1 딥러닝 소개

장점

예측 및 분류가 목적일 때
머신러닝보다 좋은 성능

비정형 데이터 처리에 좋음

자동화로 인간의 개입 최소화

단점

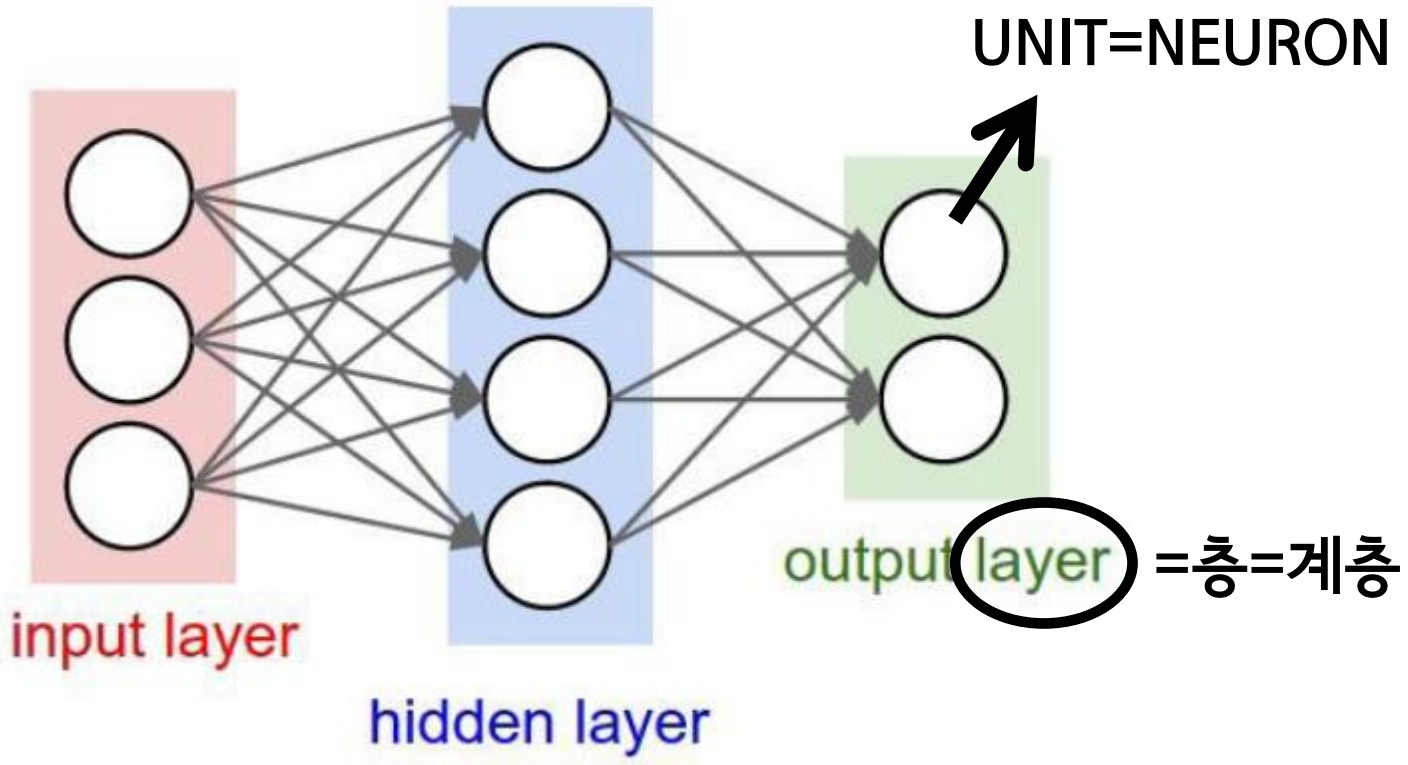
학습 데이터 확보에
시간과 비용 ↑

컴퓨터의 학습내용을
알 수 없어서 **해석 불가능**

과적합 문제 (Overfitting)

2 딥러닝 구조

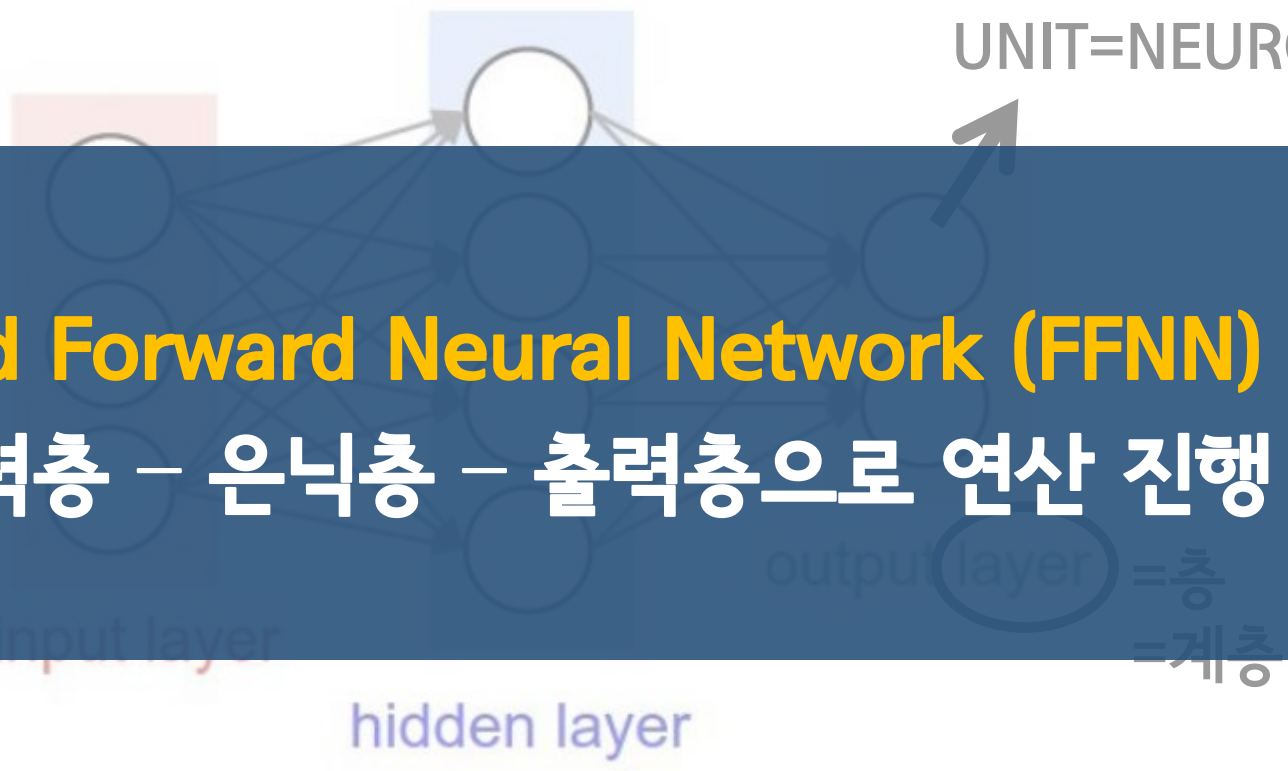
기본구조



DL은 Hidden Layer가 두 개 이상 (DNN)

2 딥러닝 구조

기본구조



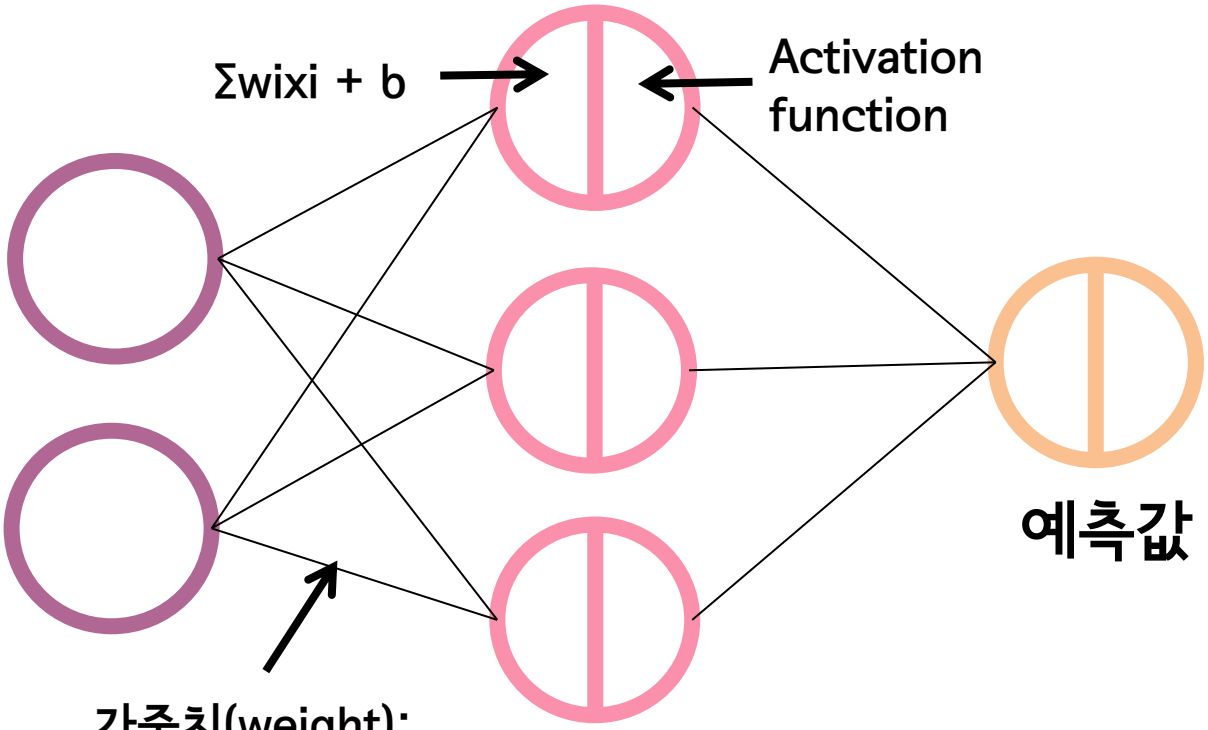
Feed Forward Neural Network (FFNN)

: 입력층 – 은닉층 – 출력층으로 연산 진행

DL은 Hidden Layer가 두 개 이상 (DNN)

2 딥러닝의 구조

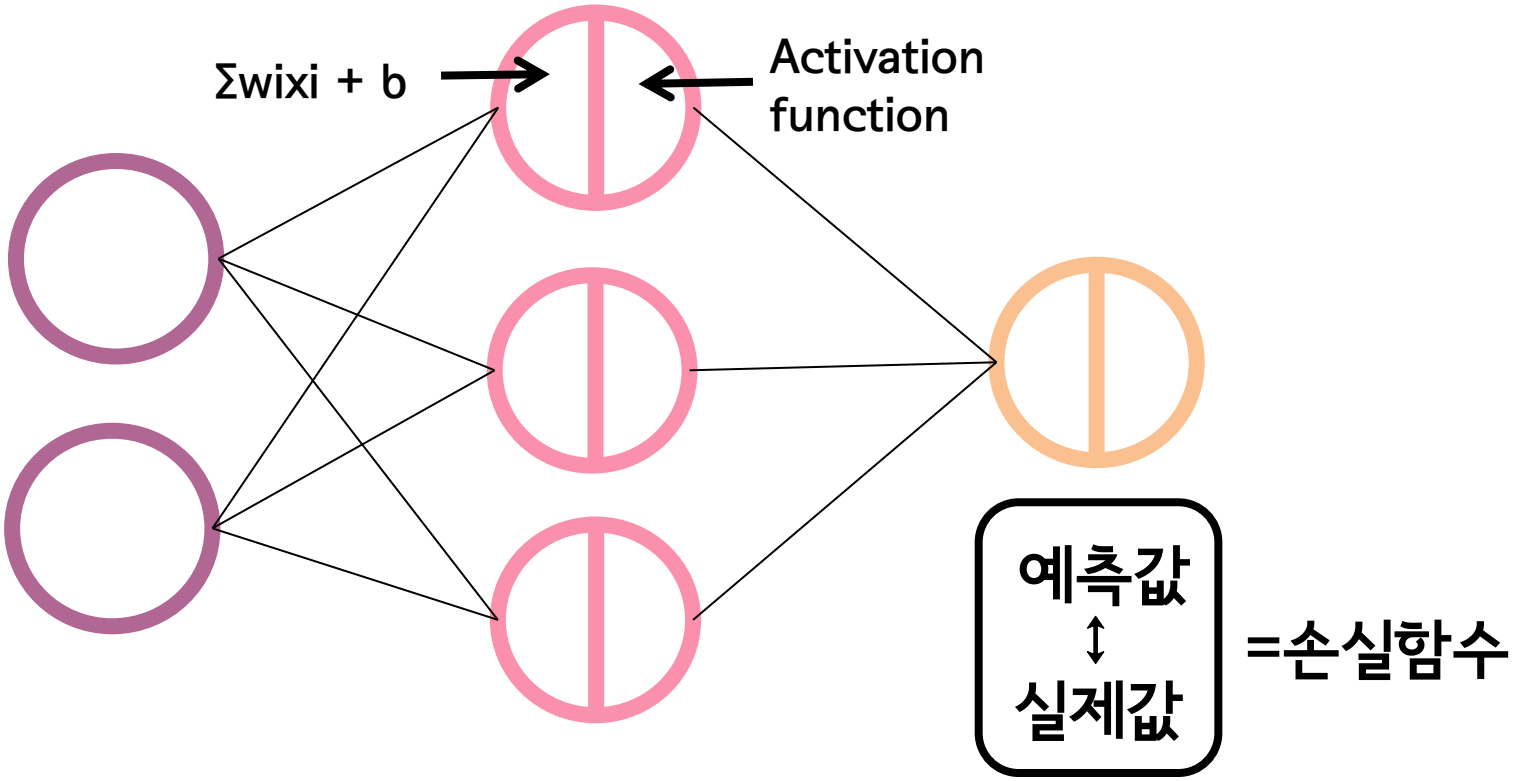
Flow Forward



1. 활성화 함수로 입력값에 대한 예측을 계산

2 딥러닝의 구조

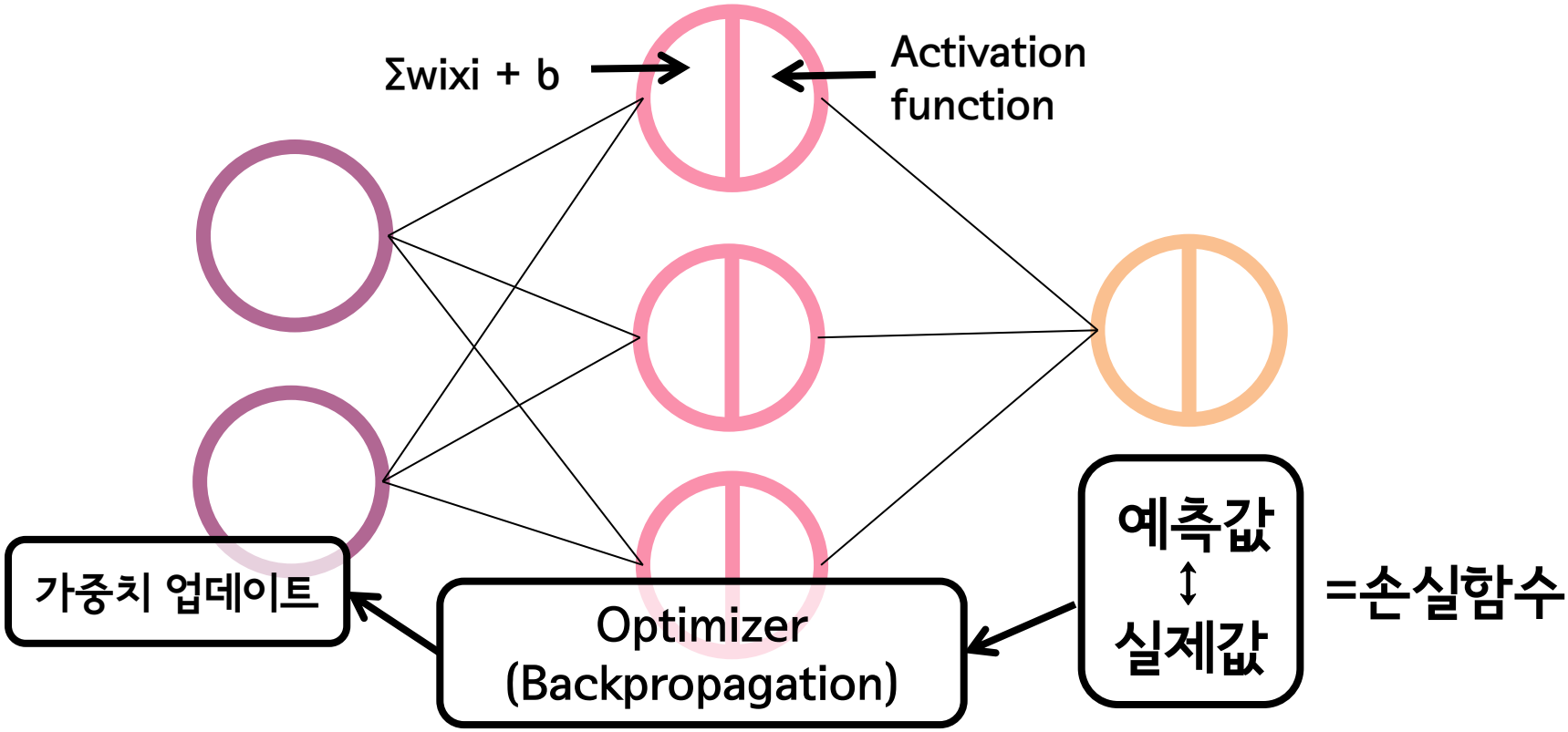
Flow Forward



2. 손실 함수로 예측값과 실제값의 오차 측정

2 딥러닝의 구조

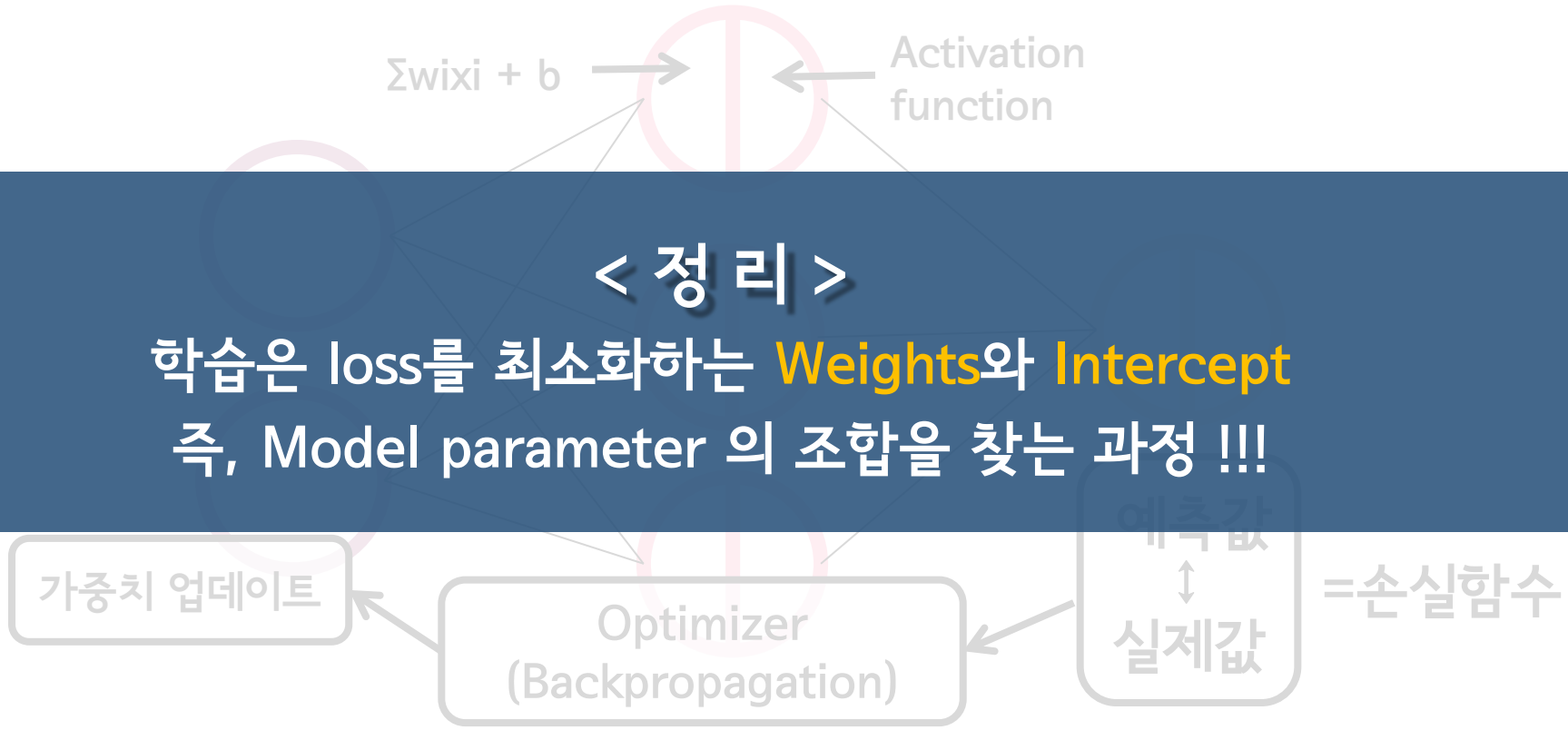
Flow Backward



3. Backpropagation: feature selection, 가중치 업데이트 (using Chain rule)

2 딥러닝의 구조

Flow Backward



3. Backpropagation: feature selection, 가중치 업데이트 (using Chain rule)

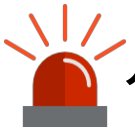
2 딥러닝의 구조

Activation Functions

Activation Function:
각 Layer의 input을
output으로 변환해주는 함수

활성화 함수로
거의 다 **비선형 함수** 사용

ex) sigmoid, relu, tanh



선형 함수를 잘 사용하지 않는 이유

층을 쌓는 이점이 **상쇄**되기 때문!

f(x) = cx를 사용하여
3개의 layer을 쌓는 경우,

y = f(f(f(x))) = c^3x와 같음

즉, layer를 여러 번 추가하더라도
한 번 추가한 것과 같게 됨

2 딥러닝은 어떻게 생겼을까?

Optimizer | Stochastic Gradient Descent

Gradient descent (경사하강법) 란?!

기울기(gradient)를 사용해 모델을 조정하는
optimization 알고리즘

가장 일반적인 optimizer

Gradient를 고려해 기울어진 방향으로 이동하는 방식

매 epoch마다 train set의 샘플 순서를 섞어
optimal weights를 계산

→ 다양하게 탐색해야 optimal weights를
제대로 찾을 수 있기 때문

2 딥러닝은 어떻게 생겼을까?

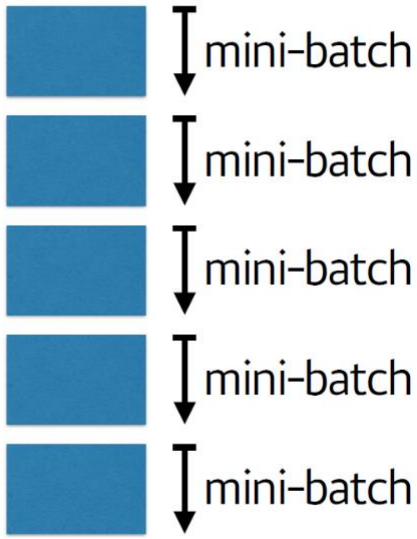
Optimizer

Stochastic Gradient Descent

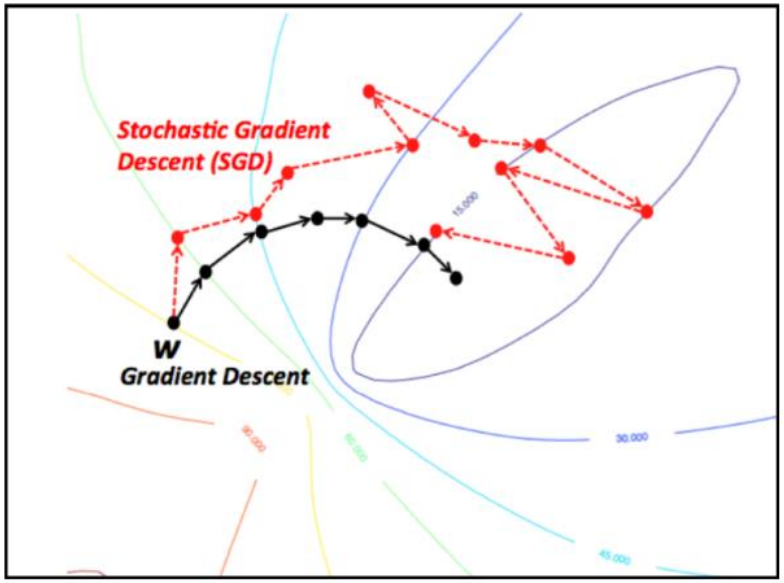
Gradient Decent

전부다 읽고나서
최적의 1스텝 간다.

Stochastic Gradient Decent



작은 토막마다
일단 1스텝간다.



3 더 좋은 딥러닝 모델

Generalization

새로운 데이터에 대해

올바른 예측 혹은 분류 가능한 모델 개발이 목적

정확한 예측 및 분류를 위해서는

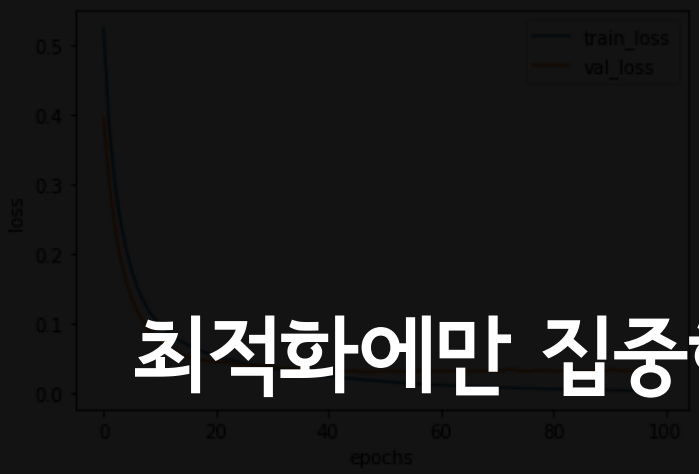
모델의 성능을 높이는 최적화 과정이 필요한데

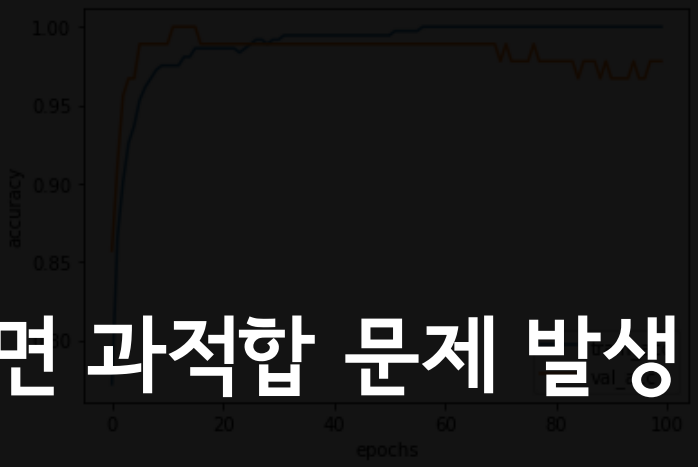
새로운 데이터에 대해서도

좋은 성능을 보이도록 하는 것이 중요!!

3 더 좋은 딥러닝 모델

Generalization





최적화에만 집중하면 과적합 문제 발생

최적화를 하는 동시에 일반화 고려

Epoch 를 늘릴수록 train set 에 대한 성능 높아짐

Validation set 에 대한 성능은 어느 순간 낮아짐

3 더 좋은 딥러닝 모델

Avoiding Overfitting

유닛 및 레이어 줄이기

모델이 너무 단순해도 성능 저하

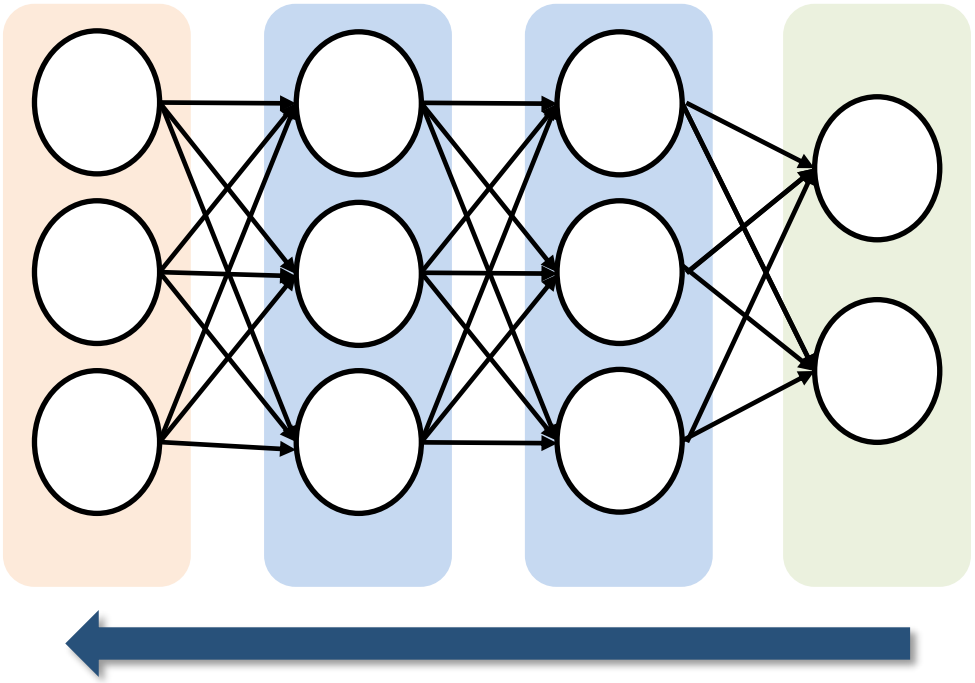
적정한 유닛과 레이어 수 결정하는 공식 없음

적은 수의 레이어와 유닛으로 시작하여

점차 모델의 크기를 조절

3 더 좋은 딥러닝 모델

Avoiding Vanishing Gradient



미분하여 기울기가
0이 되어 Gradient가
사라지지 않는 것이 중요

Backpropagation 과정에서

미분을 통해 가중치를 갱신하면서 최적화

3 더 좋은 딥러닝 모델

Avoiding Vanishing Gradient

Batch Normalization

내부 공변량 변화 문제를 해결하고 Vanishing Gradient를 막기 위해
Batch Normalization 사용

각 층에서 활성화 함수 통과하기 전 수행
배치 단위로 접근하여 정규화 진행
정규화된 값에 매개변수 γ와 β를 사용하여 스케일과 시프트 진행