



23.6.21-AI7기

지난시간 복습

코드 점검

코드 학습단계

softmax함수 - 다중분류

손실함수1 (loss function)

회귀(regression)

분류

과대적합, 과소적합

relu : hidden layer

시그모이드 이진분류

소프트맥스 다중분류

렐루 은닉층

손실함수2

경사하강법 learning rate : lr(0.01 default) (0.001~ 0.1)

하이퍼 파라미터

optimizer

sgd (경사하강법 다음으로 등장,, global minimum을 찾기 용이)

adam- 하이퍼파라미터 x

optimizers comparison (hyper parameter)

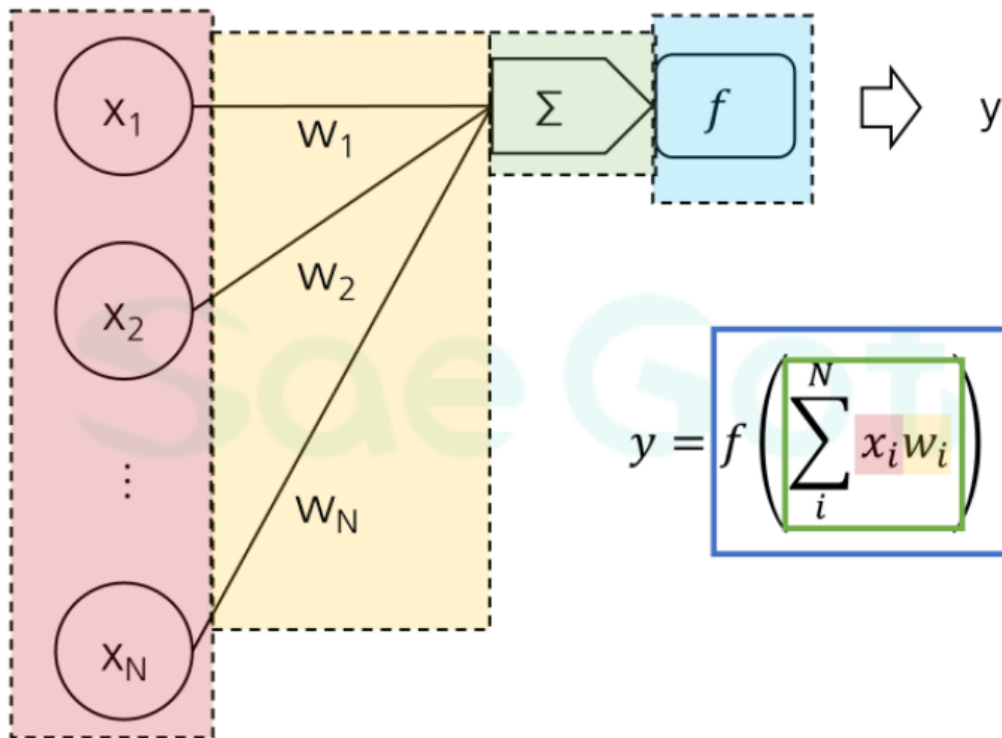
adam vs SGD

sigmoid 역전파 순전파

class

변의,,

지난시간 복습



퍼셉트론 계산식

1. MLP(multi layer perceptron)은 어떤 논리회로 문제를 해결했는가? (인공지능 겨울 참조)

= XOR

1. 퍼셉트론 그림에서 동그라미(x_n)을 뭐라고 부르는가?

= 노드

1. 퍼셉트론의 가장 마지막에 weighted sum 값을 사용하는 함수는 무엇인가?

= 활성화함수

1. ○

○ □ ⇒ 퍼셉트론이라면, 첫번째 □은 어떤수식으로 이뤄졌는가? (x, w 를 이용하라)

○ □

$$x_1 * w_1 + x_2 * w_2 + x_3 * w_3 = \square$$

1. sigmoid 대신 ReLU를 사용하는 이유는?

= 기울기소실 문제를 해결

(30분까지~)

0 ~255 ⇒ 255 =흰색

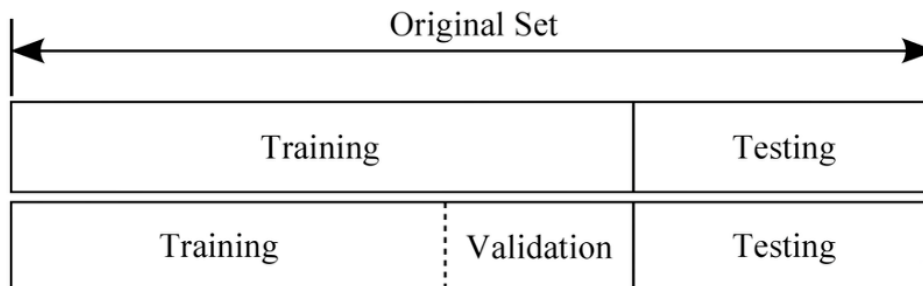
코드 점검

- MNIST - 손글씨 이미지
- 0~255 사이의 픽셀로 이뤄짐. 255 흰색/ 0 검은색

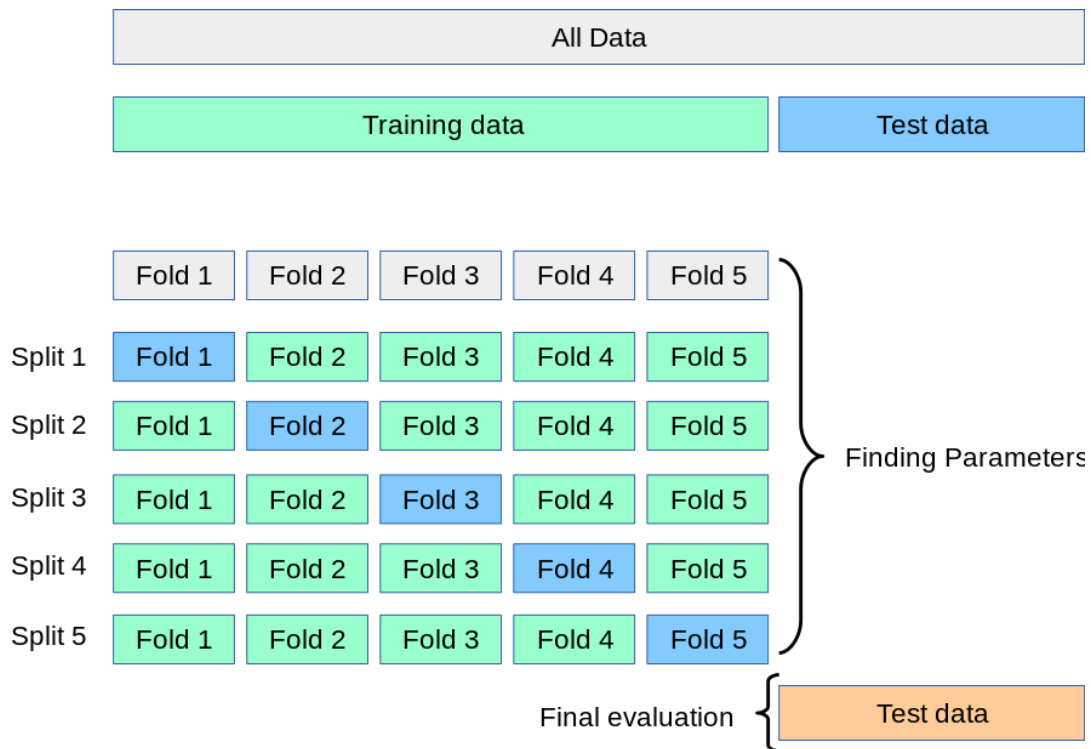
[illegible]

코드 학습단계

1. train /test 데이터를 동일한 방법으로 preprocess(전처리) 한다.
2. train : test = 8:2 로 split 한다. (validation 설정) \Rightarrow OVERFITTING(과적합 방지) = 일반화



- crossValidation



3. 학습에 사용할 모델을 가져온다. (가져오거나, 생성)

4-1. 생성

1) `tf.keras.models.Sequential()` 부터 layer를 쌓는다.

2) 중간 layer들을 쌓고, 마지막 layer에는 이진분류/다중분류 확인하고 옵션 작성.

ex) 다중분류 : `tf.keras.layers.Dense(클래스갯수, activation='softmax')`

3) compile 내용을 작성. 주로 `optimizer= adam`, `loss`는 이진분류/다중분류 따라 다르게, `metrics=['accuracy']`

이진분류일때,

- `activation = sigmoid`
- `loss function= binary_crossentropy`

```
tf.keras.layers.Dense(2, activation='sigmoid')

model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
```

다중분류일때,

- `softmax`
- `categorical_crossentropy`

```
tf.keras.layers.Dense(10, activation='softmax')
model.compile(loss='categorical_crossentropy',
```

```
optimizer='adam',
metrics=['accuracy'])
```

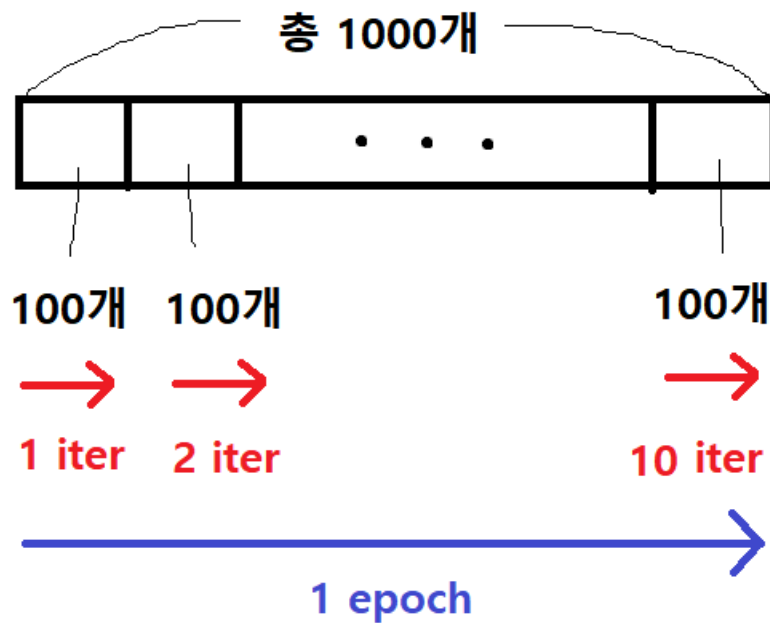
```
# sparse 한 경우, 이미지 000001000 경우
tf.keras.layers.Dense(10, activation='softmax')
model.compile(loss='sparse_categorical_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
```

4) history = model.fit (train) ⇒ train accuracy, val_acc, loss, val_loss 트랙킹

5) model.evaluate(test) : loss

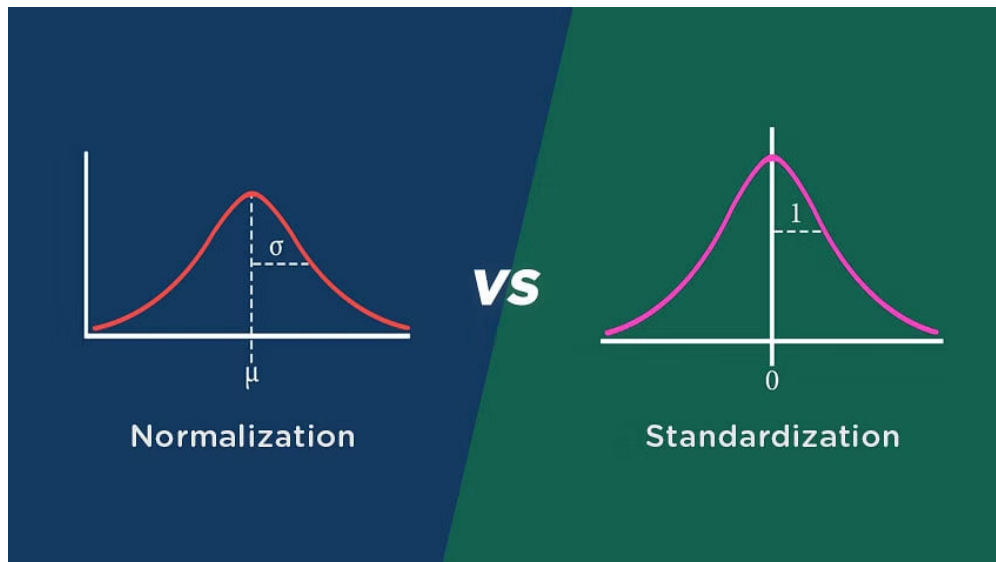
6) model.predict(test) : 특정라벨 반환, mnist: 3

- epoch: train 데이터 전체 학습 횟수
- batch사이즈: 100개 / 미니배치는 10개
- iteration : 미니배치 수와 동일, 하나의 배치 학습횟수



- 이미지데이터/문자열 데이터에서 추가되는 부분

1. 이미지데이터의 픽셀(0~255) 이므로 /255.0 취해주면 정규화(normalize) 된다. ⇒ 0~1



- 문자열 칼럼인 경우
- 1) 라벨인코더 (labelEncoder) \Rightarrow 0,1 인 경우 주로 사용 (남,녀)
- 2)원핫인코딩(get_dummies) (sparse 하다) \Rightarrow 3개 이상인 경우, 랭킹/연속성을 없애기 위해 \Rightarrow 칼럼 갯 수는 늘어나지만 컴퓨터에게는 데이터가 단순해져서 연산속도가 빠름(sparse)

ID	과일
1	사과
2	바나나
3	체리

One-Hot Encoding

ID	사과	바나나	체리
1	1	0	0
2	0	1	0
3	0	0	1

LabelEncoder

ID	과일
1	0
2	1
3	2

- 평가지표: 아주 다양하다. 혼동행렬이 기본
- **confusion matrix : 혼동행렬**

실제 0 (음성 클래스)	TN (True Negative) 0을 0으로 잘 예측한 경우	FP (False Positive) 0을 1로 잘못 예측한 경우
실제 1 (양성 클래스)	FN (False Negative) 1을 0으로 잘못 예측한 경우	TP (True Positive) 1을 1로 잘 예측한 경우
코딩재개발	예측 0 (음성 예측)	예측 1 (양성 예측)



- 평가지표는, 공모전 시에는 주최측에서 정해주지만,
- 프로젝트시에는 수식을 보고 어떤 수식을 **평가지표**로 삼을지 고려해야함.

```
def train_model():
    # TODO : MNIST를 읽어옵니다.

    (x_train, y_train), (x_test, y_test) = load_mnist(flatten= False, normalize = True)

    x_train, x_test = x_train.reshape(-1, 28, 28), x_test.reshape(-1, 28, 28)

    # TODO : 입력, 은닉, 출력을 784, 50, 10으로 합니다.
    model = tf.keras.models.Sequential([
        # (28, 28)을 (784,)로 변환 후 입력
        tf.keras.layers.Flatten(input_shape=(28, 28)),
        tf.keras.layers.Dense(50, activation='relu'),
        # 출력 10노드, softmax 사용
        tf.keras.layers.Dense(10, activation='softmax')
    ])

    # TODO : 모델을 컴파일 합니다. 컴파일시 손실 함수를 `sparse_categorical_crossentropy`로 합니다.
    model.compile(optimizer='adam',
                  # 손실 함수로 sparse_categorical_crossentropy 사용
                  loss='sparse_categorical_crossentropy',
                  metrics=['accuracy'])

    # TODO : 모델을 훈련합니다.

    history = model.fit(x_train, y_train, epochs=5)
```

```

# 모델의 손실값, 정확도를 측정합니다.
model.evaluate(x_test, y_test, verbose=2)

return history.history

def main():

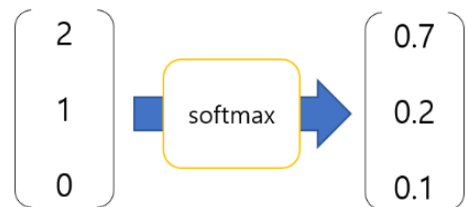
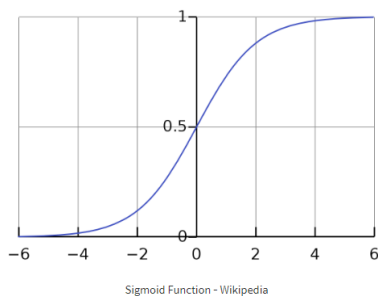
    history = train_model()

    print('loss :\n', history['loss'])
    print('accuracy :\n', history['accuracy'])

if __name__ == "__main__":
    main()

```

softmax함수 - 다중분류



Sigmoid 함수

- 모든 입력값에 대해 출력값이 실숫값으로 정의 (=Soft Decision)
- 값이 작아질수록 0, 커질수록 1에 수렴
- 출력이 0~1 사이로 확률 표현 가능 (=Binary Classification)
- 기울기소실 문제 존재

Softmax 함수

- 출력값이 N개
- 입력값을 각각 지수함수로 취하고, 이를 정규화 (=총합을 1로 만듦)
- 정규화로 인해 각 출력값은 0~1값을 가짐
- **모든 출력값의 합은 반드시 1**
- N가지 중 한 가지에 속할 확률 표현 가능 (=Multi-class Classification)

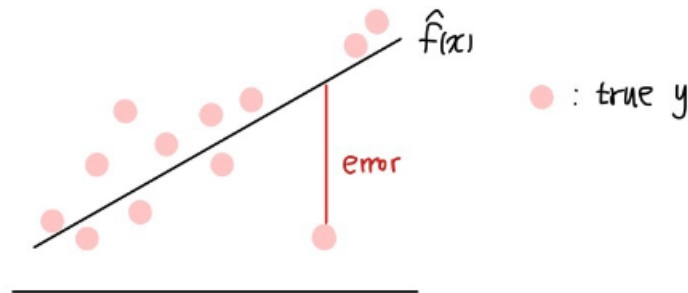
퀴즈

softmax 활성화 함수를 사용한 모델의 값이 0.3, 0.1, 0.26, x 일때, x의 값은 무엇인가?
 $= 1 - (0.3 + 0.1 + 0.26) = 1 - 0.66 = 0.34$

손실함수1 (loss function)

회귀(regression)

- 이상치 (outlier) : 이상한 데이터



모델링의 목적	목표 변수 유형	관련 모델	평가 방법
예측 / 회귀 (Prediction)	연속형	선형 회귀	MSE, RMSE, MAE, MAPE 등
분류 (Classification)	범주형	<ul style="list-style-type: none"> - 로지스틱 회귀 - 의사결정나무 - 서포트벡터머신 	정확도, 정밀도, 재현율, F1 -score

- 평가지표는 여러버전이 있고, 데이터셋에 따라서 적합한 **평가지표가 바뀐다**.
- $|y - \hat{y}|$ 값 = mae , $|y - \hat{y}|^2$ = mse , rmse

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$$

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

- 손실함수의 함수값이 **최소화** 되도록 하는 가중치(weight)와 편향(bias)를 찾는 것이 목표 \Rightarrow **미분값**

- $(x-5)^2 + 6 \Rightarrow x=5$ 최솟값이군요?
- $2x \Rightarrow$ 극값 (최댓값, 최솟값)

분류

그레디언트와 경사하강법

밑바닥부터 시작하는 딥러닝을 교재로 한 딥러닝 I 강의 내용을 정리하고자 작성한 포스팅입니다. ▶ 모델 학습의 목표: 손실함수를 최소로 만드는 것이 목표로 정확도가 높은 모델을 만들기 위한 학습이 이뤄져야 한다. > 손실함수를 최소로 만드는 것의 의미 손실함수를 최소로 만든다

📌 <https://westshine-data-analysis.tistory.com/102>

Gradient
Descent

손실 함수의 종류

Problem type	Last-layer activation	Loss function	Example
Binary classification	sigmoid	binary_crossentropy	Dog vs cat, Sentiment analysis(pos/neg)
Multi-class, single-label classification	softmax	categorical_crossentropy	MNIST has 10 classes single label (one prediction is one digit)
Multi-class, multi-label classification	sigmoid	binary_crossentropy	News tags classification, one blog can have multiple tags
Regression to arbitrary values	None	mse	Predict house price(an integer/float point)
Regression to values between 0 and 1	sigmoid	mse or binary_crossentropy	Engine health assessment where 0 is broken, 1 is new

- cross entropy

Loss function	Usage	Examples	
		Using probabilities	Using logits
		<i>from_logits=False</i>	<i>from_logits=True</i>
BinaryCrossentropy	Binary classification	y_true: 1 y_pred: 0.69	y_true: 1 y_pred: 0.8
CategoricalCrossentropy	Multiclass classification	y_true: 0 0 1 y_pred: 0.30 0.15 0.55	y_true: 0 0 1 y_pred: 1.5 0.8 2.1
Sparse CategoricalCrossentropy	Multiclass classification	y_true: 2 y_pred: 0.30 0.15 0.55	y_true: 2 y_pred: 1.5 0.8 2.1

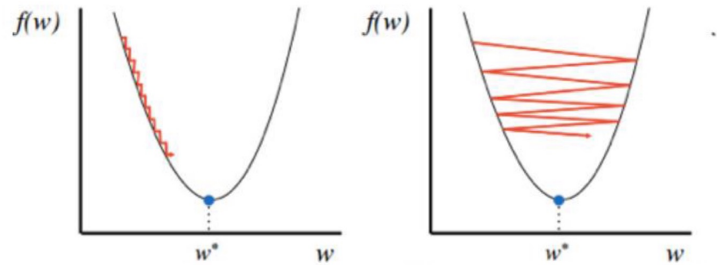
손실함수 간략 정리(예습용)

손실함수의 정의, Binary, Cross entropy, Categorical Cross entropy, MeanSquaredError

📌 https://velog.io/@yuns_u/손실함수-간략-정리

velog

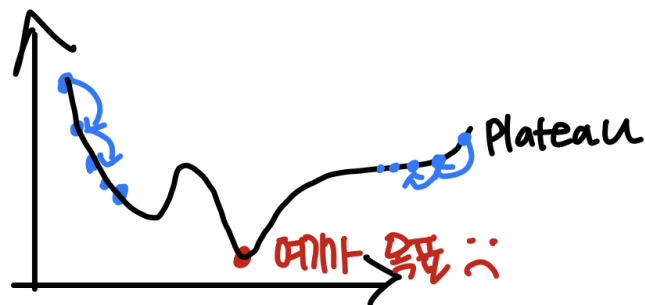
- 경사하강법 LR(learning rate) : 하이퍼파라미터 (사람이 정하는 인자)
- lr 이 작을 때 vs 클 때
- local minimum vs global minimum



최소점에 수렴하는데 너무 오랜 시간이 걸림 최소점을 찾지 못하거나 오히려 발산 될 수 있음

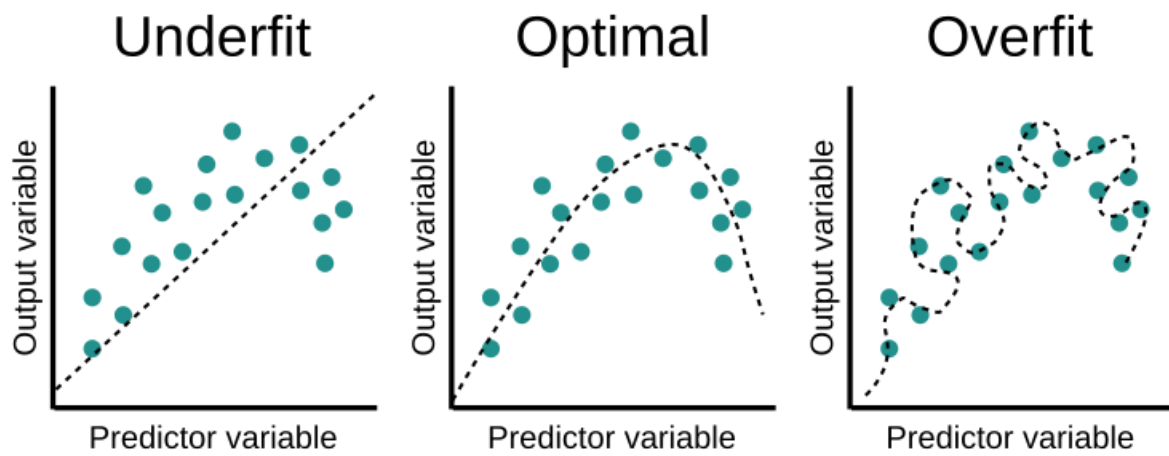
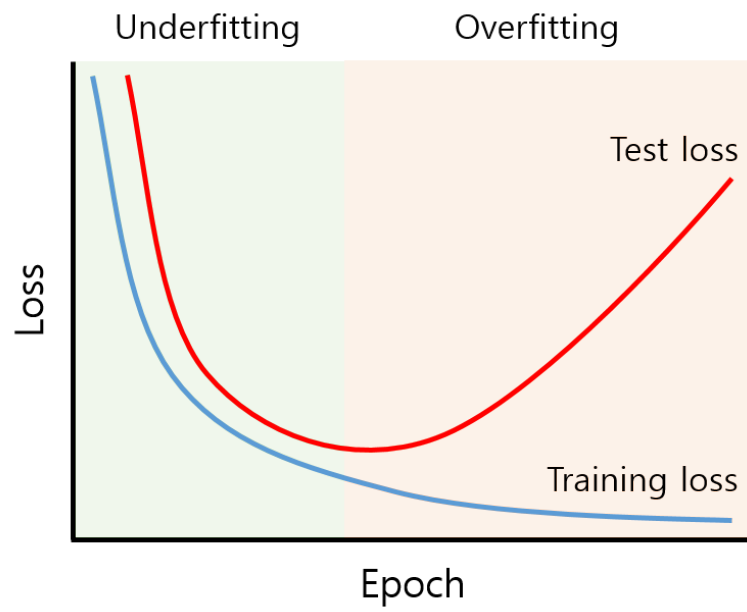
학습률이 작을 때.

학습률이 클 때.

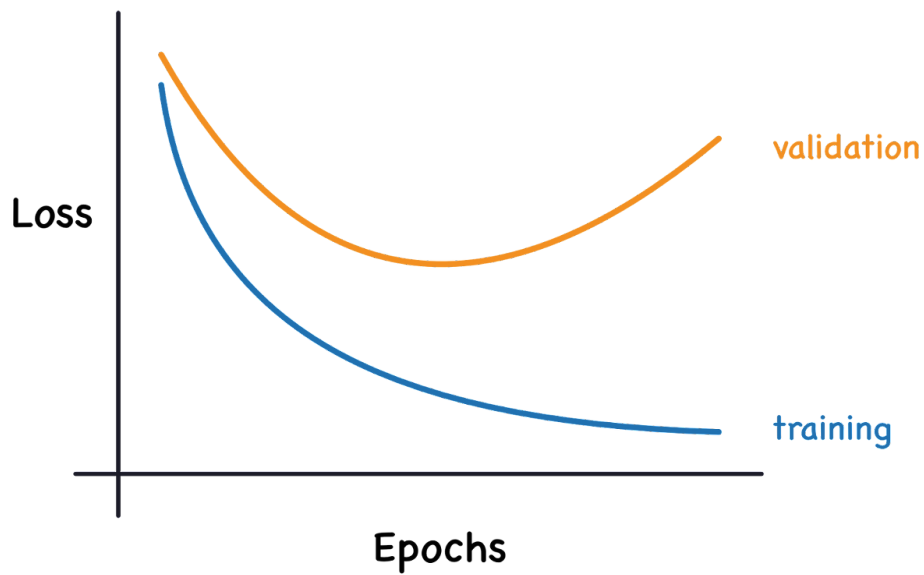


⇒ SGD 방법 등장...

과대적합, 과소적합

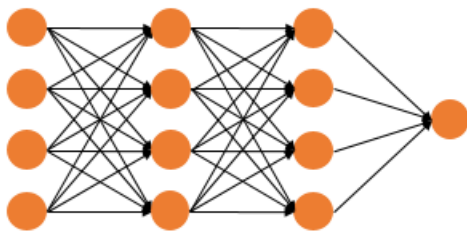


The Learning Curves

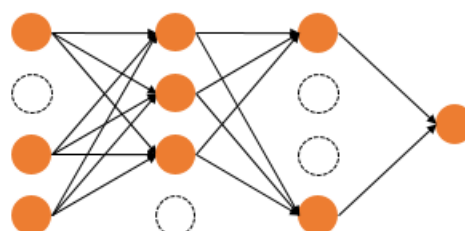


ex) validation_dataset : 유의미한 학습을 할때만 loss 감소 , 두 곡선 사이의 거리가 짧은부분이 잘 학습된 부분

ex) 드롭아웃 or Early stopping (얼리스타핑)



드롭아웃 적용 전



드롭아웃 적용 후

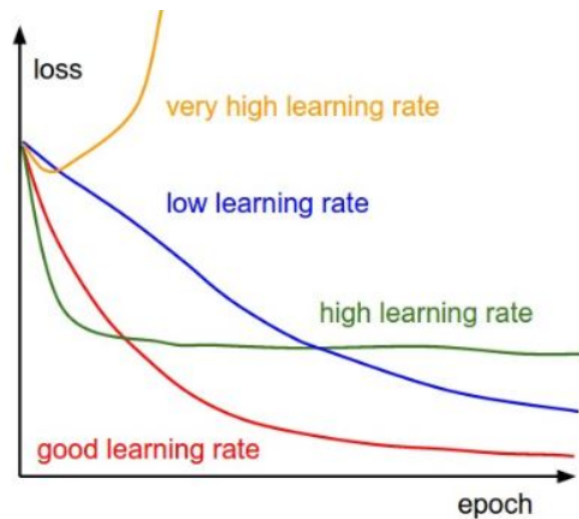
relu : hidden layer

- relu가 hidden layer에 쓰이면, 역전파 손실이 적다.
- sigmoid는 0~1 사이의 값을 전달하므로 0에 수렴하여, layer가 깊어질수록 정보손실
: $0.7 * 0.2 * 0.1 * 0.3 * 0.01 = 0$ 에 수렴
- relu는 0보다 큰 값은 자기자신을 가지므로, layer가 깊어지더라도 정보 손실이 없다.

시그모이드 이진분류
소프트맥스 다중분류
렐루 은닉층

손실함수2

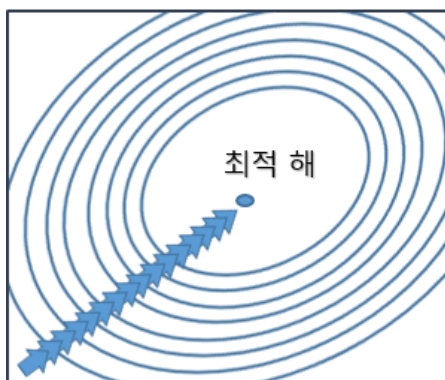
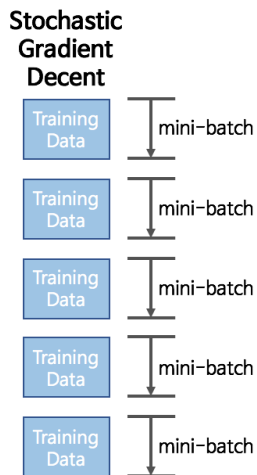
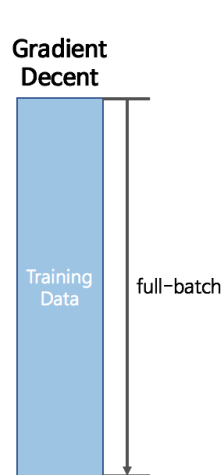
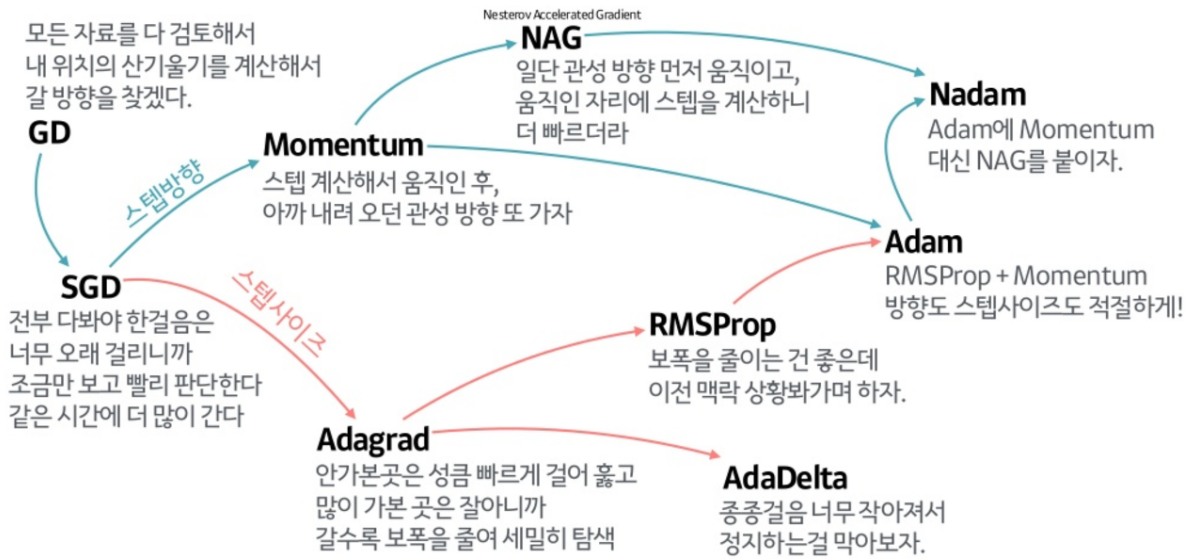
경사하강법 learning rate : lr(0.01 default) (0.001~ 0.1)



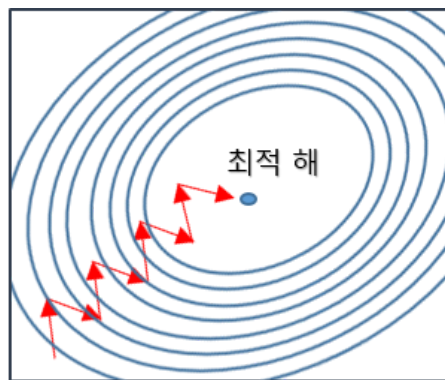
하이퍼 파라미터

- 인간이 조절하는 파라미터 (lr) → Grid Search (시간이 너~~무 오래걸림)
- 보통은 88프로? accuracy

optimizer



경사 하강법

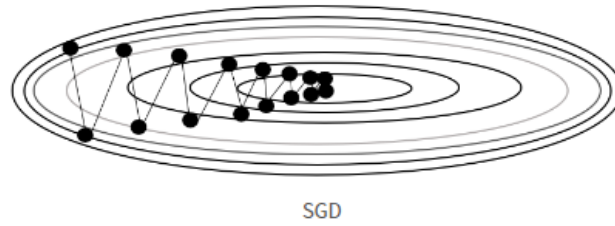


확률적 경사 하강법

- SGD는 한가지 데이터에 대해 그래디언트 계산, 훨씬 적은 연산량

sgd (경사하강법 다음으로 등장,, global minimum을 찾기 용이)

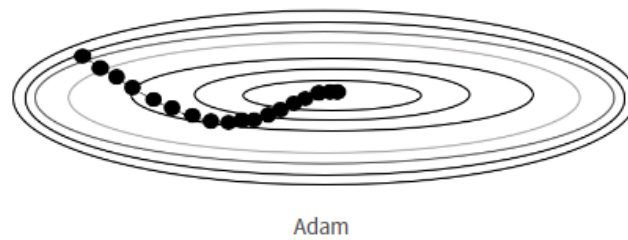
1. Stochastic Gradient Descent (SGD)



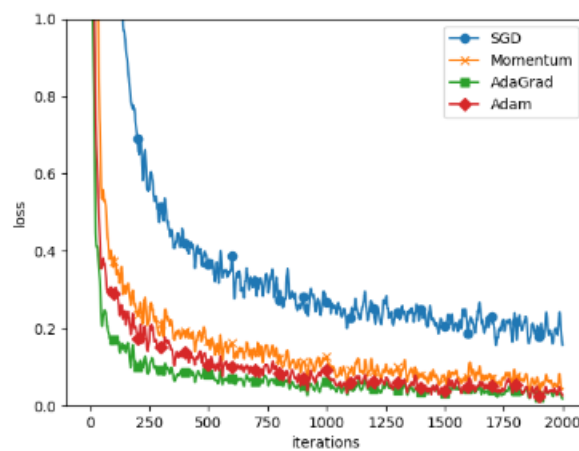
$$W = W - \eta \frac{\partial L}{\partial W}$$

adam- 하이퍼파라미터 x

5. Adam



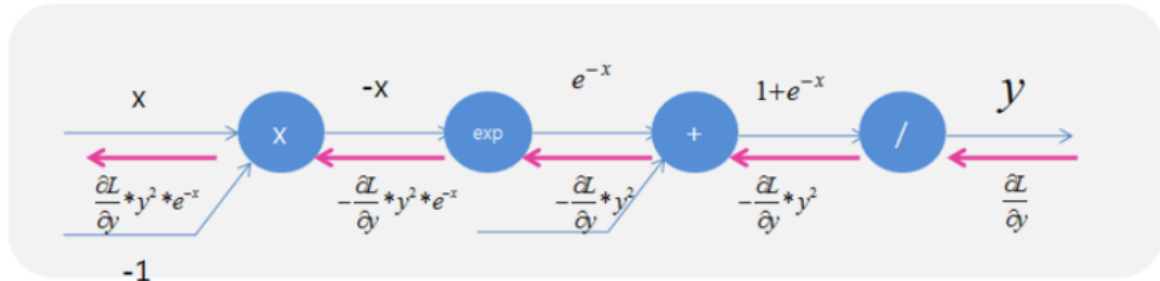
optimizers comparison (hyper parameter)



adam vs SGD

- SGD는 learning_rate를 수동으로 지정해줘야 하지만,
- adam은 알아서 파라미터를 조정하며 최적값을 정해준다.

sigmoid 역전파 순전파



class

- 모듈: a페이지의 함수를 b페이지에서 사용할 수 있게 하는것
- class: 붕어빵 틀/ 객체: 붕어빵
- class에 자주 사용되는 함수들을 저장할 수 있다.
- b페이지에서 a페이지의 함수를 사용하는 방법

: a.클래스명.함수명(변수)

번외,,

:챗gpt 활용능력 더 중요해질 예정 🙄

[AI패권전쟁 한국의 승부수] '챗GPT' 회사가 4억원 연봉 제시한 일자리...국내선 100대1 경쟁률
 생성형 인공지능(AI) 강자로 꼽히는 스타트업 뤼튼테크놀로지스(뤼튼)는 최근 국내 '공채 1호' 프롬프트 엔지니어를 선발했다. 이들은 AI에 질문을 입력해 더 좋은 답을 내놓도록 유도하는 일을 맡는다. 경쟁률은 100대1에 달했다. 많은 이들에게 생소한 직업이지만, AI 시대에 적응하기 위해 이미 많은 이들이 발 빠르게 움직이고 있다.
<https://cm.asiae.co.kr/ranking-view/20230619/2023061907213736516>



- 챗gpt에서도 파라미터가 나온다.
- 너는 마케터야, 내가 김자반이랑 김을 판매하려고 하는데, 창의적인 문구를 5개를 써줘. temperature=0~2

midjourney : 문장을 입력하면 ai가 그림을 그려준다.