04

딥러닝

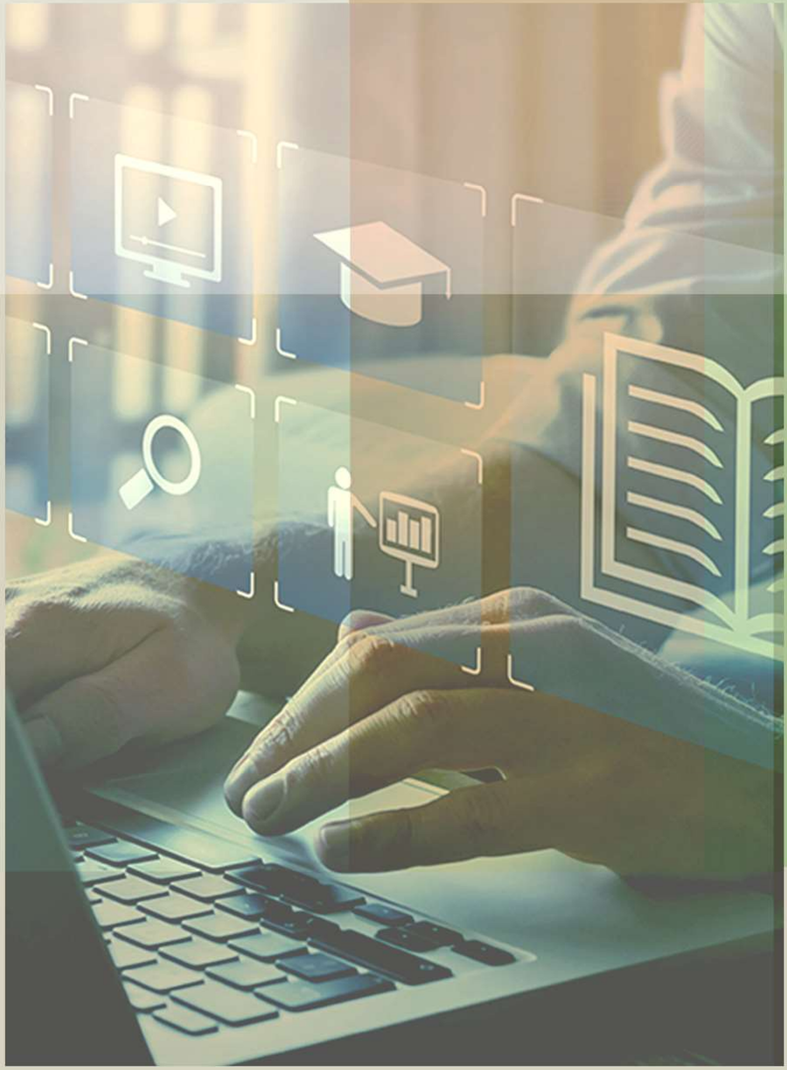
딥러닝의 학습 기술(1)

방송대 컴퓨터과학과 이병래 교수

학습목차

01

최적화와 경사 하강법



1. 경사 하강법의 개념

딥러닝의 목적

- 설계한 모델이 가장 바람직한 결론을 내릴 수 있도록 학습표본 집합을 이용하여 모델 내부의 파라미터가 최적의 값이 되게 조정하는 훈련을 하는 것

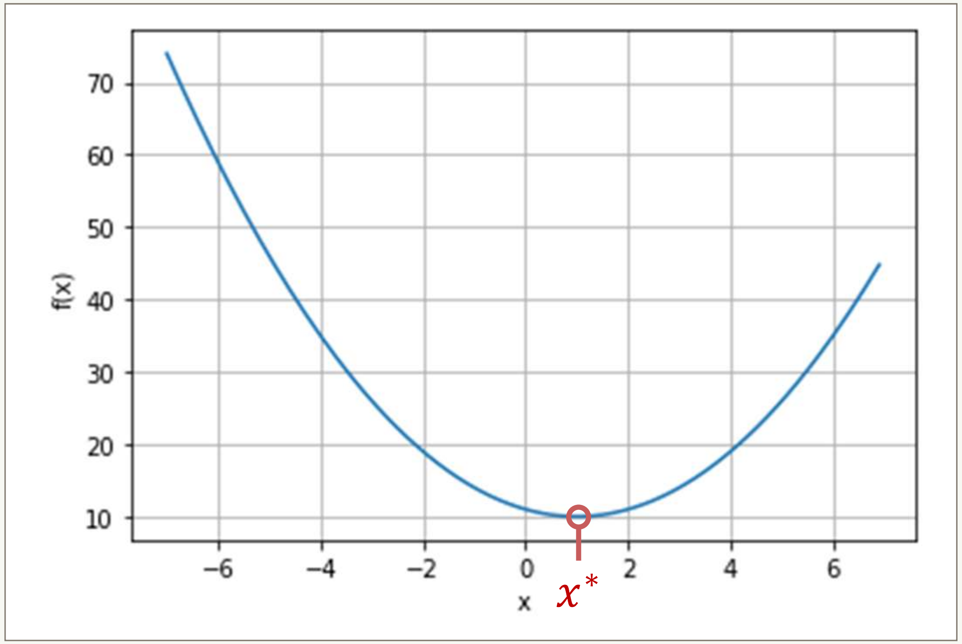
최적화 문제(optimization problem)

- 목적함수(objective function)를 최적화하는 파라미터를 결정하는 문제
  - 딥러닝 : 훈련 데이터 집합에 대한 손실함수(loss function)을 최소화하는 문제

1. 경사 하강법의 개념

볼록 최적화(convex optimization)

- 목적함수가 볼록함수(convex function)이고 해를 찾기 위한 정의역이 볼록집합(convex set)인 최적화 문제

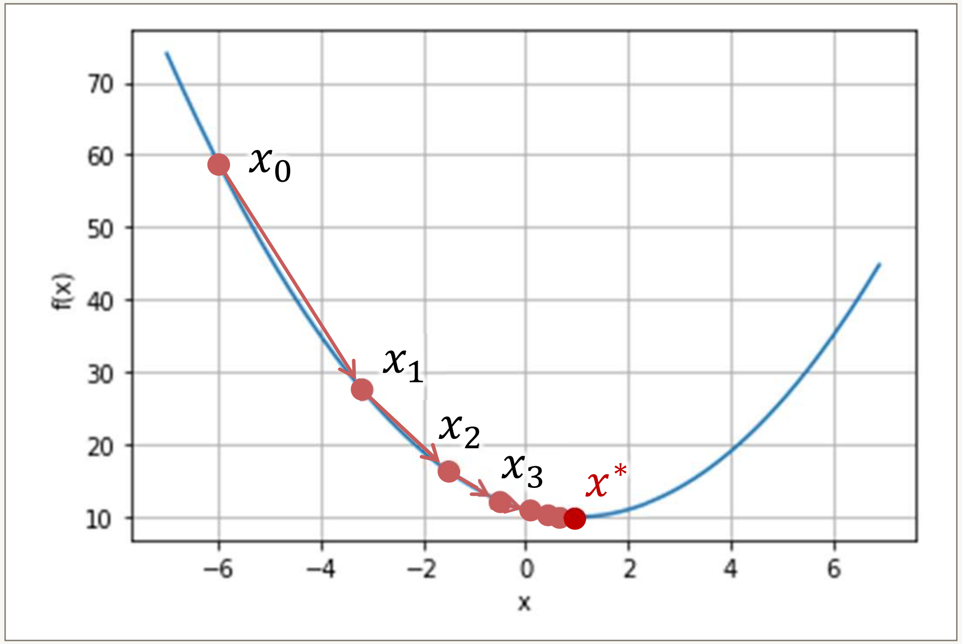


➡ 경사 하강법으로 해를 구할 수 있음

1. 경사 하강법의 개념

경사 하강법(gradient descent)

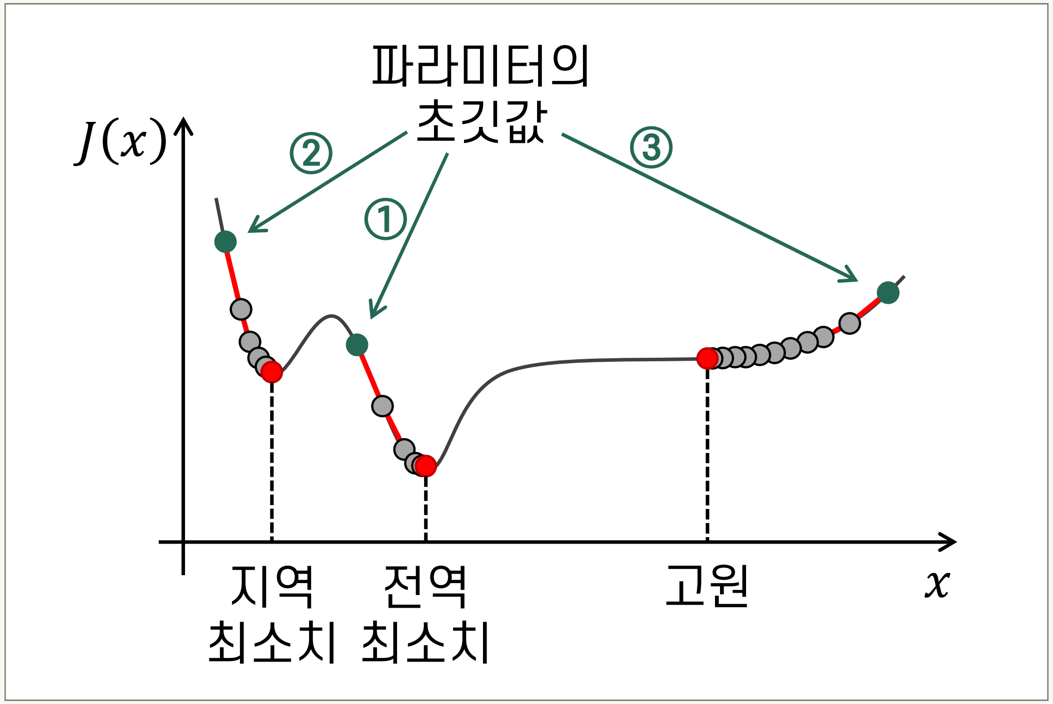
- 적절히 선택된 초깃값 x_0로부터 시작하여 경사를 따라 이동하여 목적함수의 최솟값에 해당되는 지점인 x^*에 도달하는 방법



1. 경사 하강법

경사 하강법(gradient descent)

- 목적함수가 볼록함수가 아니라면 경사 하강법이 최적화의 성공을 보장할 수 없음



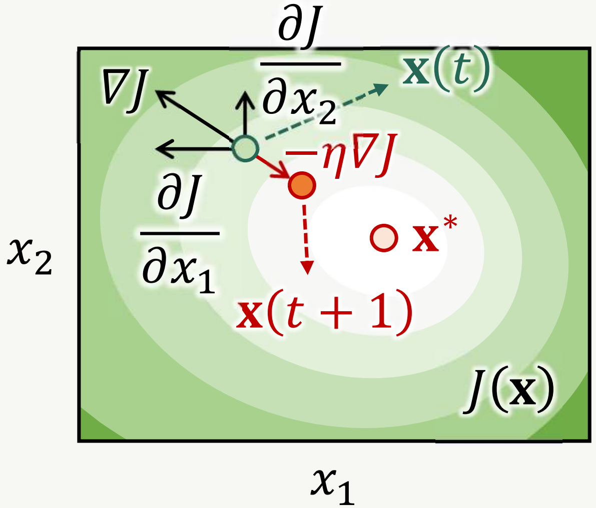
1. 경사 하강법의 개념

경사 하강법을 이용한 볼록함수의 최적화

- 실함수인 목적함수 J가 최소가 되는 파라미터 $\mathbf{x}^*$ 구하기

$$J: \mathbb{R}^d \rightarrow \mathbb{R}$$

- 초깃값 $\mathbf{x}(0)$에서 시작하여 J의 경사의 음의 방향으로 $\mathbf{x} = [x_1\ x_2\ \cdots\ x_d]^T$를 이동하는 것을 반복함



$$\mathbf{x}(t+1) = \mathbf{x}(t) - \eta\nabla J\big(\mathbf{x}(t)\big)$$

η : 학습률, t : 업데이트 횟수

$$\nabla J(\mathbf{x}) = \left[\frac{\partial J(\mathbf{x})}{\partial x_1}\ \ \frac{\partial J(\mathbf{x})}{\partial x_2}\ \ \cdots\ \ \frac{\partial J(\mathbf{x})}{\partial x_d}\right]^T$$

1. 경사 하강법의 개념

경사 하강법을 이용한 볼록함수의 최적화

예

목적함수 : $J(\mathbf{x}) = x_1^2 + 2x_2^2 - x_1 x_2$

$\mathbf{x}$의 초깃값 : $\mathbf{x}(0) = [-5 \quad 4]^T$

학습률 : $\eta = 0.1$

$$\mathbf{x}(t+1) = \mathbf{x}(t) - \eta \nabla J\big(\mathbf{x}(t)\big)$$

$$\nabla J(\mathbf{x}) = [2x_1 - x_2 \quad 4x_2 - x_1]^T$$

$$\mathbf{x}(0) = [-5 \quad 4]^T \quad \rightarrow \nabla J\big(\mathbf{x}(0)\big) = [-14 \quad 21]^T$$

$$\mathbf{x}(1) = [-3.6 \quad 1.9]^T \quad \rightarrow \nabla J\big(\mathbf{x}(1)\big) = [-9.1 \quad 11.2]^T$$

$$\mathbf{x}(2) = [-2.69 \quad 0.78]^T \quad \rightarrow \nabla J\big(\mathbf{x}(2)\big) = [-6.16 \quad 5.81]^T$$

$$\cdots\cdots$$

$$\mathbf{x}(30) = [-0.0161 \quad -0.00665]^T$$

1. 경사 하강법의 개념

경사 하강법을 이용한 볼록함수의 최적화

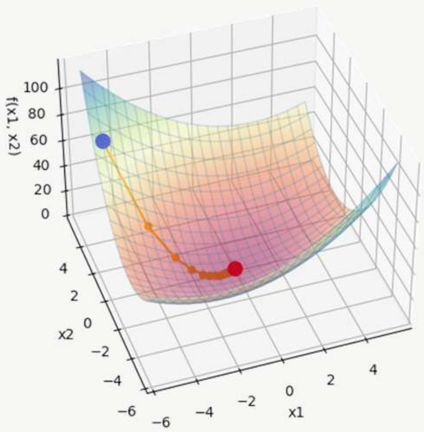
예

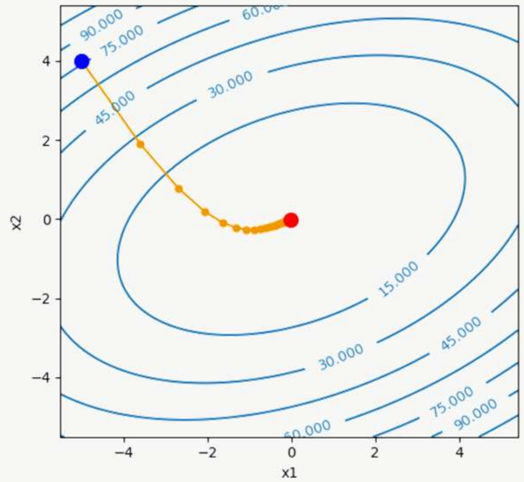
목적함수 : $J(\mathbf{x}) = x_1^2 + 2x_2^2 - x_1 x_2$

$\mathbf{x}$의 초깃값 : $\mathbf{x}(0) = [-5 \quad 4]^T$

학습률 : $\eta = 0.1$

$$\mathbf{x}(t+1) = \mathbf{x}(t) - \eta \nabla J\big(\mathbf{x}(t)\big)$$





1. 경사 하강법의 개념

경사 하강법을 이용한 볼록함수의 최적화

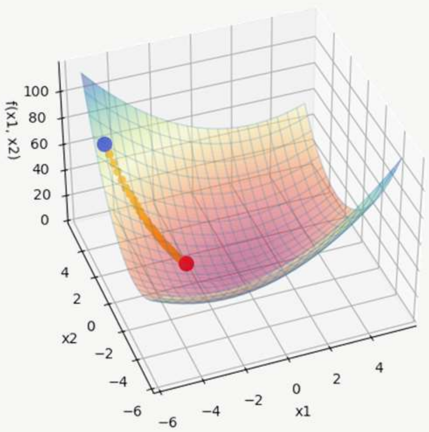
예

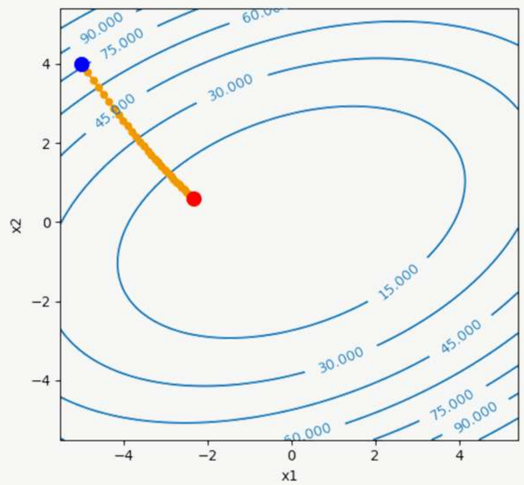
목적함수 : $J(\mathbf{x}) = x_1^2 + 2x_2^2 - x_1 x_2$

$\mathbf{x}$의 초깃값 : $\mathbf{x}(0) = [-5 \quad 4]^T$

학습률 : $\eta = 0.01$

$$\mathbf{x}(t+1) = \mathbf{x}(t) - \eta \nabla J\big(\mathbf{x}(t)\big)$$





1. 경사 하강법의 개념

경사 하강법을 이용한 볼록함수의 최적화

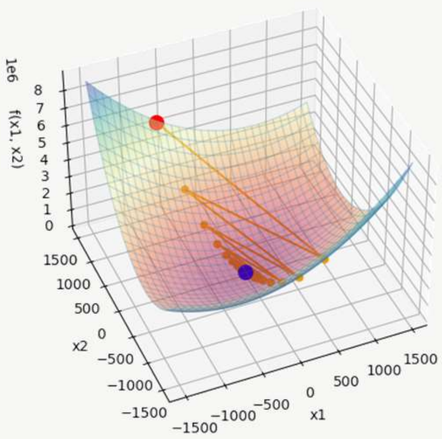
예

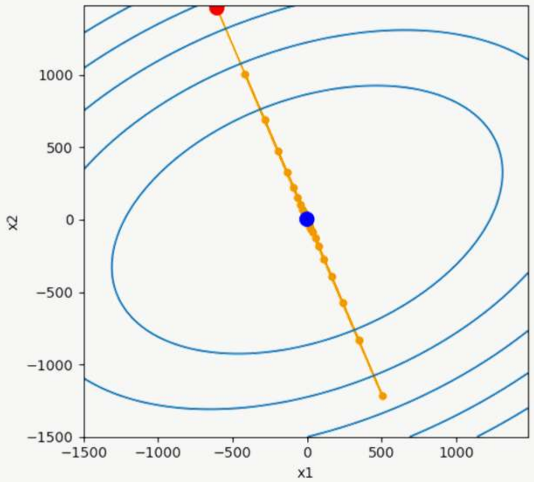
목적함수 : $J(\mathbf{x}) = x_1^2 + 2x_2^2 - x_1x_2$

$\mathbf{x}$의 초깃값 : $\mathbf{x}(0) = [-5 \quad 4]^T$

학습률 : $\eta = 0.5$

$$\mathbf{x}(t+1) = \mathbf{x}(t) - \eta\nabla J\big(\mathbf{x}(t)\big)$$



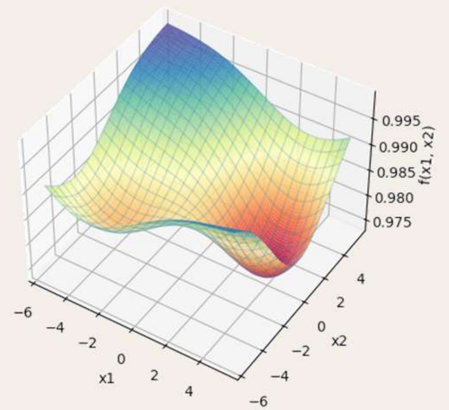


1. 경사 하강법의 개념

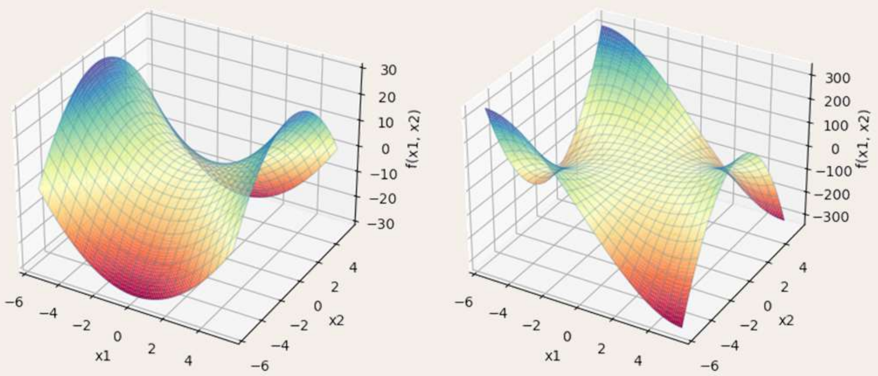
경사 하강법의 문제점

- 목적함수가 볼록함수가 아니라면 해의 탐색에 실패할 수 있음
  - 모든 파라미터에 대한 목적함수의 편미분이 0이지만 전역 최소치에 해당되지 않는 파라미터 값인 경우

지역 최소치(local minima)



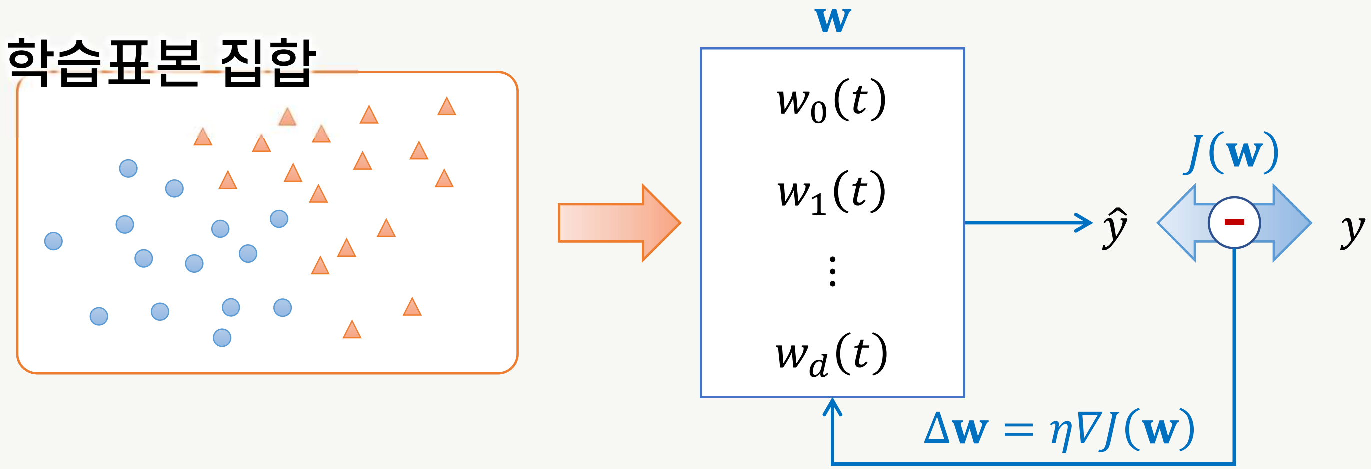
안장점(saddle point)



2. 경사 하강법의 구현

배치 경사 하강법(Batch Gradient Descent, 배치 GD)

- 모든 훈련용 표본으로 한 단계의 파라미터 업데이트를 위한 경사를 계산하는 방식



2. 경사 하강법의 구현

배치 경사 하강법(Batch Gradient Descent, 배치 GD)

- 목적함수

$$J(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N} E_i(\mathbf{w}),$$

N : 훈련에 사용되는 표본의 수

$E_i(\mathbf{w})$: i번째 표본에 대한 손실

$\mathbf{w}$: 가중치, 바이어스 등의 학습 대상 파라미터

$$\nabla J(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N} \nabla E_i(\mathbf{w}),$$

$$\mathbf{w}(t+1) = \mathbf{w}(t) - \eta \nabla J(\mathbf{w})$$

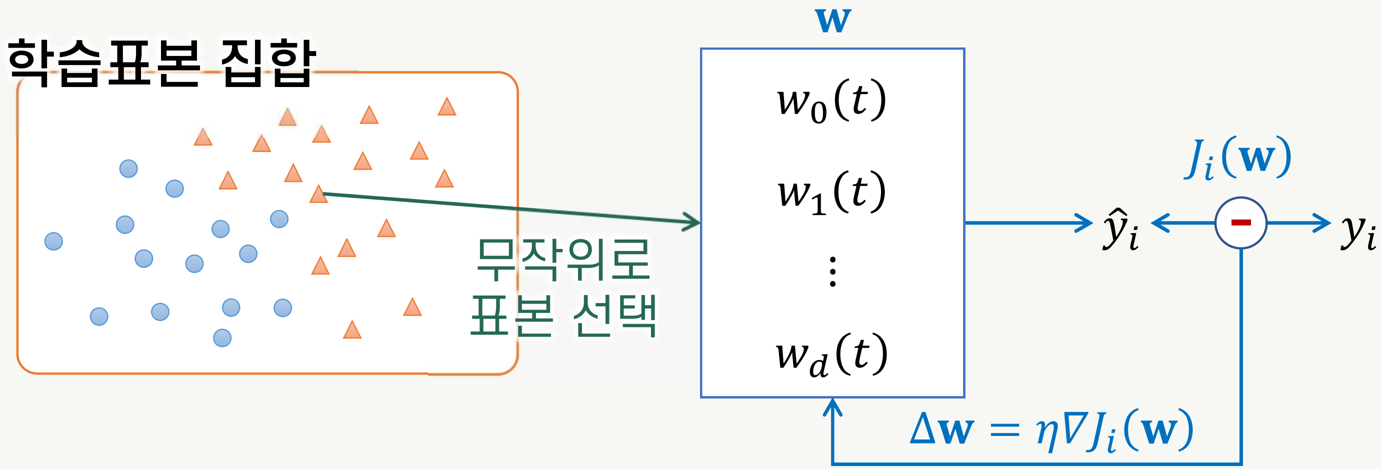
➡ 하나의 에폭(epoch)에 파라미터 한 번 업데이트됨

➡ 한 번의 업데이트에 긴 계산 시간을 소비함

2. 경사 하강법의 구현

확률적 경사 하강법(Stochastic Gradient Descent, SGD)

- 훈련 집합에서 무작위 순서로 하나씩 표본 $i \in \{1, 2, \cdots, N\}$에 대한 경사 $\nabla J_i(\mathbf{w}) = \nabla E_i(\mathbf{w})$를 계산하여 파라미터를 업데이트함

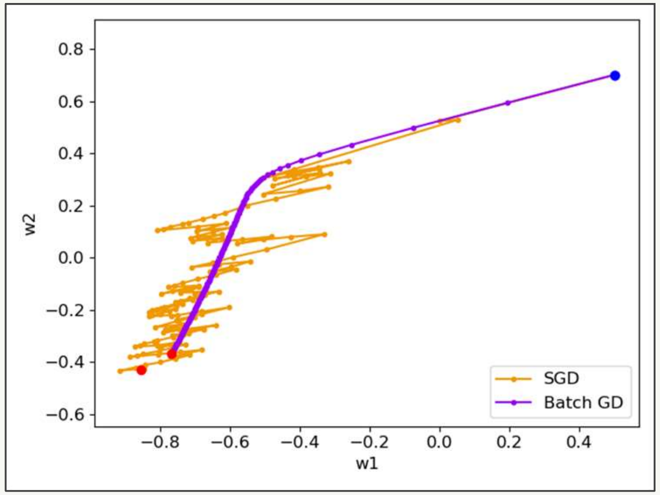


$$\mathbf{w}(t+1) = \mathbf{w}(t) - \eta\nabla J_i(\mathbf{w})$$

2. 경사 하강법의 구현

확률적 경사 하강법(Stochastic Gradient Descent, SGD)

- 1회의 에폭에 파라미터가 N번 업데이트 됨
  - ➡ 배치 방식에 비해 매우 빠르게 파라미터 업데이트가 진행됨
- 무작위로 표본을 선택하고, 표본 단위로 파라미터가 업데이트 됨
  - ➡ 배치 경사 하강법에 비해 파라미터의 변화가 불규칙하게 진행됨



- 단층 피드포워드 신경망
- 붓꽃 데이터에 대한 150회 업데이트 과정

2. 경사 하강법의 구현

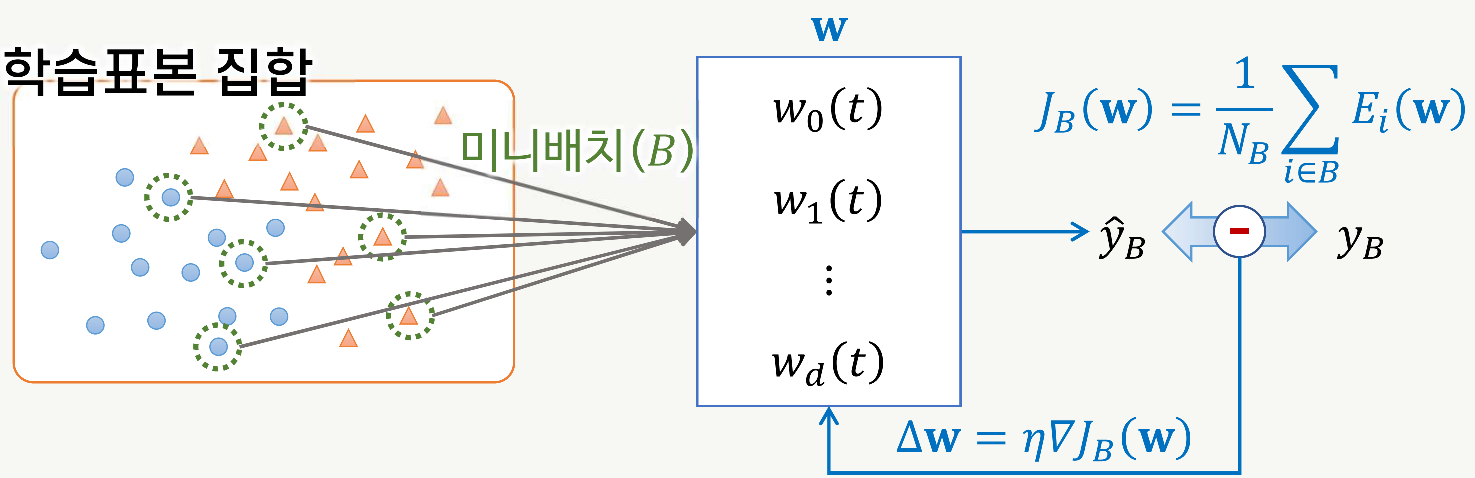
확률적 경사 하강법(Stochastic Gradient Descent, SGD)

- 1회의 에폭에 파라미터가 N번 업데이트 됨
  - ➡ 배치 방식에 비해 매우 빠르게 파라미터 업데이트가 진행됨
- 무작위로 표본을 선택하고, 표본 단위로 파라미터가 업데이트 됨
  - ➡ 배치 경사 하강법에 비해 파라미터의 변화가 불규칙하게 진행됨
  - ➡ 지역 최소치, 안장점 등에서 빠져나오는 데 도움이 될 수 있음
- 극소점 근처에 도달한 상태에서도 파라미터가 계속하여 변화
  - ➡ 최적의 파라미터로 수렴하지 않을 수 있음
  - ➡ 동적 학습률 적용 : 에폭에 따라 학습률을 점차 작은 값으로 줄임

2. 경사 하강법의 구현

미니배치 확률적 경사 하강법(mini-batch SGD)

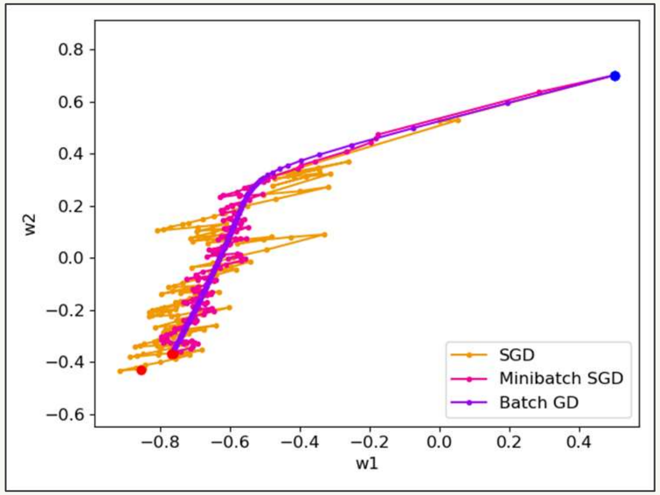
- 전체 학습표본 집합을 '미니배치'라고 하는 작은 크기의 부분집합으로 분할하여 모델을 훈련
  - 각각의 미니배치는 학습표본 집합에서 무작위로 선택함
  - 파라미터의 업데이트는 미니배치 단위로 함



2. 경사 하강법의 구현

미니배치 확률적 경사 하강법(mini-batch SGD)

- 배치 경사 하강법에 비해 빠르게 파라미터 업데이트를 진행할 수 있음
  - 하나의 에폭에 파라미터를 $\lceil N/N_B \rceil$회 업데이트함
- SGD에 비해 파라미터 업데이트의 불규칙성이 완화되어 최적값에 가깝게 파라미터 값이 결정될 수 있음



- 단층 피드포워드 신경망
- 붓꽃 데이터에 대한 150회 업데이트 과정

2. 경사 하강법의 구현

미니배치 확률적 경사 하강법(mini-batch SGD)

- 배치 경사 하강법에 비해 빠르게 파라미터 업데이트를 진행할 수 있음
  - 하나의 에폭에 파라미터를 $\lceil N/N_B \rceil$회 업데이트함
- SGD에 비해 파라미터 업데이트의 불규칙성이 완화되어 최적값에 가깝게 파라미터 값이 결정될 수 있음
- 계산 성능을 높일 수 있음
  - 특히 GPU를 사용하는 경우 미니배치 단위의 처리를 하면 행렬 연산의 최적화에 유리함

2. 경사 하강법의 구현

텐서플로(Keras)에서 경사 하강법의 구현

▪ 모델의 컴파일

예

```
bp_model_tf.compile(optimizer=optimizers.SGD(0.1, momentum=0.9),
                    loss=losses.SparseCategoricalCrossentropy(),
                    metrics=['accuracy'])
```

▪ 모델의 훈련

예

```
bp_model_tf.fit(X_tr, y_tr, batch_size=15, epochs=1000,
                verbose=2, validation_data=(X_val, y_val))
```

3. 동적 학습률

학습의 진척에 따라 점차 학습률을 줄이는 방식

① 계단형 감쇠 스케줄러

- 반복 횟수의 구간을 정하여 각 구간에 정해진 학습률을 적용함
- tf.keras.optimizers.schedules 모듈의 PiecewiseConstantDecay 클래스 인스턴스 활용

예

```
boundaries = [700, 900]
values = [0.1, 0.05, 0.01]
lr_fn = optimizers.schedules.PiecewiseConstantDecay(
    boundaries, values)
bp_model_tf.compile(
    optimizer=optimizers.SGD(lr_fn, momentum=0.9),
    loss=losses.SparseCategoricalCrossentropy(),
    metrics=['accuracy'])
```

3. 동적 학습률

학습의 진척에 따라 점차 학습률을 줄이는 방식

② 지수함수 감쇠 스케줄러

- 반복 횟수에 따라 초깃값으로부터 지수함수 형태로 감쇠함
- tf.keras.optimizers.schedules 모듈의 ExponentialDecay 클래스 인스턴스 활용

예

```
lr_fn = optimizers.schedules.ExponentialDecay(
    initial_learning_rate = 0.1,
    decay_steps = 500,
    decay_rate = 0.5,
    staircase = False)
```

$$\eta = 0.1 \cdot 0.5^{epoch/500}$$

3. 동적 학습률

학습의 진척에 따라 점차 학습률을 줄이는 방식

③ 다항식 감쇠 스케줄러

- 반복 횟수의 다항식 함수에 의해 감쇠함
- tf.keras.optimizers.schedules 모듈의 PolynomialDecay 클래스 인스턴스 활용

예

```
lr_fn = optimizers.schedules.PolynomialDecay(
    initial_learning_rate = 0.1,
    decay_steps = 900,
    end_learning_rate = 0.01,
    power = 0.5)
```

$$step = \min(epoch, 900)$$

$$\eta = (0.1 - 0.01) \cdot (1 - step/900)^{0.5} + 0.01$$

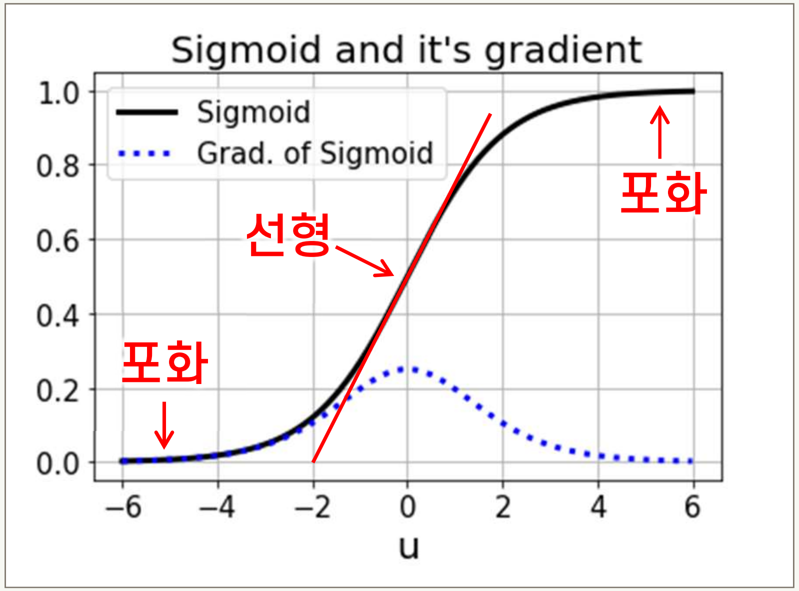
02

심층 신경망의 학습 문제

1. 불안정한 경사(unstable gradient) 문제

경사 소멸(vanishing gradient) 문제

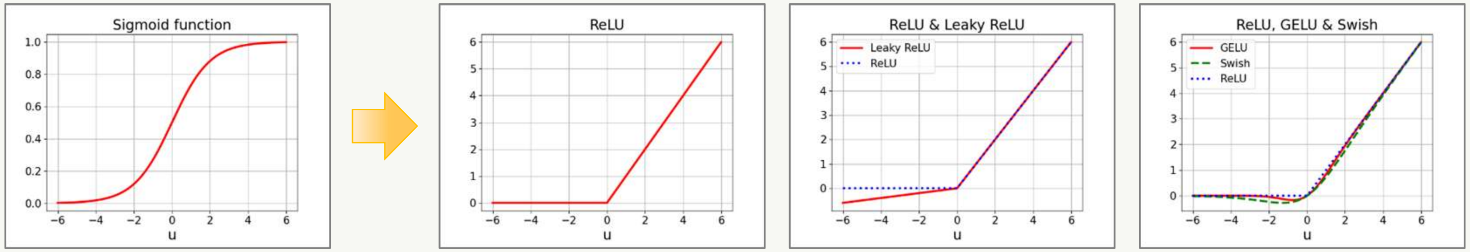
- 심층망을 학습하는 과정에서 입력층으로 갈수록 경사의 크기가 0에 근접하여 연결 가중치의 업데이트가 진행되지 않는 현상



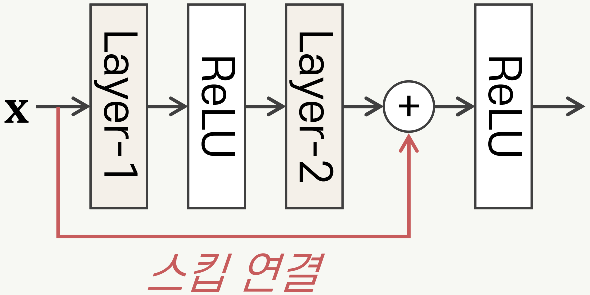
1. 불안정한 경사(unstable gradient) 문제

경사 소멸(vanishing gradient) 문제

- 경사 소멸 문제의 개선 방법
  - 활성함수 개선



  - 스킵 연결



  - 가중치의 적절한 초기화
  - 배치 정규화

1. 불안정한 경사(unstable gradient) 문제

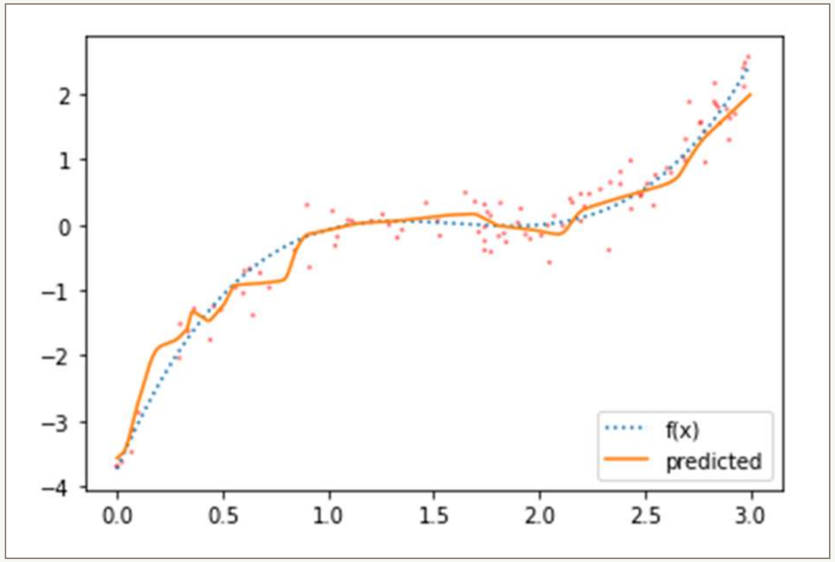
경사 폭발(exploding gradient) 문제

- 경사가 점점 더 큰 값을 가져 연결 가중치가 발산하는 상황
- 원인
  - 부적절한 가중치의 초깃값
  - 지나치게 높은 학습률 등
- 개선 방법
  - 배치 정규화
  - 경사 절단(gradient clipping)
  - 규제(regularization)
  - 최적화 알고리즘의 개선(Adma, RMSprop 등)

2. 과적합(overfitting) 문제

과적합이란?

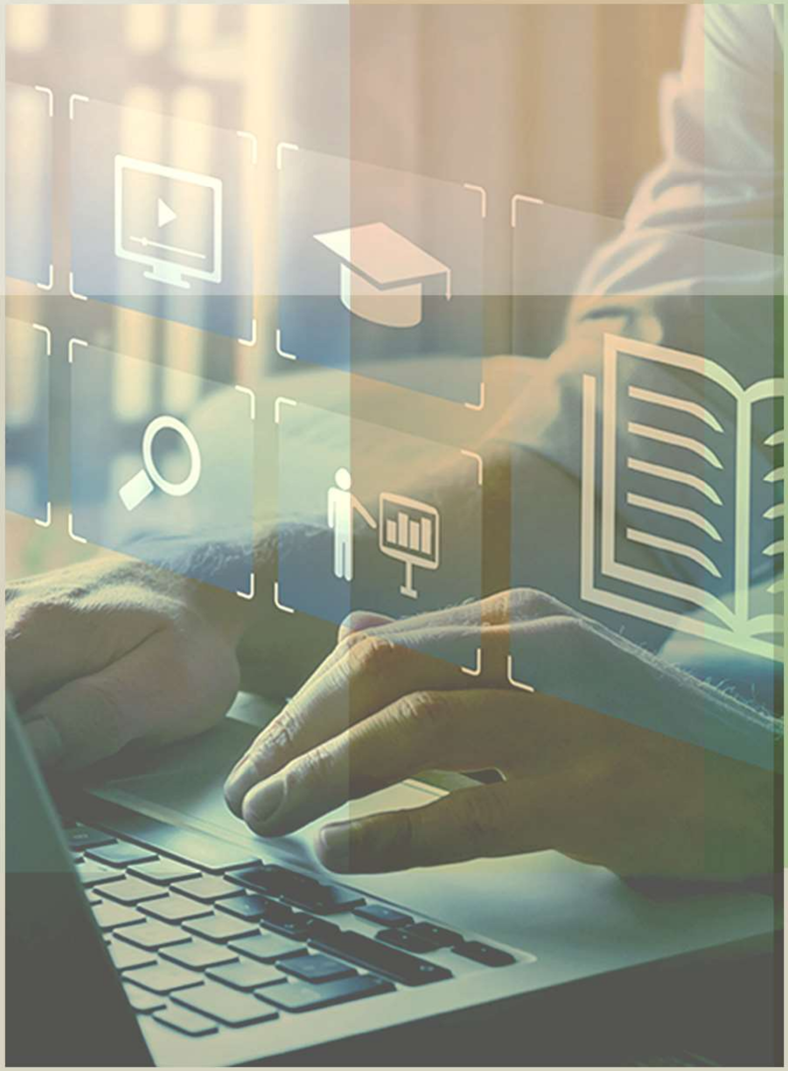
- 특정 학습 데이터 집합에 지나치게 의존적으로 학습되는 현상
  - ➡ 일반화 오류 발생



- 개선 방법 : 드롭아웃(dropout), 규제, 데이터 증강(data augmentation) 등

03

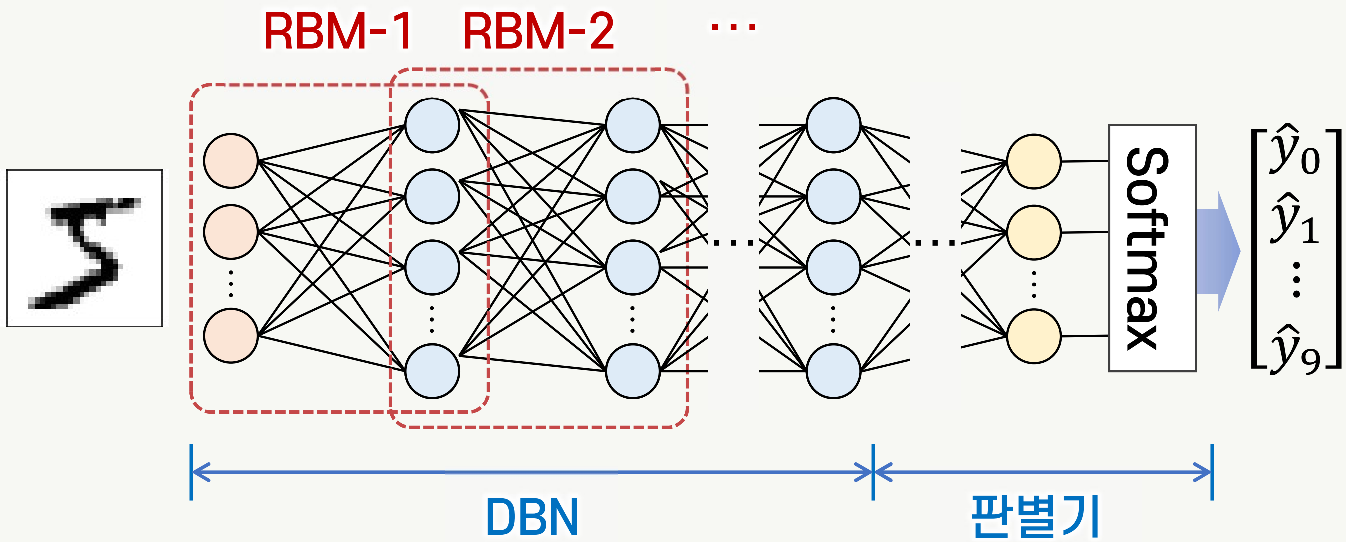
가중치의 초기화



1. 사전 학습에 의한 가중치 초기화

심층 신뢰망(deep belief nets, DBN)을 이용한 가중치 초기화

- Geoffrey Hinton(2006) : 연결가중치를 단순히 랜덤 값으로 초기화하는 것보다는 적절한 방식으로 초기화함으로써 성능을 개선할 수 있음
  - 심층 신뢰망을 사전 학습을 하여 연결가중치를 초기화



2. Glorot 초기화 방법

Xavier Glorot, Yoshua Bengio(2010)

- 연결 가중치를 뉴런의 팬-인(fan-in)과 팬-아웃(fan-out)에 따라 결정되는 값의 범위에 속하는 랜덤 값으로 초기화
- Keras의 초기화를 위한 모듈인 `initializers`에 2가지 유형의 초기화기 제공

① `tf.keras.initializers.GlorotUniform`

- `[-limit, limit]` 범위의 균등분포로 초깃값을 선택

$$\text{limit} = \sqrt{\frac{6}{fan_{in} + fan_{out}}}$$

예

```
from tensorflow.keras import layers
dense_layer = layers.Dense(10, activation='relu',
                           kernel_initializer='glorot_uniform')
```

2. Glorot 초기화 방법

Xavier Glorot, Yoshua Bengio(2010)

- 연결 가중치를 뉴런의 팬-인(fan-in)과 팬-아웃(fan-out)에 따라 결정되는 값의 범위에 속하는 랜덤 값으로 초기화
- Keras의 초기화를 위한 모듈인 `initializers`에 2가지 유형의 초기화기 제공

② `tf.keras.initializers.GlorotNormal`

- 평균이 0, 표준편차가 σ인 정규분포로 초깃값을 선택

$$\sigma = \sqrt{\frac{2}{fan_{in} + fan_{out}}}$$

예

```
from tensorflow.keras import layers
dense_layer = layers.Dense(10, activation='relu',
                           kernel_initializer='glorot_normal')
```

3. He 초기화 방법

Kaiming He, et al(2016)

- ReLU 유형의 활성함수를 사용하는 신경망에 적합한 초기화 방법을 제안
- 팬-인을 바탕으로 하여 정해지는 랜덤 값에 따라 가중치를 초기화함
- Keras의 초기화를 위한 모듈인 `initializers`에 2가지 유형의 초기화기 제공

① `tf.keras.initializers.HeUniform`

- `[-limit, limit]` 범위의 균등분포로 초깃값을 선택

$$\text{limit} = \sqrt{\frac{6}{fan_{in}}}$$

예

```
from tensorflow.keras import layers
dense_layer = layers.Dense(10, activation='relu',
                           kernel_initializer='he_uniform')
```

3. He 초기화 방법

Kaiming He, et al(2016)

- ReLU 유형의 활성함수를 사용하는 신경망에 적합한 초기화 방법을 제안
- 팬-인을 바탕으로 하여 정해지는 랜덤 값에 따라 가중치를 초기화함
- Keras의 초기화를 위한 모듈인 `initializers`에 2가지 유형의 초기화기 제공

② `tf.keras.initializers.HeNormal`

- 평균이 0, 표준편차가 σ인 정규분포로 초깃값을 선택

$$\sigma = \sqrt{\frac{2}{fan_{in}}}$$

예

```
from tensorflow.keras import layers
dense_layer = layers.Dense(10, activation='relu',
                           kernel_initializer='he_normal')
```

정리하기

- 경사 하강법은 목적함수의 최솟값에 해당되는 파라미터의 해를 구하기 위한 기법 중 하나로, 적절히 선택된 파라미터 초깃값에서 시작하여 목적함수의 경사를 따라 이동하여 해에 도달하는 방법이다.
- 목적함수가 볼록함수가 아니라면 지역 최소치, 안장점 등의 문제로 인해 단순한 경사 하강법으로 해의 탐색에 실패할 가능성이 있다.
- 배치 경사 하강법은 모든 훈련용 표본에 대한 손실함수 경사의 평균을 이용하여 한 단계의 파라미터 업데이트를 하는 과정을 반복한다.

정리하기

- 확률적 경사 하강법은 훈련 집합에서 무작위로 표본을 선택하여 이에 대한 손실함수 경사를 이용하여 파라미터 업데이트를 하는 과정을 반복한다.
- 미니배치 SGD는 전체 학습표본 집합을 작은 크기의 부분집합으로 분할한 미니배치 단위로 SGD를 수행한다.
- 미니배치 SGD는 배치 경사 하강법에 비해 빠르게 파라미터 업데이트를 진행할 수 있으며, 단순한 SGD에 비해 파라미터 업데이트의 불규칙성이 완화되어 최적값에 가깝게 파라미터 값이 결정될 수 있다.

정리하기

- 에폭이 진행됨에 따라 점차 학습률을 줄이는 동적 학습률을 활용할 수 있다.
- 심층망을 학습하는 과정에서 불안정한 경사 문제, 과적합 문제 등은 올바른 학습의 진행을 가로막는 요인이다.
- 심층망의 학습을 개선하기 위해 ReLU 등의 활성함수, 가중치의 적절한 초기화, 배치 정규화, 규제, 드롭아웃 등 다양한 기술이 활용된다.

다음시간안내

05

딥러닝의 학습 기술(2)