손에 잡히는 딥러닝

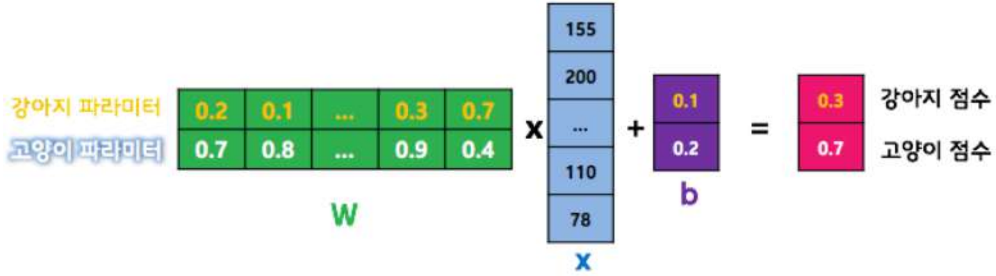
Loss function and Optimization

모두의연구소

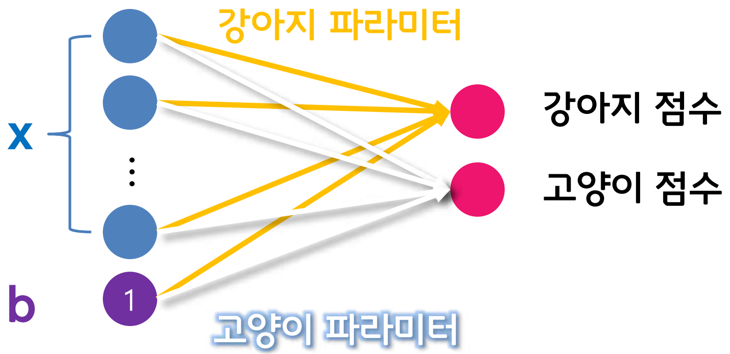
박은수 Research Director

Parametric approach

- Score function : Simple Linear Classifier



선형분류기의
뉴럴 네트워크 표현



Softmax Loss

- Cross entropy loss -

$$P(Y=k|X=x_i)=\frac{e^{s_k}}{\sum_j e^{s_j}}$$ where $$s=f(x_i;W)$$

Want to maximize the log likelihood, or (for a loss function) to minimize the negative log likelihood of the correct class:

$$L_i=-\log P(Y=y_i|X=x_i)$$

강아지 점수 2.3

고양이 점수 -1.2

---

in summary: $$L_i=-\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$

Softmax Loss

- Cross entropy loss -

$$L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$

강아지 점수	2.3	$\xrightarrow{\text{exp}}$	9.97
고양이 점수	-1.2		0.3

전부다 양수가 됨

Softmax Loss

- Cross entropy loss -

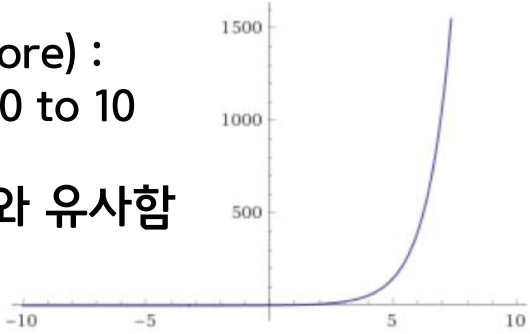
$$L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$

강아지 점수	2.3	exp →	9.97
고양이 점수	-1.2		0.3

전부다 양수가 됨

exp(score) :
from -10 to 10

Max함수와 유사함



Softmax Loss

- Cross entropy loss -

$$L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$

강아지 점수	2.3	exp →	9.97	normalize →	0.97
고양이 점수	-1.2		0.3		0.03
			전부다 양수가 됨		확률

Softmax Loss

- Cross entropy loss -

$$L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$

강아지 점수	2.3	exp	9.97	normalize	0.97
고양이 점수	-1.2		0.3		0.03
			전부다 양수가 됨		확률

Softmax 함수

Supervised Learning

A → B

집 크기 + 방 수 (B) | 집 가격 (A)

size of house (square feet)	# of bedrooms	price (1000$)
523	1	115
645	1	150
708	2	210
1034	3	280
2290	4	355
2545	4	440

B ← A

A

B 고양이

그림 인용 : Coursera - AI for Everyone : Machine learning

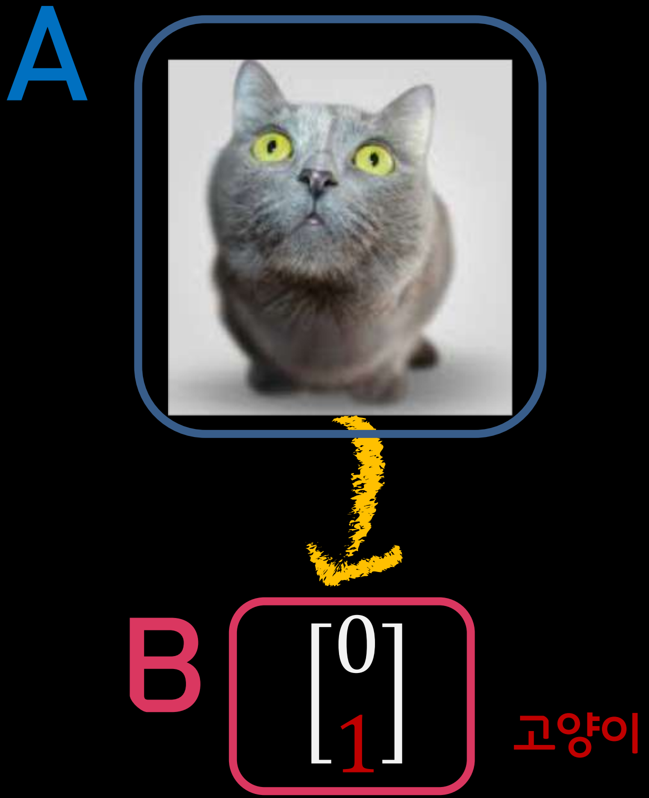
고양이 ?

A

B

고양이

One-hot Vector



One-hot Vector

A

B

$$\begin{bmatrix} 1 \\ 0 \end{bmatrix} \begin{array}{l} \text{강아지} \\ \text{고양이} \end{array}$$

Softmax Loss

- Cross entropy loss -

$$L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$

강아지 점수	2.3	exp →	9.97	normalize →	0.97	$\begin{bmatrix}0\\1\end{bmatrix}$
고양이 점수	-1.2		0.3		0.03	
			전부다 양수가 됨		추정 확률	정답

Softmax Loss

- Cross entropy loss -

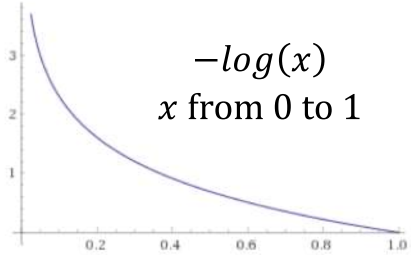
$$L_i = -\log\left(\boxed{\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}}\right)$$

강아지 점수	2.3	exp →	9.97	normalize →	0.97	$\begin{bmatrix}0\\1\end{bmatrix}$
고양이 점수	-1.2		0.3		**0.03**	
			전부다 양수가 됨		확률	정답

Softmax Loss

- Cross entropy loss -

$$L_i = -\log\left(\boxed{\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}}\right)$$



$-log(x)$

x from 0 to 1

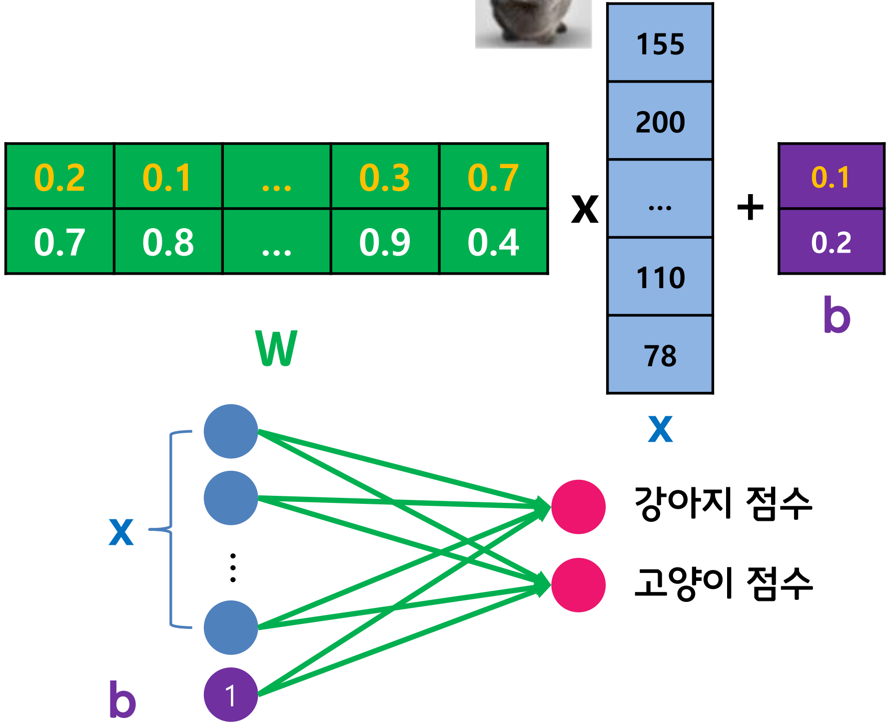
강아지 점수	2.3	exp →	9.97	normalize →	0.97
고양이 점수	-1.2		0.3		**0.03**
			전부다 양수가 됨		확률

$L_i = -log(0.03) = 3.5$

강아지와 고양이 분류해보기

- 분류기의 구성
  - Score function
  - Loss function

고양이가 입력이면 고양이
점수가 높아야 함



강아지와 고양이 분류해보기

- 분류기의 구성
  - Score function
  - Loss function

0.2	0.1	…	0.3	0.7
0.7	0.8	…	0.9	0.4

W

x

155
200
…
110
78

x

\+

0.1
0.2

b

=

Cross-entropy loss (Softmax)

		exp →		normalize →		
강아지 점수	2.3		9.97		0.97	강아지 확률
고양이 점수	-1.2		0.3		0.03	고양이 확률

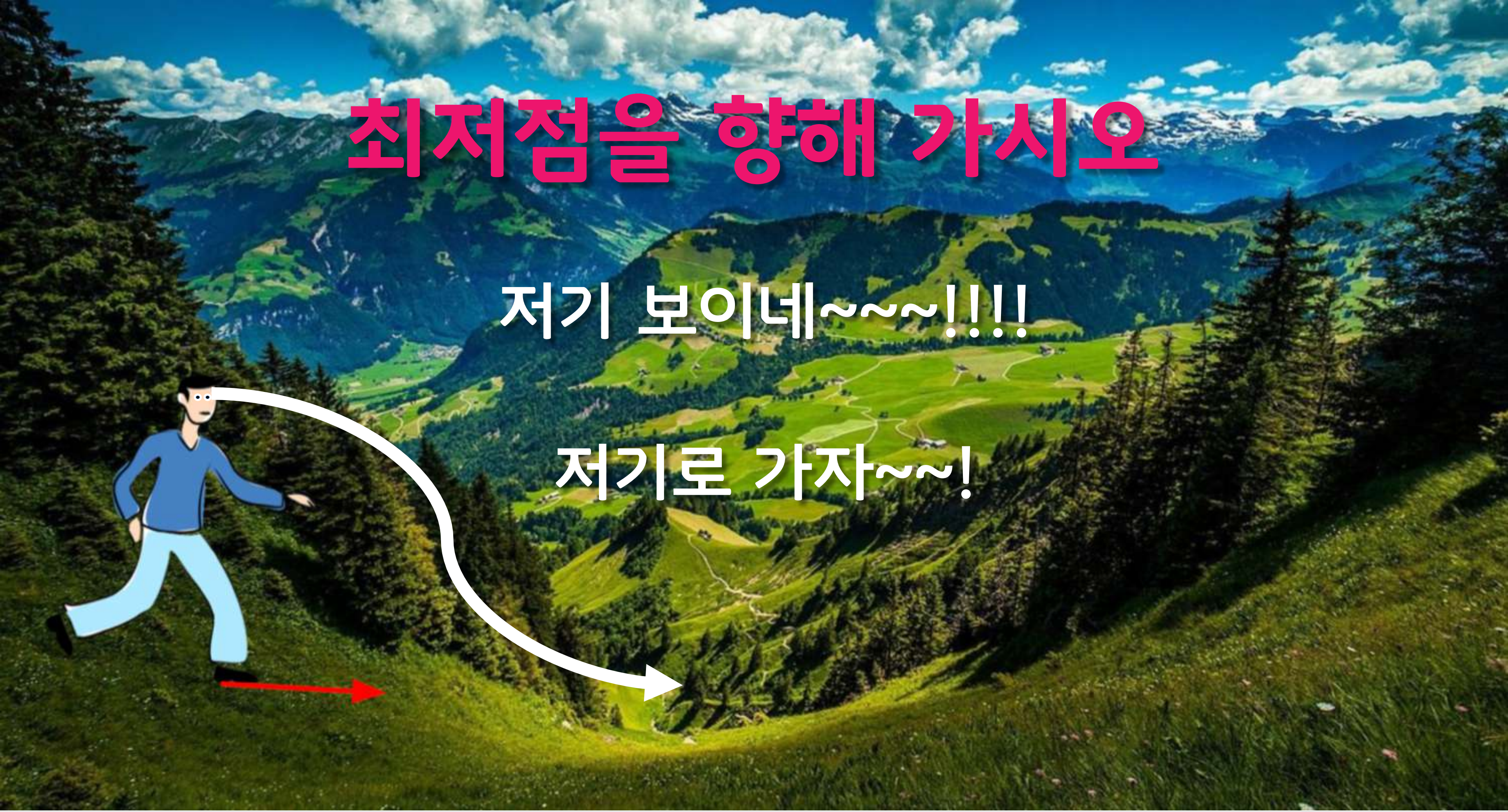
$$-log(0.03) = 3.5$$

현재의 분류기는 3.5만큼 안 좋음. 이 loss 값을 줄이는게 목표

- 분류기의 구성
  - Score function
  - Loss function
  - Optimization

이제 Loss를 최소로 하는 W를 찾는 방법만 남았습니다

최저점을 향해 가시오



From : lecture note 2 - cs231n stanford university


최저점을 향해 가시오
헉?!
눈이 없으면 ?


최저점을 향해 가시오
넘어질 것 같아..
어디까지 갈 수 있을까..
[대안]
발로 더듬더듬 해서 내리막이면 가자!
그냥 눈을 다시 그려줘 ...

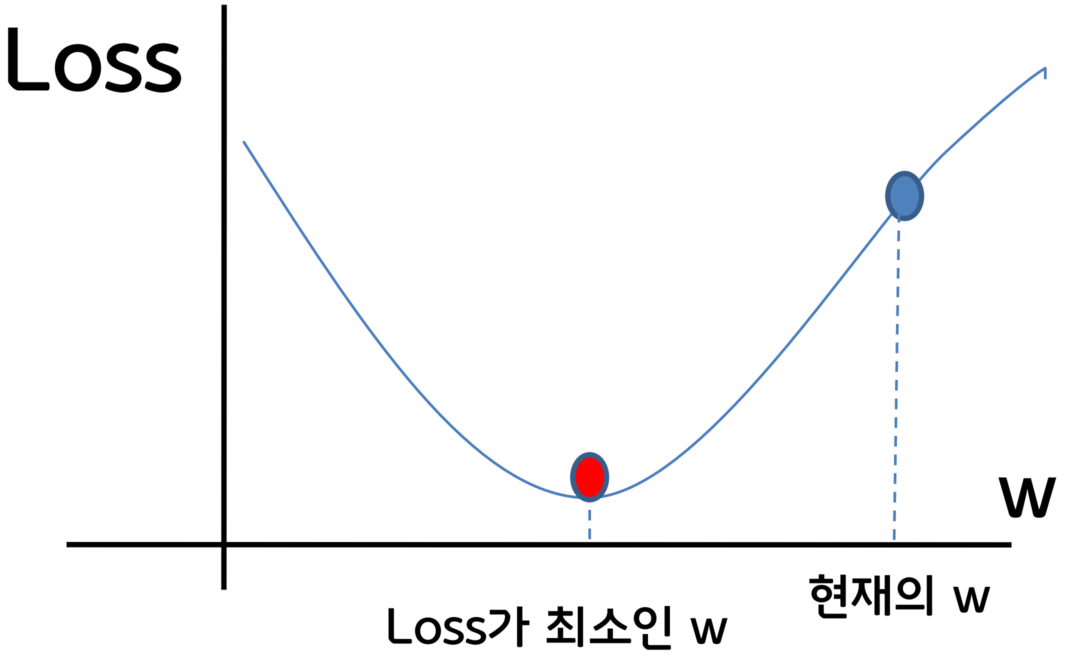
내리막을 찾는 방법

내리막 ?? == 기울기 ??

기울기를 따라 내려 가보자

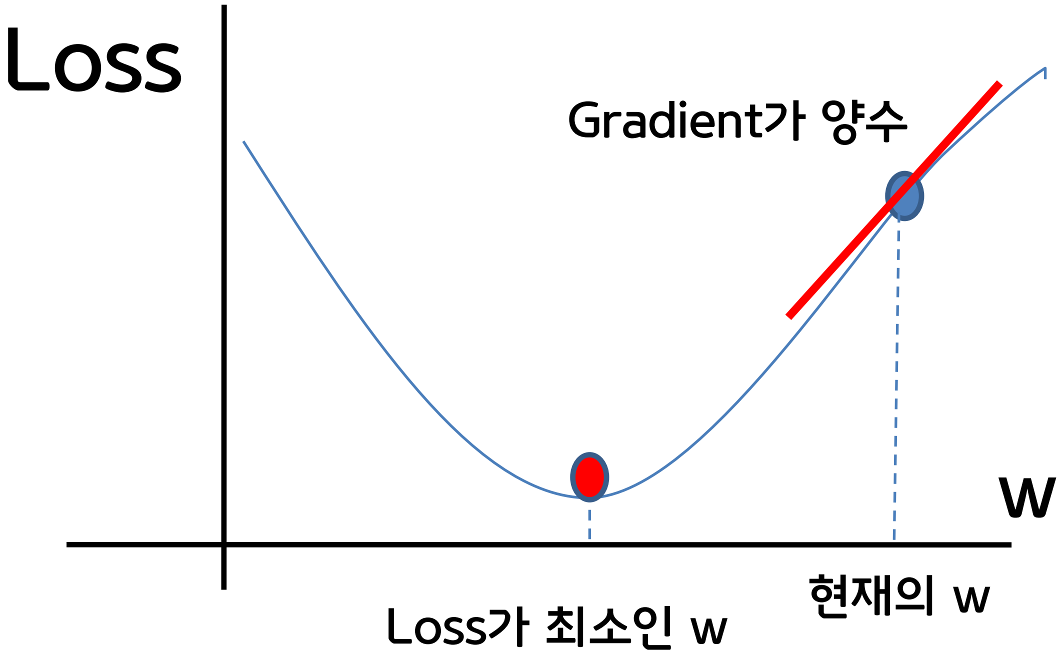
Gradient Descent

Gradient Descent



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

Gradient Descent

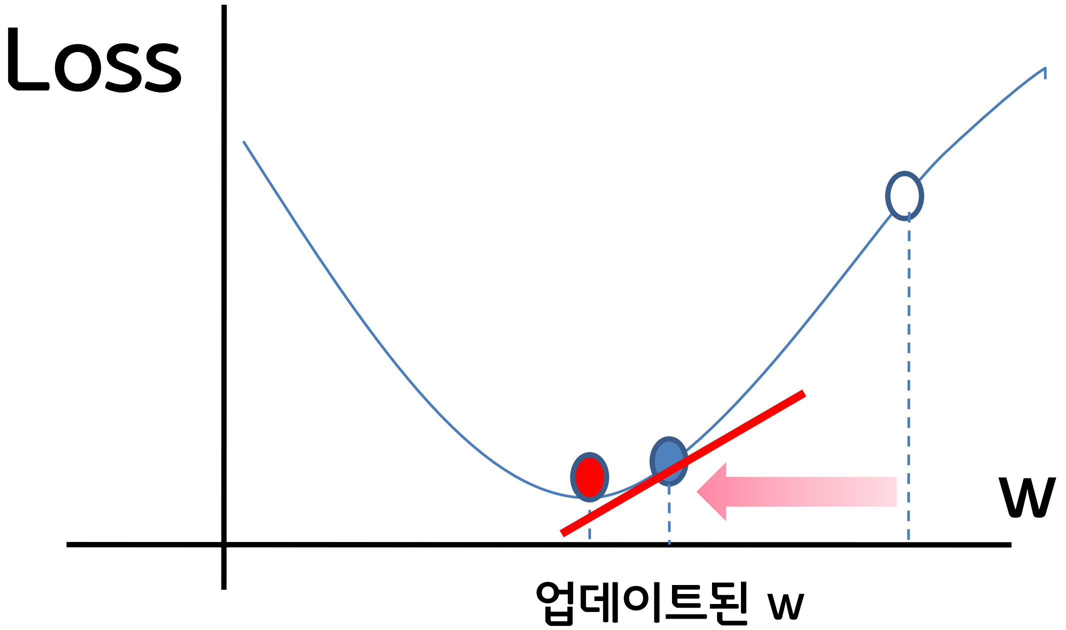


$$w = w - \eta \boxed{\frac{\partial L}{\partial w}}$$

η : learning rate

Gradient Descent

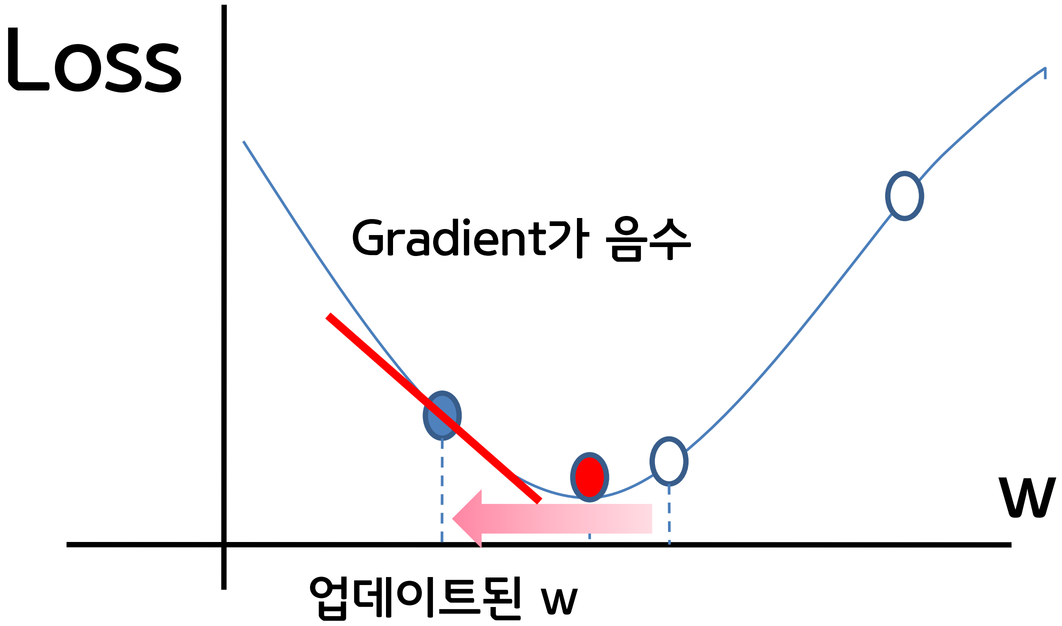
모두의연구소



$$w = \boxed{w - \eta \frac{\partial L}{\partial w}}$$

η : learning rate

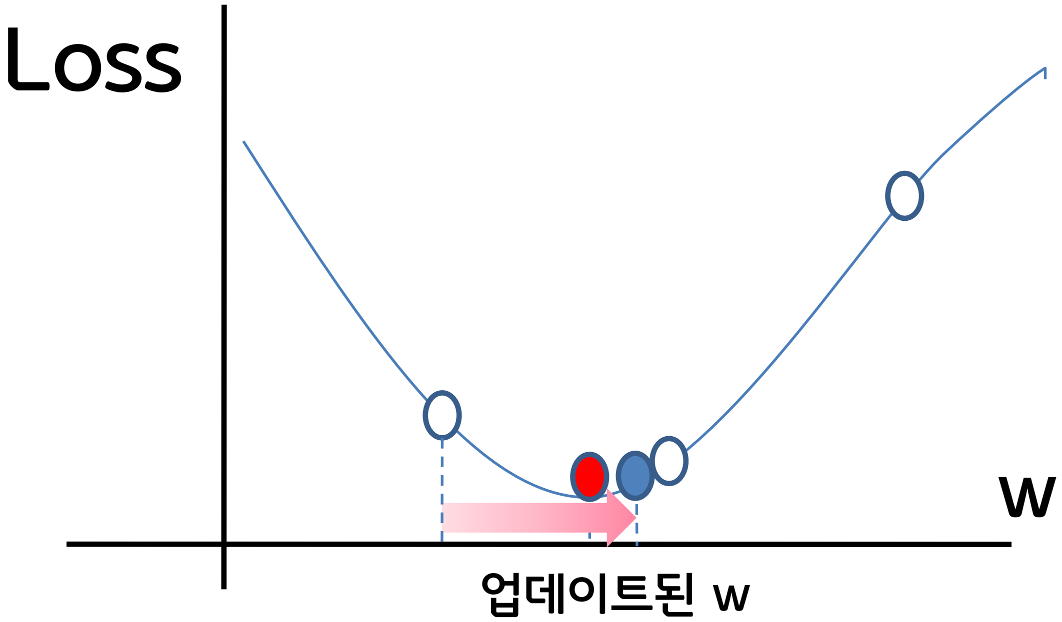
Gradient Descent



$$w = \boxed{w - \eta \frac{\partial L}{\partial w}}$$

η : learning rate

Gradient Descent



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

Loss 함수에 대한 w의 음의 Gradient 찾아서 연속적으 업데이트해 주면 되는군요

그런데 어떻게 Gradient를 찾죠?

Loss를 W로 미분하기

$L($  $, W)$

0.2	0.1	…	0.3	0.7
0.7	0.8	…	0.9	0.4

W

x

155
200
…
110
78

x

\+

0.1
0.2

b

=

강아지 점수

2.3
-1.2

고양이 점수

exp

9.97
0.3

normalize

강아지 확률

0.97
0.03

고양이 확률

$-log(0.03) = 3.5$

Score function

Loss function

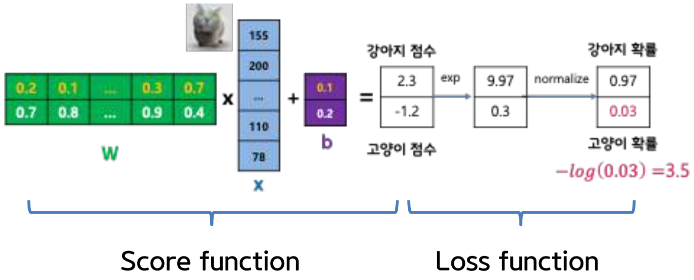
미분의 정의를 이용해 봅시다

모두의연구소



$$\frac{\partial \mathcal{L}(\quad, w_1)}{\partial w_1} = \lim_{h \to 0} \frac{\mathcal{L}(\quad, w_1+h) - \mathcal{L}(\quad, w_1)}{h}$$

$\mathcal{L}$: Loss function



$L($  $, W)$

Current W

$$\begin{pmatrix} 0.2 \\ 0.1 \\ 0.3 \\ \vdots \\ 0.9 \\ 0.4 \end{pmatrix}$$

w_1+h

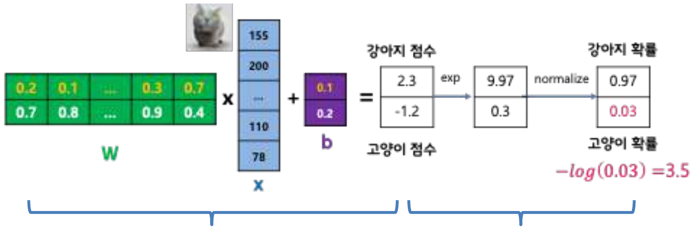
$$\begin{pmatrix} 0.2+\mathbf{0.0001} \\ 0.1 \\ 0.3 \\ \vdots \\ 0.9 \\ 0.4 \end{pmatrix}$$

Gradient dW

$$\begin{pmatrix} ? \\ ? \\ ? \\ \vdots \\ ? \\ ? \end{pmatrix}$$

Loss : $L($  $, w_1)=1.25347$

$L($  $, w_1+h)=1.25322$



$L($  $, W)$

Score function　　Loss function

Current W

$$\begin{pmatrix} 0.2 \\ 0.1 \\ 0.3 \\ \vdots \\ 0.9 \\ 0.4 \end{pmatrix}$$

Loss : $L($  $, w_1) = 1.25347$

w_1+h

$$\begin{pmatrix} 0.2+\mathbf{0.0001} \\ 0.1 \\ 0.3 \\ \vdots \\ 0.9 \\ 0.4 \end{pmatrix}$$

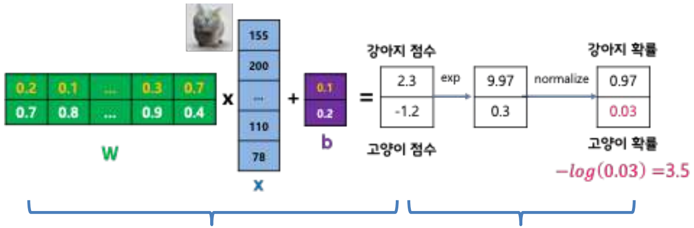
$L($  $, w_1+h) = 1.25322$

Gradient dW

$$\begin{pmatrix} -2.5 \\ ? \\ ? \\ \vdots \\ ? \\ ? \end{pmatrix}$$

$(1.25322-1.25347)/0.0001 = -2.5$

$$\lim_{h\to 0} \frac{L(\text{cat}, w_1+h) - L(\text{cat}, w_1)}{h}$$



Score function

Loss function

$L($  $, W)$

모두의연구소

Current W

$$\begin{pmatrix} 0.2 \\ 0.1 \\ 0.3 \\ \vdots \\ 0.9 \\ 0.4 \end{pmatrix}$$

w_2+h

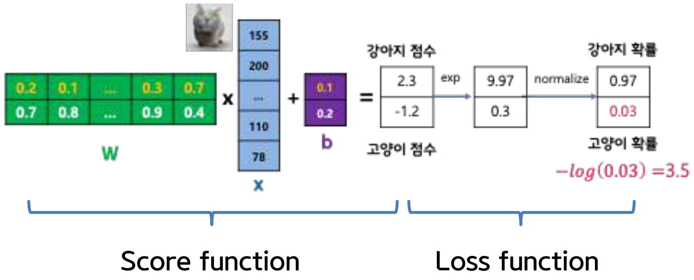
$$\begin{pmatrix} 0.2 \\ 0.1+\mathbf{0.0001} \\ 0.3 \\ \vdots \\ 0.9 \\ 0.4 \end{pmatrix}$$

Gradient dW

$$\begin{pmatrix} -2.5 \\ ? \\ ? \\ \vdots \\ ? \\ ? \end{pmatrix}$$

Loss : $L($  **$, w_1)$=1.25347**

$L($  **$, w_2+h)$ =1.25353**



$L($ [cat image] $, W)$

Current W

$$\begin{pmatrix} 0.2 \\ 0.1 \\ 0.3 \\ \vdots \\ 0.9 \\ 0.4 \end{pmatrix}$$

Loss : $L($ [cat image] $, w_1) =$ 1.25347

w_2+h

$$\begin{pmatrix} 0.2 \\ 0.1 + \mathbf{0.0001} \\ 0.3 \\ \vdots \\ 0.9 \\ 0.4 \end{pmatrix}$$

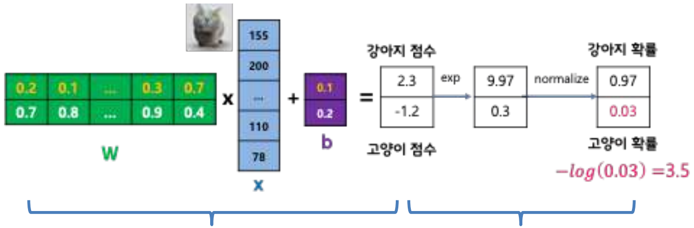
$L($ [cat image] $, w_2+h) =$ 1.25353

Gradient dW

$$\begin{pmatrix} -2.5 \\ 0.6 \\ ? \\ ? \\ ? \end{pmatrix}$$

(1.25353-1.25347)/0.0001 = 0.6

$$\lim_{h \to 0} \frac{L(\text{[cat image]}, w_1+h) - L(\text{[cat image]}, w_1)}{h}$$



Score function　　Loss function

$L($  $, W)$

Current W

$$\begin{pmatrix} 0.2 \\ 0.1 \\ 0.3 \\ \vdots \\ 0.9 \\ 0.4 \end{pmatrix}$$

w_2+h

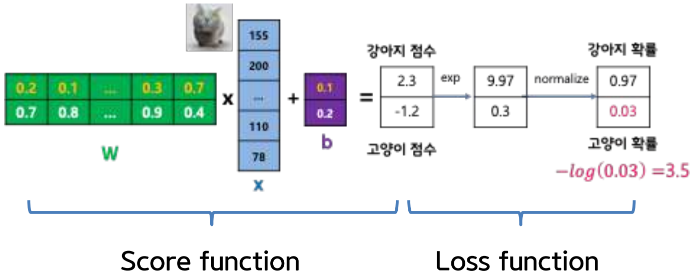
$$\begin{pmatrix} 0.2 \\ 0.1 \\ 0.3+\mathbf{0.0001} \\ \vdots \\ 0.9 \\ 0.4 \end{pmatrix}$$

Gradient dW

$$\begin{pmatrix} -2.5 \\ 0.6 \\ ? \\ \vdots \\ ? \\ ? \end{pmatrix}$$

Loss : $L($  $, w_1)$=1.25347

$L($  $, w_2+h)$ =1.25347



$L($, $W)$

Current W

$$\begin{pmatrix} 0.2 \\ 0.1 \\ 0.3 \\ \vdots \\ 0.9 \\ 0.4 \end{pmatrix}$$

Loss : $L($, $w_1) =$ 1.25347

w_2+h

$$\begin{pmatrix} 0.2 \\ 0.1 \\ 0.3 + \mathbf{0.0001} \\ \vdots \\ 0.9 \\ 0.4 \end{pmatrix}$$

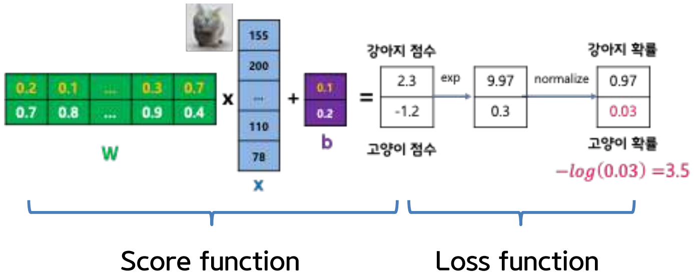
$L($, $w_2+h) =$ 1.25347

Gradient dW

$$\begin{pmatrix} -2.5 \\ 0.6 \\ 0 \\ \vdots \\ ? \\ ? \end{pmatrix}$$

(1.25347-1.25347)/0.0001 = 0

$$\lim_{h \to 0} \frac{L(\ , w_1+h) - L(\ , w_1)}{h}$$



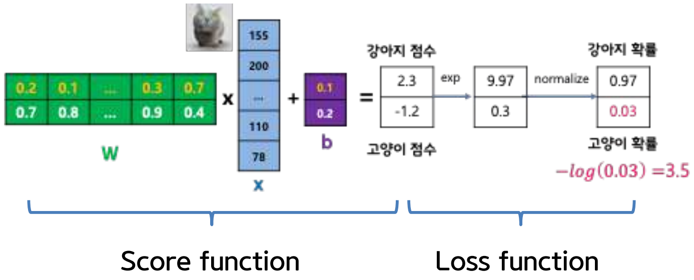
$L($  $, W)$

Current W

$$\begin{pmatrix} 0.2 \\ 0.1 \\ 0.3 \\ \vdots \\ 0.9 \\ 0.4 \end{pmatrix}$$

Gradient dW

$$\begin{pmatrix} -2.5 \\ 0.6 \\ 0 \\ \vdots \\ -0.2 \\ 0.5 \end{pmatrix}$$



$L($$, W)$

Current W

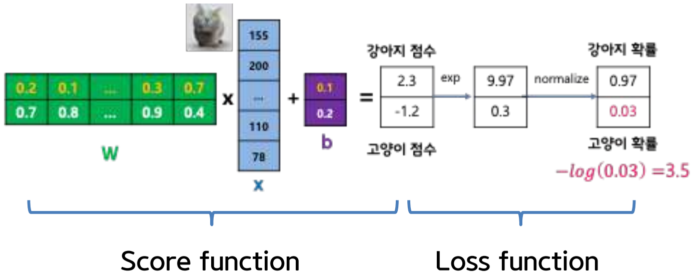
$$\begin{pmatrix} 0.2 \\ 0.1 \\ 0.3 \\ \vdots \\ 0.9 \\ 0.4 \end{pmatrix}$$

Update

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

Gradient dW

$$\begin{pmatrix} -2.5 \\ 0.6 \\ 0 \\ \vdots \\ -0.2 \\ 0.5 \end{pmatrix}$$

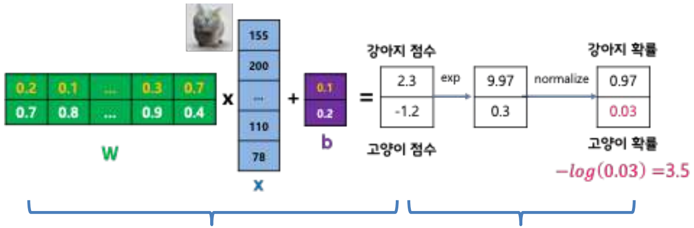


$L($  $, W)$

New W		Current W			Gradient dW

$$\begin{pmatrix} 0.225 \\ 0.094 \\ 0.3 \\ \vdots \\ 0.902 \\ 0.395 \end{pmatrix} = \begin{pmatrix} 0.2 \\ 0.1 \\ 0.3 \\ \vdots \\ 0.9 \\ 0.4 \end{pmatrix} - 0.01 \times \begin{pmatrix} -2.5 \\ 0.6 \\ 0 \\ \vdots \\ -0.2 \\ 0.5 \end{pmatrix}$$

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



Score function Loss function

$L($  $, W)$

New W **Current W** **Gradient dW**

$$\begin{pmatrix} 0.225 \\ 0.094 \\ 0.3 \\ \vdots \\ 0.902 \\ 0.395 \end{pmatrix} = \begin{pmatrix} 0.2 \\ 0.1 \\ 0.3 \\ \vdots \\ 0.9 \\ 0.4 \end{pmatrix} - 0.01 \times \begin{pmatrix} -2.5 \\ 0.6 \\ 0 \\ \vdots \\ -0.2 \\ 0.5 \end{pmatrix}$$

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

미분을 이렇게 하는 것의 구현은 쉽지만 시간이 오래걸립니다

또한 부정확합니다

↓

오차역전파법 (Backpropagation)

$$x^3$$

미분하면 ?

일단 미분공식을 이용해서 쉽게 미분할 겁니다

$$3x^2$$

$$x^n \rightarrow nx^{n-1}$$

오차역전파 법으로 Loss를 W로 미분하기

$L($  $, W)$

0.2	0.1	…	0.3	0.7
0.7	0.8	…	0.9	0.4

W

x

155
200
…
110
78

x

+

0.1
0.2

b

=

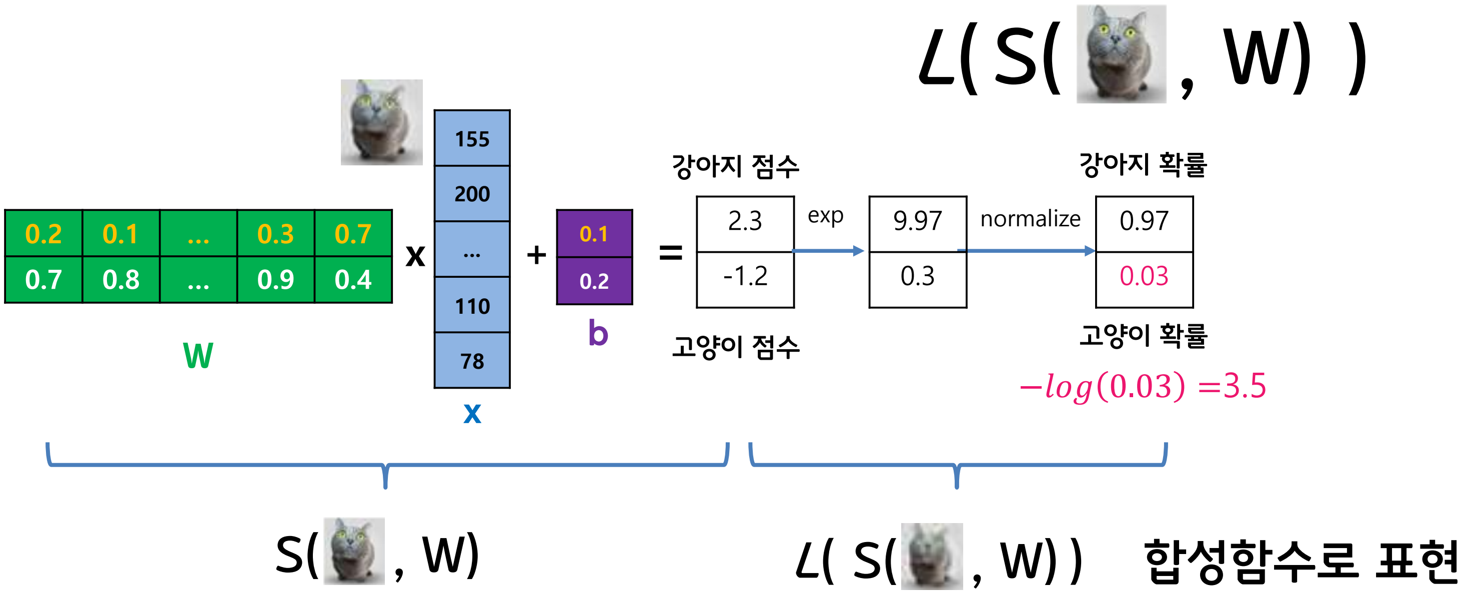
	점수		exp		normalize		확률
강아지	2.3	→		9.97	→	0.97	강아지
고양이	-1.2			0.3		0.03	고양이

$-log(0.03) = 3.5$

Score function

Loss function

오차역전파 법으로 Loss를 W로 미분하기



오차역전파 법으로 Loss를 W로 미분하기

Loss를 Score로 미분

Numerator (left): $\partial\, L(S($  $, W))$

Denominator (left): $\partial\, W$

$=$

Numerator (right): $\partial\, L(S($  $, W))$

Denominator (right): $\partial\, S($  $, W)$

합성합수의 미분법

오차역전파 법으로 Loss를 W로 미분하기

Loss를 Score로 미분 Score를 W로 미분

$\partial L(S($  $, W))$ / ∂W = $\partial L(S($  $, W))$ / $\partial S($  $, W)$ · $\partial S($  $, W)$ / ∂W

합성합수의 미분법

곱으로 표현됨

Optimization

$$W = W - \eta \frac{\partial L(S(\quad , W))}{\partial W}$$

Optimization

$$\underset{\text{New W}}{\begin{pmatrix} 0.225 \\ 0.094 \\ 0.3 \\ \vdots \\ 0.902 \\ 0.395 \end{pmatrix}} = \underset{\text{Current W}}{\begin{pmatrix} 0.2 \\ 0.1 \\ 0.3 \\ \vdots \\ 0.9 \\ 0.4 \end{pmatrix}} - \eta \times \underset{\text{Gradient dW}}{\begin{pmatrix} -2.5 \\ 0.6 \\ 0 \\ \vdots \\ -0.2 \\ 0.5 \end{pmatrix}}$$

TensorFlow로 미분을 계산해 봅시다

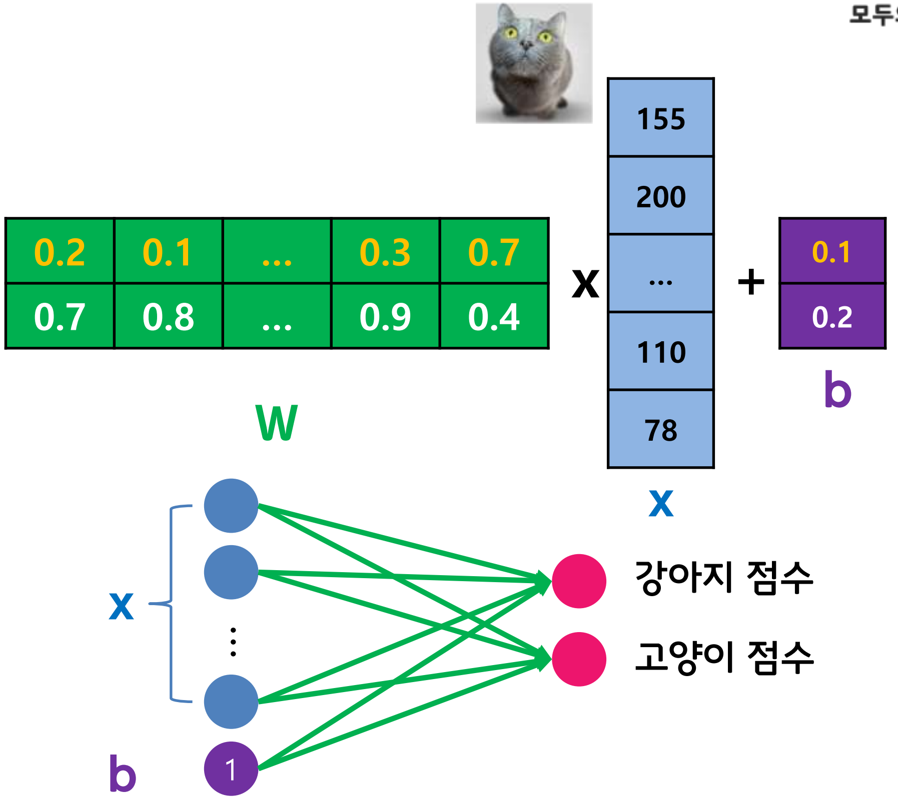
Gradient Tape

1_automatic_differentiation.ipynb

강아지와 고양이 분류해보기

모두의연구소

- 분류기의 구성
  - Score function
  - Loss function
  - Optimization



고양이가 입력이면 고양이 점수가 높아야 함

강아지와 고양이 분류해보기

- 분류기의 구성
  - Score function
  - Loss function
  - Optimization

$$\underbrace{\begin{bmatrix} 0.2 & 0.1 & \cdots & 0.3 & 0.7 \\ 0.7 & 0.8 & \cdots & 0.9 & 0.4 \end{bmatrix}}_{W} \times \underbrace{\begin{bmatrix} 155 \\ 200 \\ \cdots \\ 110 \\ 78 \end{bmatrix}}_{x} + \underbrace{\begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}}_{b} =$$

Cross-entropy loss (Softmax)

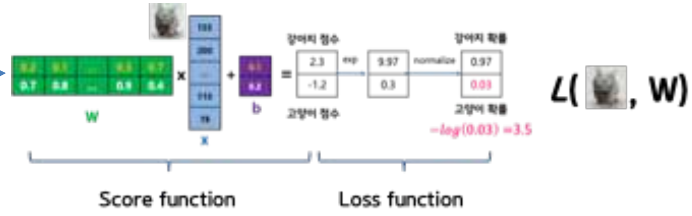
강아지 점수	2.3	exp →	9.97	normalize →	0.97	강아지 확률
고양이 점수	-1.2		0.3		0.03	고양이 확률

$-log(0.03) = 3.5$

현재의 분류기는 3.5만큼 안 좋음. 이 loss 값을 줄이는게 목표

강아지와 고양이 분류해보기

- 분류기의 구성
  - Score function
  - Loss function
  - Optimization



$$\underbrace{\begin{pmatrix} 0.225 \\ 0.094 \\ 0.3 \\ \vdots \\ 0.902 \\ 0.395 \end{pmatrix}}_{\text{New W}} = \underbrace{\begin{pmatrix} 0.2 \\ 0.1 \\ 0.3 \\ \vdots \\ 0.9 \\ 0.4 \end{pmatrix}}_{\text{Current W}} - 0.01 \times \underbrace{\begin{pmatrix} -2.5 \\ 0.6 \\ 0 \\ \vdots \\ -0.2 \\ 0.5 \end{pmatrix}}_{\text{Gradient dW}}$$

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

정리

- 분류기의 구성
  - Score function : Wx+b
  - Loss function : Score Function의 잘못 분류된 정도를 측정
  - Optimization : Loss function의 값을 줄이는 방향으로 파라미터 업데이트

$$\mathbf{w} = \mathbf{w} - \eta \frac{\partial L}{\partial \mathbf{w}}$$

모두의연구소

박 은 수 Research Director

E-mail : es.park@modulabs.co.kr