5장 인공신경망

@쯔아누 @2023년 2월 23일

예제 코드

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/5bbd667d-6620-45ec-8ef0-1809b616bf23/5.1_Linear_Regression.ipynb

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/0dae2232-e770-4361-9fc5-80170368de5d/5.1_MLP.ipynb

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/5094e870-845a-4803-abc3-c52dfc716a0c/5.4_Optimizers.ipynb

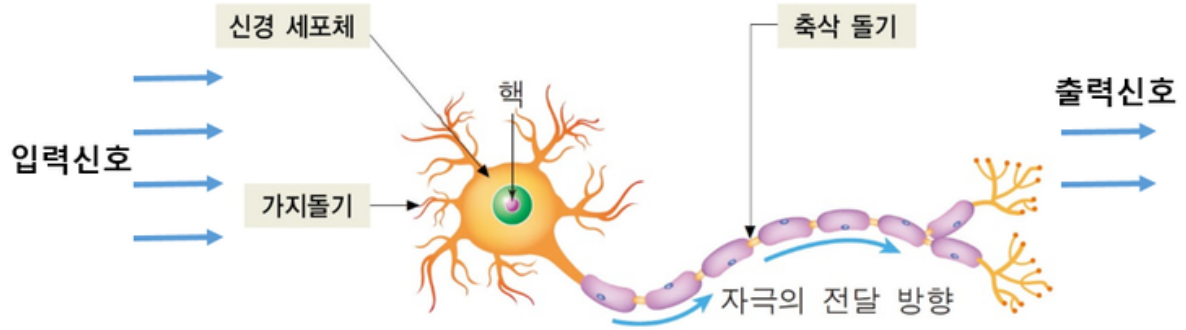
https://s3-us-west-2.amazonaws.com/secure.notion-static.com/d4ac0c42-8ff2-4a71-a3b5-d6953f6922de/5.5_Cross-Validation.ipynb

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/3dc320da-37c0-427c-8a62-661a70d7cf8c/5.6_Model_parameters.ipynb

다층 퍼셉트론

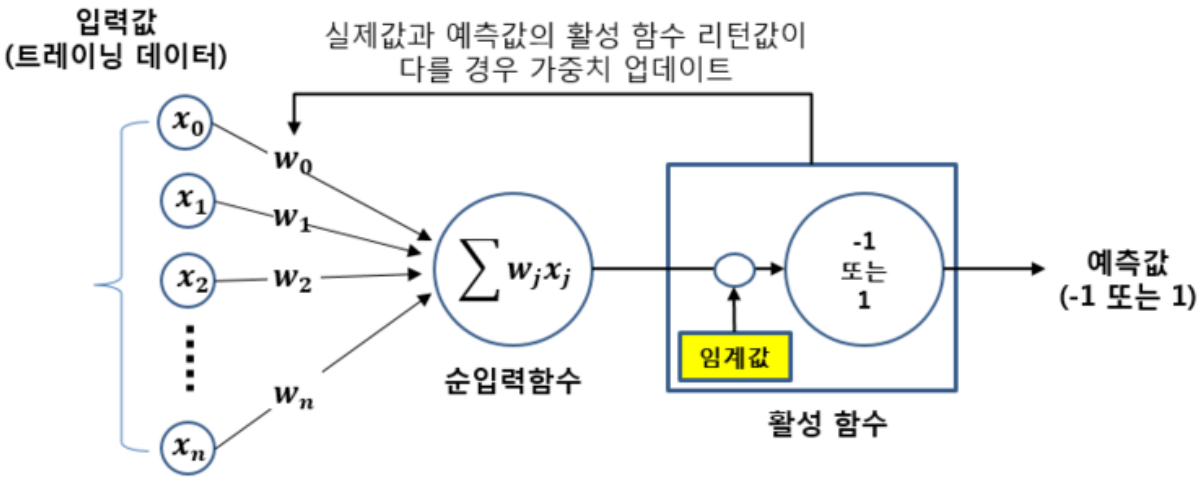
퍼셉트론이란?

인간의 뇌를 구성하는 세포인 뉴런은 가지돌기, 신경세포체, 축삭돌기로 구성되어 있다. 이러한 뉴런을 단순화 된 원리로 설명하는 MCP 뉴런 모델을 소개한다.



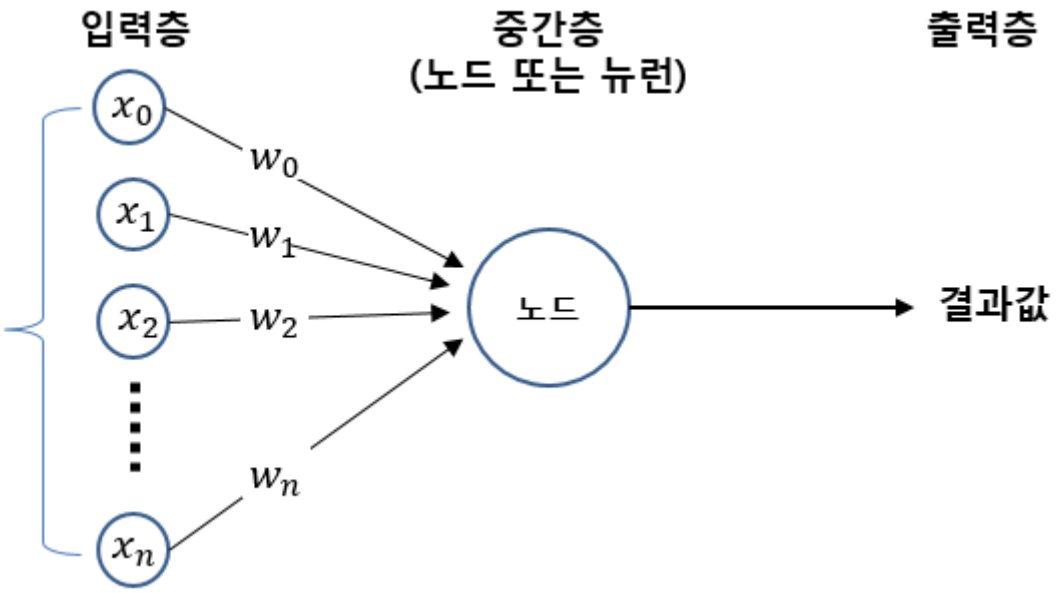
MCP 뉴런에서 시냅스라는 입력신호가 가지돌기로 들어오면 신경세포체에서 이를 하나의 신호로 통합하고, 통합된 신호가 어떤 임계값을 초과하면 하나의 단일 신호가 생성되어 축삭 돌기를 통해 다른 신경세포로 전달된다.

그리고 MCP 뉴런 모델을 기초로 **퍼셉트론** 학습 규칙이라는 개념이 고안된다.



위에서 설명한 뉴런 모델에 빗대어 설명하면 입력신호로 x 값들이 들어오고 신경세포체 대신 순입력함수에서 들어온 신호들을 하나의 통합된 신호로 변환한다. 그리고 통합된 신호가 어떠한 임계값을 초과하면 1, 그렇지 않으면 -1이 반환되어 다른 퍼셉트론의 입력으로 들어가게 된다. 일반적으로 x 값들로 데이터의 특성(features)이 들어오기 때문에 모델을 학습시킬 때 x 값들 대신 w 값들을 업데이트한다.

이 모델을 단순하게 바꿔 다시 표현해보면 아래와 같이 나타낼 수 있다.

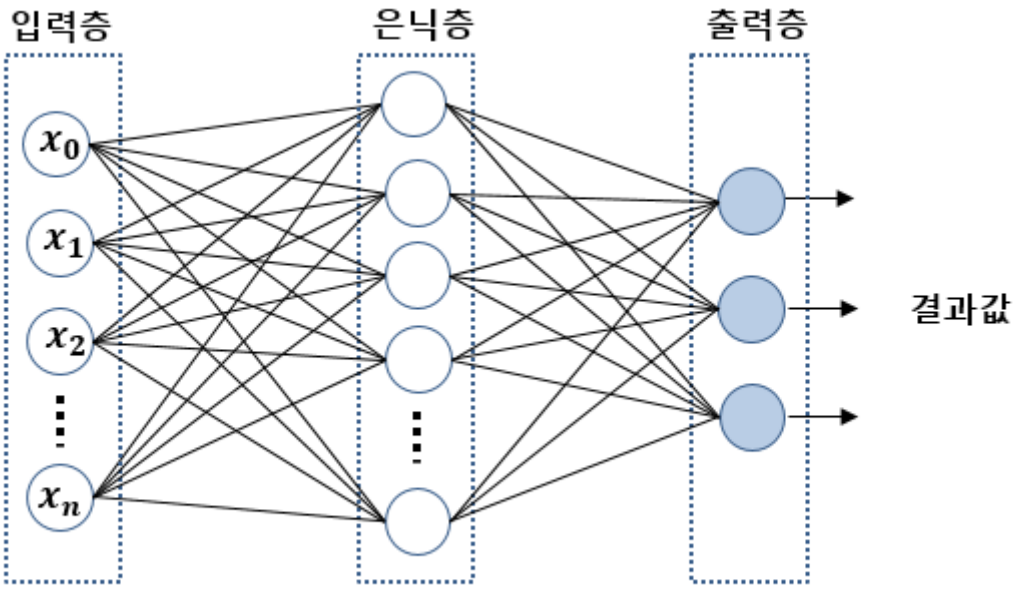


노드라는 개념이 새롭게 정의되었다. 노드는 순입력함수와 활성함수를 포함하고 여러개의 입력 데이터에 대해 하나의 출력 데이터를 반환한다. 입력 데이터는 이전 노드의 출력 데이터를 사용하고 현재 노드의 출력 데이터는 다음 노드의 입력 데이터로 사용된다.

다층 퍼셉트론

앞서 다룬 퍼셉트론은 하나의 노드에 대해 여러 입력값들이 들어오면 하나의 출력값을 반환하는 단층 퍼셉트론으로, 1과 -1으로 주어진 어떤 데이터에 대한 선형 분류가 가능했으나 비선형 분류는 불가능한 단점이 있다.

이를 극복하고자 입력층과 출력층 사이에 하나 이상의 중간층을 두어 비선형적으로 분리되는 데이터에 대해서도 학습이 가능하게 한 것이 다층 퍼셉트론이다.



이렇게 입력층과 출력층 사이에 존재하는 중간층을 다른 말로 은닉층이라고도 하며 여러개의 은닉층이 있는 인공신경망을 심층 신경망(deep neural network)이라고 한다. 우리가 흔히 알고 있는 딥러닝(deep learning)이란 분야는 이러한 심층 신경망을 주로 다루는 분야를 의미한다.

추가 1: 순전파(feedforward)란?

순전파란 입력층의 각 노드로부터 전달되는 값이 은닉층의 모든 노드로 전달되고 마찬가지로 은닉층의 각 노드로부터 전달되는 값이 출력층의 모든 노드로 향하는 것을 말한다. 즉 루프를 돌지 않으면서 각 층의 출력값이 다음 층의 입력값이 되고, 이는 이후 나올 RNN(Recurrent Neural Network)과 대비되는 개념이다.

추가 2: 학습이 어떻게 이루어지는가?

단층 퍼셉트론의 경우 활성함수의 결과값이 기대한 값과 다를 경우 입력값들 각각에 적용되는 가중치의 값을 변경해주면 되었다. 하지만 다층 퍼셉트론의 경우 하나 이상의 은닉층이 존재하고 각 은닉층에서 어떤 결과값이 나오는 것이 옳은지 알 수 없다. 따라서 이 문제를 해결하기 위한 방법으로 역전파(backpropagation)이란 학습 방법을 사용한다. 이에 대한 내용은 생략한다.

활성화 함수

활성화 함수는 기본적으로 선형인 함수를 비선형 꼴로 만들어주는 함수이다. 연산을 복잡하게 만드는 과정이 왜 필요할까 생각할수도 있겠지만 선형 함수를 미분하면 언제나 상수가 나

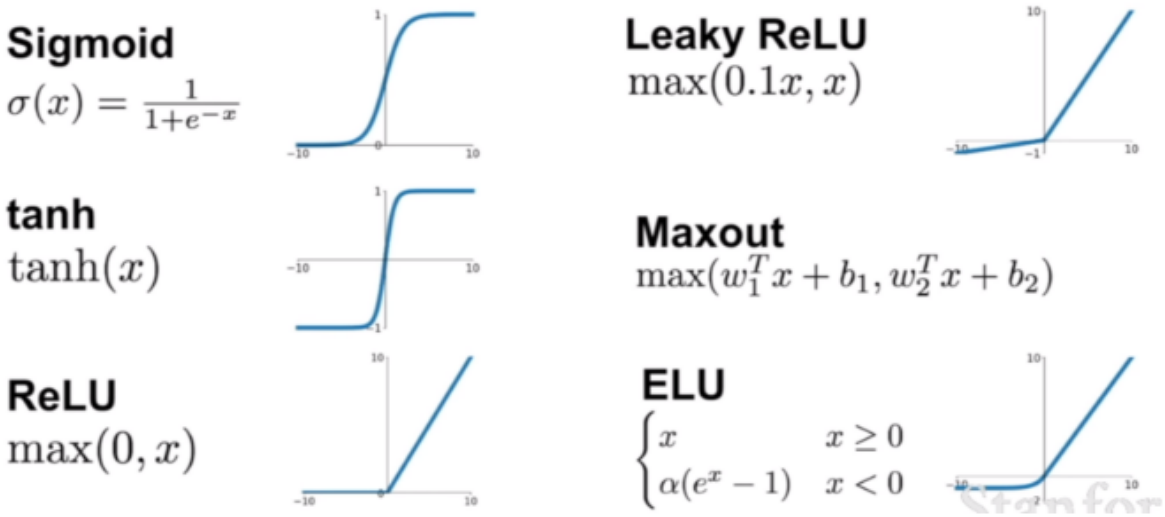
온다는 점에 집중할 필요가 있다.

일반적으로 통계적모델이나 딥러닝모델을 학습하는 즉 최적화하는 방법으로 손실함수의 최소값을 찾는 방법을 사용한다. 그 방법으로 손실함수를 미분하여 0이 되는 점을 찾고 최소값인지 확인하거나 뉴튼-랩슨, 가우스 반복법 등을 사용할 수 있다. 그러기 위해선 미분한 결과값이 중요한데 만약 그 결과값이 입력값과는 상관없이 언제나 상수가 나온다면 학습이 불가능할 것이다. 따라서 학습을 위하여 비선형함수 꼴로 만들어주는 활성화 함수가 필요한 것이다.

또한 딥러닝 모형의 경우 여러 개의 층을 쌓게 되는데 선형 함수의 경우 아무리 쌓아도 선형함수 꼴로 나타나므로 층을 쌓는 의미가 손실된다.

$$W_1 x_1 + b_1 = x_2$$
$$W_2(W_1 x_1 + b_1) + b_2 = W_2 x_2 + b_2$$
$$W_n(W_{n-1}(...) + b_{n-1}) = W_n x_n$$

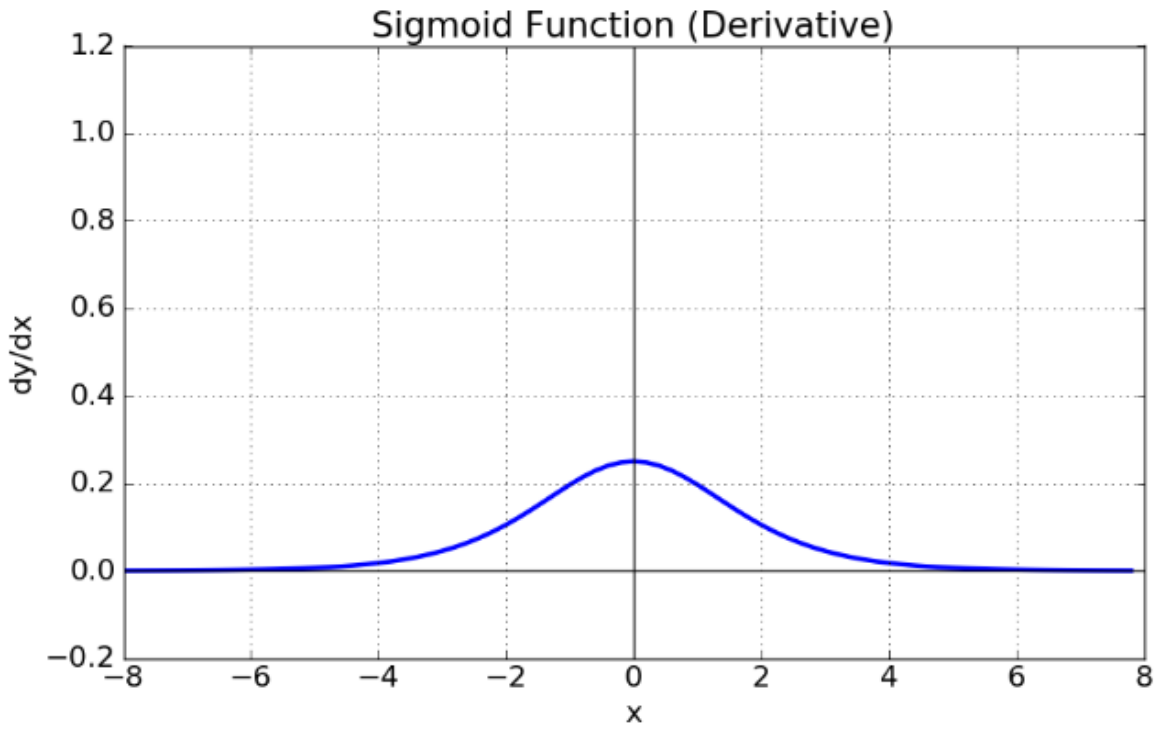
흔히 사용하는 활성화 함수로는 아래 6가지의 함수가 있다. 이 중에서도 왼쪽의 3가지를 일반적으로 사용하곤 한다. 함수마다 약간의 차이는 있지만 큰 틀에서의 용도는 같기 때문에 왼쪽의 세가지 함수만 짚고 넘어간다.



Sigmoid

$$sigmoid(x) = \frac{1}{1 + e^{-x}}$$

시그모이드 함수의 결과값은 0~1 사이의 값으로 나타난다. 그러나 그래프의 개형을 보면 알 수 있듯이 x의 절대값이 4만 넘어가도 기울기의 변화가 미미해지므로 큰 x 값이 들어오면 기울기를 소실하는 vanishing gradient 문제가 발생할 수 있다. 아래의 도함수 그림을 보면 더 명확하다.



또한, 함수의 중심이 0이 아니고 지수함수를 사용하기 때문에 그로부터 발생하는 학습 속도의 감소가 있을 수 있다. (*함수의 중심이 0이 아닌 것이 학습 속도의 감소로 이어지는 정확한 원인은 모르겠다.*)

그럼에도 이 함수를 많이 사용하는 이유는 미분 결과가 간결할 뿐더러 해석도 용이하기 때문이다. 시그모이드 함수는 로지스틱 함수라고 불리기도 하는데, 로지스틱 회귀분석에서 성공이 나올 확률을 표현할 때 주로 사용한다.

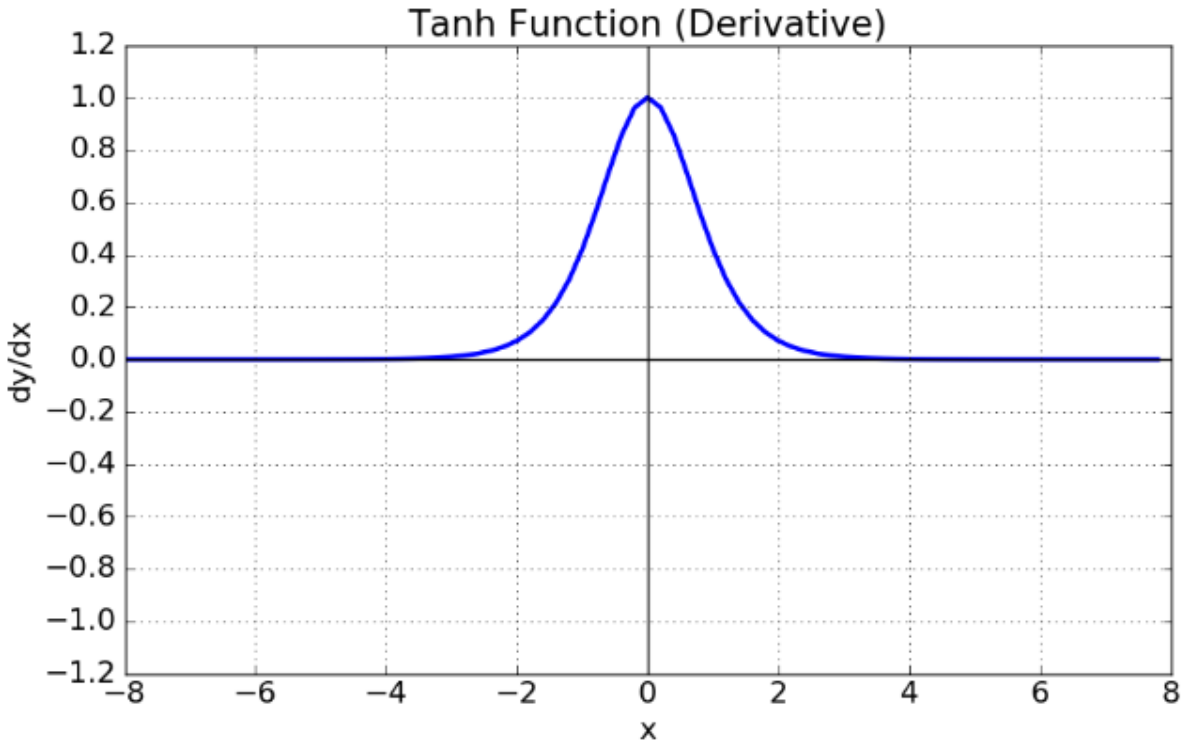
Tanh

$$tanh(x) = 2\sigma(2x) - 1$$

$$tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$$

$$tanh^{'}(x) = 1 - tanh^{2}(x)$$

Tanh 함수의 결과값은 -1~1 사이로 중심이 0이기 때문의 학습 속도의 감소를 줄일 수 있다. 하지만 이 역시 지수 함수를 사용하기 때문에 그에 따른 학습 속도의 감소가 발생한다. 뿐만 아니라 이 역시 x의 절대값이 3을 넘어가는 경우 기울기의 변화가 미미해지므로 여러 층을 쌓을 경우 vanishing gradient 문제를 발생시킬 수 있다.



ReLU

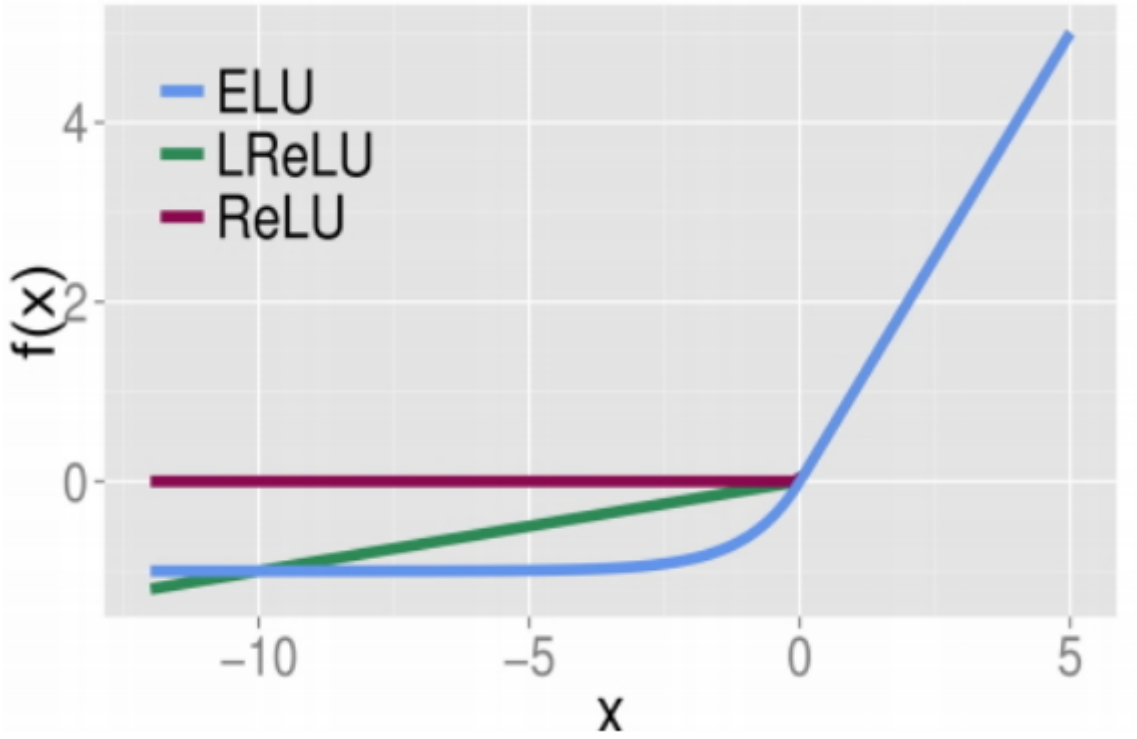
$$f(x) = max(0, x)$$

최근 많이 사용하는 활성화 함수인 ReLU 함수는 양수의 입력값에 대해서 그대로 반환하기 때문에 처리 속도가 굉장히 빠르고 vanishing gradient 문제도 어느정도 해소할 수 있다. 이에 대한 미분값은 다음과 같다.

- 1 (x > 0)
- 0 (x ≤ 0)

추가: ELU, LReLU

ELU와 Leaky ReLU는 기존의 ReLU 함수가 음수에 대해 기울기가 소멸하는 문제를 해결하기 위한 방법으로 사용한다.



손실 함수

수학적 의미에서의 학습이란?

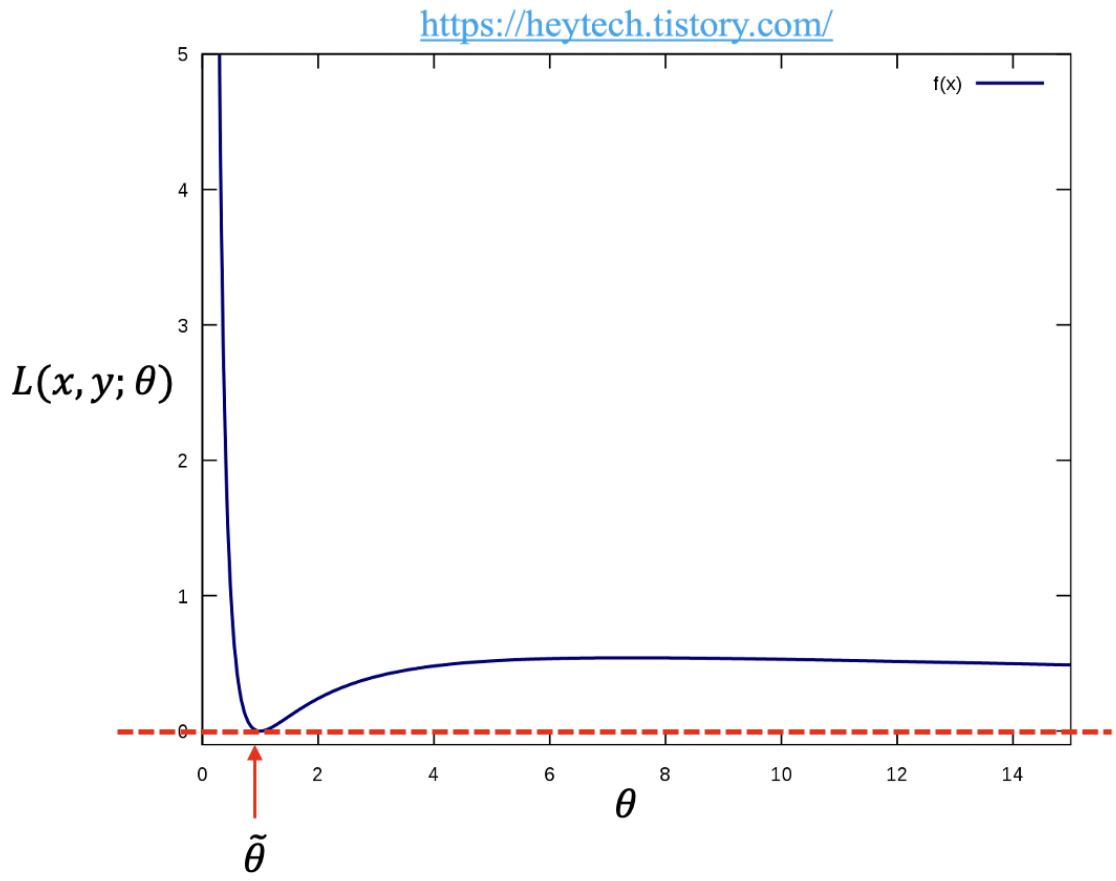
먼저 다음 예시를 살펴보자.

$$\tilde{\theta} = \arg\min_{\theta} L(x, y; \theta)$$

수식 내 각각의 의미는 다음과 같습니다.

- L: 손실 함수
- $\arg\min$: arguments of minimum을 축약한 수학적 표현으로, 목적 함수를 최소화하는 입력값을 찾는 역할(목적 함수: L)
- x: 학습 데이터의 입력값으로, x로 얻어낸 예측값($\hat{y}$)은 정답(y)과 비교
- y: 학습 데이터의 정답
- θ: 알고리즘 학습 시 사용되는 모든 파라미터의 벡터
- $\tilde{\theta}$: 추정된 최적의 파라미터

알고리즘을 학습한다는 것은 알고리즘이 예측한 값인 y hat 이 정답인 y 와 최대한 근접하게 만드는 것을 의미한다. 그리고 y 와 y hat 사이의 차이를 계산하는 함수를 손실함수라고 정의하면 앞의 문장을 손실함수를 최소화 하는 파라미터를 찾는다는 것으로 이해할 수 있다.



자주 사용하는 손실함수로는 아래 세 가지가 있다. 하나씩 소개해보려 한다.

- MAE

- MSE
- Cross Entropy Loss

MAE (L1 Error)

$$L = \frac{1}{n}\Sigma_{i=1}^{n}|y_i - \hat{y}_i|$$

- 손실이 오차에 비례하여 일정하게 커지므로 이상치 값에 대해 강건하다.

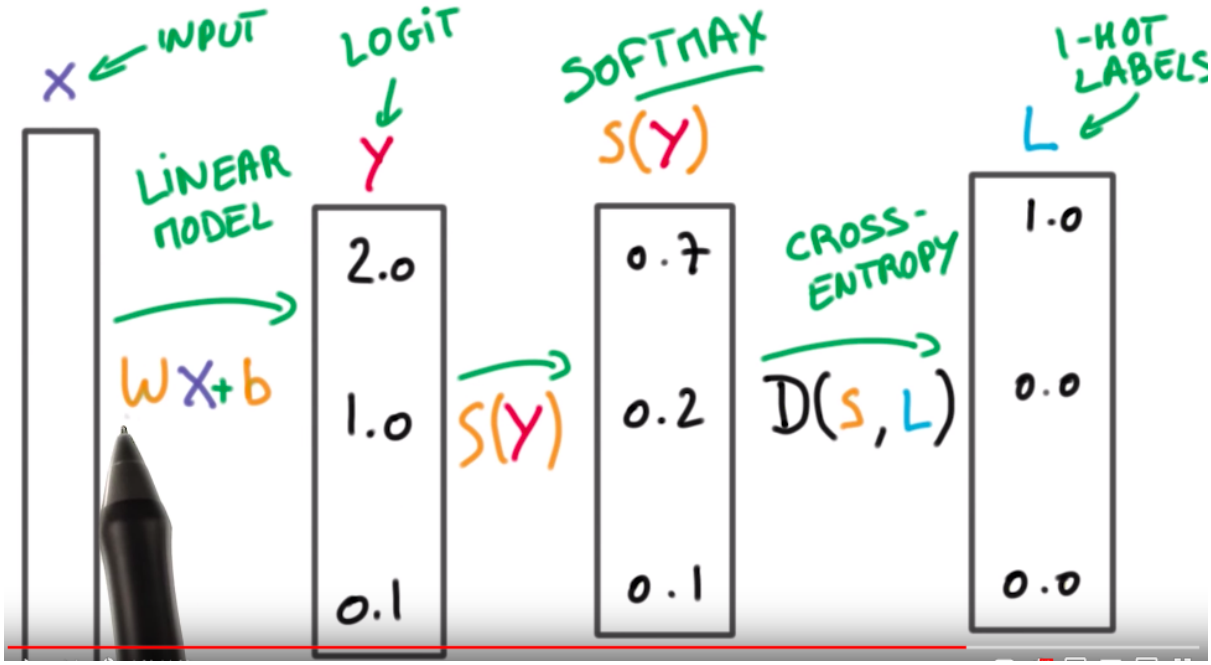
MSE (L2 Error)

$$L = \frac{1}{n}\Sigma_{i=1}^{n}(y_i - \hat{y}_i)^2$$

- 손실이 오차가 증가함에 따라 제곱배만큼 커지므로 이상치 값에 민감하다.

Cross Entropy Loss

$$X : \text{input}$$
$$y = WX + b : \text{score}$$
$$s = softmax(y)$$
$$L = crossEntropy(s, class)$$
$$class : \text{OneHot vector}$$



- Cross entropy 손실함수는 주로 둘 이상의 클래스를 분류할 때 사용한다.
- CrossEntropy(s, class)를 계산하는 방법은 여러가지가 있는데 보통은 간단한 아래와 같은 방법을 사용한다.
  - i : 정답 class의 index
  - L = -log(s_i)
    - softmax 결과는 0~1 사이의 값을 가지므로 -log(s)의 결과는 0~oo 사이의 값을 갖는다.

추가: Regression vs Classification

Problem Type	Output Type	Final Activation Function	Loss Function
Regression	Numerical value	Linear	Mean Squared Error (MSE)
Classification	Binary outcome	Sigmoid	Binary Cross Entropy
Classification	Single label, multiple classes	Softmax	Cross Entropy
Classification	Multiple labels, multiple classes	Sigmoid	Binary Cross Entropy

최적화 기법

Todo

SGD

Momentum

AdaGrad

RMSProp

Adam

교차검증

기존의 학습은 train data를 활용하여 모델을 학습시키고 test data로 모델의 성능을 평가한다. 그러나 train data가 고정된 상태에서 학습을 여러 횟수 거치게 되면 train data에 과적합

된 모델이 만들어질 수 있다. 과적합의 여부는 train data에 대한 성능은 올라가지만 test data에 대한 성능이 감소하는 것으로 판단할 수 있다.

이를 해결하기 위해 train data를 학습을 위한 train data와 학습 단계에서 평가하기 위한 validation data 또는 test data로 나누는 방법을 교차검증(Cross Validatoin)이라고 한다. 교차검증을 활용한다면 학습 단계에서 train data에 대한 성능과 validation data에 대한 성능을 비교하고 과적합 여부를 판단할 수 있다.

- 장점: 과적합을 방지하고 일반화된 모델을 생성할 수 있다.
- 단점: 모델 훈련 및 평가에 소요되는 시간이 증가한다.

Hold-out Cross Validation

가장 기초적인 교차검증 방법. 특정 비율로 train/test data를 분할한다.

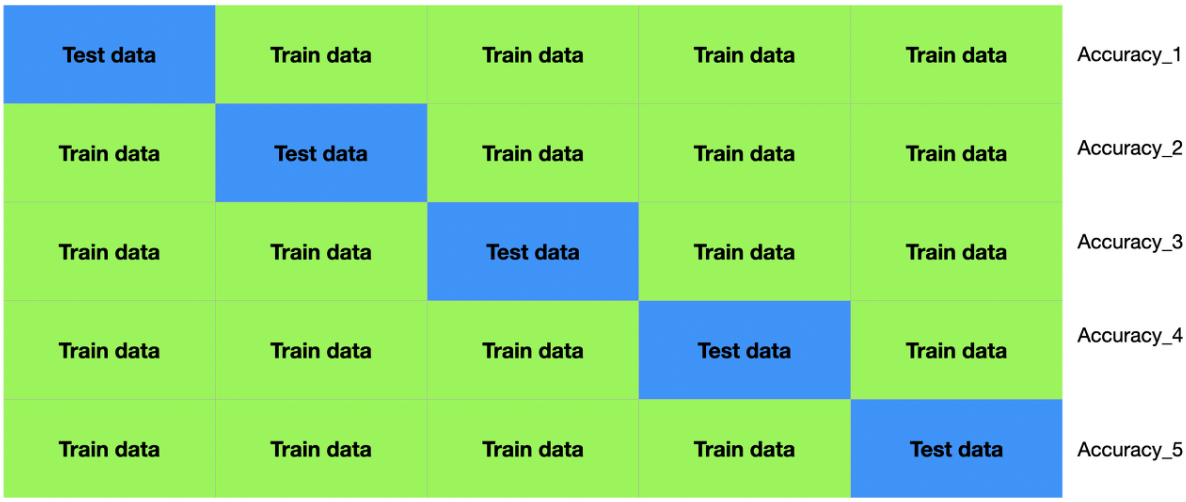
https://heytech.tistory.com/

K-Fold Cross Validation

전체 데이터 셋을 K개의 fold로 나누고 K번의 반복동안 1개의 test fold를 선택하고 나머지 (K-1)개의 fold를 train data로 사용한다. 그리고 최종 결과값은 모든 반복동안 나타난 결과값의 평균으로 나타낸다.

K가 클수록 많은 양의 데이터를 학습에 사용하고 K가 작을수록 많은 양의 데이터를 평가에 사용한다.

- 장점: 모든 데이터를 train과 test 데이터로 활용할 수 있다.
- 단점: 순서가 고려된 데이터가 shuffle 되어 있지 않은 경우 성능이 악화될 수 있다.



Accuracy = Accuracy1 + Accuracy2 + … + Accuracy5

https://heytech.tistory.com/

LOOCV

- K=n 인 K-Fold CV, 1개의 fold가 1개의 data인 경우를 의미한다.
- 데이터의 개수가 적은 경우 효과적인 방법론이다.

모델 구조 및 가중치 확인

💡 todo