

Machine Learning

2장 딥러닝 데이터와 세 가지 기본신경망

고려대학교 통계학과
박유성



Contents

01 딥러닝의 개념 및 분석절차

02 딥러닝에 사용되는 데이터의 형태

03 세 가지 핵심 신경망

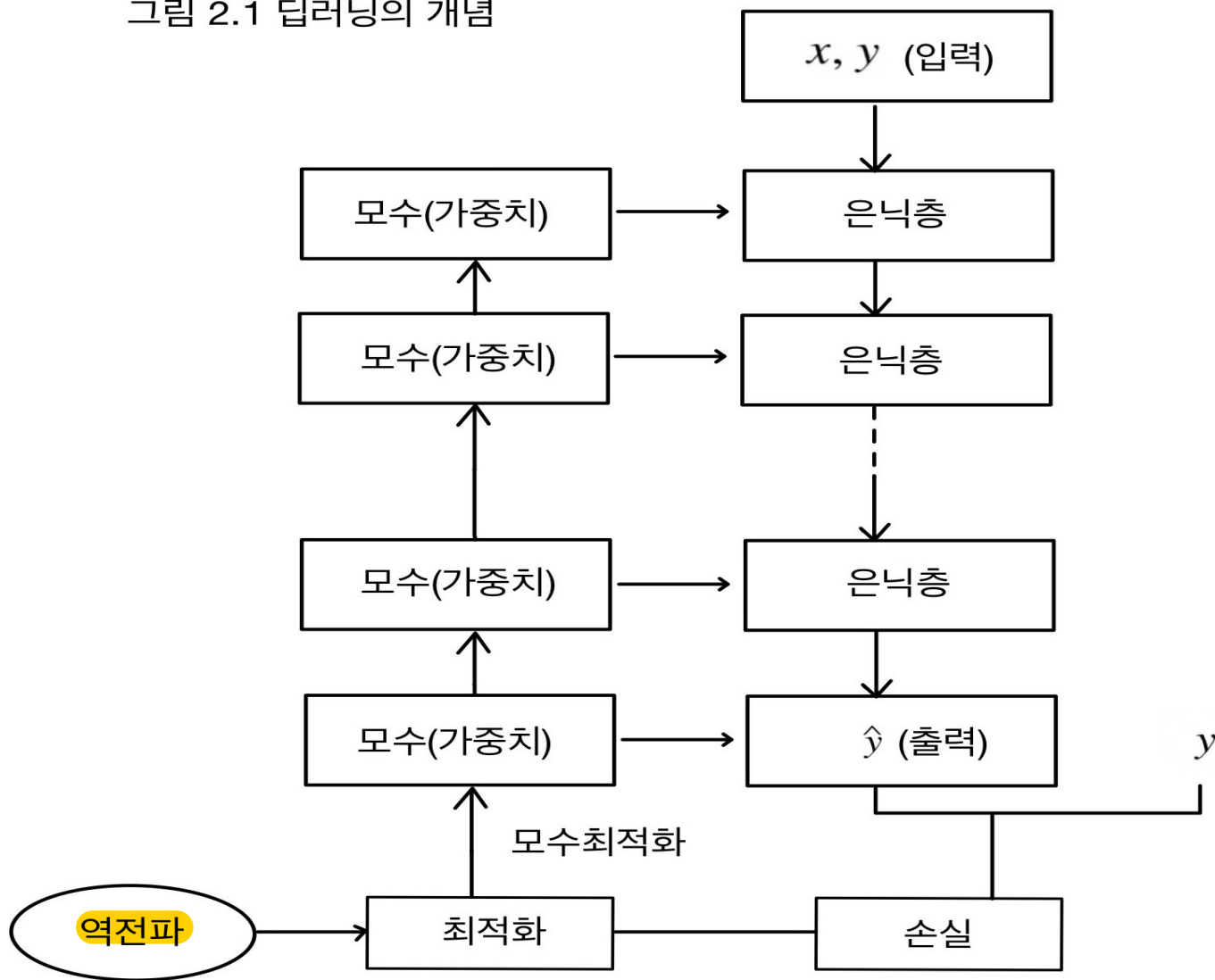
01 딥러닝의 개념 및 분석절차

딥러닝의 영역

1. **지도학습**: 사진과 같은 이미지 자료에 있는 개, 고양이, 건물, 사람 등과 같은 객체를 구별하고 인식하는 객체인식, 언어나 문장에 의한 주제 분류, 번역 및 통역
2. 비지도학습: 트윗이나 구매내역에 따른 고객분류, 흐릿한(잡음이 있는) 영상이나 이미지의 복원, 단어들의 상관관계를 구현하는 특성변수의 변환, 차원축소를 위한 특성변수의 축약(compression)
3. **스스로 지도학습**(self-supervised learning): 모네 그림을 그대로 재현. 여러 사람의 이미지로부터 새로운 사람의 이미지를 생성, 새로운 소설, 영화
4. **강화학습**(reinforcement learning): 주어진 환경에서 최대의 보상을 받는 행동(action)을 학습하는 머신러닝 영역

딥러닝의 개념

그림 2.1 딥러닝의 개념



딥러닝 분석절차

- 학습데이터를 k 개로 분할. 이를 batch하고 함. 배치의 크기는 일반적으로 32 또는 64개의 표본으로 구성한다
- 1. 딥러닝 모델을 구축하고 딥러닝 모델의 모수에 임의의 값을 할당한다.
- 2. 학습데이터로부터 한 배치의 X, y 를 추출. X 를 특성변수, y 를 목적변수.
- 3. 특성변수 X 로 목적변수 y 를 예측. 실제 y 와 예측치의 거리를 측정하는 손실 함수 계산
- 4. 모수에 대해 손실함수의 미분 값을 산출한 후, 손실함수 미분값의 음의 방향으로 모수값을 최신회한다.
- 5. 2~4를 k 번 반복. 이를 1 에폭(epoch)이 완성되었다고 한다.

딥러닝 분석절차

6. 2~5를 반복하여 모형의 정밀도를 원하는 수준만큼 높이고 검증데이터를 통해 초모수를 조절한다든가, 모수에 제한 조건을 부여하여 모형을 튜닝한다.
7. 시험데이터를 이용하여 모형의 과대적합(overfitting)을 점검한다.
8. 과대적합이 발생하면 모수의 규제화, dropout, 배치정규화(batch normalization) 등을 통해 과대적합문제를 해결한다.

데이터의 사전정리과정

- 자료의 벡터화(vectorization): one-hot coding
- 특성변수는 정규화(normalization)을 원칙으로 한다.

$$\frac{x_j - \bar{x}}{sd(x_j)}$$

- 특성변수의 표준화(standardization)

$$\frac{x_j - \min(x_j)}{\max(x_j) - \min(x_j)}$$

- 특성변수는 결측치가 없어야 한다. 딥러닝에서는 0으로 대체가 안전함.
- 모수추정에 사용하는 역전파에서 특성변수값 0은 모수의 최신화(update)에 기여하지 않기 때문이다.

딥러닝에 사용되는 데이터의 형태

- 딥러닝 데이터는 텐서(tensor)로 입력되고 출력된다.

- 텐서는 행렬의 일반화

스칼라: 0차원 텐서(0-dimensional tensor, 0D텐서)

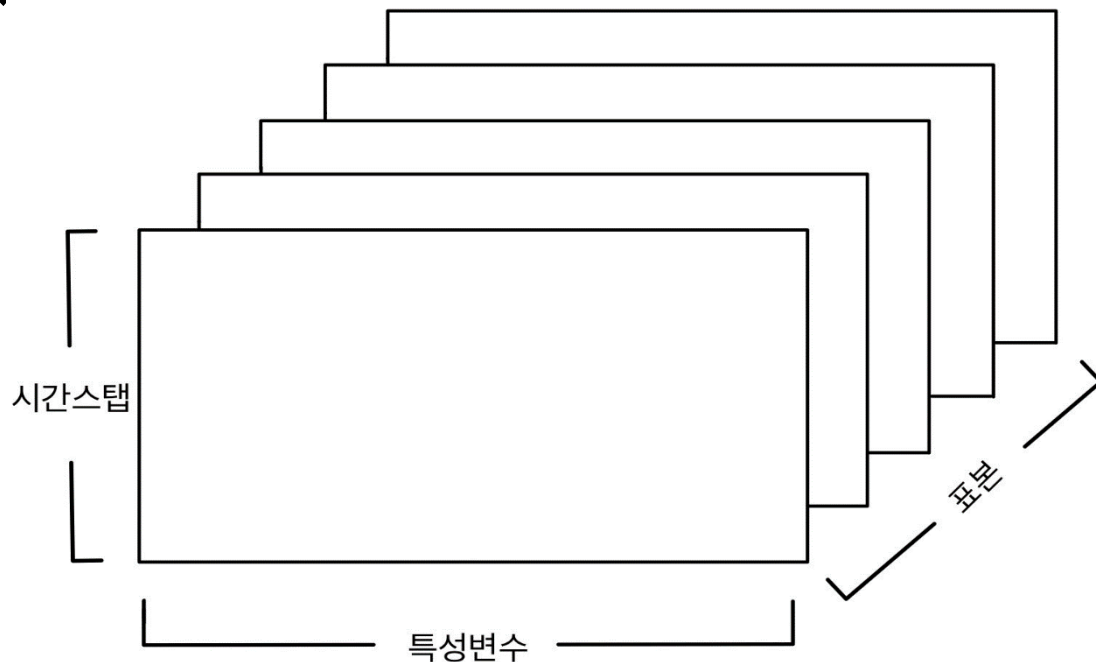
벡터: 1차원 텐서(1D 텐서),

행렬: 2차원 텐서(2D텐서)

- 3차원 데이터는 3D텐서이고 4차원 데이터는 4D텐서가 된다.

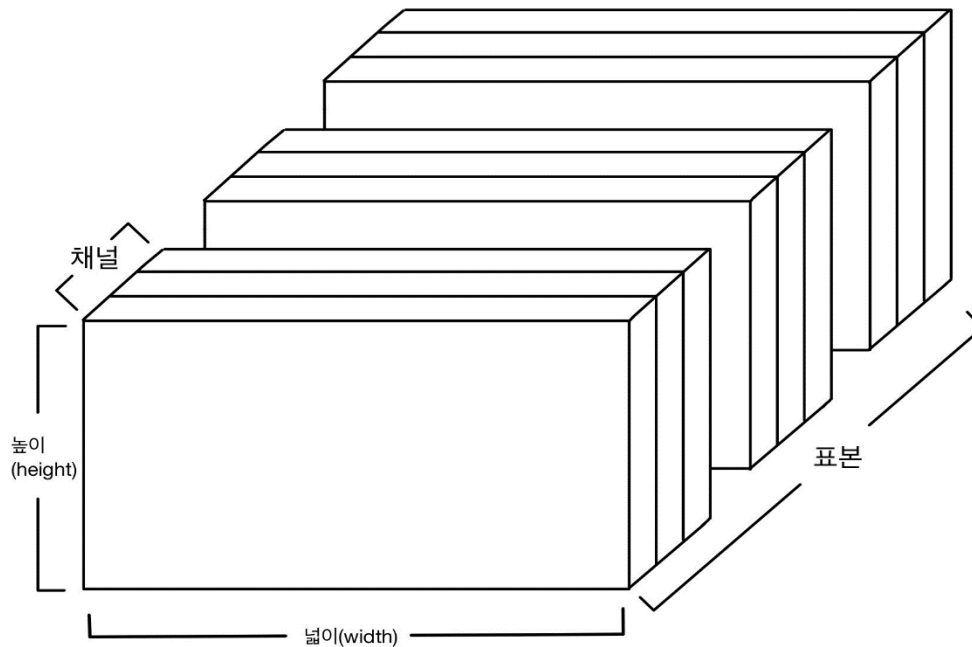


- (표본수, 특성변수수)로 표현된다.
- 100명의 사람 각각에 대한 연령, 성별, 수입 자료: (100, 3) 2D텐서
- 20,000개의 단어로 표현된 500개의 문서: (500, 20000) 2D텐서
- keras에서는 표본의 수를 나타내는 첫 번째 축의 값은 생략함
- 즉, (100,3)은 (3,)으로 입력하고 (500,20000)은 (20000,)으로 입력
- 2D텐서는 MLP 모형에서 사용



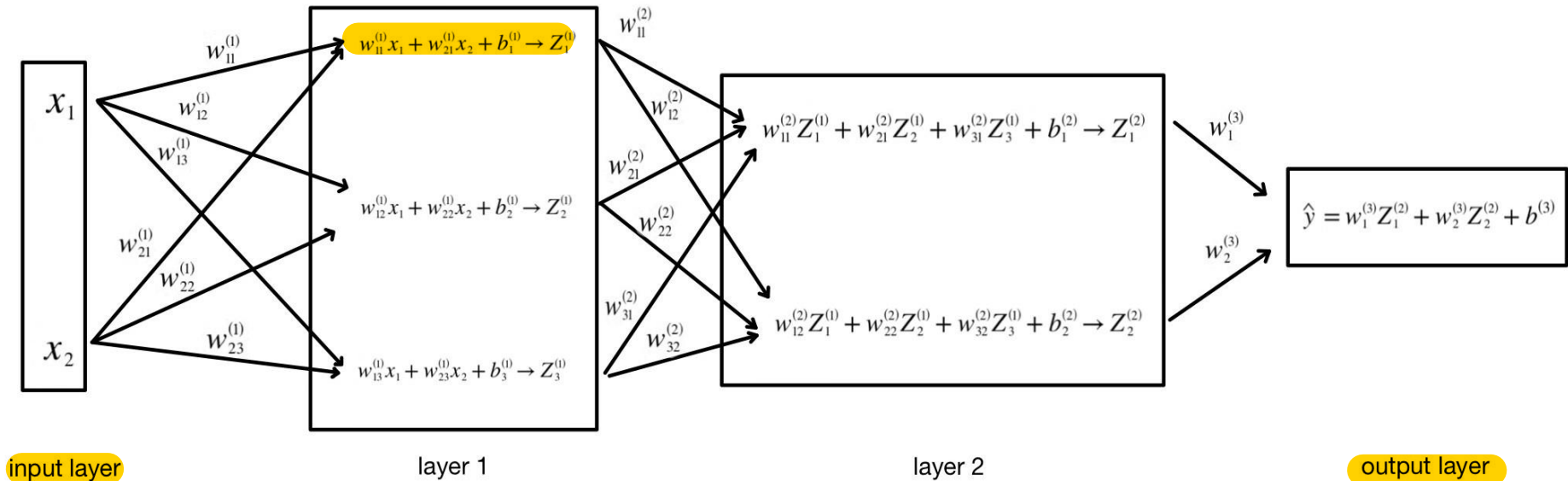
- 시간스텝(time steps): 시간의 순서나 단어의 순서가 자료 분석에 중요한 정보를 가지고 있음.
- 하루 거래 시간 기준으로 390분 동안의 주식가격, 최대가격, 최소가격 데이터를 100일 동안 관측: 표본 수는 100, 시간순서는 390, 그리고 특성변수의 수는 3이 되어 크기가 (100,390,3)인 3D 텐서.

4D 텐서



- 위 그림은 3장의 컬러이미지 자료를 도식화한 것임.
- (표본수, 높이, 넓이, 채널수) 또는 (표본수, 채널수, 높이, 넓이) 형태로 제공되지만 (표본수, 높이, 넓이, 채널수) 형태가 좀 더 일반적으로 사용된다.
- 78× 78 픽셀 컬러이미지가 5,000장: (5000,78,78,3) 4D 텐서
- 이 이미지 자료가 흑백: (5000,78,78,1) 4D 텐서
- keras에서는 표본수를 제외하고 (78,78,3) 또는 (78,78,1)로 자료가 입력.

MLP모형의 구조



$$Z_i^{(1)} = \sigma(w_{1i}^{(1)}x_1 + w_{2i}^{(1)}x_2 + b_i^{(1)})$$

$$i = 1, 2, 3$$

$$Z_i^{(2)} = \sigma(w_{1i}^{(2)}Z_1^{(1)} + w_{2i}^{(2)}Z_2^{(1)} + w_{3i}^{(2)}Z_3^{(1)} + b_i^{(2)})$$

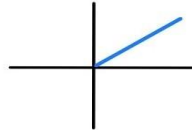
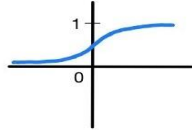
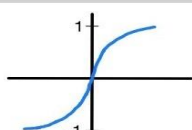
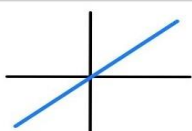
$$i = 1, 2$$

- 1D텐서인 특성변수가 입력되면 이 특성변수를 선형결합하고 이 선형결합에 활성화함수(activation function)를 적용하여 비선형 변환한다. 이 값은 0D 텐서(즉, 스칼라)이며 은닉층(hidden layer)의 첫 번째 출력값이 된다.

MLP

- 또 다른 선형결합과 활성화함수에 의한 비선형변환 해서 은닉층의 두 번째 출력 값을 만든다.
- 이를 n_1 번 반복하여 크기가 n_1 인 1D텐서를 출력한다.
- 모수의 수 계산: 입력의 크기가 m 인 1D텐서이므로 선형결합을 위해 m 개의 모수가 필요하고 1개의 bias를 추가. 그러므로 노드당 $(m+1)$ 개의 모수
- 노드가 n_1 개 있으므로 MLP 은닉층 1개의 총 모수는 $n_1(m+1)$ 개임.
- Activation function은 모수가 없음
- 출력층의 노드수는 범주형이면 범주의 수, 연속형이면 목적변수 y 의 차원

Activation Function

활성함수	함수식	그래프
ReLU	$\sigma(z) = \max(z, 0)$	
Logistic (sigmoid)	$\sigma(z) = \frac{1}{1 + e^{-z}}$	
tanh	$\sigma(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	
linear	$\sigma(z) = z$	

- **ReLU (Rectified linear unit)**: MLP, CNN, 그리고 RNN에서 가장 빈번하게 사용되는 함수로 입력값이 0보다 작으면 0을 출력하고 입력이 양의 값을 가지면 출력은 입력값을 그대로 출력하는 함수

Activation Function

- Logistic: 목적변수가 범주형일 때 0~1 사이값을 출력하여 주로 최종 출력층에 사용하며 범주에 속할 확률을 출력한다.
- Tanh: RNN과 MLP에서 유용하게 사용되며 $-1 \sim 1$ 사이의 값을 출력한다.
- linear는 목적변수 y 가 연속형인 회귀일 때 출력층에 주로 사용한다.
- Activation function을 사용하는 이유: 활성화함수를 적용하지 않는다면 MLP는 선형결합을 반복적으로 적용하여 결과적으로 하나의 선형결합을 쓴 결과와 동일한 결과를 초래하게 되어, 여러 개의 은닉층을 쓸 이유가 없게됨

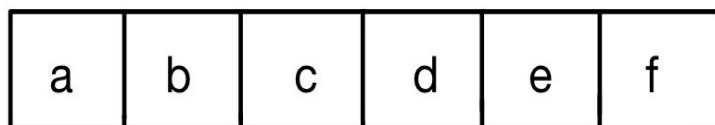
MLP

- 입력데이터의 특성변수가 1D 텐서가 아니더라도 MLP적 모수연결(dense)이 가능.
- 컬러이미지처럼 3D 텐서자료를 1D 텐서로 전환하여 MLP적 연결
- (28,28,3) 3D텐서를 옆붙이기로 크기가 28X28X3인 1D텐서로 전환.
⇒은닉층의 노드가 64개이면 총 모수 수는 $(2352+1) \times 64 = 150,592$ 개!!!
⇒과대적합문제 발생
- 2의 배수인 노드수를 줄이면 정보의 손실과 병목현상(bottleneck)

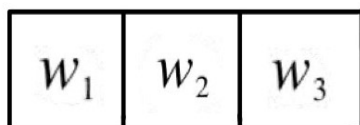
CNN (convolutional neural networks)

- CNN 모형의 설계는 MLP 모형설계의 일반화로 해석할 수 있다.
- CNN은 2D텐서나 3D텐서 특성변수로 제공되는 이미지자료 분석에 특화된 모형임.
- CNN 모형도 MLP 모형과 동일하게 입력층, 은닉층, 출력층을 연결하는 선형 결합과 활성화함수로 구성되어 있다.
- CNN 모형은 선형결합에 필요한 모수의 개수를 획기적으로 줄이는 1D, 2D, 그리고 3D convolution을 사용
- 1D, 2D, 3D convolution은 각각 1D, 2D, 3D 선형결합결과를 출력하는 선형결합 함수.

1D convolution

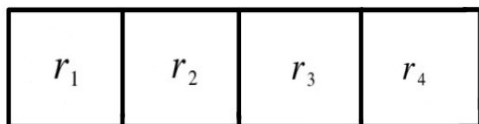


⊗



(1,3) 커널

||



$$r_1 = w_1a + w_2b + w_3c,$$

$$r_2 = w_1b + w_2c + w_3d,$$

$$r_3 = w_1c + w_2d + w_3e,$$

$$r_4 = w_1d + w_2e + w_3f$$

- 크기가 6인 1D텐서를 3개의 모수를 가진 (1,3) kernel로 크기가 4인 1D텐서로 변환
- 그러므로 MLP는 (1,6) kernel을 사용한 1D convolution으로 해석
- 6개의 모수를 3개로 줄임

1D convolution

- 6개의 특성변수가 입력되었으므로 6개의 출력을 만들고 싶으면 입력자료 양 끝에 0을 추가하면 크기가 (1,6)인 1D 텐서를 출력하게 된다. 이와 같이 커널이 움직이는 양방향으로 0을 채워주는 것을 패딩(padding)이라고 한다.
- 커널이 한 칸씩 오른쪽으로 움직였지만 3칸씩 움직인다면 크기가 크기가 2인 1D텐서를 출력함
- 이와 같이 이동하는 칸수를 스트라이드(stride)라고 한다.
- 물론, 특성변수가 열벡터로 공급되면 커널을 열벡터로 정의하고 커널은 위에서 아래로 움직이면서 선형결합을 출력하게 된다.

2D텐서 자료에 대한 1D convolution의 적용

a	b	c	d	e
f	g	h	i	j
k	l	m	n	o

 $\otimes (w_{11}, w_{12}, w_{13})$
 $\otimes (w_{21}, w_{22}, w_{23}) =$

r_1	r_2	r_3
-------	-------	-------

 $\otimes (w_{31}, w_{32}, w_{33})$

$$r_1 = w_{11}a + w_{12}b + w_{13}c + w_{21}f + w_{22}g + w_{23}h + w_{31}k + w_{32}l + w_{33}m$$

$$r_2 = w_{11}b + w_{12}c + w_{13}d + w_{21}g + w_{22}h + w_{23}i + w_{31}l + w_{32}m + w_{33}n$$

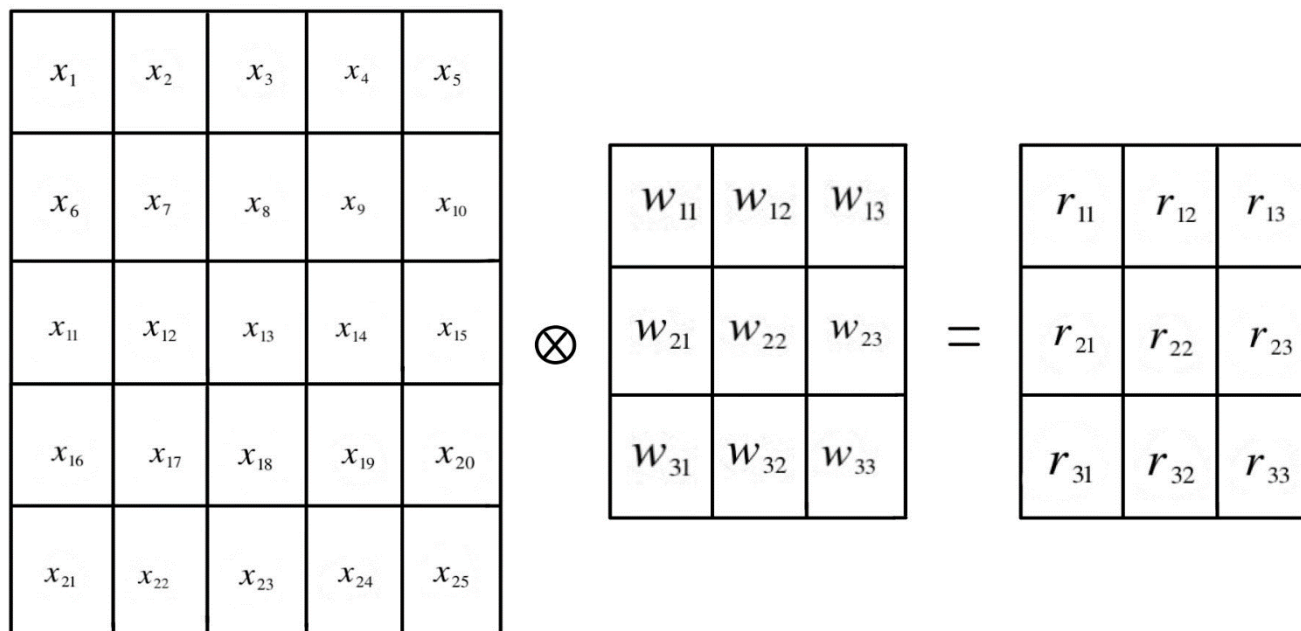
$$r_3 = w_{11}c + w_{12}d + w_{13}e + w_{21}h + w_{22}i + w_{23}j + w_{31}m + w_{32}n + w_{33}o$$

- (1,3) kernel을 적용.
- 모든 열이 동시에 오른쪽으로 1칸씩 움직여야 하므로 3개의 (1,3)kernel이 필요함
- 9개의 모수가 필요함(MLP는 15개 필요함)

2D텐서 자료에 대한 1D convolution의 적용

- CNN의 은닉층에 10개의 노드가 있으면 앞과 같은 세 쌍의 (1,3) kernel이 10개 있다는 말과 동일함. CNN에서는 노드를 filter라고 함.
- 그러므로 10쌍의 서로 다른 (r_1, r_2, r_3) 을 출력하고
- 각 kernel 마다 1개의 bias를 추가하여 활성화함수를 적용하여 최종 출력하고 다음 층의 입력으로 전달됨.
- 총 모수의 수 $(9+1) \times 10 = 100$ 개
- MLP의 경우, 160개의 모수가 필요함.

2D convolution

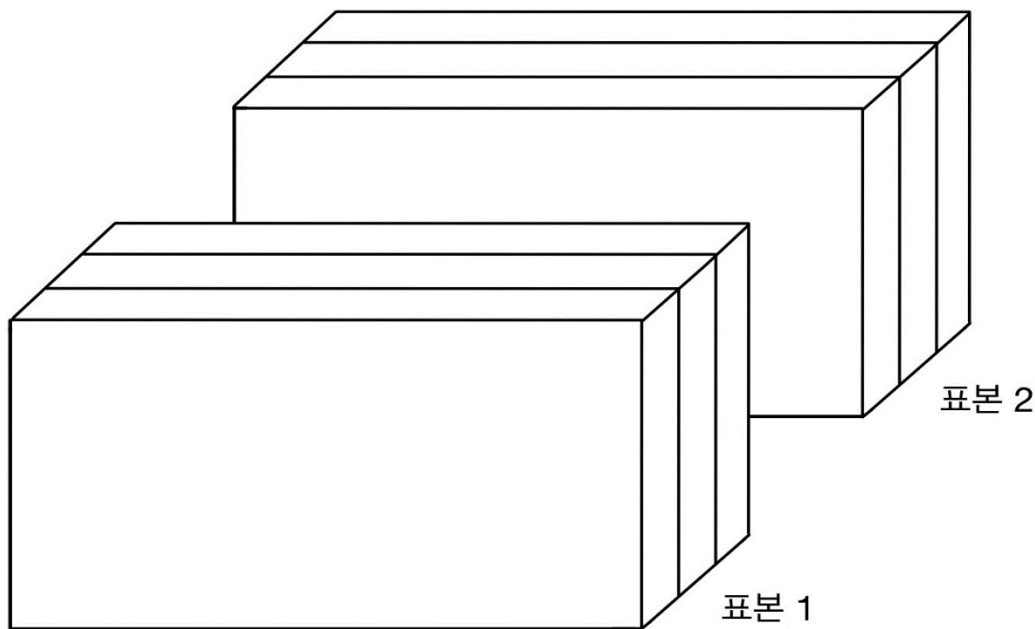


- CNN에서 일반적으로 사용하는 convolution임.
- 2D convolution은 2D 커널을 사용하므로 적용할 수 있는 입력특성변수는 2D텐서 이상이어야 하며 출력은 2D텐서이다.
- 위 그림은 입력은 (5,5) 2D텐서이고 이 자료에 (3,3) kernel을 적용한 예

2D convolution

- (3,3) kernel을 좌에서 우로 한 칸씩, 위에서 아래로 한 칸씩 이동하면서 계산
- (3,3) 2D텐서를 출력함
- 입력과 동일한 (5,5) 2D텐서를 출력하려면 좌,우,상,하 끝 한 칸에 0을 padding하면 됨.
- Stride=2를 주면 2칸씩 이동하므로 (2,2) 2D텐서를 출력함
- stride=m을 주면 입력 2D텐서의 크기를 $1/m$ 으로 줄여줌.
- Filter=10이면 필터당 (9+1)개의 모수가 필요하므로 총 100개의 모수가 필요
- 위의 예제에서의 출력은 (3,3,10) 3D텐서임.

2D convolution



- 위 그림은 컬러이미지 자료를 형상화한것임.
- 컬러이미지 자료는 크기가 (height, width, channel)인 3D자료임
- 위 그림은 (2, height,width, 3)인 4D텐서임. 첫 번째 축은 표본 수, 마지막 축은 channel수 임. 컬러의 channel=3임.

2D convolution

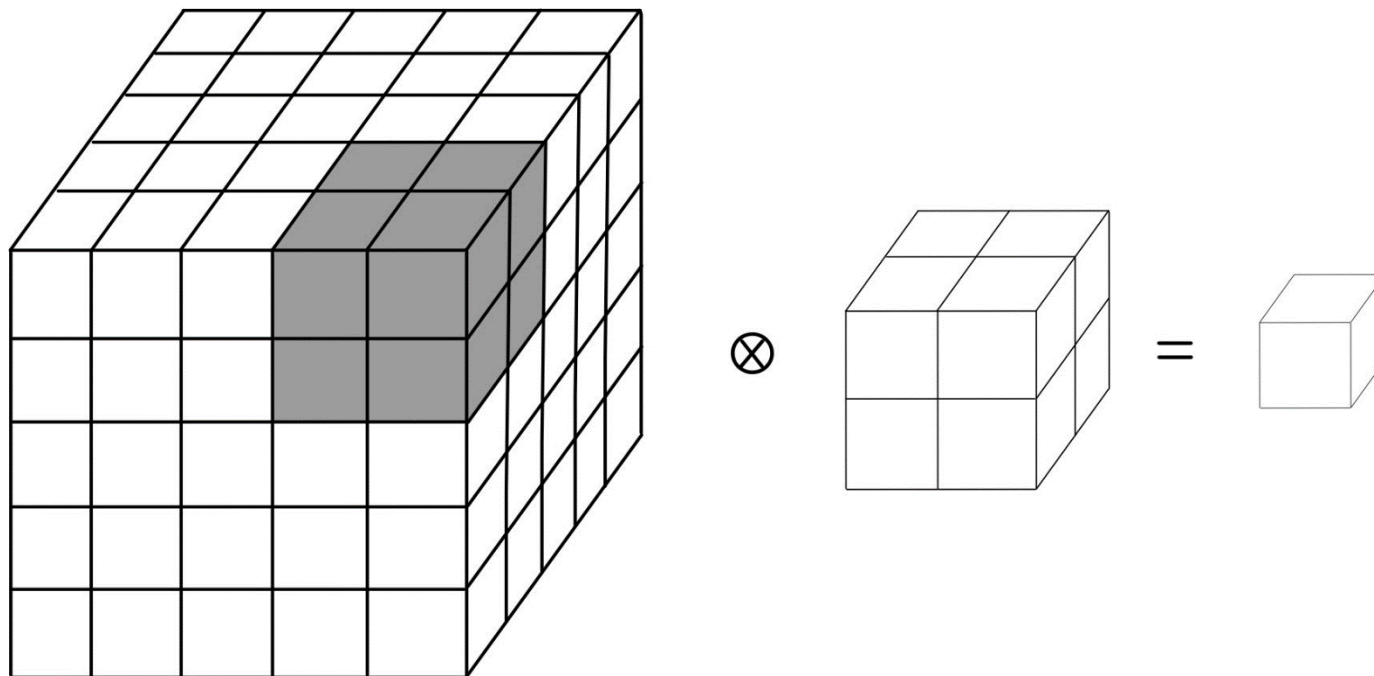
$$\begin{array}{c}
 \begin{array}{|c|c|c|c|c|} \hline x_1^{(1)} & x_2^{(1)} & x_3^{(1)} & x_4^{(1)} & x_5^{(1)} \\ \hline x_6^{(1)} & x_7^{(1)} & x_8^{(1)} & x_9^{(1)} & x_{10}^{(1)} \\ \hline x_{11}^{(1)} & x_{12}^{(1)} & x_{13}^{(1)} & x_{14}^{(1)} & x_{15}^{(1)} \\ \hline x_{16}^{(1)} & x_{17}^{(1)} & x_{18}^{(1)} & x_{19}^{(1)} & x_{20}^{(1)} \\ \hline x_{21}^{(1)} & x_{22}^{(1)} & x_{23}^{(1)} & x_{24}^{(1)} & x_{25}^{(1)} \\ \hline \end{array} & \otimes & \begin{array}{|c|c|c|} \hline w_{11}^{(1)} & w_{12}^{(1)} & w_{13}^{(1)} \\ \hline w_{21}^{(1)} & w_{22}^{(1)} & w_{23}^{(1)} \\ \hline w_{31}^{(1)} & w_{32}^{(1)} & w_{33}^{(1)} \\ \hline \end{array} \\
 \\
 \begin{array}{|c|c|c|c|c|} \hline x_1^{(2)} & x_2^{(2)} & x_3^{(2)} & x_4^{(2)} & x_5^{(2)} \\ \hline x_6^{(2)} & x_7^{(2)} & x_8^{(2)} & x_9^{(2)} & x_{10}^{(2)} \\ \hline x_{11}^{(2)} & x_{12}^{(2)} & x_{13}^{(2)} & x_{14}^{(2)} & x_{15}^{(2)} \\ \hline x_{16}^{(2)} & x_{17}^{(2)} & x_{18}^{(2)} & x_{19}^{(2)} & x_{20}^{(2)} \\ \hline x_{21}^{(2)} & x_{22}^{(2)} & x_{23}^{(2)} & x_{24}^{(2)} & x_{25}^{(2)} \\ \hline \end{array} & \otimes & \begin{array}{|c|c|c|} \hline w_{11}^{(2)} & w_{12}^{(2)} & w_{13}^{(2)} \\ \hline w_{21}^{(2)} & w_{22}^{(2)} & w_{23}^{(2)} \\ \hline w_{31}^{(2)} & w_{32}^{(2)} & w_{33}^{(2)} \\ \hline \end{array} & = & \begin{array}{|c|c|c|} \hline r_{11} & r_{12} & r_{13} \\ \hline r_{21} & r_{22} & r_{23} \\ \hline r_{31} & r_{32} & r_{33} \\ \hline \end{array} \\
 \\
 \begin{array}{|c|c|c|c|c|} \hline x_1^{(3)} & x_2^{(3)} & x_3^{(3)} & x_4^{(3)} & x_5^{(3)} \\ \hline x_6^{(3)} & x_7^{(3)} & x_8^{(3)} & x_9^{(3)} & x_{10}^{(3)} \\ \hline x_{11}^{(3)} & x_{12}^{(3)} & x_{13}^{(3)} & x_{14}^{(3)} & x_{15}^{(3)} \\ \hline x_{16}^{(3)} & x_{17}^{(3)} & x_{18}^{(3)} & x_{19}^{(3)} & x_{20}^{(3)} \\ \hline x_{21}^{(3)} & x_{22}^{(3)} & x_{23}^{(3)} & x_{24}^{(3)} & x_{25}^{(3)} \\ \hline \end{array} & \otimes & \begin{array}{|c|c|c|} \hline w_{11}^{(3)} & w_{12}^{(3)} & w_{13}^{(3)} \\ \hline w_{21}^{(3)} & w_{22}^{(3)} & w_{23}^{(3)} \\ \hline w_{31}^{(3)} & w_{32}^{(3)} & w_{33}^{(3)} \\ \hline \end{array}
 \end{array}$$

- 위 그림은 (5,5,3)인 컬러이미지자료 1개를 펼쳐 놓은 상태에서 (3,3) kernel을 적용한 예제임

3D텐서에 2D convolution 적용

- Channel이 3개 있으므로 세 쌍의 (3,3) kernel이 필요.
- 선형결합을 위해 모수는 $3 \times 3 \times 3 + 1$ 개가 필요.
- 노드가 10개 있으면 총 모수 수는 280개가 소요됨

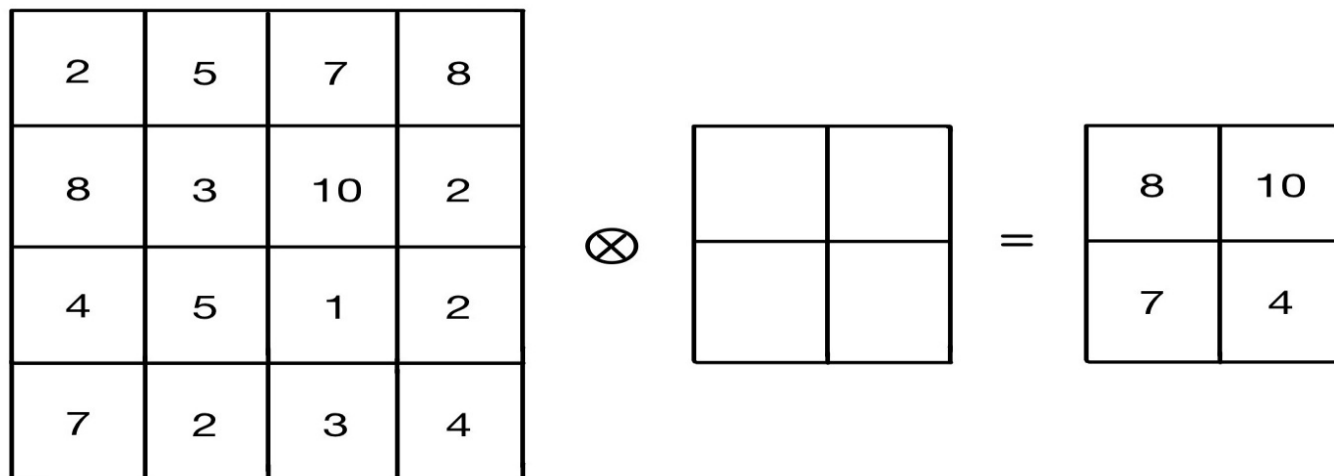
3D convolution



- (5,5,5) 3D텐서자료에 (2,2,2) kernel을 적용
- 좌에서 우로, 상에서 하로, 안에서 밖으로 한 칸씩 이동하면서 kernel
- (4,4,4) 3D텐서 출력

Pooling

Stride=(2,2)인 maxpooling



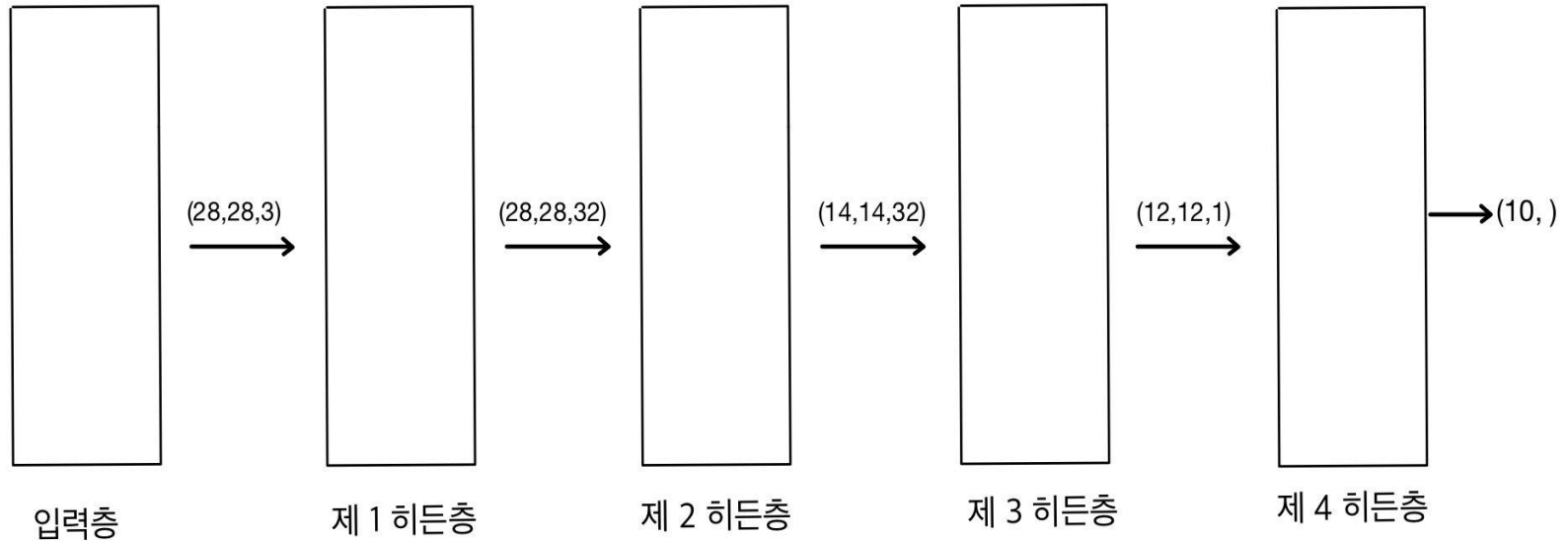
$$8 = \max \left(\begin{array}{|c|c|} \hline 2 & 5 \\ \hline 8 & 3 \\ \hline \end{array} \right) \quad 10 = \max \left(\begin{array}{|c|c|} \hline 7 & 8 \\ \hline 10 & 2 \\ \hline \end{array} \right)$$

$$7 = \max \left(\begin{array}{|c|c|} \hline 4 & 5 \\ \hline 7 & 2 \\ \hline \end{array} \right) \quad 4 = \max \left(\begin{array}{|c|c|} \hline 1 & 2 \\ \hline 3 & 4 \\ \hline \end{array} \right)$$

Pooling

- 이미지 자료의 특성을 좀 더 드러나게 하기 위한 커널
- 2D 커널인 경우, stride=2를 가진 (2,2) 또는 stride=3인 (3,3) pooling 커널을 주로 사용한다
- stride의 크기와 pooling 커널의 크기가 일치하는 이유: pooling 커널 적용 시 겹치는 부분이 없도록 하기 위함이고 원래 텐서의 크기를 1/2 또는 1/3로 되도록 하기 위함이다
- 입력 pooling 커널은 모수의 수를 줄일 뿐만 아니라 CNN의 성능을 향상시키는 역할을 한다.
- pooling 커널은 선형결합이 아니므로 커널 자체의 모수가 없다.

CNN



- 제 1히든층: $(3,3)$ 커널 +padding, 모수 : $(3 \times 3 \times 3 + 1) \times 32 = 896$
- 제 2히든층: $(2,2)$ max-pooling
- 제 3히든층: $(3,3)$ 커널, 모수: $3 \times 3 \times 32 + 1 = 289$
- 제 4히든층: $(3,1)$ 커널: 모수: $3 \times 12 + 1 = 37$

RNN (recurrent neural networks)

- 문장이나 말하기(speech)는 단어순서에 의해 이해된다.
- 이해된다는 의미는 어떤 문장을 분류하거나(예를 들어, 신문기사의 주제가 경제, 문화, 사회 등의 분류, 특정 주제에 대한 트윗이 찬성인지 반대인지를 구별하기 등), 한 언어에서 다른 언어로 번역한다는 것을 말한다.
- 최근의 일주일 동안 또는 한 달 동안의 주식가격이 시간단위 또는 일일 단위의 순서로 제공되면 내일의 주식가격을 예측하는데 매우 유용하게 사용될 것이다.
- RNN은 이처럼 순서정보가 분류나 회귀 등의 분석에 중요한 요인이 될 때 사용하는 신경망모형이다.

RNN의 예

- 총 100,000 개의 문서에서 가장 빈번하게 사용된 5000개의 단어로 각 문서의 첫 1,000개의 단어를 one-hot coding을 하면, 각 문서는 (1000,5000) 2D텐서이고 전체데이터는 (100000,1000,5000) 3D텐서가 된다.
- 매 1분마다 주식종목 A에 대한 주식가격, 최고가격, 최저가격을 측정하면 하루에 390개의 (주식가격, 최고가격, 최저가격)이 관측된다. 이를 100일 동안 관측했다고 하자.
- x_t 를 1분단위 시점 t에서의 (주식가격,최고가격,최저가격)이고 y_t 를 목적변수인 시점 t에서의 주식가격이라고 정의하자.
- $(x_{t+1}, x_{t+2}, \dots, x_{t+390})$ 으로 y_{t+391} 을 예측하고자 함.

RNN의 예

전체 데이터구조

$[(x_1, x_2, \dots, x_{390}), y_{391}], [(x_{391}, x_{392}, \dots, x_{780}), y_{781}], \dots, [(x_{38611}, x_{38662}, \dots, x_{39000}), y_{39001}]$

- 각 표본은 (390,3) 2D텐서 입력자료, 0D텐서 출력자료로 구성됨
- 전체 입력데이터는 (100,390,3) 3D텐서, 출력은 (100,1) 2D텐서
- $x_{t-390}, x_{t-389}, \dots, x_{t-1}$ 로 예측한 주식가격을 \hat{y}_t 라고 할 때
- $y_t - \hat{y}_t$ 가 i.i.d 해야 함. 이를 위한 조건은 시계열자료가 stationary해야 함.
- 일반적으로 시계열 자료는 이를 만족하지 못함.
- 특히, finance data는 비정상시계열임.
- 정상성을 만족하면

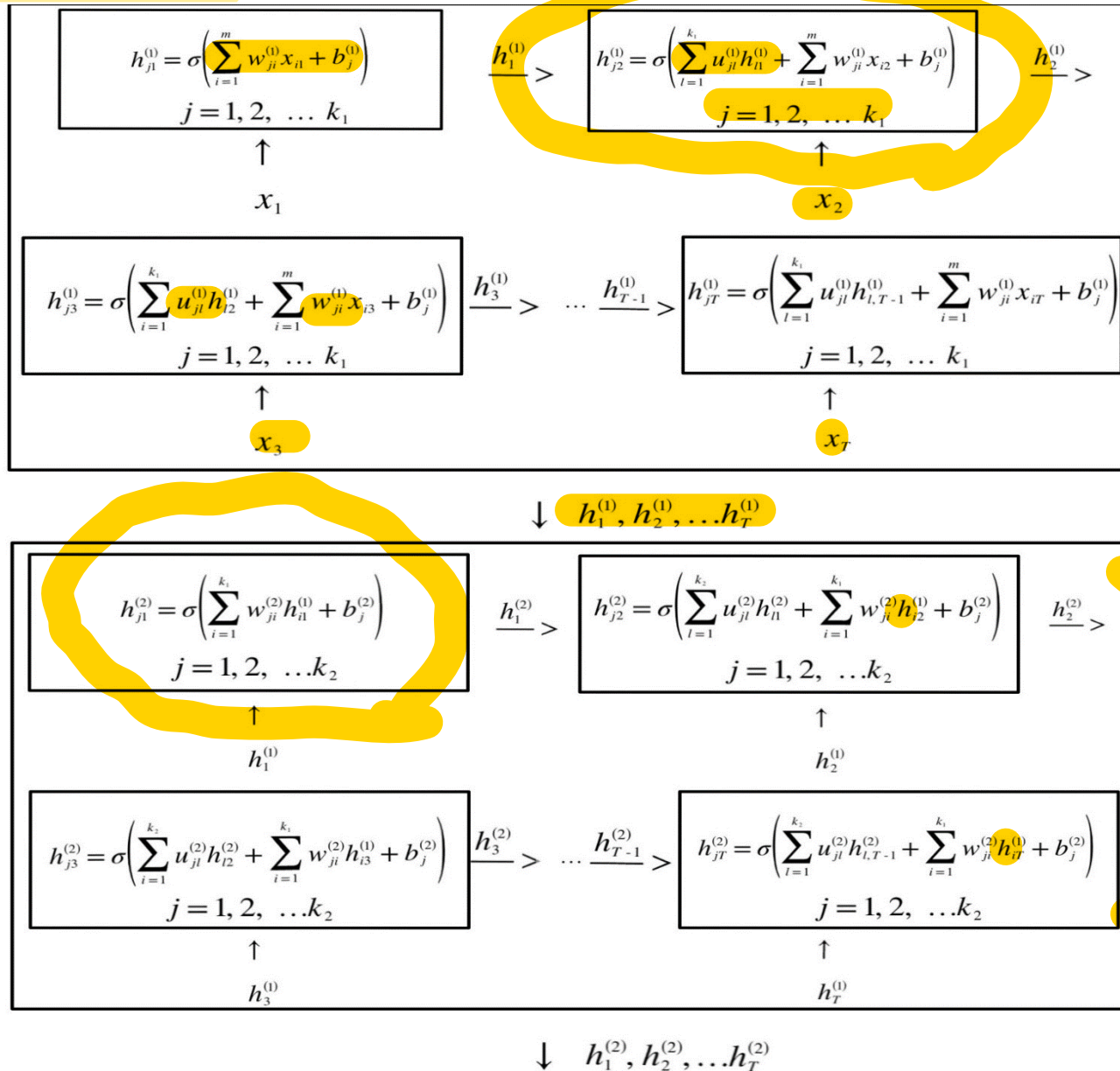
$[(x_1, x_2, \dots, x_{390}), y_{391}], [(x_2, x_3, \dots, x_{391}), y_{392}], \dots, [(x_{38661}, x_{38662}, \dots, x_{39000}), y_{39001}]$

으로 데이터를 38661개로 증가시켜 모형의 성능을 대폭 개선할 수 있음

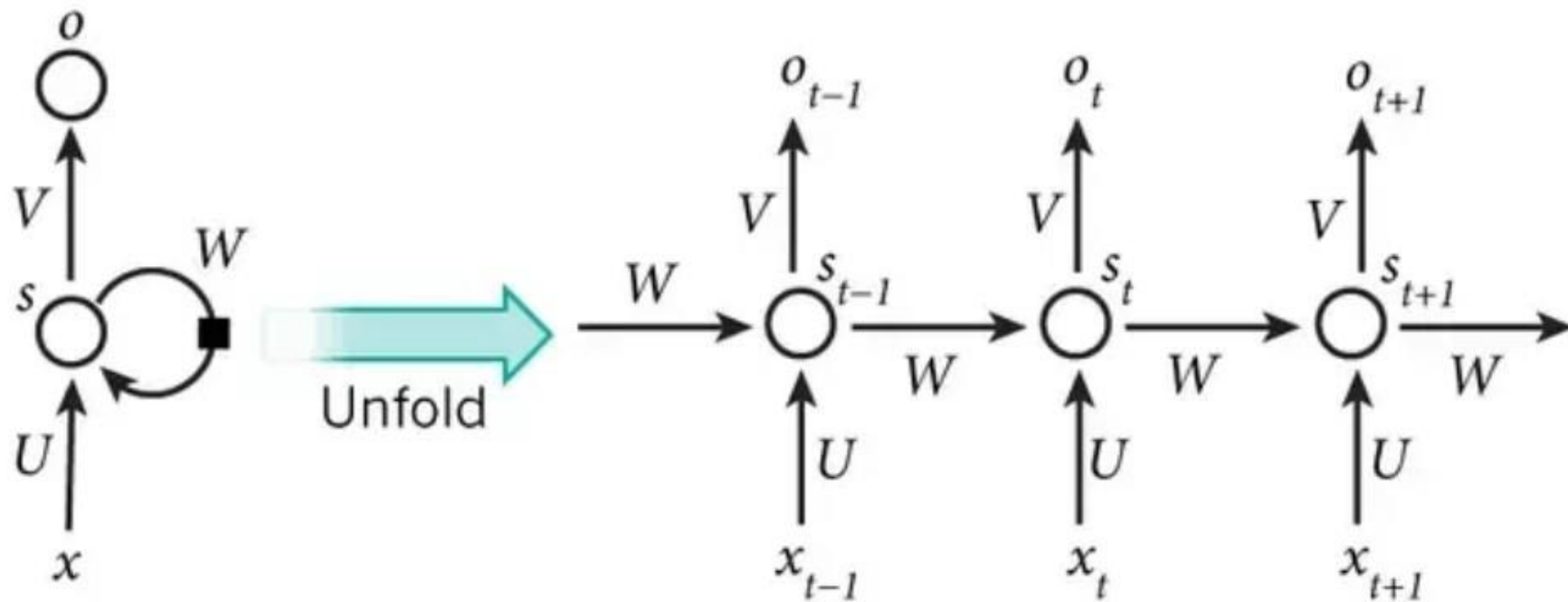
RNN

- DNN의 입력자료는 (표본수, 시간스텝, 특성변수의 수)로 3D텐서
- 주식가격의 예에서 첫 번째 자료구조에서는 (100,390,3)이고 두 번째 자료구조에서는 (38611,390,3)이다.
- CNN의 흑백이미지 데이터와 동일한 3D텐서이지만 DNN 입력데이터에 2D 또는 3D convolution을 적용하면 390분 동안 시간순서로 제공된 (주식가격, 최고가격, 최소가격)에 대한 시간정보가 손실되게 된다.
- x_t 는 x_{t+1} 에 영향을 주고 x_{t+1} 은 x_{t+2} 에 영향을 주므로 RNN은 이러한 의존관계를 반영하여야 한다.

Simple RNN



Simple RNN



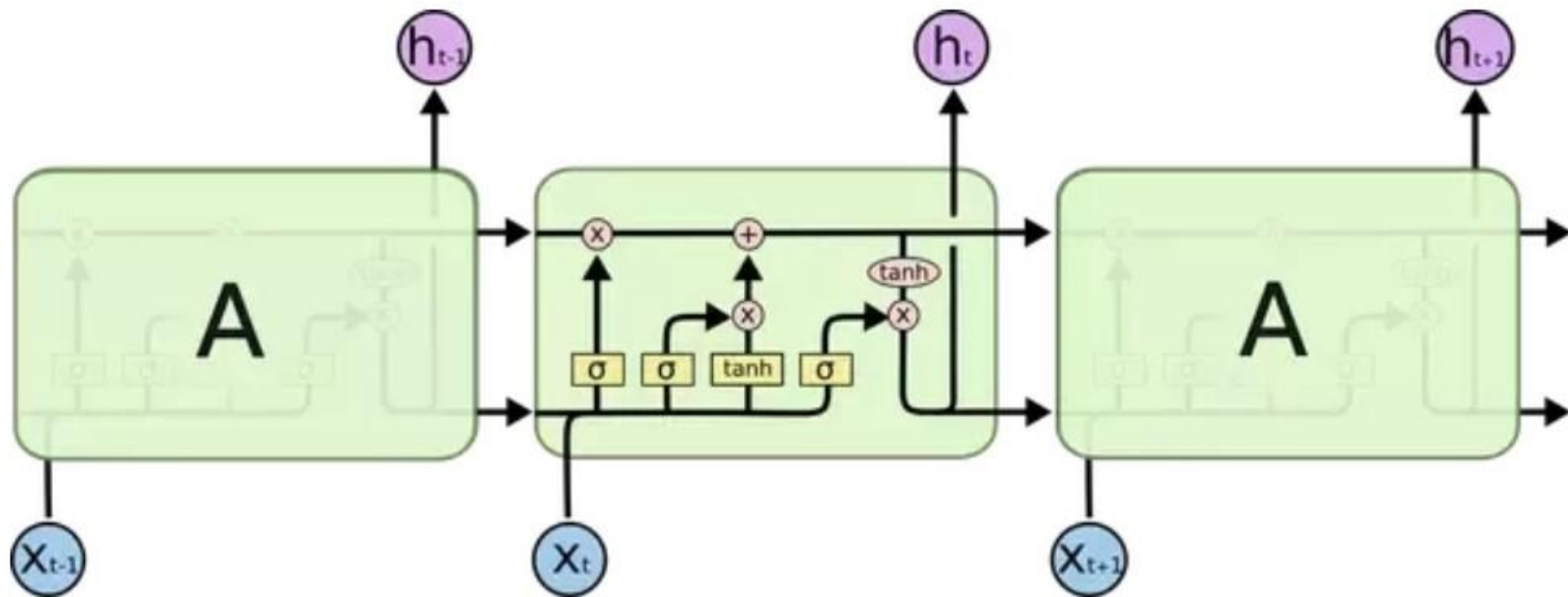
Simple RNN

- 외부층(outer layers)과 내부층(inner layers)으로 구성되어 있다.
- 오직 2개의 외부층을 보여주고 있지만 3개 이상의 외부층을 가질 수 있다.
- 간단한 RNN은 이름에 비해 매우 복잡해 보이지만 만약 $T=1$ 이면 각 외부층은 오직 하나의 내부층만 가지게 되어 은닉층이 2개인 MLP 모형과 동일함을 알 수 있다.
- 입력변수 x_t 와 h_t 의 선형결합을 위한 모수 $w_{ji}^{(1)}$ 와 $w_{ji}^{(2)}$, h_{t-1} 의 선형결합을 위한 모수 $u_{jl}^{(1)}$ 와 $u_{jl}^{(2)}$, 그리고 bias $b_j^{(1)}$ 와 $b_j^{(2)}$ 는 시점 t 에 의존하지 않음
- 이는 입력변수 x_t 가 정상성(stationary)하다는 가정을 충족해야함
- (입력 특성변수의 수 + 은닉층의 노드수 + bias) x 은닉층의 노드수
- $h_t^{(1)}$, $t=1,2,...,T$ 는 다음 은닉층의 입력변수로 전달

Simple RNN

- Simple RNN의 약점은 h_T 는 h_{T-1} 의 함수이고 h_{T-1} 은 h_{T-2} 의 함수이며 계속 반복하여 h_2 는 h_1 의 함수가 된다는 것이다.
- 이들은 동일한 u_{jl} 를 사용하기 때문에, backpropagation에 의한 u_{jl} 의 최적값은 h_T 부터 h_1 까지의 미분값을 통해 최신화를 한다.
- 그러므로 h_T 에서 멀리 떨어진 h_2 나 h_1 의 미분값은 u_{jl} 에 의존하여 u_{jl} 의 절대값이 1보다 크면 무한대로 접근하고 1보다 작으면 0으로 접근한다.
- 이를 RNN에서 미분값의 폭발(exploding) 또는 사라짐(vanishing) 현상이라고 한다.
- 이러한 현상이 일어나는 근본적인 이유는 h_t 가 h_{t-1} 에 직접 연결되기 때문
- 이 문제를 해결한 RNN 아키텍처가 LSTM (long short-term memory)와 GRU (gated recurrent unit)이다.

LSTM



LSTM(long short term memory)

- LSTM과 GRU 모두 h_t 와 h_{t-1} 을 간접적으로 연결
- GRU(gated recurrent units)는 LSTM의 특수한 경우
- LSTM과 GRU는 모두 <그림 2-15>의 간단한 RNN 구조를 가지고 있지만 내부층을 구성하는 h_t 가 좀더 복잡하다.
- LSTM을 구성하는 네 개의 gates:

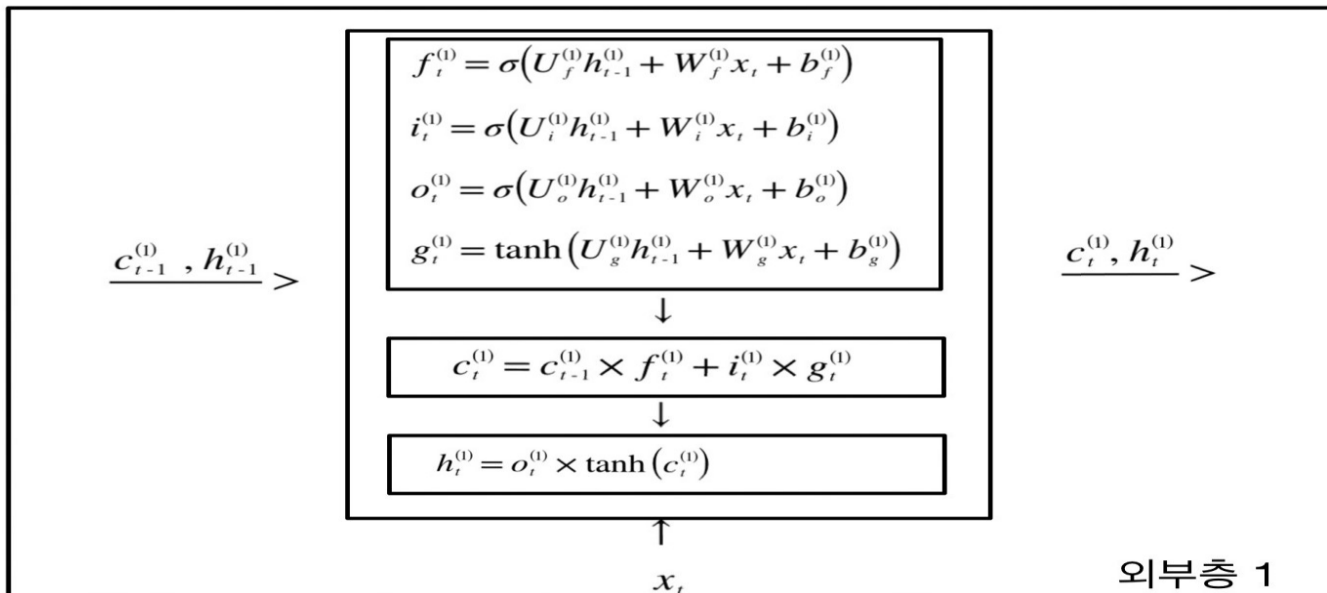
$$f_t = \sigma(U_f h_{t-1} + W_f x_t + b_f), i_t = \sigma(U_i h_{t-1} + W_i x_t + b_i)$$

$$o_t = \sigma(U_o h_{t-1} + W_o x_t + b_o), g_t = \tanh(U_g h_{t-1} + W_g x_t + b_g)$$

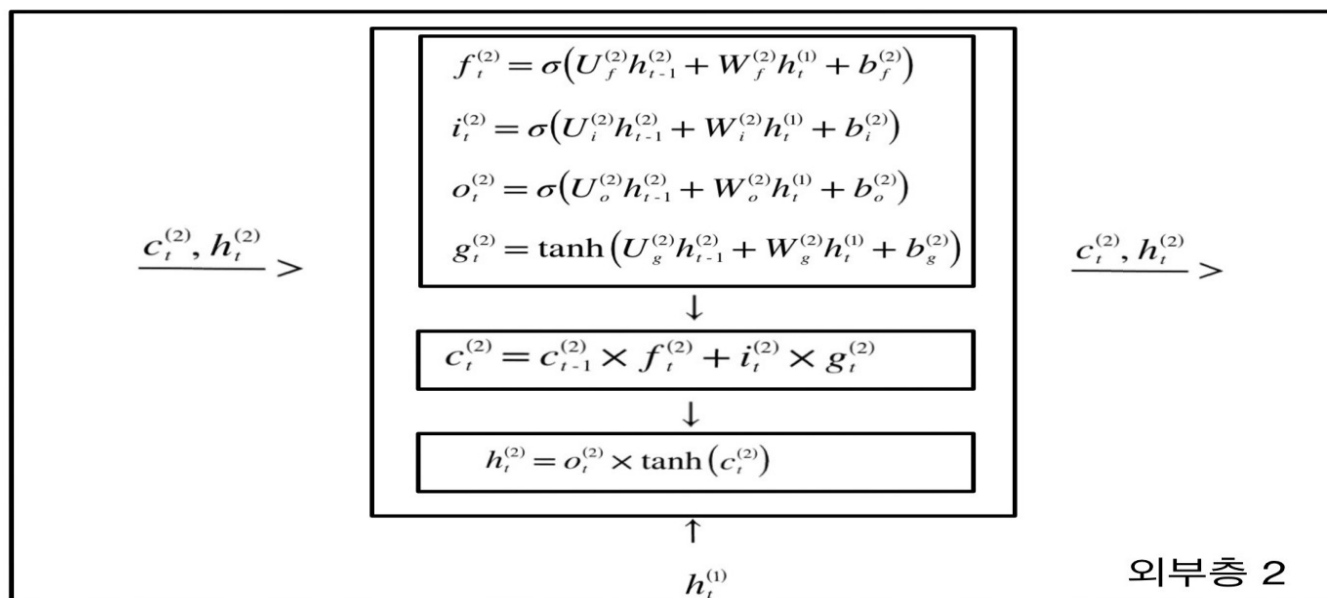
- 상태벡터 c_t 와 출력벡터 h_t 는 다음과 같이 정의된다.

$$c_t = c_{t-1} \times f_t + i_t \times g_t, h_t = o_t \times \tanh(c_t)$$

LSTM



↓ $(h_i^{(1)} h_2^{(1)}, \dots, h_T^{(1)}), h_T^{(1)}, c_T^{(1)}$



↓ $(h_i^{(2)} h_2^{(2)}, \dots, h_T^{(2)}), h_T^{(2)}, c_T^{(2)}$

LSTM(long short term memory)

- LSTM모형의 모수 수는 simple RNN의 4배
(입력변수 수+은닉층의 노드수+1)x은닉층의 노드수x4
- 첫 번째 외부층: $x_t(m \times 1), h_t^{(1)}(k_1 \times 1) \rightarrow (k_1 + m + 1) \times k_1$
- 두 번째 외부층: $h_t^{(2)}(k_2 \times 1) \rightarrow (k_2 + k_1 + 1) \times k_2$

GRU(gated recurrent units)

- g_t 와 f_t 만을 이용하여

$$h_t = (1 - f_t) \times g_t + f_t \times h_{t-1}$$
- 그러므로 모수는 LSTM의 $\frac{1}{2}$

Summary

- 세 개의 신경망에 있는 모수는 표본에 의존하지 않고, 추가적으로 RNN의 모수는 시간스텝에 의존하지 않는다
- 이 의미는 표본은 서로 간에 독립이어야 한다는 것을 말하고, RNN의 시간스텝은 최소한 정상성조건을 만족하여야 한다는 것을 말한다.
- 세 개의 신경망에서 나타나는 노드수는 은닉층에서 만들어진 특성변수의 수로 이해하여야 한다.
- 입력으로 사용하는 특성변수는 사람이 부여하고 이 특성변수가 은닉층에 전달되어 은닉층의 특성변수를 만들어 내는데 노드 수가 은닉층의 특성변수 수가 된다.

Summary

- MLP모형은 1D텐서를 입력으로 받고 1D텐서를 출력하므로 **출력의 차원이 은닉층의 노드가 된다**. 즉, 은닉층에 입력되는 특성변수를 은닉층의 노드수를 조정하여 새로운 특성변수로 전환하고 특성변수의 수도 줄이거나 늘리게 된다.
- CNN모형에서는 특성변수가 3D텐서로 입력되고 convolution과 channel수로 특성변수 수를 조절한다. 가장 일반적인 2D convolution을 사용하면 2D변수가 만들어지고 channel수를 조절하여 새로운 3D텐서 특성변수를 출력하게 된다.
- 그러므로 3D텐서 내 하나의 셀이 하나의 특성변수가 되므로 특성변수의 크기가 너무 커지는 것을 방지하기 위해 **channel를 늘리면 convolution를 조절하여 픽셀의 크기를 줄여준다**.

Summary

- RNN은 2D텐서를 입력 특성변수로 받고 2D텐서 특성변수를 출력한다
- 각 시간순서는 1D특성변수로 구성되어 있으므로 2D텐서 중 시간스텝인 행은 변함이 없고 은닉층의 노드수에 따라, 시간스텝의 특성변수 수를 조절하게 된다.

Q & A