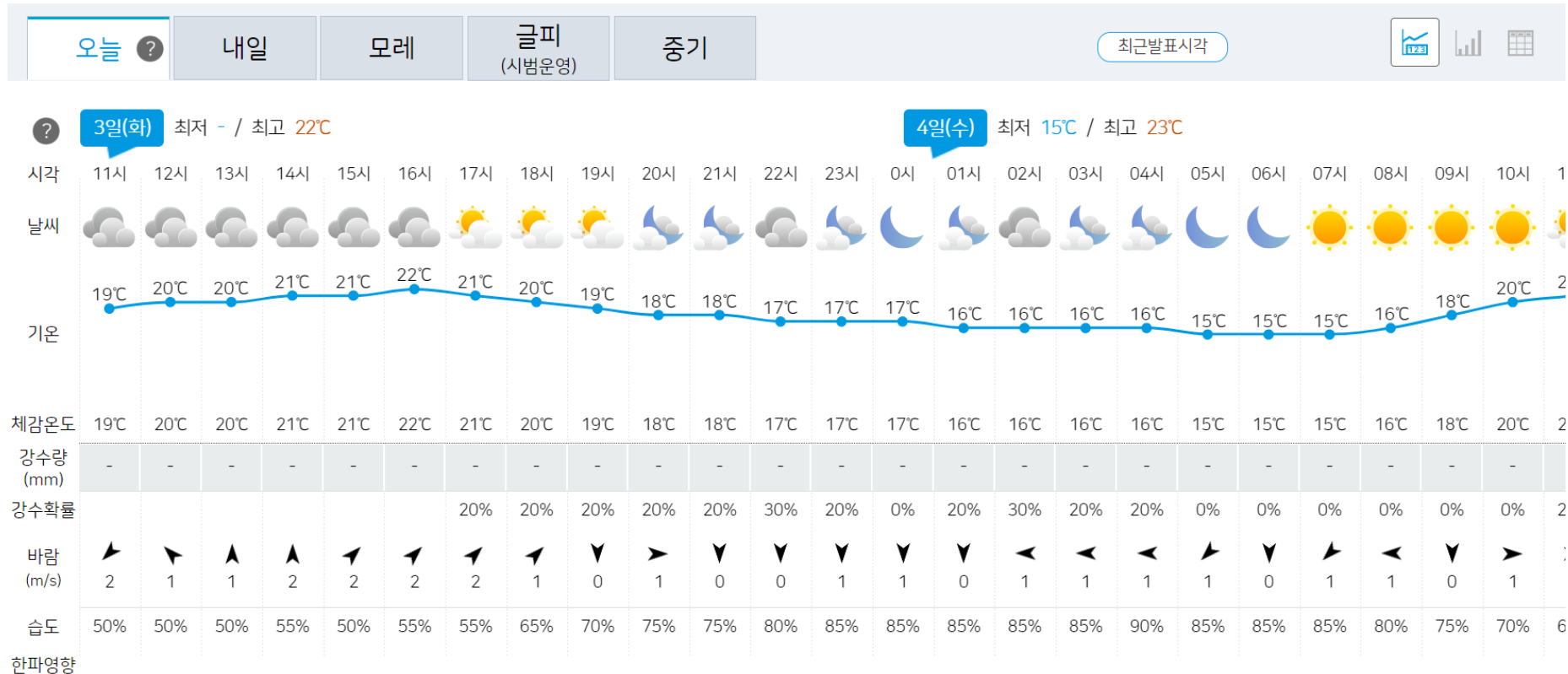
- 텍스트나 시계열 데이터와 같이 순서에 의미가 있는 데이터

- 순환 신경망

- 순차 데이터에 잘 맞는 인공 신경망
- 순환 신경망에서는 순환층을 '셀' 이라 부름
 - 하나의 셀은 여러 개의 뉴런으로 구성이 됨
- 순환 신경망에서는 셀의 출력을 **은닉 상태**라고 부름

LSTM을 활용한 일기예보

- LSTM은 시계열 데이터의 학습에 뛰어난 딥러닝 모델



1. 데이터셋 확인하기

- 2009년 1월 1일 ~ 2016년 12월 31일까지 데이터가 포함 (420,523개)
 - 14개의 특징 (features)
 - 기압, 온도, 등.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Date Time	p (mbar)	T (degC)	Tpot (K)	Tdew (degC)	rh (%)	VPmax (mbar)	VPact (mbar)	VPdef (mbar)	sh (g/kg)	H2OC (mmol/mc)	rho (g/m**3)	wv (m/s)	max. wv (m/s)	wd (deg)
2	01.01.2009 00:1	996.52	-8.02	265.4	-8.9	93.3	3.33	3.11	0.22	1.94	3.12	1307.75	1.03	1.75	152.3
3	01.01.2009 00:2	996.57	-8.41	265.01	-9.28	93.4	3.23	3.02	0.21	1.89	3.03	1309.8	0.72	1.5	136.1
4	01.01.2009 00:3	996.53	-8.51	264.91	-9.31	93.9	3.21	3.01	0.2	1.88	3.02	1310.24	0.19	0.63	171.6
5	01.01.2009 00:4	996.51	-8.31	265.12	-9.07	94.2	3.26	3.07	0.19	1.92	3.08	1309.19	0.34	0.5	198
6	01.01.2009 00:5	996.51	-8.27	265.15	-9.04	94.1	3.27	3.08	0.19	1.92	3.09	1309	0.32	0.63	214.3
7	01.01.2009 01:0	996.5	-8.05	265.38	-8.78	94.4	3.33	3.14	0.19	1.96	3.15	1307.86	0.21	0.63	192.7
8	01.01.2009 01:1	996.5	-7.62	265.81	-8.3	94.8	3.44	3.26	0.18	2.04	3.27	1305.68	0.18	0.63	166.5
9	01.01.2009 01:2	996.5	-7.62	265.81	-8.36	94.4	3.44	3.25	0.19	2.03	3.26	1305.69	0.19	0.5	118.6
10	01.01.2009 01:3	996.5	-7.91	265.52	-8.73	93.8	3.36	3.15	0.21	1.97	3.16	1307.17	0.28	0.75	188.5
11	01.01.2009 01:4	996.53	-8.43	264.99	-9.34	93.1	3.23	3	0.22	1.88	3.02	1309.85	0.59	0.88	185
12	01.01.2009 01:5	996.62	-8.76	264.66	-9.66	93.1	3.14	2.93	0.22	1.83	2.94	1311.64	0.45	0.88	183.2
13	01.01.2009 02:0	996.62	-8.88	264.54	-9.77	93.2	3.12	2.9	0.21	1.81	2.91	1312.25	0.25	0.63	190.3
14	01.01.2009 02:1	996.63	-8.85	264.57	-9.7	93.5	3.12	2.92	0.2	1.82	2.93	1312.11	0.16	0.5	158.3
15	01.01.2009 02:2	996.74	-8.83	264.58	-9.68	93.5	3.13	2.92	0.2	1.83	2.93	1312.15	0.36	0.63	184.8
16	01.01.2009 02:3	996.81	-8.66	264.74	-9.46	93.9	3.17	2.98	0.19	1.86	2.99	1311.37	0.33	0.75	155.9
17	01.01.2009 02:4	996.81	-8.66	264.74	-9.5	93.6	3.17	2.97	0.2	1.85	2.98	1311.38	0.07	0.5	272.4
18	01.01.2009 02:5	996.86	-8.7	264.7	-9.55	93.5	3.16	2.95	0.21	1.85	2.96	1311.64	0.32	0.63	219.2
19	01.01.2009 03:0	996.84	-8.81	264.59	-9.66	93.5	3.13	2.93	0.2	1.83	2.94	1312.18	0.18	0.63	167.2
20	01.01.2009 03:1	996.87	-8.84	264.56	-9.69	93.5	3.13	2.92	0.2	1.83	2.93	1312.37	0.07	0.25	129.3
21	01.01.2009 03:2	996.97	-8.94	264.45	-9.82	93.3	3.1	2.89	0.21	1.81	2.9	1313.01	0.1	0.63	115.3
22	01.01.2009 03:3	997.08	-8.94	264.44	-9.8	93.4	3.1	2.9	0.2	1.81	2.9	1313.15	0.3	0.75	149.3
23	01.01.2009 03:4	997.1	-8.86	264.52	-9.76	93.1	3.12	2.9	0.22	1.81	2.91	1312.78	0.29	0.75	149.7
24	01.01.2009 03:5	997.06	-8.99	264.39	-9.99	92.4	3.09	2.85	0.23	1.78	2.86	1313.39	0.12	0.63	231.7
25	01.01.2009 04:0	996.99	-9.05	264.34	-10.02	92.6	3.07	2.85	0.23	1.78	2.85	1313.61	0.1	0.38	240
26	01.01.2009 04:1	997.05	-9.23	264.15	-10.25	92.2	3.03	2.79	0.24	1.74	2.8	1314.62	0.1	0.38	203.9
27	01.01.2009 04:2	997.11	-9.49	263.89	-10.54	92	2.97	2.73	0.24	1.71	2.74	1316.02	0.34	0.75	159.7
28	01.01.2009 04:3	997.19	-9.5	263.87	-10.51	92.3	2.97	2.74	0.23	1.71	2.75	1316.16	0.43	0.88	66.16
29	01.01.2009 04:4	997.24	-9.35	264.02	-10.29	92.8	3	2.79	0.22	1.74	2.79	1315.47	0.4	0.88	105
30	01.01.2009 04:5	997.37	-9.47	263.89	-10.46	92.4	2.97	2.75	0.23	1.72	2.75	1316.25	0.37	0.75	125.8
31	01.01.2009 05:0	997.46	-9.63	263.72	-10.65	92.2	2.94	2.71	0.23	1.69	2.71	1317.19	0.4	0.88	157
32	01.01.2009 05:1	997.43	-9.67	263.68	-10.63	92.6	2.93	2.71	0.22	1.69	2.72	1317.35	0.36	0.75	132.5
33	01.01.2009 05:2	997.42	-9.68	263.67	-10.73	92	2.92	2.69	0.23	1.68	2.7	1317.4	0.09	0.5	143.2
34	01.01.2009 05:3	997.53	-9.9	263.45	-10.98	91.7	2.87	2.64	0.24	1.64	2.64	1318.68	0.29	1	72.5
35	01.01.2009 05:4	997.6	-9.91	263.43	-10.9	92.4	2.87	2.65	0.22	1.66	2.66	1318.81	0.5	1	60.72
36	01.01.2009 05:5	997.62	-9.51	263.83	-10.37	93.4	2.96	2.77	0.2	1.73	2.77	1316.77	0.31	0.75	147.4
37	01.01.2009 06:0	997.71	-9.67	263.66	-10.62	92.7	2.93	2.71	0.21	1.69	2.72	1317.71	0.05	0.5	146
38	01.01.2009 06:1	997.81	-9.59	263.74	-10.47	93.2	2.95	2.74	0.2	1.71	2.75	1317.44	0.24	0.75	161.6
39	01.01.2009 06:2	997.86	-9.15	264.17	-10.02	93.3	3.05	2.85	0.2	1.78	2.85	1315.26	0.18	0.5	125.7
40	01.01.2009 06:3	998	-8.91	264.4	-9.89	92.5	3.11	2.87	0.23	1.79	2.88	1314.23	0.45	0.88	67.1

2. 데이터 전처리

- 가. 모든 데이터가 0~1사이의 값을 가지도록 정규화 (normalization)
 - A. 온도의 경우 음수를 포함 →
 - 최소값을 뺀 후, 최대값으로 나누어 전처리 함
- 나. 데이터 윈도우 (windowing)
 - 미래를 예측하기 위해 사용되는 기법
 - 윈도우잉이란 ?

원본 데이터

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

Frame 1

1	2	3	4	5	6
---	---	---	---	---	---

Frame 2

2	3	4	5	6	7
---	---	---	---	---	---

Frame 3

3	4	5	6	7	8
---	---	---	---	---	---

2. 데이터 전처리

- 미래예측에 활용

- 과거 데이터 vs. 미래 데이터

원본 데이터



Frame 1



Frame 2



Frame 3

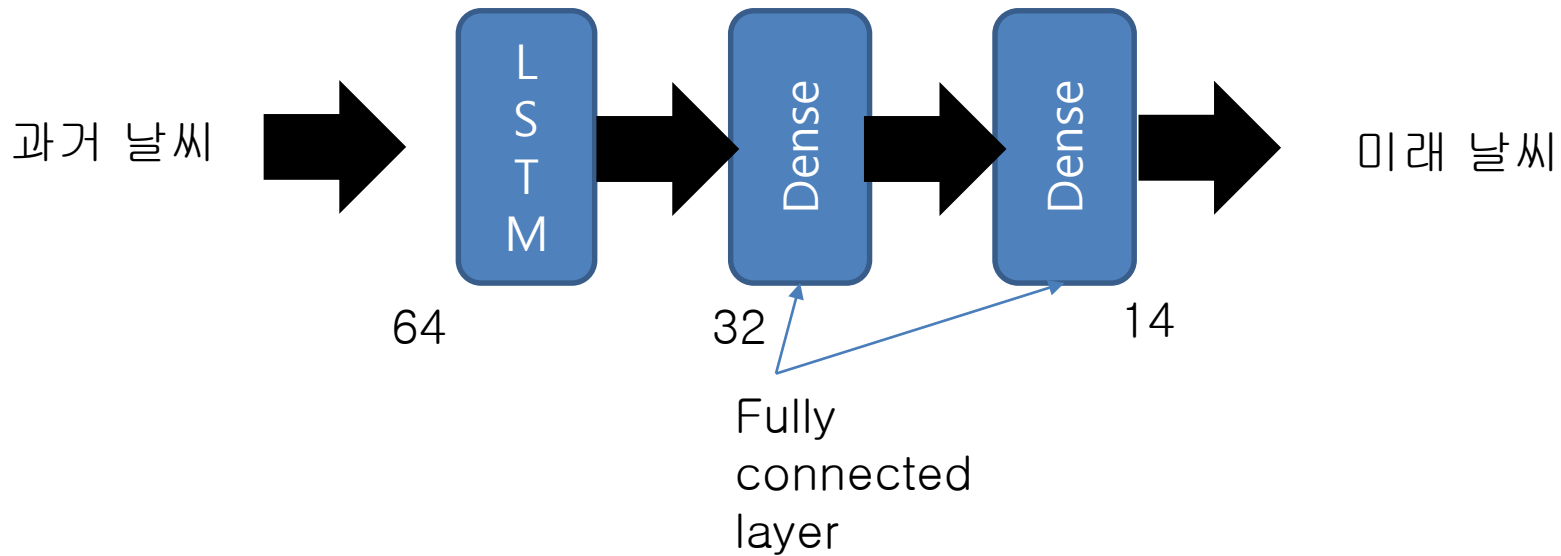


과거 데이터

미래 데이터

즉, AI 가 과거 데이터를 학습하게 하여, 미래 데이터를 예측하는 모델을 만들 것임

3. 인공지능 모델



* 본 예는 간략화하기 위해 별도의 활성화 함수를 사용하지 않음

데이터 준비하기

- Jena_climate_2009_2019.csv 파일 업로드
 - 마우스로 드래그

파일



- ..
- sample_data
- data_reader.py
- jena_climate_2009_2016.csv
- result.png
- train_history.png

파일 업로드에 시간이
꽤 걸림

+ 코드 + 텍스트

✓
0초

[43]

```
plt.title("Regression Result")
plt.xlabel("Ground Truth")
plt.ylabel("AI Predict")
plt.scatter(X, Y)
plt.plot([minval, maxval], [minval, maxval], "
fig.savefig("result.png")

train_history = history.history["loss"]
validation_history = history.history["val_loss"]
fig = plt.figure(figsize=(8, 8))
plt.title("Loss History")
plt.xlabel("EPOCH")
plt.ylabel("LOSS Function")
plt.plot(train_history, "red")
plt.plot(validation_history, 'blue')
```

프로그램 코드



```
#import data_reader
from tensorflow import keras
```

```
# 몇 에포크 만큼 학습을 시킬 것인지 결정합니다.
```

```
EPOCHS = 50 # 예제 기본값은 50입니다
```

DataReader class은 첨부파일 참고

```
# 데이터를 읽어옵니다.
```

```
dr = DataReader(12)
```

Param "12"는 사용할 과거 데이터 갯수

```
# 인공신경망을 제작합니다.
```

```
model = keras.Sequential([
    keras.layers.Bidirectional(keras.layers.LSTM(64, return_sequences=True)),
    keras.layers.Dense(32),
    keras.layers.Dense(14),
])
```

딥러닝 모델

```
# 인공신경망을 컴파일합니다.
```

```
model.compile(optimizer="adam", metrics=["mae"], loss="mse")
```

컴파일 옵션

```
# 인공신경망을 학습시킵니다.
```

```
print("\n\n***** TRAINING START ***** ")
early_stop = keras.callbacks.EarlyStopping(monitor='val_loss', patience=10)
history = model.fit(dr.train_X, dr.train_Y, epochs=EPOCHS,
                    validation_data=(dr.test_X, dr.test_Y),
                    callbacks=[early_stop])
```

train

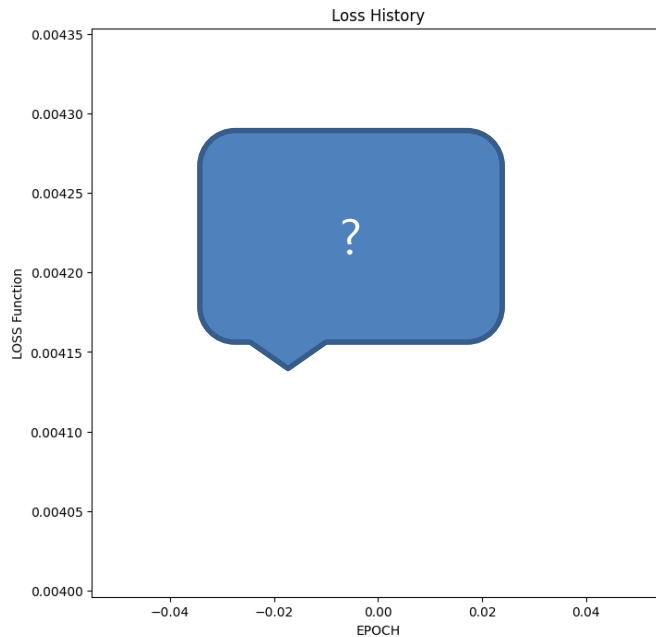
```
# 학습 결과를 그래프로 출력합니다.
```

```
draw_graph(model(dr.test_X[:200]), dr.test_Y[:200], history)
```

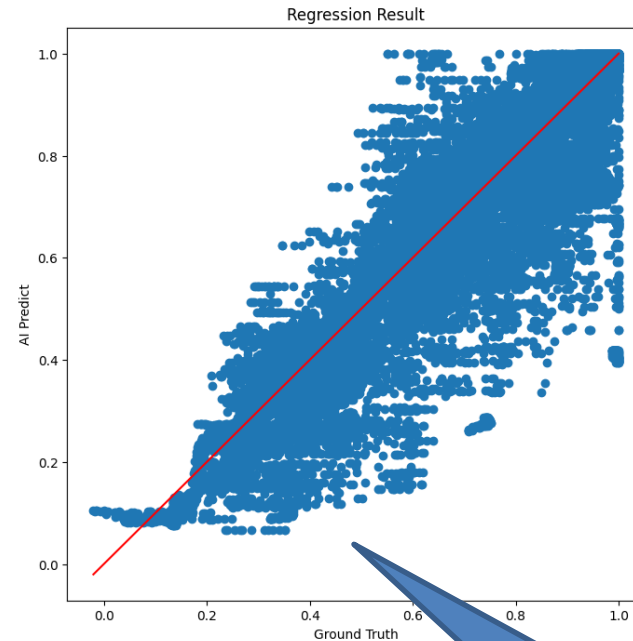
학습결과 그래프
(200개 데이터만)

결과

- Loss 가 어떻게 나오는지 확인
 - 30 에포크에서 학습이 중단됨을 알 수 있음
 - 원하는 성능을 조기 이룸



LSTM이 97%의 정확도로
날씨를 예측



직선을 따라 점들이
고르게 분포

실습해보기

인공지능 (Artificial Intelligence)

시계열 딥러닝

Hyuntae Cho

Dept. of Digital Media Engineering

Tongmyong University

LSTM을 활용한 주가 예측

- 애플의 주가 예측해보기



1. 데이터셋 확인하기

- 2010년 1월 4일 ~ 2020년 8월 10일까지 나스닥 거래 주가
 - 6개의 특징 (features)

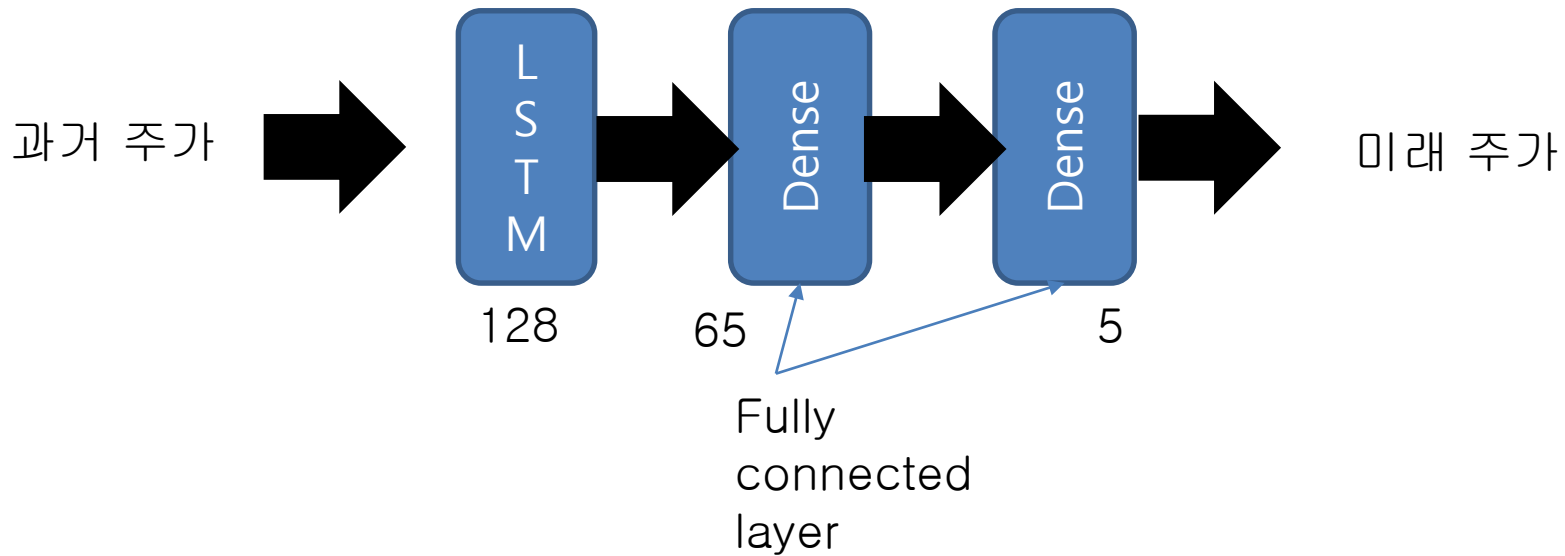
A	B	C	D	E	F	G	H
Date	Open	High	Low	Close	Adj Close	Volume	
2010-01-04	30.49	30.642857	30.34	30.572857	26.419203	123432400	
2010-01-05	30.657143	30.798571	30.464285	30.625713	26.464878	150476200	
2010-01-06	30.625713	30.747143	30.107143	30.138571	26.043921	138040000	
2010-01-07	30.25	30.285715	29.864286	30.082857	25.995777	119282800	
2010-01-08	30.042856	30.285715	29.865715	30.282858	26.168606	111902700	
2010-01-11	30.4	30.428572	29.778572	30.015715	25.937763	115557400	
2010-01-12	29.884285	29.967142	29.488571	29.674286	25.642714	148614900	
2010-01-13	29.695715	30.132856	29.157143	30.092857	26.004419	151473000	
2010-01-14	30.015715	30.065714	29.860001	29.918571	25.853821	108223500	
2010-01-15	30.132856	30.228571	29.41	29.418571	25.421745	148516900	
2010-01-19	29.761429	30.741428	29.605715	30.719999	26.546362	182501900	
2010-01-20	30.701429	30.792856	29.928572	30.247143	26.137741	153038200	
2010-01-21	30.297142	30.472857	29.601429	29.724285	25.685919	152038600	
2010-01-22	29.540001	29.642857	28.165714	28.25	24.411934	220441900	
2010-01-25	28.93	29.242857	28.598572	29.01	25.068684	266424900	
2010-01-26	29.421429	30.530001	28.940001	29.42	25.422976	466777500	
2010-01-27	29.549999	30.082857	28.504286	29.697144	25.662466	430642100	
2010-01-28	29.275715	29.357143	28.385714	28.469999	24.602047	293375600	
2010-01-29	28.725714	28.885714	27.178572	27.437143	23.709522	311488100	
2010-02-01	27.481428	28	27.328571	27.818571	24.039122	187469100	
2010-02-02	27.987143	28.045713	27.625713	27.98	24.178619	174585600	
2010-02-03	27.881428	28.6	27.774286	28.461428	24.594633	153832000	
2010-02-04	28.104286	28.338572	27.367144	27.435715	23.708282	189413000	
2010-02-05	27.518572	28	27.264286	27.922857	24.129242	212576700	
2010-02-08	27.955715	28.268572	27.714285	27.731428	23.96382	119567700	
2010-02-09	28.059999	28.214285	27.821428	28.027143	24.219357	158221700	
2010-02-10	27.984285	28.085714	27.751429	27.874287	24.087267	92590400	
2010-02-11	27.84	28.535715	27.722857	28.381428	24.525509	137586400	
2010-02-12	28.301428	28.805714	27.928572	28.625713	24.736601	163867200	
2010-02-16	28.848572	29.098572	28.788572	29.057142	25.109415	135934400	
2010-02-17	29.17	29.187143	28.694286	28.935715	25.004488	109099200	
2010-02-18	28.804285	29.127142	28.702858	28.99	25.051394	105706300	
2010-02-19	28.837143	29.028572	28.73	28.809999	24.895855	103867400	

2. 데이터셋 전처리

- 0~1로 정규화 (normalization)
- 훈련데이터 95%, 테스트 데이터 5%
- 윈도우크기 14
 - (14일간 학습, 14일 예측)
- 6개의 특징을 입력 받아 5개의 특징을 출력 (예측)하고
자 함
 - 미래의 거래량은 의미 없으므로 제외

3. 인공지능 모델

- 전과 동일



* 본 예는 간략화하기 위해 별도의 활성화 함수를 사용하지 않음

프로그램 코드

```
from tensorflow import keras

# 몇 에포크 만큼 학습을 시킬 것인지 결정합니다.
EPOCHS = 100 # 예제 기본값은 100입니다.

# 데이터를 읽어옵니다.
dr = DataReader(14)

# 인공신경망을 제작합니다.
model = keras.Sequential([
    keras.layers.Bidirectional(keras.layers.LSTM(128, return_sequences=True)),
    keras.layers.Dense(64, activation="relu"),
    keras.layers.Dense(5)
])

# 인공신경망을 컴파일합니다.
model.compile(optimizer="adam", loss="mae", metrics=["mse"])

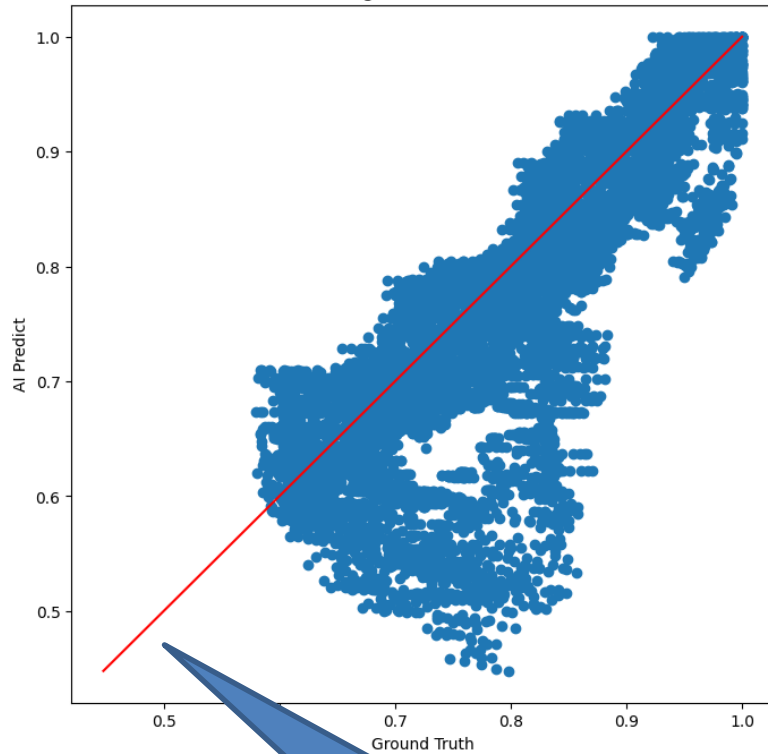
# 인공신경망을 학습시킵니다.
print("\n\n***** TRAINING START ***** ")
early_stop = keras.callbacks.EarlyStopping(monitor='val_loss', patience=20)
history = model.fit(dr.train_X, dr.train_Y, epochs=EPOCHS,
                    validation_data=(dr.test_X, dr.test_Y),
                    callbacks=[early_stop])

# 학습 결과를 그래프로 출력합니다.
draw_graph(model(dr.test_X), dr.test_Y, history)
```

결과

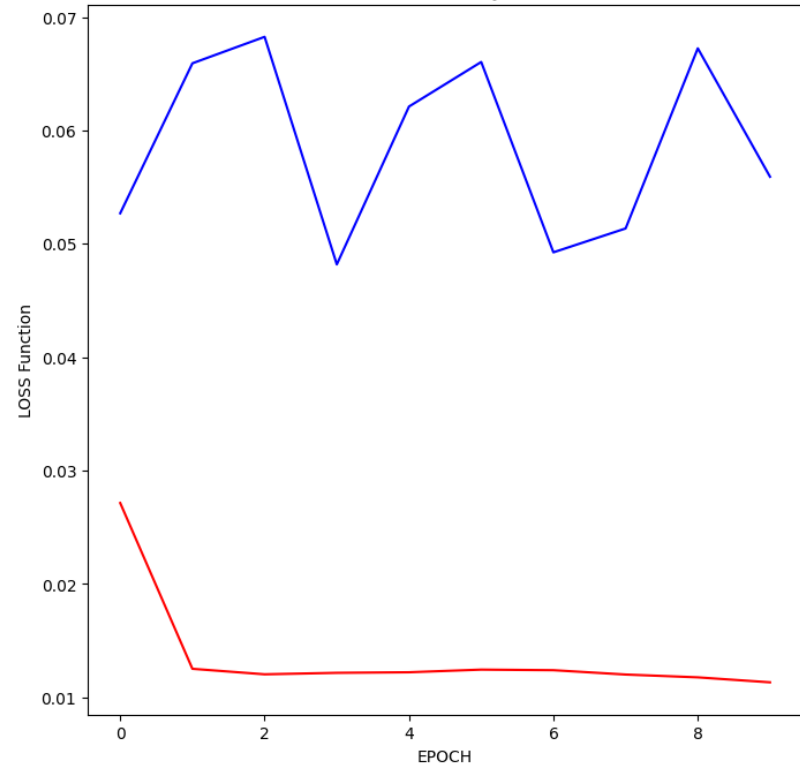
89 에포크에서 중단

Regression Result



직선을 따라 점들이
고르게 분포

Loss History



요약

- 인공지능의 최적 성능은 MAE 0.31%,
- 99.7% 수준의 정확도로 주가를 예측할 수 있음을 의미

실습해보기

Q&A
