

딥러닝팀

1팀

김예찬

윤지영

채소연

한지원

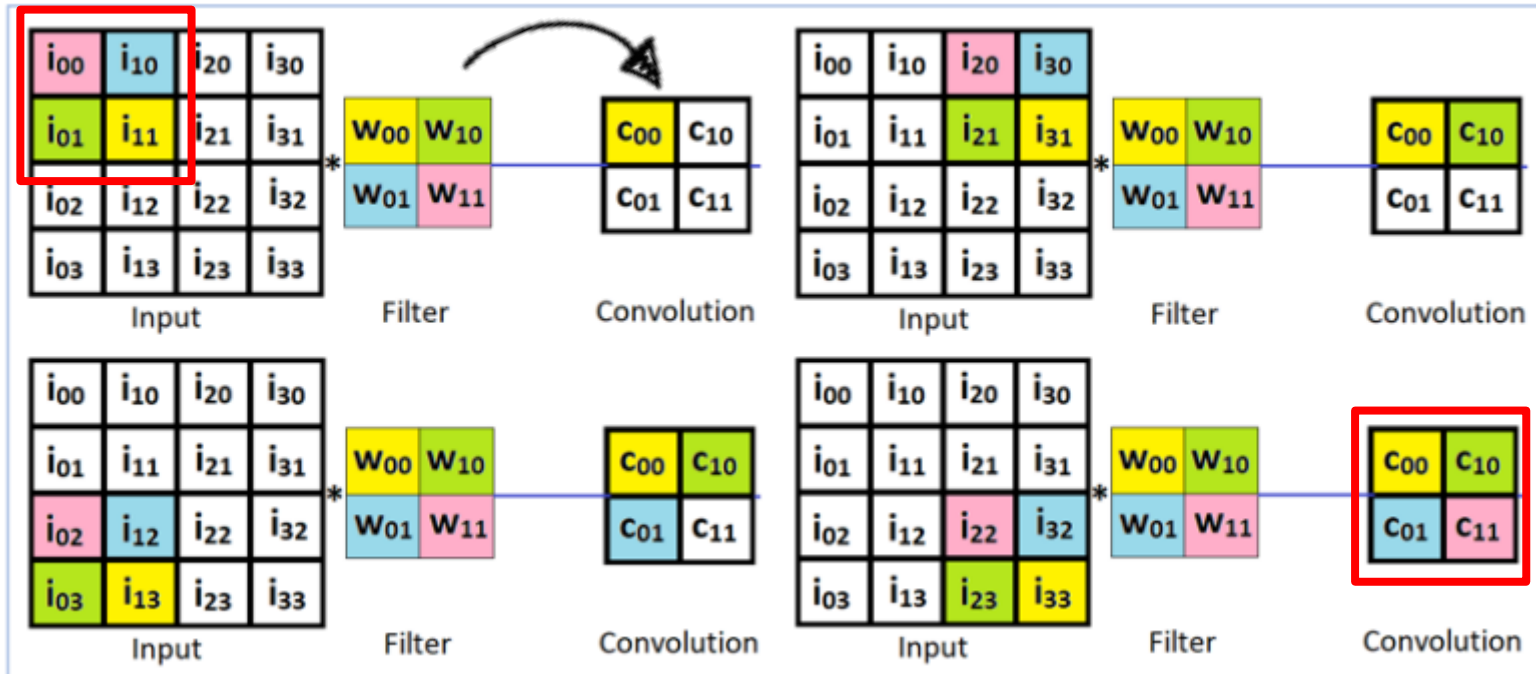
홍지우

1

CNN

1 CNN(Convolutional Neural Network)

- Convolution Layer



Feature Map : 반환된 결과

1 CNN(Convolutional Neural Network)

- Feature Map

Feature Map

하이퍼 파라미터	Pytorch 표현	역할
Input Channel	in_channels	입력 데이터의 채널 개수 지정
Output Channel	out_channels	출력의 채널 개수 지정
필터 사이즈	kernel_size	필터의 크기 지정
Stride	stride	필터의 이동 간격 지정
Padding	padding	입력 데이터 주변에 붙일 padding 의 수 지정

1 CNN(Convolutional Neural Network)

- Pooling Layer

Pooling Layer란?

: 입력 이미지의 중요한 특징들을 더욱 강조하는 동시에 데이터의 크기를 줄이는 것에 더 중점을 둔 Layer

Convolution
Layer

가중치 존재

Pooling
Layer

가중치 없음

2

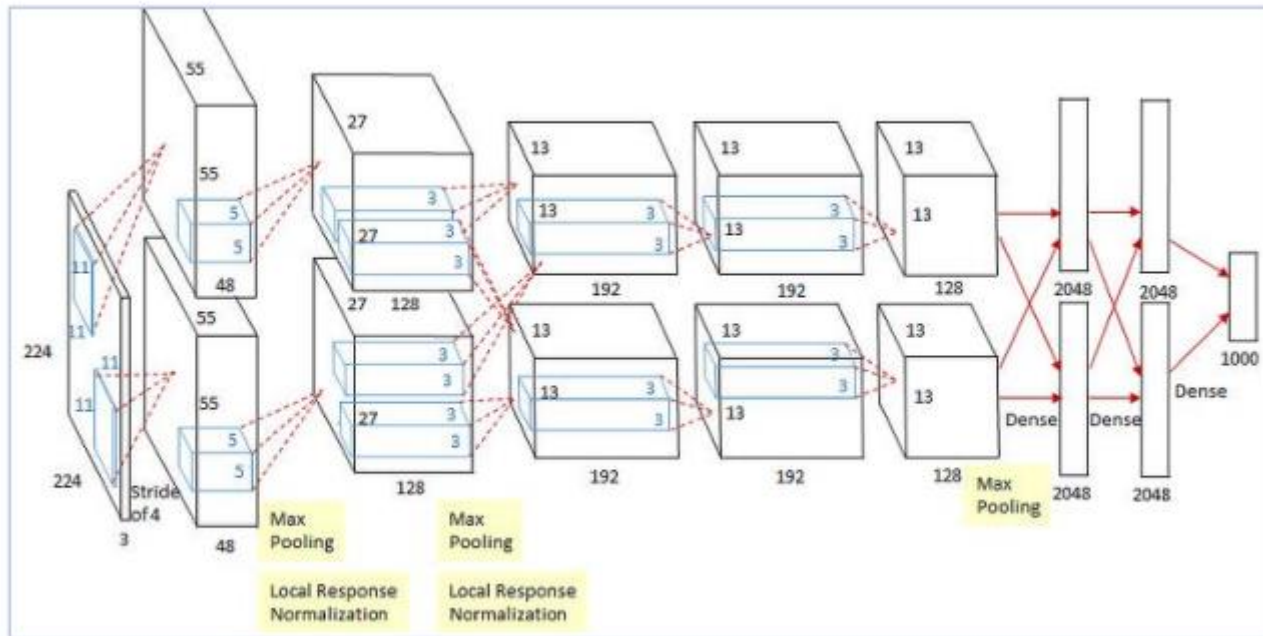
CNN의 발전 과정

2 CNN의 발전 과정

● AlexNet

AlexNet의 특징

- ILSVRC-2012에서 압도적 1등 차지 - 오류율이 큰 폭으로 감소
- 224 * 224의 컬러 이미지 처리를 위해 병렬 처리 ➡ 2개의 GPU를 병렬적으로 이용



2 CNN의 발전 과정

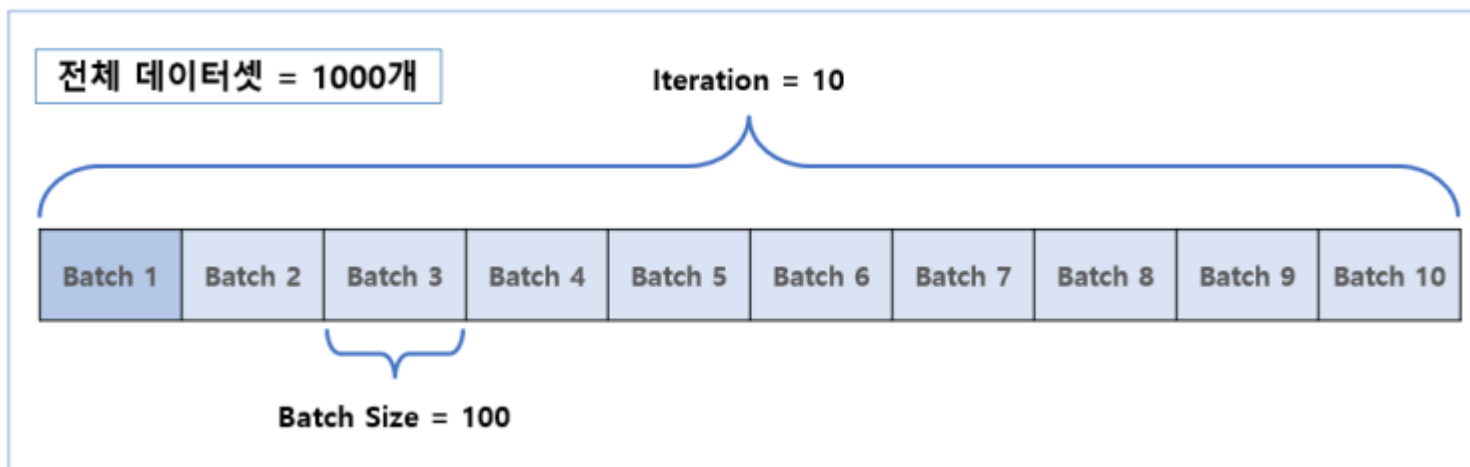
● 배치 정규화

배치 정규화

- 배치의 크기에 따라 Iteration의 크기가 결정
- 모집단의 분포와 표본의 분포가 같지 않은 것처럼, 배치마다 데이터의 분포가 달라질 수 있음



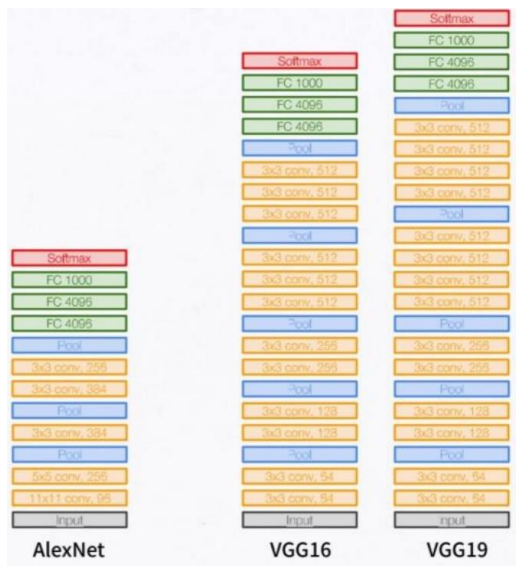
Batch 별 데이터를 평균과 분산으로 정규화 해주는 것



2 CNN의 발전 과정

- VGGNet

VGGNet의 구조와 특징

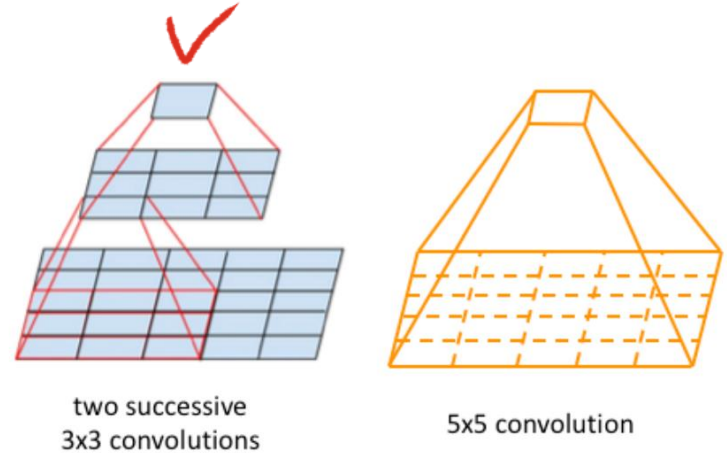


더 많은 layer 사용

How?

파라미터 수:
 $2 \times 3^2 \times \text{채널 수}$

< 5² x 채널 수



- 작은 필터 여러 번 사용
- 파라미터 수 감소 ↓ 비선형성 ↑

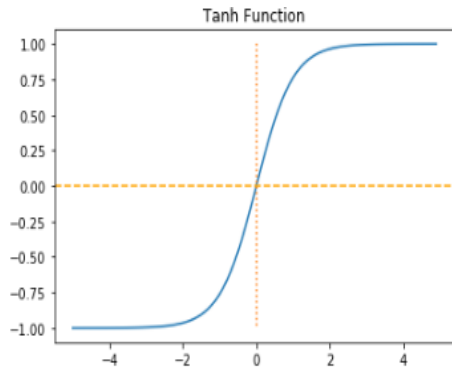
3

RNN

3 RNN (Recurrent Neural Network)

● Vanilla RNN

Hidden State



$$h_t = \tanh(W_x \times x_t + W_h \times h_{t-1})$$

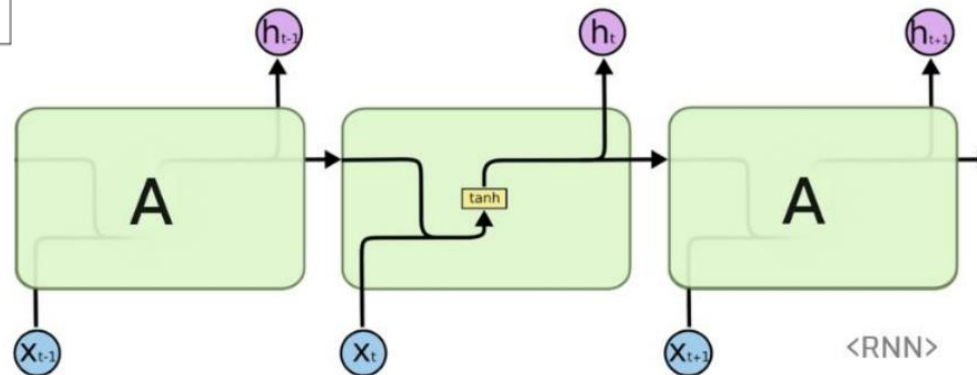
$$o_t = f(W_o h_t)$$

Vanilla RNN의 가중치

$$W_x: x_t \rightarrow h_t$$

$$W_h: h_t \rightarrow h_{t+1}$$

$$W_o: h_t \rightarrow o_t$$

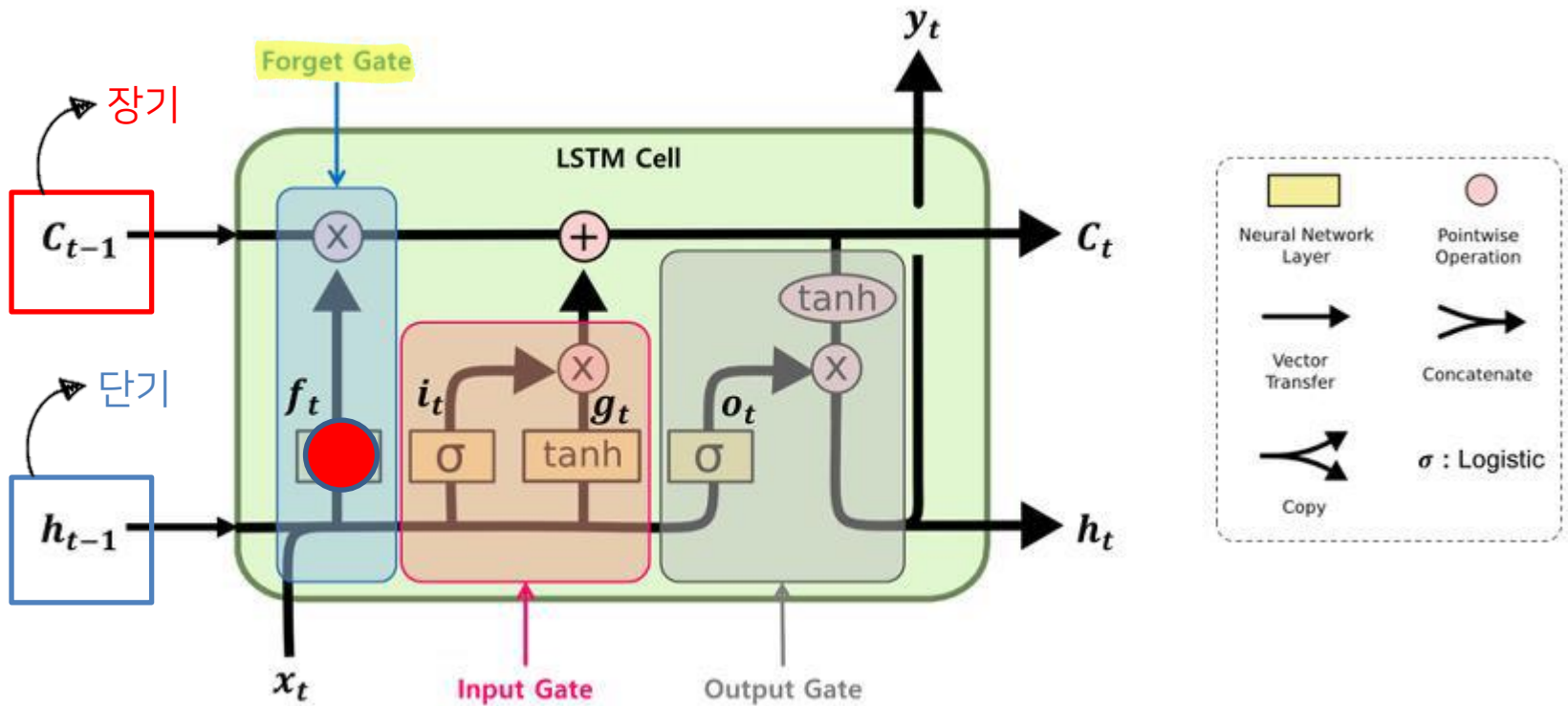


<RNN>

3 RNN (Recurrent Neural Network)

- Hidden State 와 Cell State

Forget Gate

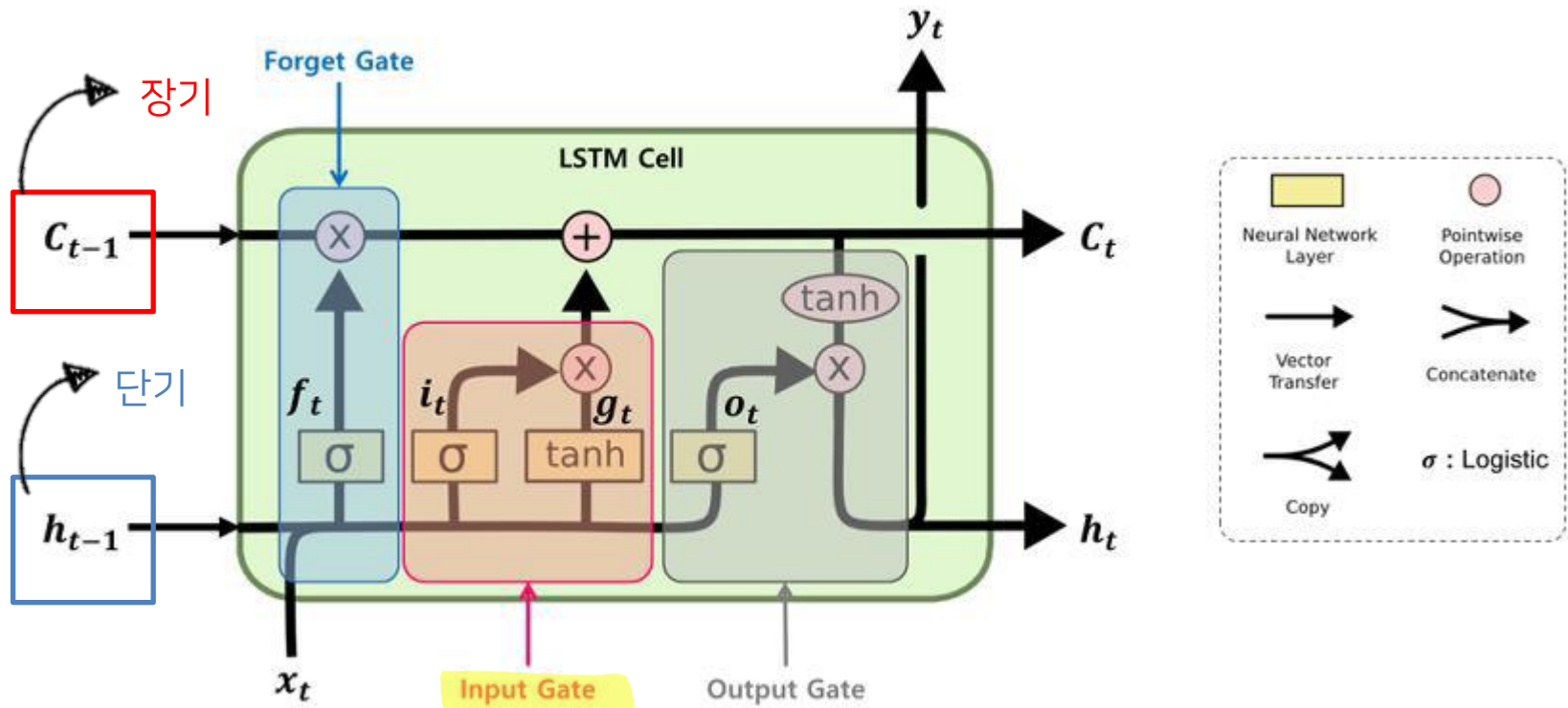


이전 시점의 Hidden State 와 현재 시점의 입력을 가중치와 곱해 시그모이드 함수에 전달

3 RNN (Recurrent Neural Network)

- Hidden State 와 Cell State

Input Gate

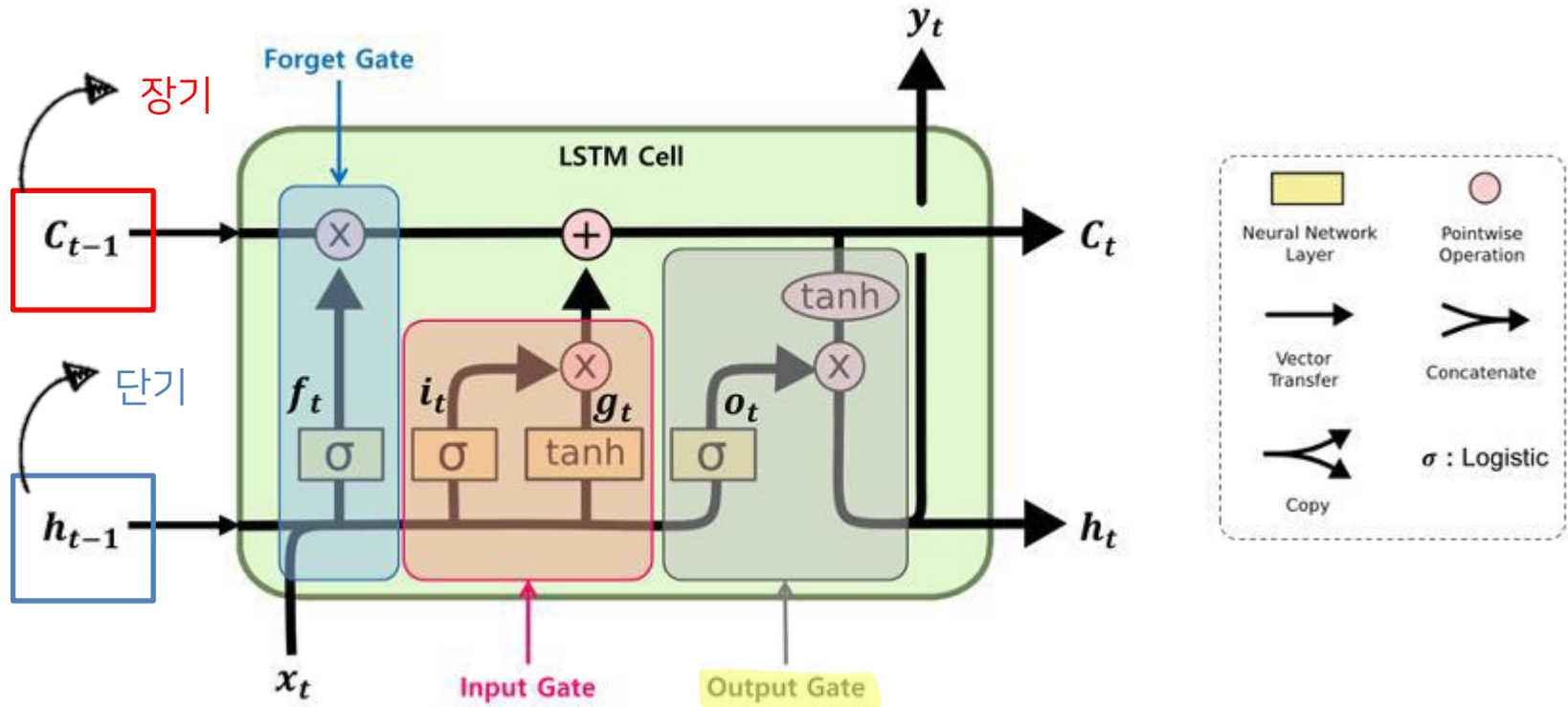


이전 시점의 Hidden State 와 현재 시점의 입력을 가중치와 곱해 시그모이드 또는 tanh 함수에 전달

3 RNN (Recurrent Neural Network)

- Hidden State 와 Cell State

Output Gate



현재까지의 정보들 중 어떤 정보를 얼마나 활용할 것인지 결정

4

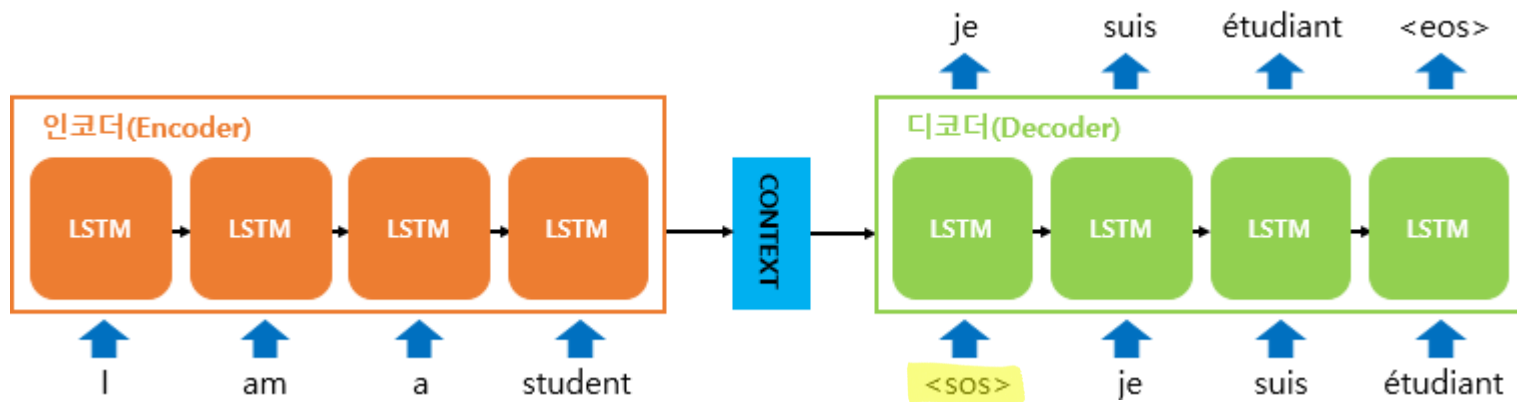
RNN 모델의 응용

4 RNN 모델의 응용

● Seq2Seq

Encoder-Decoder란?

- Encoder : Sequential Data를 입력받아 압축된 하나의 벡터로 만드는 역할
- Decoder : Encoder의 마지막 Hidden State와 입력을 바탕으로 Encoder와 달리 매 시점마다 출력을 내보냄

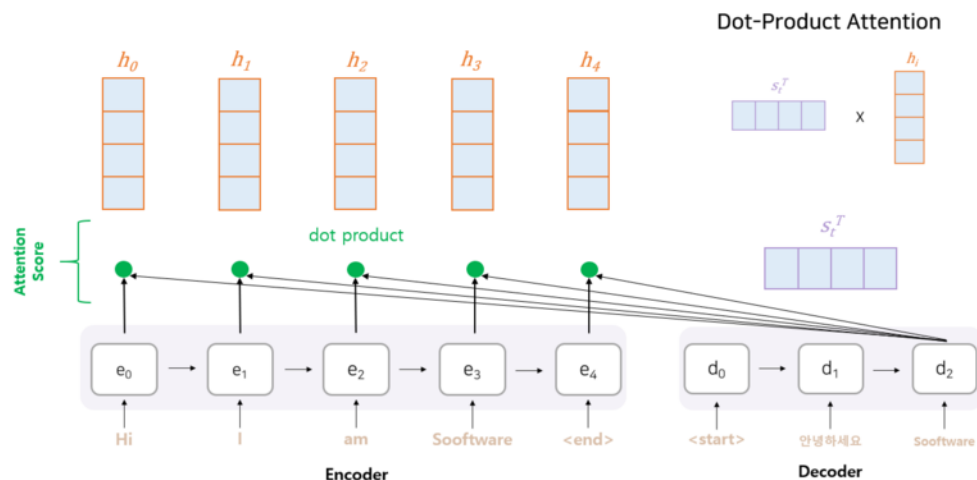


4 RNN 모델의 응용

● Attention

Attention의 계산

- Decoder의 각 시점 Hidden State와 Encoder의 모든 시점 Hidden State의 유사도를 구함



$$score(s_t, h_i) = s_t^T h_i$$

$$e^t = [s_t^T h_1, s_t^T h_2, \dots, s_t^T h_N]$$