AWS DeepRacer 와 강화학습으로 배우는 자율주행 AI 만들기

인공지능과 자율주행

동양미래대학교

학습목표

- 자율주행에 이용되는 인공지능 기술에 대해 이해한다.
- 머신러닝 및 딥러닝의 작동 원리를 이해한다.
- 딥러닝에 자주 등장하는 단어들과 익숙해진다.



01 자율주행

02 인공지능

03 머신러닝

04 모델

05 Loss

06 Optimizer

07 Overfitting

08 하이퍼파라미터



인공지능과 자율주행

01. 자율주행

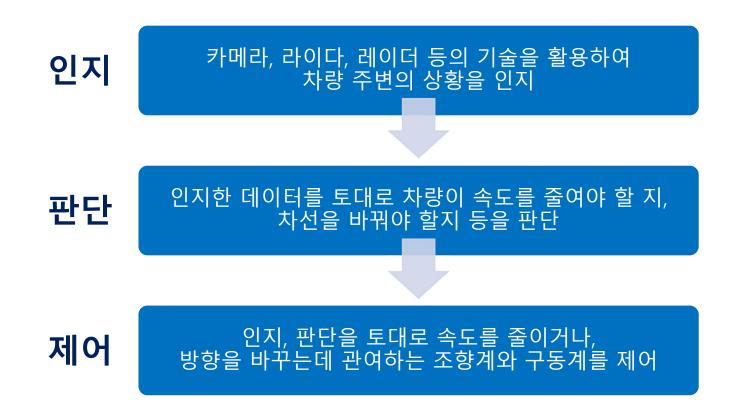
: 자율주행이란



차량을 운전하는 사람의 개입없이 주변환경을 <mark>인식하고,</mark> 상황을 종합하여 <mark>판단하며</mark> 차량을 <mark>제어하면서</mark> 스스로 최종목적지까지 <u>주행</u>하는 차

01. 자율주행

: 자율주행은 어떻게 작동하는가?



* 이외의 기술 : 측위 (차량이 도로위에 어느 위치에 있는지 계산하는 기술, 흔히 GPS)

01. 자율주행

: 자율주행 자동화단계

< 운전 자동화의 단계적 구분 >

	Level 0	Level 1	Level 2	Level 3	Level 4	Level 5
레벨 구분						
	운전자 보조 기능			자율주행 기능		
명칭	無 자율주행 (No Automation)	운전자 지원 (Driver Assistance)	부분 자동화 (Partial Automation)	조건부 자동화 (Conditional Automation)	고도 자동화 (High Automation)	완전 자동화 (Full Automation)
자동화 항목	없음(경고 등)	조향 or 속도	조향 & 속도	조향 & 속도	조향 & 속도	조향 & 속도
운전주시	항시 필수	항시 필수	항시 필수 (조향핸들 상시 잡고 있어야함)	시스템 요청시 (조향핸들 잡을 필요X, 제어권 전환 시만 잡을 필요)	작동구간 내 불필요 (제어권 전환X)	전 구간 불필요
자동화 구간	-	특정구간	특정구간	특정구간	특정구간	전 구간
시장 현황	대부분 완성차 양산	대부분 완성차 양산	7~8개 완성차 양산	1~2개 완성차 양산	3~4개 벤처 생산	없음
예시	사각지대 경고	차선유지 또는 크루즈 기능	차선유지 및 크루즈 기능	혼잡구간 주행지원 시스템	지역(Local) 무인택시	운전자 없는 완전자율주행

참조: https://www.korea.kr/news/reporterView.do?newsId=148885852



인공지능과 자율주행

02. 인공지능

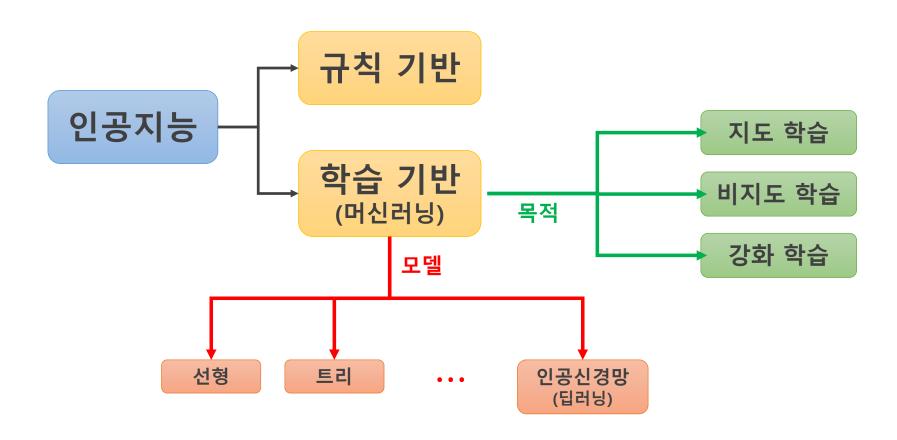
: 인공지능 정의

인공지능은 인텔리전트한 기계를 만드는 과학과 공학이며, 특히 인텔리전트한 컴퓨터 프로그램

- 존 매카시 -

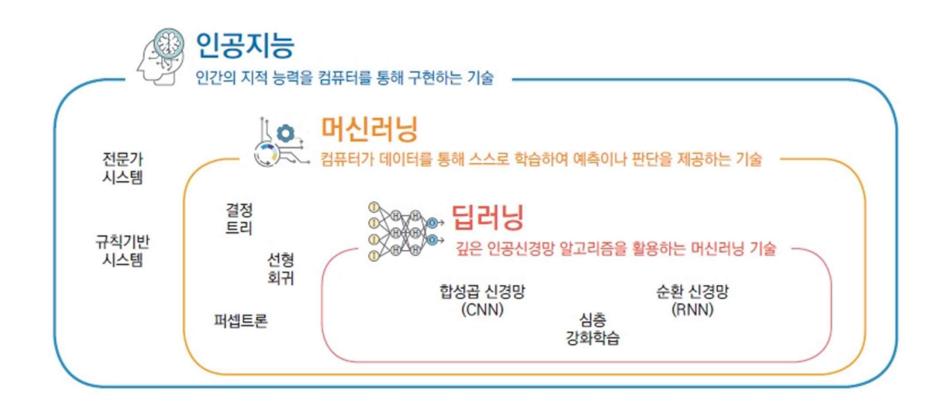
02. 인공지능

: 인공지능의 종류



02. 인공지능

: 인공지능의 종류





- 1 머신러닝 맛보기
- 2 머신러닝 정의
- ③ 머신러닝 종류

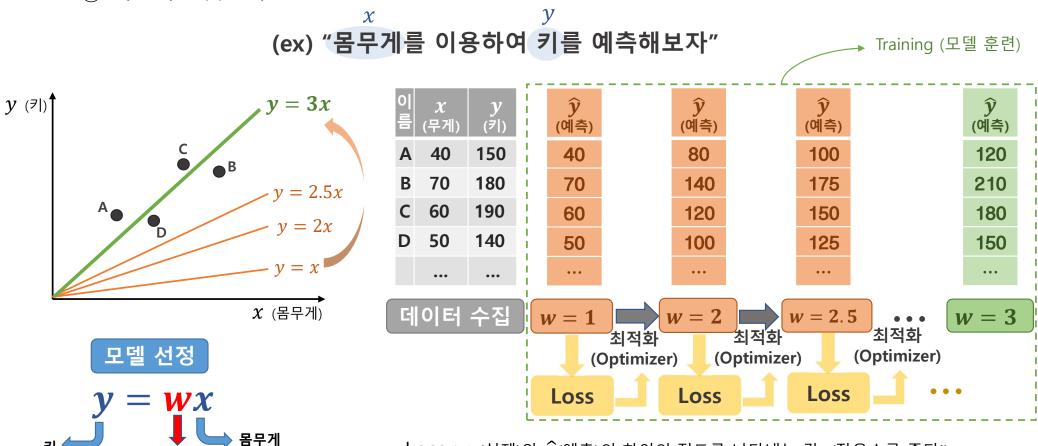
인공지능과 자율주행

: ① 머신러닝 맛보기

<u>파라미터</u>

(output)

(input)



- Loss : y(실제)와 \hat{y} (예측)의 차이의 정도를 나타내는 값 (작을수록 좋다!)
- Optimizer : Loss가 줄어들도록 가중치를 수정하는 과정

: ① 머신러닝 맛보기

"앞에서 배운 핵심 키워드를 정리합시다."

모델

데이터의 구조를 정의하는 식

Loss

실제 값과 예측 값의 차이의 정도 (작을수록 좋다!) 파라미터

모델의 패턴을 결정하는 변수

Optimizer

Loss 가 작아지도록 파라미터를 수정하는 것

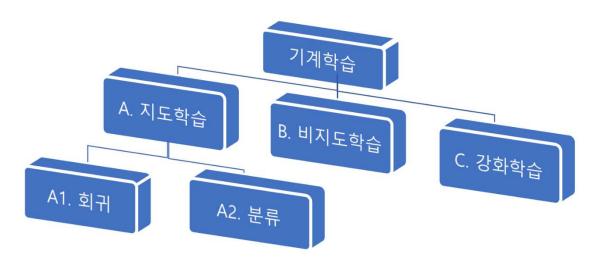
: ② 머신러닝 정의

"기계가 학습한다는 것은 무엇을 의미할까?"

모델의 좋은 파라미터를 찾는 과정

Loss 가 작아지도록 모델의 파라미터를 Optimizer 하는 과정

: ③ 머신러닝 종류



- **A. 지도학습** y 를 예측하는 것에 관심이 있다.
 - **회귀** (Regression) : *y* 값이 연속형 이다. (ex) 키, 몸무게, ...
 - **분류 (Classification)**: *y* 값이 범주형이다. (ex) 고양이 vs 강아지, 시험 등급,
- **B. 비지도학습**》 y 가 없다. X 간의 관계에 관심이 있다. (ex) 군집, 연관규칙,
- c. 강화학습》보상의 합이 최대가 되도록 하는 것에 관심이 있다.

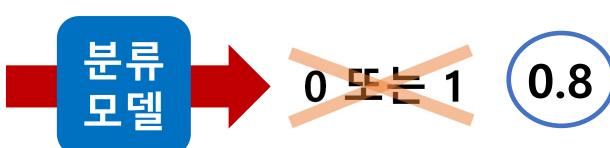
오개념 바로잡기

분류 모델의 Output 값은 Label 이 아니라 확률로 내뱉는다.

<Ex> 강아지 (Label: 0) 와 고양이 (Label: 1) 을 구분하는 분류 모델

input Output





즉, Label 1 (고양이) 가 나올 확률 0.8



1 선형 모델

- 2 로지스틱 모델
- ③ 의사결정나무모델

- 4 인공신경망 모델
- 5 그 외

인공지능과 자율주행

: ① 선형 모델 (Linear)

A. 정의

$$\hat{y} = w_0 + w_1 X_1 + w_2 X_2 + \cdots + w_n X_n$$

위와 같이 선형결합을 이용한 모델을 선형모델이라 한다.

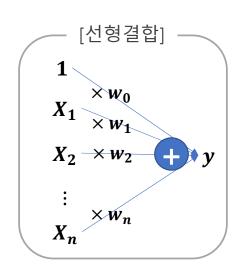
- $-\hat{y}$: output (예측 값) $-X_i$: input (특성 값)
- w_j : 파라미터 또는 가중치 (특히, w_0 를 편향이라고 함)

B. 특징

- 주로 회귀에서 사용한다. 따라서 선형회귀모델 이라고도 부른다.

C. 예제

선형 모델 $\hat{y} = 3 + 2X_1 + 3X_2 - X_3$ 에 대하여 $X_1 = 1, X_2 = 2, X_3 = 3$ 을 입력했을 때, 이 모델의 예측 값은?



: ② 로지스틱 모델 (Logistic)

A. 정의

$$\widehat{y} = \sigma(w_0 + w_1 X_1 + w_2 X_2 + \cdots + w_n X_n)$$

꼴의 모델을 로지스틱 모델이라 한다.

- $-\sigma(t) = \frac{1}{1+exp(-t)}$ 를 시그모이드(Sigmoid) 함수라 한다. 이때 t를 로짓 또는 로그-오즈라 한다.
- ŷ : Output (예측 값) Xi : Input (특성 값)
- w_i : 파라미터 또는 가중치 (특히, w_0 를 편향이라고 함)

B. 특징

- 로지스틱 모델의 \hat{y} 의 값은 항상 0과 1 사이다. ($:: \sigma$ 의 치역이 0과 1 사이)
- 주로 분류에서 사용한다. 여기서 \hat{y} 의 값은 특정 범주에 속할 확률을 의미한다.

C. 예제

모델 M 은 사진(Input)을 입력하면 고양이 (범주 0) 또는 강아지 (범주 1) 를 예측(Output)하는 로지스틱 모델이다. 모델 M에 어떤 사진을 입력하였더니 Output 값이 0.8 이 나왔다. 이것이 의미하는 것은?

: ③ 의사결정나무 (Decision Tree)

A. 정의

우측의 그림처럼 가지치기 형태의 모델을 의사결정나무 모델이라 한다.

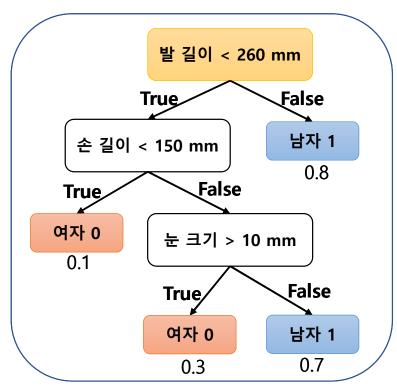
- 각각의 사각형 박스를 노드(node)라 한다. 최상위 박스 (노란색) 을 루트 노드(Root node)라 한다. 말단 박스 (파랑, 빨강) 을 리프 노드(Leaf node)라 한다.
- 리프 노드 하단의 숫자는 남자(1) 범주에 속할 확률이다.

B. 특징

- 주로 분류에서 사용한다. (회귀에서도 사용은 가능하다)
- 분류에서 예측 값(Output)은 범주에 속할 확률로 나온다.

C. 예제

0.3 X_1 , X_2 , X_3 를 각각 발 길이, 손 길이, 눈 크기라 하자. 우측 위 모델에서 X_1 = 250, X_2 = 100 , X_3 = 15 를 입력했을 때 Output 을 구하고 이것이 의미하는 것은 무엇인지 말하시오.

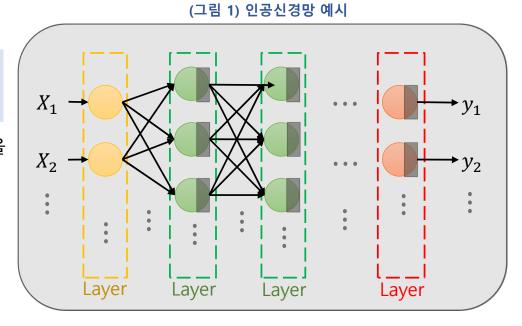


: ④ 인공신경망 (Artificial Neural Network, ANN)

A. 정의

오른쪽 (그림 1)처럼 사람의 신경망 구조를 모방하여 만든 모델을 인공신경망 모델이라 한다.

- 인공신경망 모델은 기본적으로 여러 개의 Layer를 쌓아 올 린 구조로 되어있다.
 - * 점선 박스 = Layer (층)
 - * 노란색 점선 박스 = Input-layer (입력 층)
 - * 초록색 점선 박스 = Hidden-layer (은닉 층)
 - * 빨간색 점선 박스 = Ouput-layer (출력 층)
- Layer 내부는 여러 개의 Node 의 집합으로 구성된다.
 - * 원 모양 = Node
- 이전 Layer의 Node 값들은 선형결합하여 다음 Layer의 Node로 들어온다. 그리고 이 선형결합 된 값은 activation 을 통해 수정되어 Node의 최종 값이 된다.
 - * 화살표 = Node 선형결합의 가중치
 - * Node에 붙어있는 회색 박스 = Activation (활성 함수)
- 여러 개의 Layer로 구성 된 인공신경망을 **심층 신경망**(Deep Neural Network, **DNN**) 이라 한다. 또한 심층 신경망을 이용한 머신러닝을 **딥러닝**이라 한다.



▶ (그림 2) 활성함수 예시

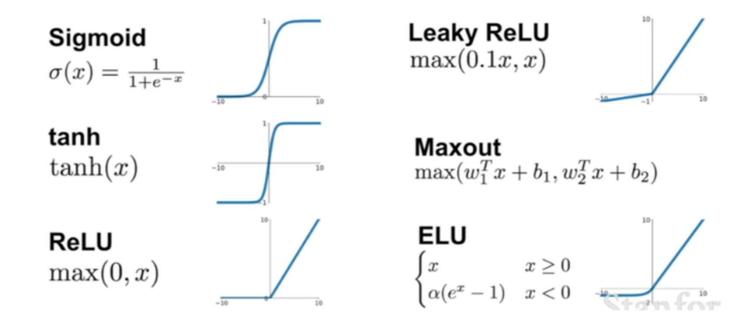
-3 -2

-1

: ④ 인공신경망 (Artificial Neural Network, ANN)

B. Activation

- Activation (활성함수)에 의하여 인공신경망은 복잡한 비선형 구조로 되어있다.
- 활성함수에는 아래와 같은 것들이 있다. (이 외에도 많다...)



: ④ 인공신경망 (Artificial Neural Network, ANN)

C. Layer

C. Layer		
- Layer 는 다양	한 종류가 있다. 아래 3가지는 꼭 기억해두자.	
Layer	설명	
Dense (밀집)	- 가장 일반적인 형태의 레이어	
Convolutional (합성곱)	- 이미지 분석에서 많이 사용 - 이것을 활용한 모델을 CNN 이라 한다.	0 15 10 10 10 1 1 1 1 1 1 1 1 1 1 1 1 1
Recurrent (순환)	 시계열 분석, 자연어 처리에서 자주 사용 이것을 활용한 모델을 RNN 이라 한다. RNN 변종으로 LSTM, GRU 가 있다. 	
		(Y_T) (y_{T-1}) (y_t) (y_{T-1})
RNN	LSTM GRU	$= \begin{array}{c ccccccccccccccccccccccccccccccccccc$
i ₂₋₁	$h_{t-1} = \begin{pmatrix} c_t & c_t \\ c_t & c_t \\ c_t & c_t \end{pmatrix}$	$\begin{array}{cccccccccccccccccccccccccccccccccccc$

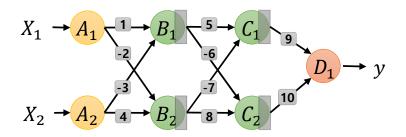
: ④ 인공신경망 (Artificial Neural Network, ANN)

D. 예제

우측의 '인공신경망 모델 구조'와 '활성함수'로 이루어진 모델 M이 있다. 모델 M에서 $X_1=10, X_2=20$ 을 Input 으로 입력했을 때 Output (y)을 구하려고 한다. 다음 물음에 차례로 답하시오.

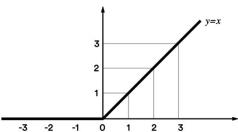
- C1. Input-layer 의 Node A_1 , A_2 의 출력값을 구하시오.
- C2. 첫번째 Hidden-layer 의 Node B_1 , B_2 의 출력값을 구하시오.
- C3. 두번째 Hidden-layer 의 Node C_1 , C_2 의 출력값을 구하시오.
- C4. Output-layer 의 Node D_1 의 출력값을 구하시오.
- C5. 모델 M의 Output (y) 를 구하시오.

[인공신경망 모델 구조]



[Activation (활성함수)]

$$f(x) = \begin{cases} x & if \ x > 0 \\ 0 & otherwise \end{cases}$$



: ⑤ 그 외 모델들

SVM

배깅

랜덤 포래스트

나이브 베이즈

K-평균

DBSCAN

•••••

: ⑤ 그 외 모델들



1996년에 발표한 유명한 논문에서 데이비드 월퍼트는 데이터에 관해 완벽하게 어떤 가정도 하지 않으면 한 모델을 다른 모델보다 선호할 근거가 없음을 보였다.

경험하기 전에 더 잘 맞을 거라고 보장할 수 있는 모델은 없다 어떤 모델이 최선인지 확실히 아는 유일한 방법은 모든 모델을 평가해보는 것 뿐





1 회귀 손실함수



Review

- Loss 는 실제 값과 예측한 값의 차이의 정도를 나타내는 값이다.
- 즉, Loss 가 작을 수록 좋은 모델이라고 할 수 있다.
- 모델을 학습시키는 과정은 Loss 가 최소가 되도록 최적화하는 과정이다.

인공지능과 자율주행

: ① 회귀 손실함수

A. 회귀 손실함수 종류

회귀에서 많이 사용하는 손실함수는 다음과 같다.

Loss Name	Definition	
MSE (Mean Square Error)	$mse = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$	
MAE (Mean Absolute Error)	$mae = \frac{1}{N} \sum_{i=1}^{N} y_i - \hat{y}_i $	
Huber	MSE 와 MAE 를 조합한 형태 (식 생략)	
$y_i =$	실제 값, \hat{y}_i = 예측 값, N = 셈플 개수	

- MSE 는 제곱을 이용하여 계산하기 때문에 이상치에 민감하다.
- 일반적으로 MSE 를 많이 사용하지만, 이상치가 많을 경우 MAE 또는 Huber를 사용하는 것이 좋다.

: ① 회귀 손실함수

B. 예제

B1. 오른쪽 표를 보고 MSE 와 MAE 를 각각 계산해 보시오.

y (실제)	ŷ (예측)
2	1
5	7
8	5
10	11

B2. 키를 예측하는 모델 M1, M2 를 손실함수 MSE 를 이용하여 훈련시켰다. M1의 Loss는 9이고 M2의 Loss는 16이다. 어느 모델이 더 좋다고 할 수 있는가?

B3. 키를 예측하는 모델 M1, M2 를 각각 손실함수 MSE, MAE 를 이용하여 훈련시켰다. M1의 Loss는 100이고 M2의 Loss는 36이다. 어느 모델이 더 좋다고 할 수 있는가?

: ② 분류 손실함수

A. 분류 손실함수 종류

분류에서 손실함수로 주로 Cross-entropy 를 이용한다. Cross-entropy 는 다음과 같이 두 종류가 있다.

Loss Name	Definition
Binary-cross-entropy	이진 분류에서 사용하는 손실함수 (식은 생략)
Categorical-cross -entropy	다중 분류에서 사용하는 손실함수 (식은 생략)

- Cross-entropy 값의 범위는 항상 0 이상이며 0 에 가까울 수록 분류를 잘 하는 모델이다.

이진 분류 : 범주가 2개인 분류이다. 출력 값의 개수는 1개이며 범위는 0~1 이다.

- (ex 1) 강아지 0 vs 고양이 1 → 출력 값 = 0.2 → 해석 : 고양이 일 확률은 0.2 (즉, 강아지로 예측)
- (ex 2) 실패 0 vs 성공 1 → 출력 값 = 0.7 → 해석 : 성공할 확률은 0.7 (즉, 성공할 것이라고 예측)

<u>다중 분류</u>: 범주가 3개 이상인 분류이다. 출력 값의 개수는 범주의 개수와 동일하며 합은 항상 1이다.

- (ex) 하수 0 vs 중수 1 vs 고수 2
 - → 출력 값 = [0.1, 0.6, 0.3] → 해석 : 하수, 중수, 고수일 확률은 각각 0.1, 0.6, 0.3 (즉, 중수일 것이라고 예측)

: ② 분류 손실함수

B. 예제

B1. 강아지(0)와 고양이(1)를 분류하는 두 모델 M1, M2의 Binary-cross-entropy 손실값이 각각 0.5, 10 이 나왔다. 어느 모델이 더 좋은 모델이라고 할 수 있는가?

B2. 분류 모델 M 의 출력 값이 0.3이 나왔다. 이 모델은 이진 분류인지 다중 분류인지 답하시오. 또한 출력 값이 의미하는 것을 서술하시오.

B3. 분류 모델 M 의 출력 값이 [0.1, 0.2, 0.7]이 나왔다. 이 모델은 이진 분류인지 다중 분류인지 답 하시오. 또한 출력 값이 의미하는 것을 서술하시오.



1 Gradient 와 Optimizer

- **2** Optimizer 종류
- 3 Learning rate (학습률) 이해

Review

Optimizer는 Loss가 줄어들도록 파라미터(가중치)를 수정하는 것

인공지능과 자율주행

06. Optimizer (최적화, 최적화 함수)

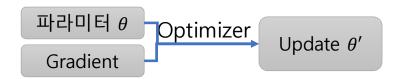
: ① Gradient (기울기)

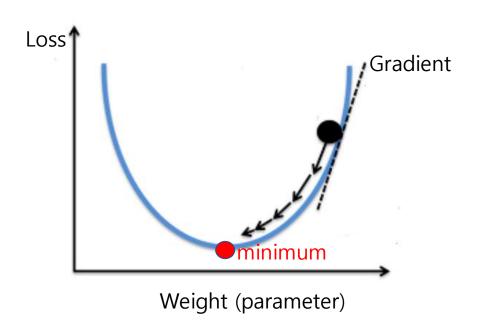
A. Gradient 의 정의

- Gradient란 목적함수 (Loss)를 파라미터 θ 로 미분한 값이다. 즉, Gradient := ∇_{θ} 목적함수
- Gradient에는 Loss 가 줄어들기 위해서 파라미터 θ 가 어떻게 수정되어야 하는지에 대한 정보가 담겨있다.

B. Optimizer 의 작동 원리

- Optimizer는 Gradient 값을 기반으로 파라미터를 수정한다.
- 구체적으로 Optimizer의 구조는 아래와 같다.





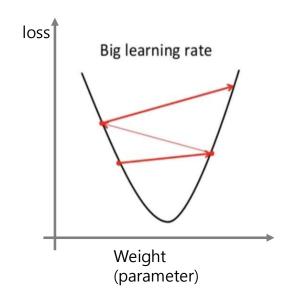
06. Optimizer (최적화, 최적화 함수)

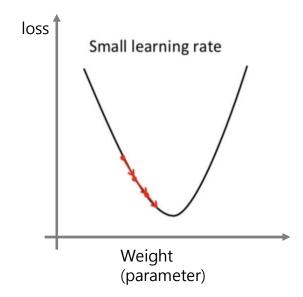
: ② Learning rate

- Learning rate 는 Optimizer 를 생성할 때 지정하는 옵션값이다. Optimizer 가 파라미터를 수 정할 때 Gradient를 얼마나 반영할지에 대한 비율을 의미한다. (범위는 0~1)
- 예를 들어 Learning rate 를 α 라 하면, 아래와 같은 방식으로 Optimizer 작업이 일어난다.

$$\theta \leftarrow \theta - \alpha \times \nabla Loss$$

- Learning rate 가 너무 크면 최적 값에 수렴하지 못하고 맴돌게 된다. 반면에, 너무 작으면 최 적 값에 수렴하기 까지 시간이 오래 걸린다.





06. Optimizer (최적화, 최적화 함수)

: ③ optimizer 종류

optimizer	수렴 속도	수렴 품질
SGD (Stochastic Gradient Descent)	*	***
SGD + Momentum	**	***
SGD + Momentum + Nesterov	***	***
Adagrad	***	*
RMSprop	***	***
Adam	***	***
Nadam	***	***
AdaMax	***	***



- 1 Overfiting 이란
- 2 테스트와 검증
- ③ 규제, 조기종료

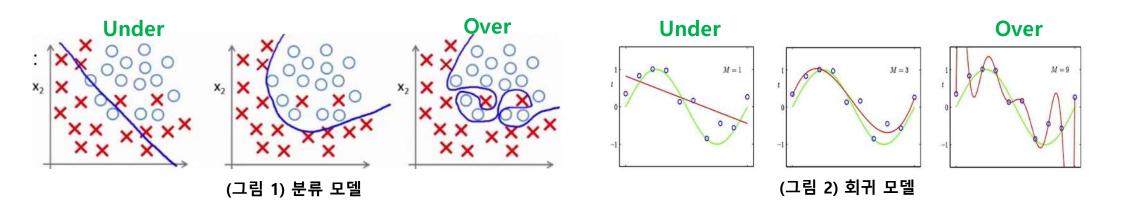
인공지능과 자율주행

07. Overfitting (과적합, 과대적합)

: ① Overfitting 이란

Overfitting 은 분석 모델이 주어진 데이터에 지나치게 적합 되어 새로운 데이터에서는 모델의 성능이 저하되는 현상을 의미한다.

- Overfitting과는 반대로 적합이 너무 덜 된 것을 Underfitting (과소적합) 이라고 한다.
- 특히 딥러닝과 같은 복잡한 모델에서 Overfitting 문제가 자주 발생한다.



07. Overfitting (과적합, 과대적합)

- : ② 테스트와 검증
- 모델의 Overfitting 여부를 확인하기 위해서는 훈련에 사용하지 않은 데이터가 필요하다.
- 따라서 전체 데이터를 훈련에 사용하지 않고 <u>훈련용 데이터</u>와 Overfitting을 확인 할 데이터로 나눈다.

A. 두 개의 세트로 나누는 방법 - 두 개의 세트로 나누어 ①로 훈련하고 ②로 평가한다. - ① 을 Training Set (훈련 세트)라 하고 ② 를 Test Set (테스트 세트) 라 한다. - Training Set 과 Test Set 의 비율은 일반적으로 7:3 또는 8:2를 사용한다. Training Set 과 Test Set 의 비율은 일반적으로 7:3 또는 8:2를 사용한다.

B. 세 개의 세트로 나누는 방법

- 세 개의 세트로 나누어 ①로 훈련하면서 ②로 모니터링하고 훈련이 끝나면 ③으로 최종적으로 평가한다.
- ① 을 Training Set (훈련 세트)라 하고 ② 를 Validation Set (검증 세트),
 - ③ 을 Test Set (테스트 세트) 라 한다.
- 이러한 방식을 **홀드아웃 검증** (Holdout Validation) 이라 한다.



07. Overfitting (과적합, 과대적합)

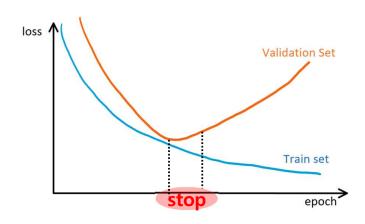
- : ③ 규제, 조기종료
- Overfitting 을 막는 기법으로 규제와 조기종료가 있다.

A. 규제 (Regularization)

- : 모델이 Overfitting 되지 않도록 사전에 방지하는 방법
 - (ex 1) 선형 모델에서 가중치가 커지면 패널티를 부여하여 L:oss 값이 커지도록 한다.
 - (ex 2) 트리 모델에서 깊이에 제한을 두어 복잡한 모델을 만들지 못하게 한다.

B. 조기종료 (Early Stopping)

- : 모델이 Overfitting 되기 시작하면 훈련을 종료하는 방법
 - (ex) Validation set 의 Loss 값이 커지기 시작하면 훈련을 종료한다.
 - 시간이 경과함에 따라 Train set 과 Validation set 의 Loss 값은 오른쪽 그림과 같은 양상을 보이는 경우가 많다.
 - Train set 의 Loss 가 꾸준히 감소하는 이유는
 Train set 의 데이터를 기반으로 훈련하고 있기 때문이다.
 - 하지만 Validation set의 Loss 는 어느 순간 증가하기 시작한다. 그리고 그 시점이 Overfitting 되는 시점이므로 훈련을 중단한다.





인공지능과 자율주행

08. 하이퍼파라미터

- 모델을 훈련을 하기 위해 미리 지정해야 하는 요소들을 **하이퍼파라미터**라 한다
- 파라미터는 모델이 학습을 통해 스스로 찾아가는 반면에
 하이퍼파라미터는 학습 전에 미리 지정해야 한다는 점에서 차이가 있다.

하이퍼파라미터	요소
Loss	MSE, MAE, Huber
Optimizer	SGD, Adam,
Activation	ELU, Sigmoid,
Layer 및 개수	Dense, CNN, RNN,
Node 개수	Layer 층의 Node 개수
Learning rate	0 ~ 1
규제	L1, L2,
Epoch	
Batch Size	8, 16, 32, 64,

08. 하이퍼파라미터

Note

Epoch (에포크) , Batch Size (배치크기)

- 일반적으로 모델이 훈련을 할 때, 학습 데이터를 한번에 학습하는 것이 아니라 학습 데이터를 분할하여 나누어 학습한다.
- 이렇게 분할 된 데이터 하나를 mini Batch 라 한다.
- 하나의 mini Batch 에 포함된 데이터의 개수를 Batch Size 라 한다.
- 분할 된 mini Batch를 모두 한 번씩 훈련에 사용하는 것을 1 Epoch 라 한다.



1 Epoch