

Deep Learning Programming

Lecture 2.2: Linear Classifier

Sangryul Jeon

School of Computer Science and Engineering

srjeonn@pusan.ac.kr

1.

Linear Classifier

Linear Classifier에 대한 정의와 해석

1.1 매개변수적 접근

Linear Classifier

모든 훈련 예제를 외우는 대신

입력(이미지 x)을 레이블 점수(클래스 y)에 매핑하는 함수 f 를 생각해 봅시다.

이미지 x



레이블 y

airplane
automobile
bird
cat
deer
dog
frog
horse
ship
truck

$$f(\mathbf{x})$$

1.1 매개변수적 접근: Linear Classifier

Linear Classifier

Q : 이 Function f 는 어떤 형태여야 할까요?

A : 여러 가지 방법이 있을 수 있지만 간단한 선형 함수부터 시작해 보겠습니다!

→ 입력 픽셀의 가중치 합계

이미지 x



=

```
[[105 112 108 111 104 99 106 99 96 103 112 119 104 97 93 87]
 [ 91 98 102 106 104 79 98 103 99 105 123 136 110 105 94 85]
 [ 76 85 90 105 128 105 87 96 95 99 115 112 106 103 99 85]
 [ 99 81 93 120 131 127 100 95 98 102 99 96 93 101 94]
 [106 91 61 64 69 91 88 85 101 107 109 98 75 84 96 95]
 [114 108 85 55 55 69 64 54 64 87 112 129 98 74 84 91]
 [133 137 147 103 65 81 80 65 52 54 74 84 102 93 85 82]
 [128 137 144 140 109 95 86 70 62 65 63 63 60 73 86 101]
 [125 133 148 137 119 121 117 94 65 79 80 65 54 64 72 98]
 [127 125 131 147 133 127 126 131 111 96 89 75 61 64 72 84]
 [115 114 109 123 150 148 131 118 113 109 100 92 74 65 72 78]
 [ 89 93 90 97 108 147 131 118 113 114 113 109 106 95 77 80]
 [ 63 77 86 81 77 79 102 123 117 115 117 125 125 130 115 87]
 [ 62 65 82 89 78 71 80 101 124 126 119 101 107 114 131 119]
 [ 63 65 75 88 89 71 62 81 120 138 135 105 81 98 110 118]
 [ 87 65 71 87 106 95 69 45 76 130 126 107 92 94 105 112]
 [118 97 82 86 117 123 116 66 41 51 95 93 89 95 102 107]
 [164 146 112 80 82 120 124 104 76 48 45 66 88 101 102 109]
 [157 170 157 120 93 86 114 132 112 97 69 55 70 82 99 94]
 [130 128 134 161 139 100 109 118 121 134 114 87 65 53 69 86]
 [128 112 96 117 150 144 120 115 104 107 102 93 87 81 72 79]
 [123 107 96 86 83 112 153 149 122 109 104 75 80 107 112 99]
 [122 121 102 80 82 86 94 117 145 148 153 102 58 78 92 107]
 [122 164 148 103 71 56 78 83 93 103 119 139 102 61 69 84]]
```

각 픽셀에 해당하는
매개변수 W 의 가중치들

```
[[0.7 0.7 0. 0.3 0.7 0.6 0.4 0.9 0. 0.6 0. 0.4 0.8 0. 0. 0.1 0. 0.5]
 [0.5 0.8 0.6 0.7 0.2 0.3 0.5 0.9 0.9 0.6 0.2 0.2 0.6 0.3 0. 0.5 0.1 0.4]
 [0.2 0.3 0. 0.7 0.4 0.7 0. 0.4 0.6 0.6 0.2 0.3 0.9 0. 0.2 0.2 0.6 0.3]
 [0.2 0.4 0. 0.1 0. 0.3 0.9 0.6 0.6 0.1 0.5 0.9 0.3 0. 0.8 0.1 0.9]
 [0.8 0.2 0.3 0.5 0.4 0.9 0.7 0.3 0.4 0.3 0.4 0.7 0.4 0.7 0. 0.6 0.3 0.8]
 [0.4 0.4 0.4 0.5 0.1 0.8 0.5 0.8 0.1 0.9 0.3 0.1 0.2 0.7 0. 0.9 0.7 0.5]
 [0.9 0.7 0.7 0.3 0.7 0.3 0.1 0.3 0.1 0.1 0.8 0. 0.4 0.6 0.1 0.7 0.2 0.3 0.4]
 [0.5 0.2 0.8 0.7 0.3 0.9 0.4 0.8 0.1 0.2 0.5 0.6 0.1 0.1 0.8 0.5 0.8 0.3]
 [0.6 0. 0. 0.8 0. 0.2 0. 0.3 0.1 0.9 0. 0.9 0. 0.9 0.2 0.4 0.1 0. ]
 [0.8 0.1 0.5 0.1 0.5 0.2 0.3 0.9 0.5 0.8 0.2 0.4 0.1 0.1 0.5 0.9 0.4 0.2]
 [0.6 0.4 0.6 0.5 0.2 0.5 0.6 0.8 0. 0.2 0.5 0.2 0.3 0.6 0.5 0.3 0.7]
 [0.6 0.6 0.8 0. 0.5 0.6 0.2 0.5 0.2 0.7 0.4 0.3 0.8 0.7 0.9 0.8 0.4 0.8]
 [0.6 0.2 0.5 0. 0.9 0.2 0.4 0.5 0.9 0.9 0.4 0.7 0.1 0.7 0. 0.8 0.6 0. ]
 [0.7 0.8 0.2 0.4 0.5 0.9 0.1 0.7 0. 0.8 0.2 0.2 0.2 0.2 0.7 0. 0.2 0.5]
 [0.6 0.4 0.2 0.1 0.5 0.9 0.6 0.9 0.8 0.9 0.4 0.5 0.5 0.9 0.7 0.5 0.4 0.7]
 [0.3 0.7 0. 0.3 0.5 0.9 0.3 0. 0.8 0.2 0.6 0.4 0.8 0.1 0.9 0.3 0.2 0.6]
 [0.1 0.7 0.3 0. 0.9 0.1 0.2 0.3 0.6 0.5 0.4 0.3 0.5 0.1 0. 0.2 0.7 0.8]
 [0.2 0.8 0.5 0.1 0.8 0.4 0. 0.7 0.5 0.1 0.4 0.8 0.8 0.4 0.6 0.1 0.5 0.6]
 [0.4 0.5 0.4 0.8 0.9 0.8 0.7 0.1 0.3 0.5 0.3 0.7 0.9 0.7 0.3 0.7 0.5 0.6]
 [0.7 0.7 0.1 0.9 0.3 0. 0.7 0.2 0.6 0.6 0.8 0. 0.2 0.3 0.1 0.2 0.8 0.8]
 [0.6 0. 0.3 0.5 0.2 0.8 0.4 0.7 0.4 0.5 0.6 0.5 0.6 0.8 0.4 0.4 0.8 0.8]
 [0.3 0.9 0.4 0.7 0.9 0.5 0.1 0.8 0.9 0.7 0.4 0.6 0.6 0.9 0.7 0.7 0. 0.4 0.6]
 [0.2 0.7 0.1 0.7 0.9 0.5 0. 0.3 0.9 0.2 0.6 0.6 0.6 0.5 0.8 0.6 0. 0.6]
 [0.7 0.3 0. 0. 0.2 0.9 0.7 0.6 0. 0.9 0.1 0.9 0.3 0.4 0.7 0.3 0.7 0.3]]]
```

1.1 매개변수적 접근: Linear Classifier

Linear Classifier

Q : 이 Function f 는 어떤 형태여야 할까요?

A : 여러 가지 방법이 있을 수 있지만 간단한 선형 함수부터 시작해 보겠습니다!

→ 입력 픽셀의 가중치 합계 : $W_{1,1}x_{1,1} + W_{1,2}x_{1,2} + \dots + W_{M,N}x_{M,N}$

이미지 x

```
[[105 112 108 111 104 99 106 99 96 103 112 119 104 97 93 87]
 [ 91 98 102 106 104 79 98 103 99 105 123 136 110 105 94 85]
 [ 76 85 90 105 128 105 87 96 95 99 115 112 106 103 99 85]
 [ 99 81 81 93 120 131 127 100 95 98 102 99 96 93 101 94]
 [106 91 61 64 69 91 88 85 101 107 109 98 75 84 96 95]
 [114 108 85 55 55 69 64 54 64 87 112 129 98 74 84 91]
 [133 137 147 103 65 81 80 65 52 54 74 84 102 93 85 82]
 [128 137 144 140 109 95 86 70 62 65 63 63 60 73 86 101]
 [125 133 148 137 119 121 117 94 65 79 80 65 54 64 72 98]
 [127 125 131 147 133 127 126 131 111 96 89 75 61 64 72 84]
 [115 114 109 123 150 148 131 118 113 109 100 92 74 65 72 78]
 [ 89 93 90 97 108 147 131 118 113 114 113 109 106 95 77 80]
 [ 63 77 86 81 77 79 102 123 117 115 117 125 125 130 115 87]
 [ 62 65 82 89 78 71 80 101 124 126 119 101 107 114 131 119]
 [ 63 65 75 88 89 71 62 81 120 138 135 105 81 98 110 118]
 [ 87 65 71 87 106 95 69 45 76 130 126 107 92 94 105 112]
 [118 97 82 86 117 123 116 66 41 51 95 93 89 95 102 107]
 [164 146 112 80 82 120 124 104 76 48 45 66 88 101 102 109]
 [157 170 157 120 93 86 114 132 112 97 69 55 70 82 99 94]
 [130 128 134 161 139 100 109 118 121 134 114 87 65 53 69 86]
 [128 112 96 117 150 144 120 115 104 107 102 93 87 81 72 79]
 [123 107 96 86 83 112 153 149 122 109 104 75 80 107 112 99]
 [122 121 102 80 82 86 94 117 145 148 153 102 58 78 92 107]
 [122 164 148 103 71 56 78 83 93 103 119 139 102 61 69 84]]
```

가중치 혹은 파라미터 W

```
[[0.7 0.7 0. 0.3 0.7 0.6 0.4 0.9 0. 0.6 0. 0.4 0.8 0. 0. 0.1 0. 0.5]
 [0.5 0.8 0.6 0.7 0.2 0.3 0.5 0.9 0.9 0.6 0.2 0.2 0.6 0.3 0. 0.5 0.1 0.4]
 [0.2 0.3 0. 0.7 0.4 0.7 0. 0.4 0.6 0.6 0.2 0.3 0.9 0. 0.2 0.2 0.6 0.3]
 [0.2 0.4 0. 0.1 0. 0.3 0.9 0.9 0.6 0.6 0.1 0.5 0.9 0.3 0. 0.8 0.1 0.9]
 [0.8 0.2 0.3 0.5 0.4 0.9 0.7 0.3 0.4 0.3 0.4 0.7 0.4 0.7 0. 0.6 0.3 0.8]
 [0.4 0.4 0.4 0.5 0.1 0.8 0.5 0.8 0.1 0.9 0.3 0.1 0.2 0.7 0. 0.9 0.7 0.5]
 [0.9 0.7 0.7 0.3 0.7 0.3 0.1 0.3 0.1 0.8 0. 0.4 0.6 0.1 0.7 0.2 0.3 0.4]
 [0.5 0.2 0.8 0.7 0.3 0.9 0.4 0.8 0.1 0.2 0.5 0.6 0.1 0.1 0.8 0.5 0.8 0.3]
 [0.6 0. 0. 0.8 0. 0.2 0. 0.3 0.1 0.9 0. 0.9 0. 0.9 0.2 0.4 0.1 0. ]
 [0.8 0.1 0.5 0.1 0.5 0.2 0.3 0.9 0.5 0.8 0.2 0.4 0.1 0.1 0.5 0.9 0.4 0.2]
 [0. 0.6 0.4 0.6 0.5 0.2 0.5 0.6 0.8 0. 0.2 0.5 0.2 0.3 0.6 0.5 0.3 0.7]
 [0.6 0.6 0.8 0. 0.5 0.6 0.2 0.5 0.2 0.7 0.4 0.3 0.8 0.7 0.9 0.8 0.4 0.8]
 [0.6 0.2 0.5 0. 0.9 0.2 0.4 0.5 0.9 0.9 0.4 0.7 0.1 0.7 0. 0.8 0.6 0. ]
 [0.7 0.8 0.2 0.4 0.5 0.9 0.1 0.7 0. 0.8 0.2 0.2 0.2 0.2 0.7 0. 0.2 0.5]
 [0.6 0.4 0.2 0.1 0.5 0.9 0.6 0.9 0.8 0.9 0.4 0.5 0.5 0.9 0.7 0.5 0.4 0.7]
 [0.3 0.7 0. 0.3 0.5 0.9 0.3 0. 0.8 0.2 0.6 0.4 0.8 0.1 0.9 0.3 0.2 0.6]
 [0.1 0.7 0.3 0. 0.9 0.1 0.2 0.3 0.6 0.5 0.4 0.3 0.5 0.1 0. 0.2 0.7 0.8]
 [0.2 0.8 0.5 0.1 0.8 0.4 0. 0.7 0.5 0.1 0.4 0.8 0.8 0.4 0.6 0.1 0.5 0.6]
 [0.4 0.5 0.4 0.8 0.9 0.8 0.7 0.1 0.3 0.5 0.3 0.7 0.9 0.7 0.3 0.7 0.5 0.6]
 [0.7 0.7 0.1 0.9 0.3 0. 0.7 0.2 0.6 0.6 0.8 0. 0.2 0.3 0.1 0.2 0.8 0.8]
 [0.6 0. 0.3 0.5 0.2 0.8 0.4 0.7 0.4 0.5 0.6 0.5 0.6 0.8 0.4 0.4 0.8 0.8]
 [0.3 0.9 0.4 0.7 0.9 0.5 0.1 0.8 0.9 0.7 0.4 0.6 0.9 0.7 0.7 0. 0.4 0.6]
 [0.2 0.7 0.1 0.7 0.9 0.5 0. 0.3 0.9 0.2 0.6 0.6 0.5 0.8 0.6 0. 0.6]
 [0.7 0.3 0. 0. 0.2 0.9 0.7 0.6 0. 0.9 0.1 0.9 0.3 0.4 0.7 0.3 0.7 0.3]]]
```

클래스 airplane에 해당하는 점수

레이블 y

airplane

automobile

bird

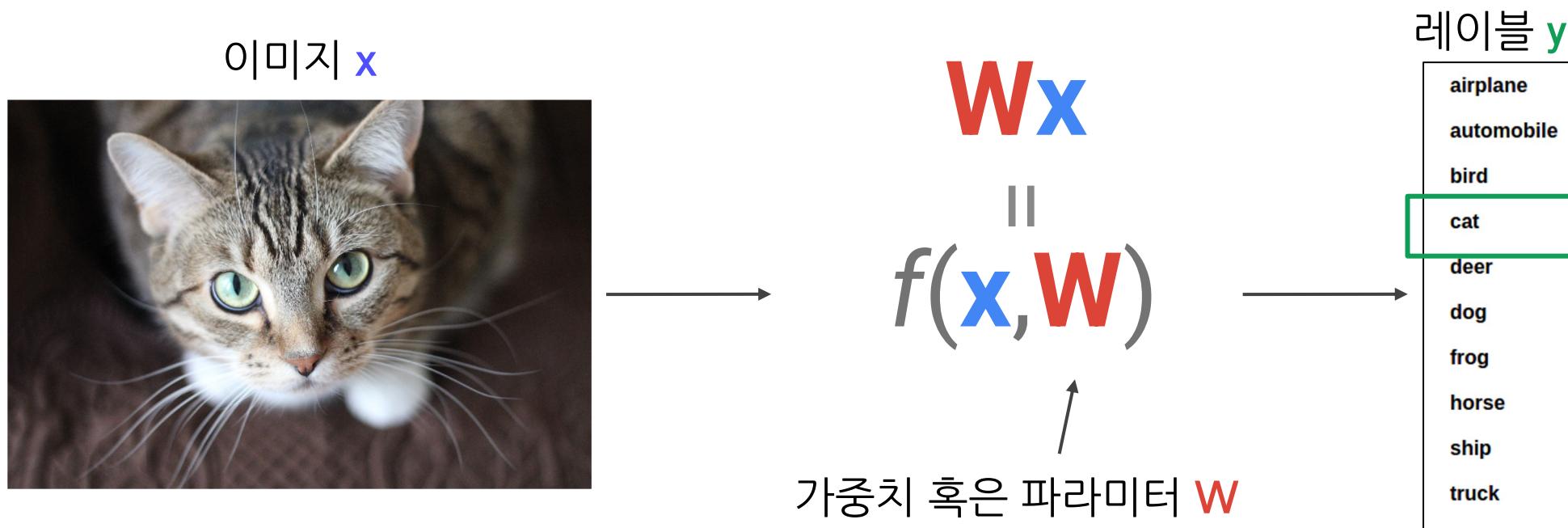
cat

deer

1.1 매개변수적 접근: Linear Classifier

Linear Classifier

각 클래스가 올바른 이미지에 대해 가장 높은 점수를 받도록 W 의 값을 결정해야 합니다.
이 예제에서는 'Cat' 클래스의 점수가 가장 높아야 합니다.



1.1 매개변수적 접근: Linear Classifier

Linear Classifier

각 변수의 차원을 확인해 봅시다:

이미지 x



$32 \times 32 \times 3$
(=3072개)

$$f(x, W)$$

10×1

각 클래스별로

10개의 독립적인 classifier f

레이블 y

airplane
automobile
bird
cat
deer
dog
frog
horse
ship
truck

1.1 매개변수적 접근: Linear Classifier

Linear Classifier

각 변수의 차원을 확인해 봅시다:

이미지 x



$32 \times 32 \times 3$
 $(=3072\text{개})$

가중치 혹은 파라미터 W

3072×1

$$f(\mathbf{x}, \mathbf{W}) = \mathbf{W}\mathbf{x}$$

10×1 10×3072

각 클래스별로

10개의 독립적인 classifier f

레이블 y

airplane
automobile
bird
cat
deer
dog
frog
horse
ship
truck

1.1 매개변수적 접근: Linear Classifier

Linear Classifier

각 변수의 차원을 확인해 봅시다:

이미지 x



$32 \times 32 \times 3$
(=3072개)

가중치 혹은 파라미터 W

3072×1

$$f(\mathbf{x}, \mathbf{W}) = \mathbf{Wx} + \mathbf{b}$$

10×1 10×3072 10×1

각 클래스별로

10개의 독립적인 classifier f

편향 b :

데이터 x 와 상호 작용하지
않고 출력에 영향을 미칩니다.

레이블 y

airplane
automobile
bird
cat
deer
dog
frog
horse
ship
truck

1.1 매개변수적 접근: Linear Classifier

Linear Classifier

$$f(\mathbf{x}, \mathbf{W}) = \mathbf{W}\mathbf{x} + \mathbf{b} = [\mathbf{w} \ \mathbf{b}] \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}$$

10 × 1 10 × 3072 3072 × 1
10 × 3073 3073 × 1

$\mathbf{W}' \quad \mathbf{x}'$

$$f(\mathbf{x}, \mathbf{W}) = \mathbf{W}'\mathbf{x}'$$

1.1 매개변수적 접근: Linear Classifier

Linear Classifier

우리의 최종적 Linear Classifier 모델은 다음과 같습니다 :

$$f(\mathbf{x}, \mathbf{W}) = \mathbf{W}\mathbf{x}$$

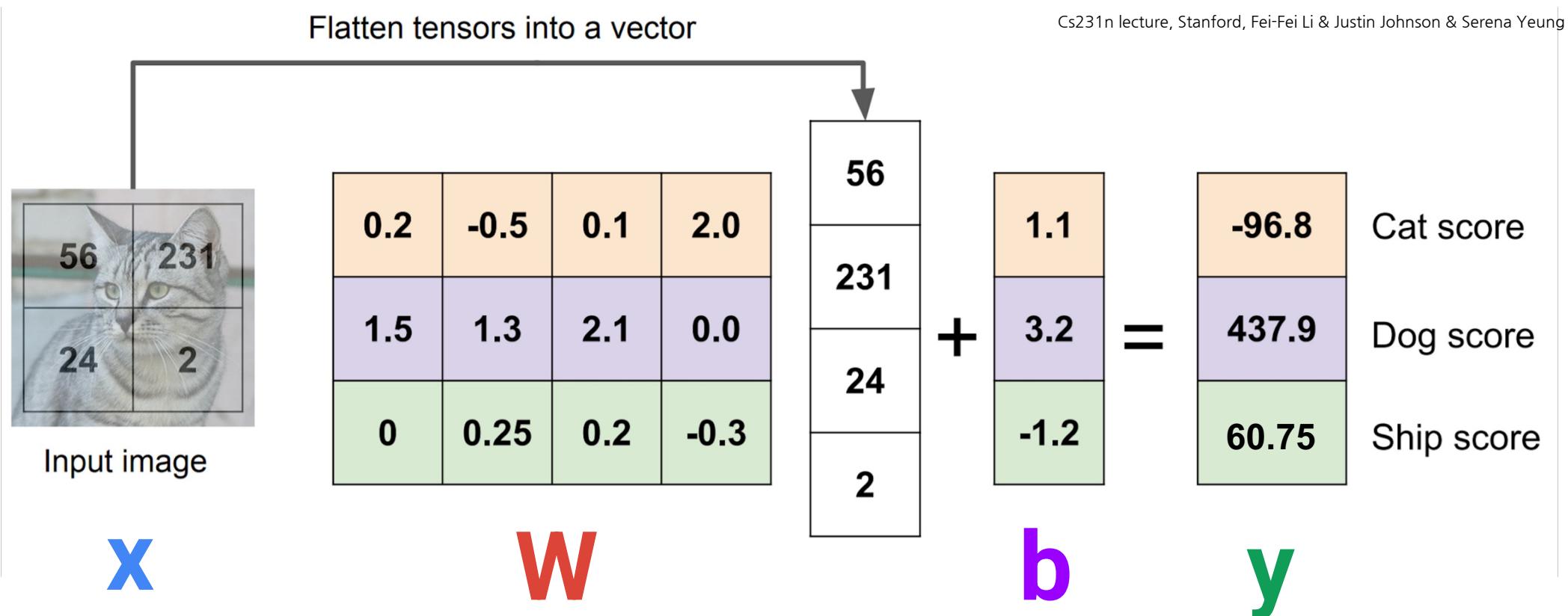
매개변수적 모델의 장점 :

- 학습이 완료되면 가중치 \mathbf{W} 만을 필요로 합니다. 방대한 학습 데이터 셋을 저장할 필요가 없습니다.
→ 공간 효율성이 높습니다.
- 테스트 시 단일 행렬-벡터 곱($\mathbf{W}\mathbf{x}$)으로 예제를 평가할 수 있습니다.
→ 모든 훈련 데이터와 비교하는 것보다 훨씬 빠릅니다.

1.2 Linear Classifier의 예시

Linear Classifier

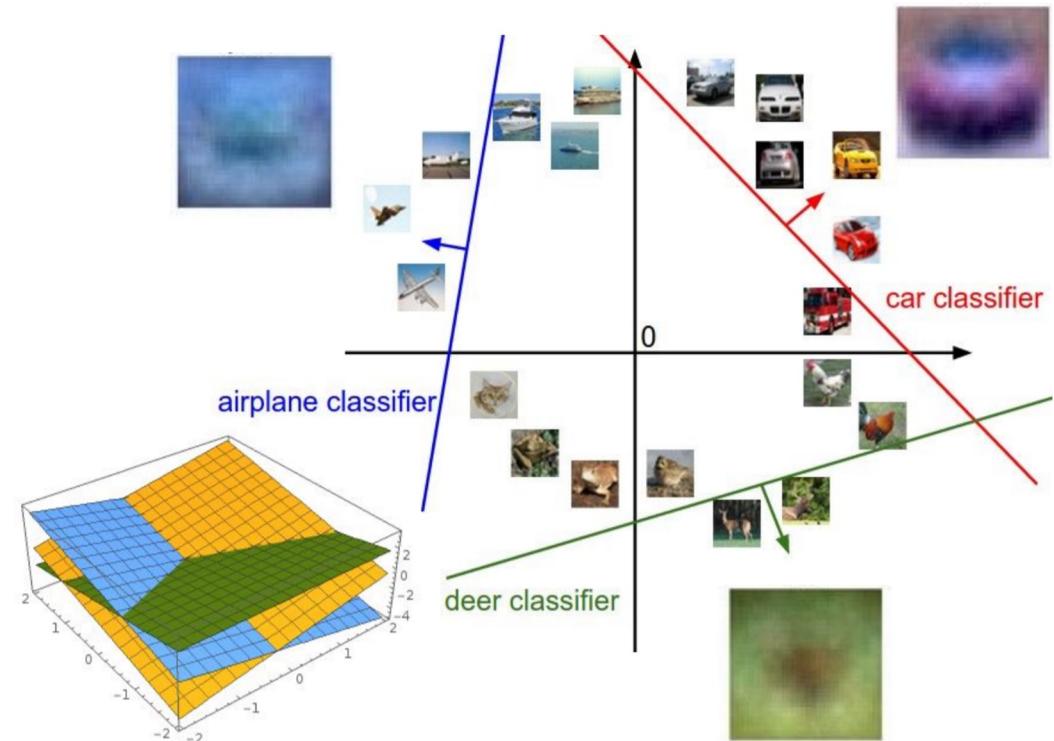
4 픽셀과 3 클래스를 가진 이미지를 이용한 예시(cat / dog / ship) :



1.3 Linear Classifier 해석하기 : 기하학적 관점

Linear Classifier

- 각 선형 바운더리는 **W**의 해당 행에서 나옵니다.
- W**의 값이 변경되면 해당 decision 바운더리가 회전합니다.
- b**가 변경되면 해당 decision 바운더리가 위/아래로 이동합니다.

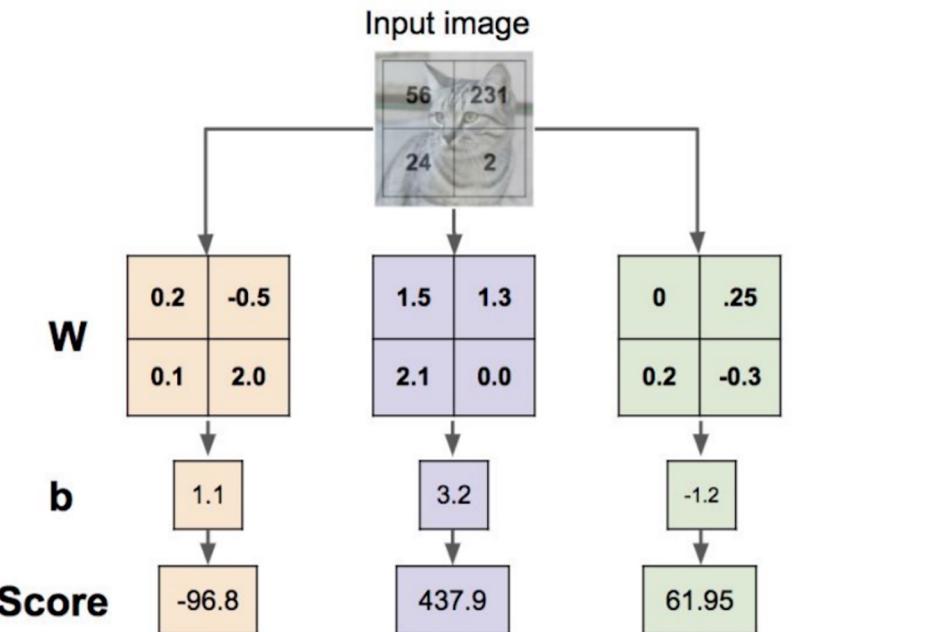


1.3 Linear Classifier 해석하기 : 시각적 관점

Linear Classifier

Alex Krizhevsky, "Learning Multiple Layers of Features from Tiny Images", Technical Report, 2009.

Cs231n lecture, Stanford, Fei-Fei Li & Justin Johnson & Serena Yeung



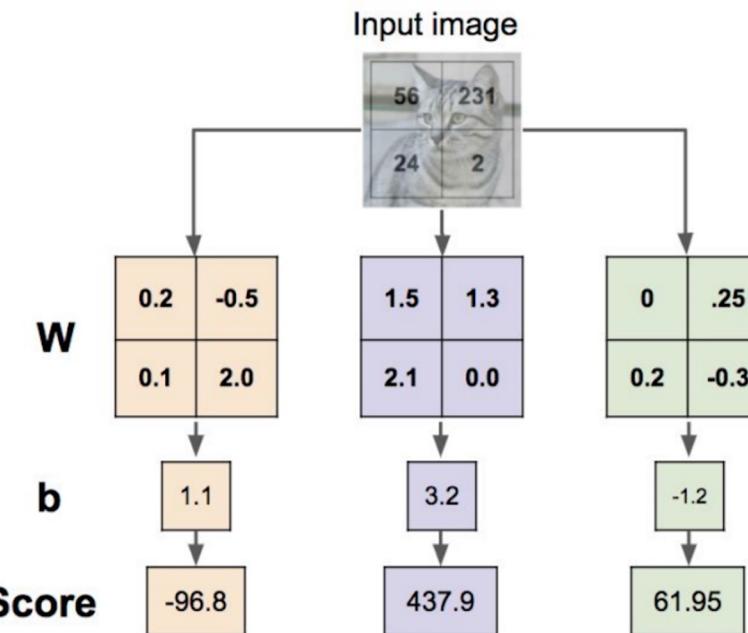
1.3 Linear Classifier 해석하기 : 시각적 관점

Linear Classifier

Linear Classifier가 하는 일은 :

- 훈련 시: 훈련 데이터에서 템플릿 학습
- 테스트 시: 새로운 예제로 템플릿 매칭

Cs231n lecture, Stanford, Fei-Fei Li & Justin Johnson & Serena Yeung



2.

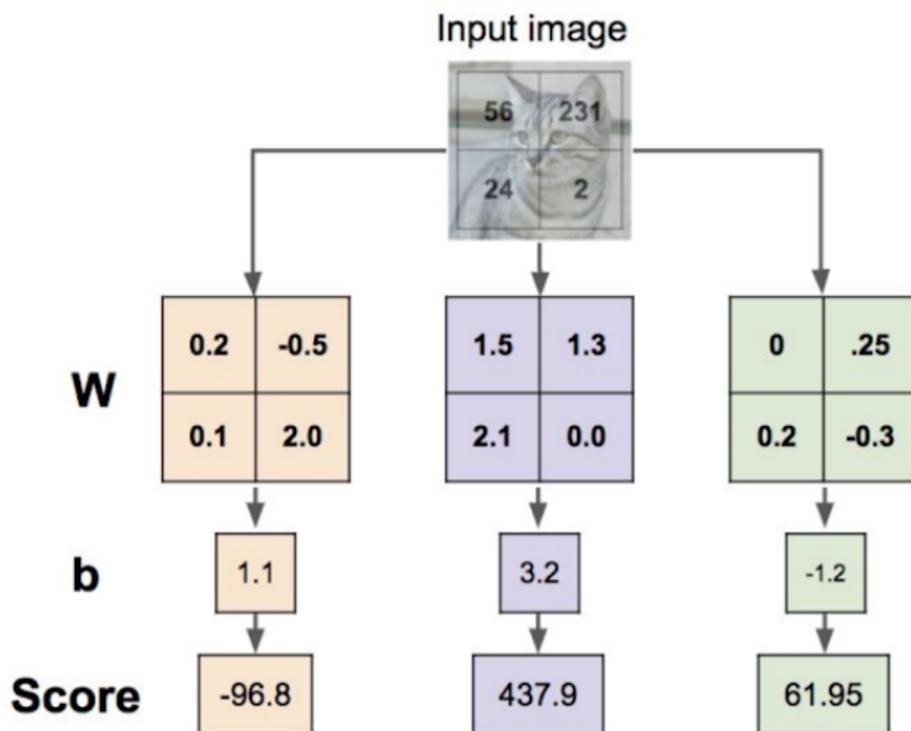
Softmax Classifier

Softmax Classifier의 필요성과 사용방법

2.1 Linear Classifier의 한계

Softmax Classifier

‘점수’란 어떠한 의미를 가지나요?



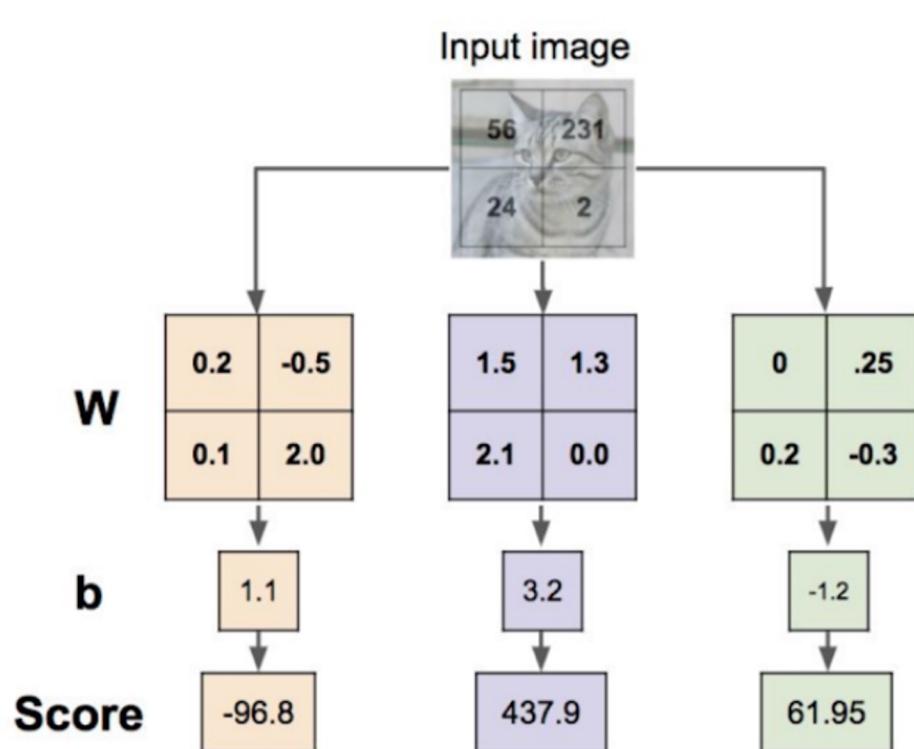
- **제한이 없음**: 임의로 커질 수 있습니다.
- **해석하기 어려움**: 437.9의 의미는 무엇인가요?
얼마나 좋을까요?

0과 1 사이의 경계 점수를 얻어서
확률로 해석할 수 있다면 더 좋을 것입니다.

2.1 Linear Classifier의 한계

Softmax Classifier

‘점수’란 어떠한 의미를 가지나요?



이미지가 각 클래스에 속할 확률에 대해 정의해 보겠습니다.

- 2개의 클래스가 있다고 가정할 때 :
 $s_1 > s_2$ 라면: 해당 이미지가 클래스2보다 클래스1에 있을 확률이 높습니다.
- 클래스 간 갭($s_1 - s_2$)이 더 클수록, $x \in c_1$ 일 확률이 더 높습니다.
- 반대의 경우도 마찬가지입니다.
 $s_2 - s_1$ 이 클수록 $x \in c_2$ 일 가능성이 높습니다.

2.2 Sigmoid 함수

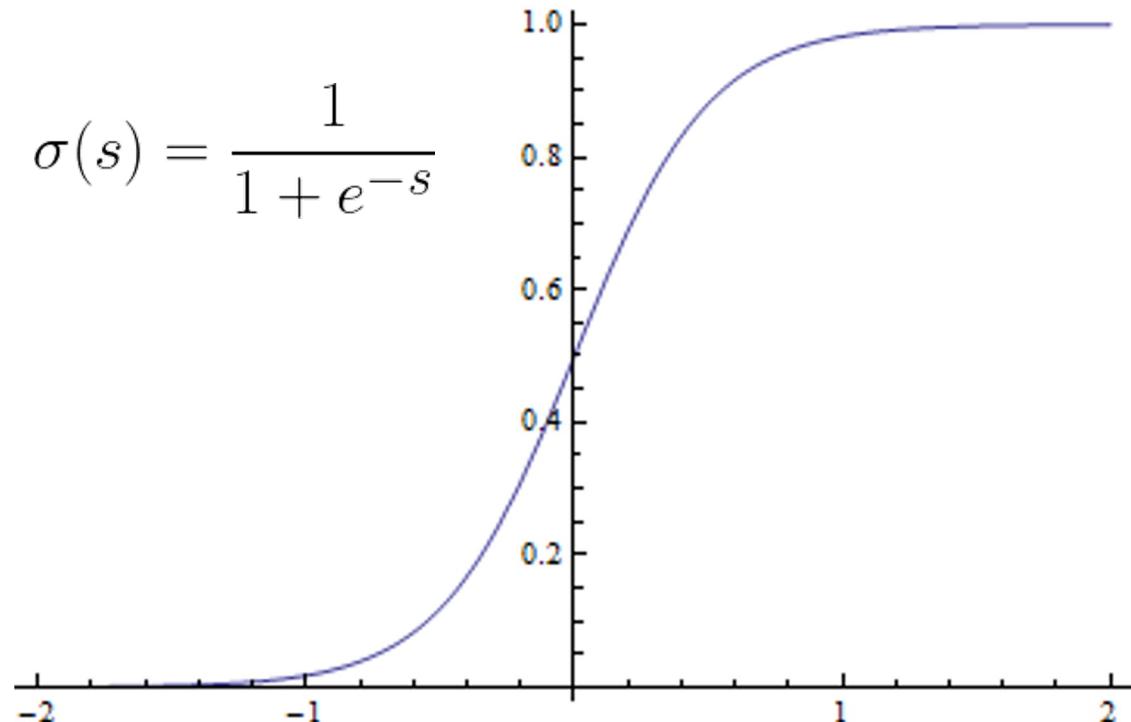
Softmax Classifier

점수의 차이가 $s (= s_1 - s_2)$ 라고 주어질 때, 우리는 다음과 같은 함수를 필요로 합니다.

$$p(y = c_1 | \mathbf{x}) = \frac{1}{1 + e^{-(s_1 - s_2)}}$$

$$p(y = c_2 | \mathbf{x}) = \frac{1}{1 + e^{-(s_2 - s_1)}}$$

$$\sigma(s) = \frac{1}{1 + e^{-s}}$$



2.3 Softmax Classifier

Softmax Classifier

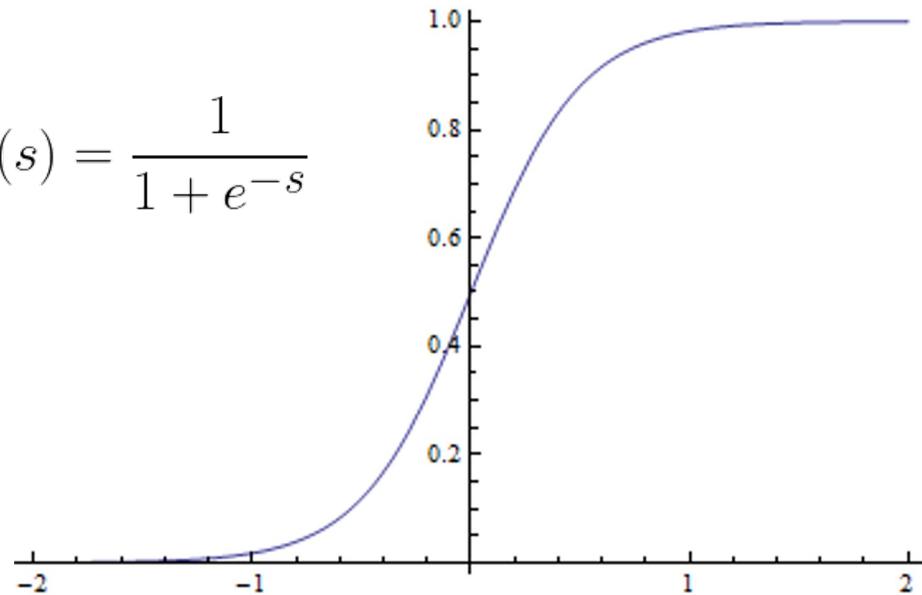
이 값이 정말 확률에 해당하는지 확인해봅시다.:

- 1) $0 \leq p(y=c_n|x) \leq 1$ (for $n = 1, 2$)
- 2) $p(y=c_1|x) + p(y=c_2|x) = 1$

$$p(y = c_1|x) = \frac{1}{1 + e^{-(s_1 - s_2)}}$$

$$p(y = c_2|x) = \frac{1}{1 + e^{-(s_2 - s_1)}}$$

$$\sigma(s) = \frac{1}{1 + e^{-s}}$$



클래스가 $n > 2$ 인 경우에 대해 일반화한다면:

https://en.wikipedia.org/w/index.php?title=Logistic_regression&oldid=755697139#As_a_log-linear_model

$$p(y = c_i|x) = \frac{e^{s_i}}{e^{s_1} + e^{s_2} + \dots + e^{s_n}} = \frac{e^{s_i}}{\sum_j e^{s_j}}$$

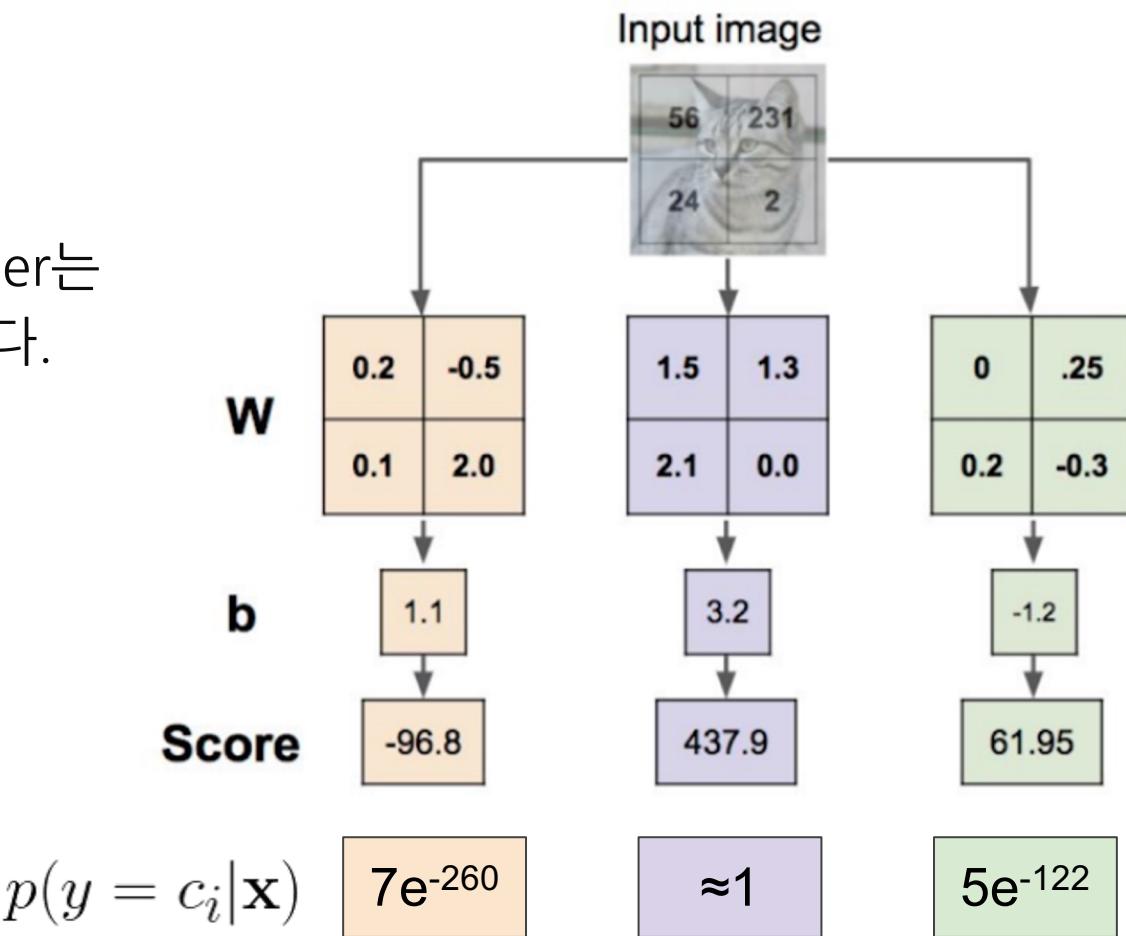
Softmax
Function

2.3 Softmax Classifier

Softmax Classifier

이 예시로 다시 돌아와서,

측정된 확률 값에 따라, 우리의 softmax classifier는 클래스 2(dog)를 약 100%의 확률로 선택합니다.



2.4 가중치는 어떻게 설정해야 할까요?

Softmax Classifier

아직 가장 중요한 문제에 대해 이야기하지 않았습니다.

W(매개변수들)의 값을 어떻게 정할 것인가?

머신 러닝은 데이터 기반의 접근입니다.

- 우리는 모델의 형식(e.g., $f(\mathbf{x}, \mathbf{W}) = \mathbf{W}\mathbf{x}$)을 디자인하고
- 랜덤하게 매개변수(\mathbf{W})의 값을 설정합니다.
- 그 후, 학습 데이터 \mathbf{x} 를 입력해 라벨 $\hat{\mathbf{y}}$ 을 예측합니다.
- 추정값 $\hat{\mathbf{y}}$ 을 기준값 라벨 \mathbf{y} 와 비교하여 현재값이 얼마나 좋은지/나쁜지를 비교합니다.
- 손실값에 따라 매개변수(\mathbf{W})를 업데이트하고,
- $\hat{\mathbf{y}} \approx \mathbf{y}$ 가 될 때까지 이 과정을 반복합니다.

최적화

손실 함수

3.

손실 함수

손실함수에 대한 정의와 종류 소개

3.1 손실 함수

손실 함수

손실함수는 해당 머신 러닝 모델이 얼마나 좋은지, 혹은 나쁜지를 정량화합니다.

- 추정값 \hat{y} 와 기준값 레이블 y 에 해당하는 함수: $\mathcal{L}(\hat{y}, y)$
- \hat{y} 과 y 가 어떻게 다른지에 따라 모델에 패널티를 주는 양수를 출력합니다.
 - $\hat{y} = y$ 인 경우 모델에 패널티를 주고 싶지 않으므로 손실은 (약) 0이 되어야 합니다.
 - $\hat{y} \approx y$ 인 경우 모델에 패널티를 주어 미세 조정할 수 있습니다.
 - \hat{y} 와 y 사이의 격차가 크면 크게 패널티를 주어야 합니다.

3.1 손실 함수: 차별적(Discriminative) 설정

손실 함수

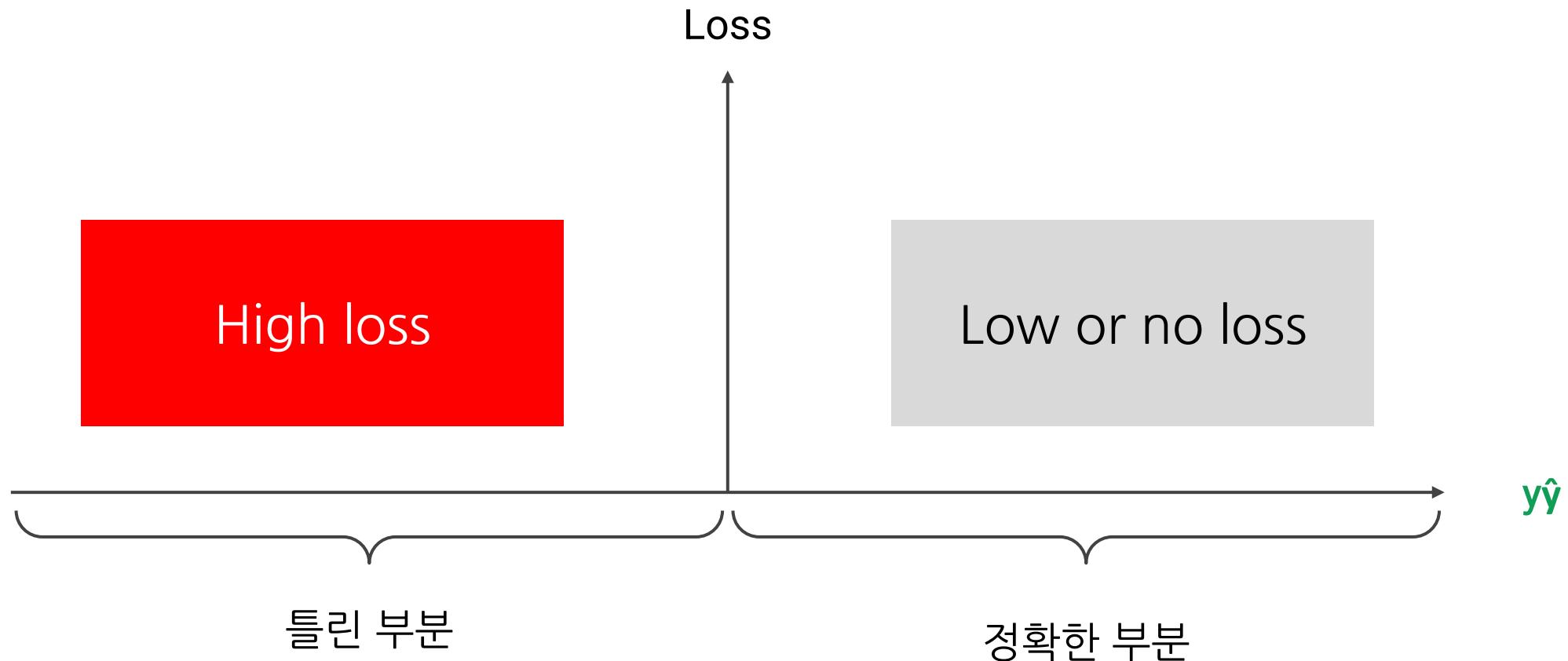
이진 분류의 경우, 기준값(ground truth)은 $y = \{+1, -1\}$ 입니다.

- 모델은 하나의 점수 $\hat{y} \in \mathbb{R}$ 를 예측합니다.
- \hat{y} 이 0보다 크면 포지티브 클래스로, 그렇지 않으면 네거티브 클래스로 분류합니다.
- **마진 기반 손실:**
 - 손실은 $y\hat{y}$ 에 따라 결정됩니다.
 - 부호가 같으면(즉, 분류가 정확하면) 손실이 작아지거나 0이 됩니다.
 - 부호가 다르면(잘못된 분류) 손실이 커집니다.

3.2 마진 기반 손실

손실 함수

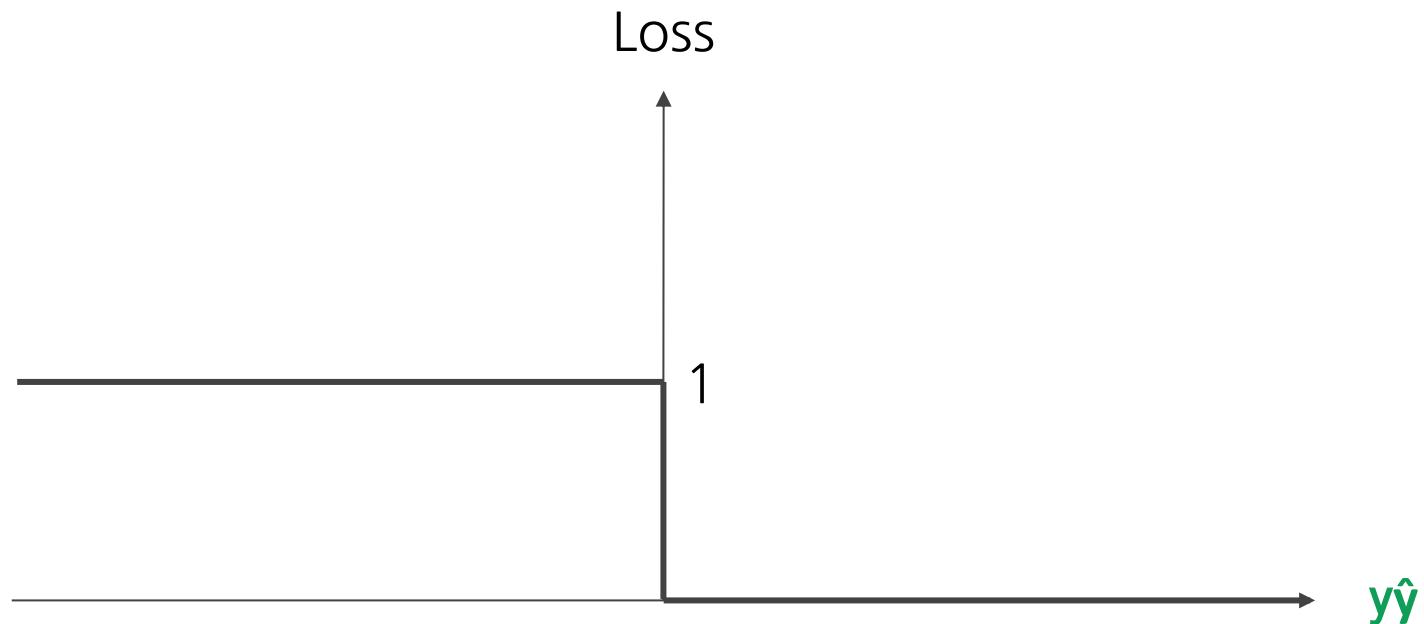
손실함수는 해당 머신 러닝 모델이 얼마나 좋은지, 혹은 나쁜지를 정량화합니다.



3.2 마진 기반 손실: 0/1 손실

손실 함수

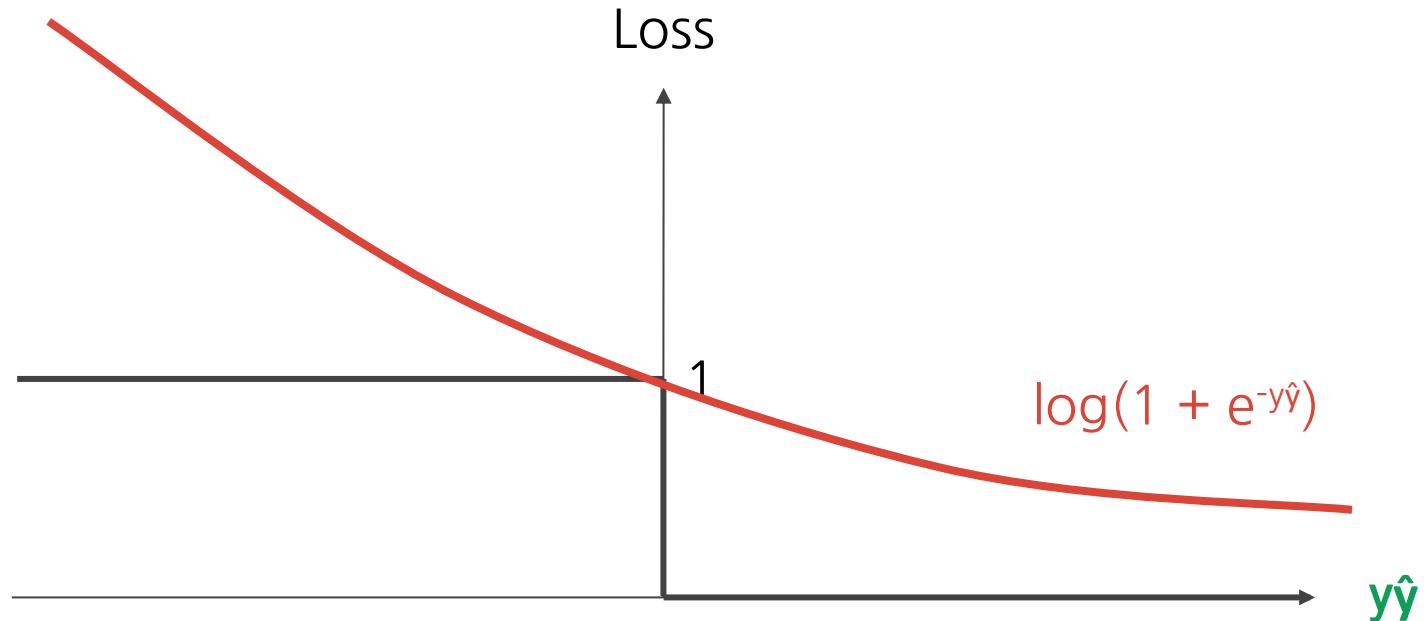
모델이 예제를 잘못 분류하면 일정한 손실이 발생합니다.
모델이 예제를 올바르게 분류하면 손실이 없습니다.



3.2 마진 기반 손실: 로그 손실

손실 함수

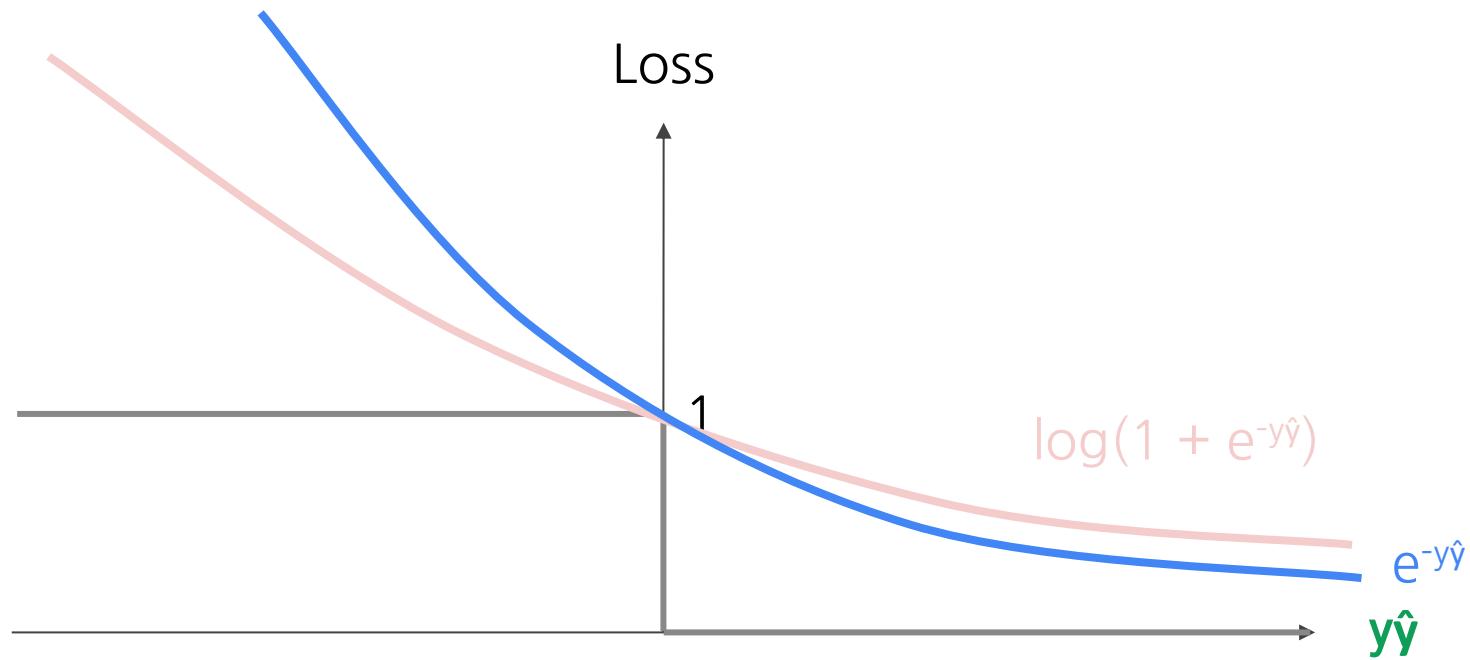
예측이 정확할수록 페널티는 작아집니다. (0은 아닙니다.)
연속 함수 → 어느 시점에서나 **미분 가능!**



3.2 마진 기반 손실: 지수 손실

손실 함수

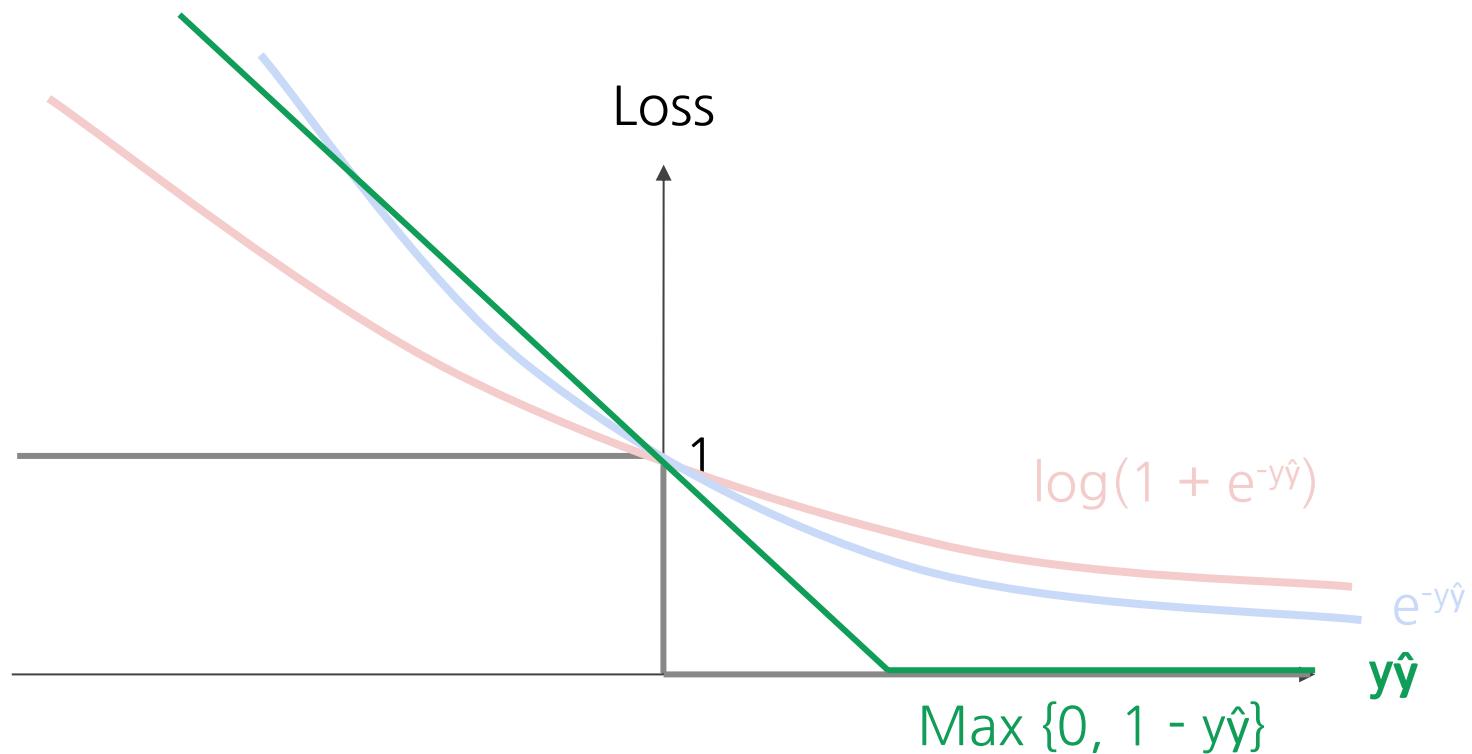
로그 손실과 비슷하지만,
잘못된 경우에는 더 엄격하게 패널티를 주고 올바른 경우에는 패널티를 적게 줍니다.
지수 손실 또한 어느 지점에서나 **미분가능**합니다.



3.2 마진 기반 손실: Hinge 손실

손실 함수

오류에 대한 패널티가 선형적으로 증가합니다.
오차 범위 내에서 정답인 경우에도 약간의 패널티를 받습니다.



3.4 Cross Entropy

손실 함수

- 정보 이론에 따르면 집합에서 추출한 이벤트를 식별하는데 필요한 평균 비트 수를 측정합니다.

- 일반적 정의 :

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \sum_{k=1}^K y_{ik} \log(\hat{y}_{ik})$$

- 이진 정의 :

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

3.4 Cross Entropy

손실 함수

- 모든 클래스(K)에 걸쳐 합산하더라도 단 하나의 k 에 대해 $y_{ik} = 1$ 이 있으므로 한 항만 남습니다.

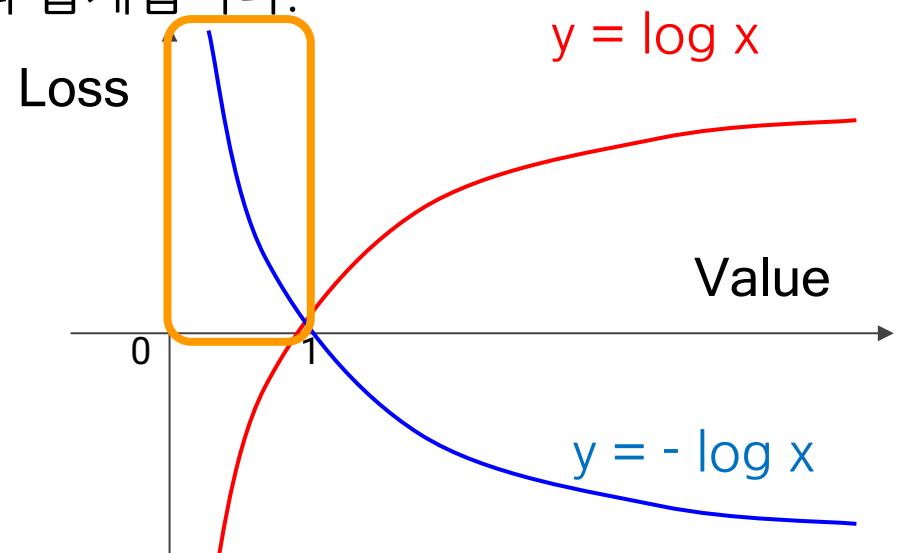
$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad \longrightarrow \quad \mathcal{L} = -\frac{1}{N} \sum_{i=1}^N \log(\hat{y}_{iT_i})$$

T_i : ground truth
index for example
i

→ 모든 샘플에 대한 $-\log$ (“올바른 클래스에 대한 예측 확률”)의 합계입니다.

“예측 확률”은 0에서 1 사이이므로 왼쪽에 표시된 주황색 범위에 해당합니다.

→ 추정치가 1에 가까워지면 손실이 0에 수렴하고, 0에 가까워지면 손실이 증가합니다.



4.

최적화

최적화의 필요성과 예시

4.1 가중치는 어떻게 설정해야 할까요?

최적화

아직 가장 중요한 문제에 대해 이야기하지 않았습니다.

W(매개변수들)의 값을 어떻게 정할 것인가?

머신 러닝은 데이터 기반의 접근입니다.

- 우리는 모델의 형식(e.g., $f(\mathbf{x}, \mathbf{W}) = \mathbf{W}\mathbf{x}$)을 디자인하고
- 랜덤하게 매개변수(\mathbf{W})의 값을 설정합니다.
- 그 후, 학습 데이터 \mathbf{x} 를 입력해 라벨 $\hat{\mathbf{y}}$ 을 예측합니다.
- 추정값 $\hat{\mathbf{y}}$ 을 기준값 라벨 \mathbf{y} 와 비교하여 현재값이 얼마나 좋은지/나쁜지를 비교합니다.
- 손실값에 따라 매개변수(\mathbf{W})를 업데이트하고,
- $\hat{\mathbf{y}} \approx \mathbf{y}$ 가 될 때까지 이 과정을 반복합니다.

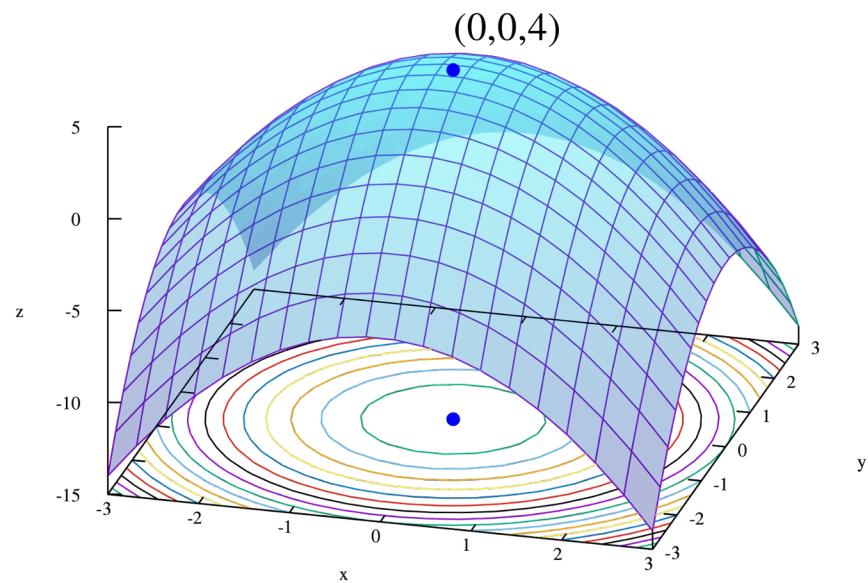
이 파트에 대해 이야기 해봅시다

이 문제에 대해서 이야기 했으니

4.2 최적화란?

최적화

[Wikipedia] Mathematical optimization or mathematical programming is the selection of a best element (with regard to some criterion) from some set of available alternatives.



https://en.wikipedia.org/wiki/Mathematical_optimization

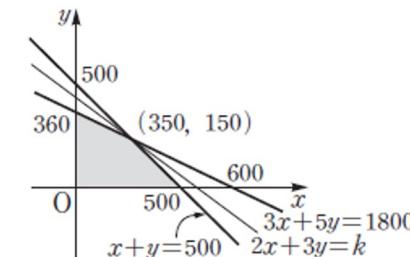
$$\begin{cases} 3x+5y \leq 1800 \\ x+y \leq 500 \\ x \geq 0 \\ y \geq 0 \end{cases}$$

따라서 연립부등식을 만족시키는 영역은 오른쪽 그림의 어두운 부분(경계선 포함)과 같다.

이때 포만감은 $2x+3y$ 이므로 $2x+3y=k$ (k 는 상수)로 놓으면 이 직선이 두 직선

$3x+5y=1800$, $x+y=500$ 의 교점 $(350, 150)$ 을 지날 때, k 의 값은 최대가 된다.

따라서 식품 A를 350 g 섭취하는 것이 적당하다.

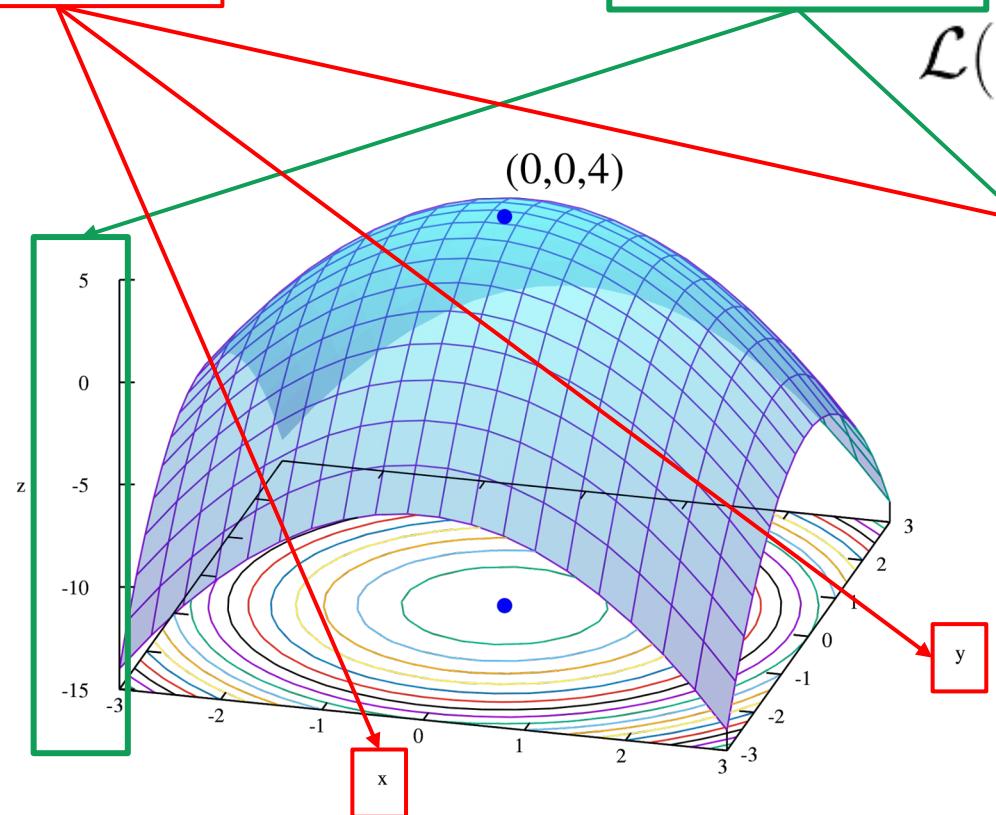


답 ⑤

4.2 최적화!

최적화

[Wikipedia] Mathematical optimization or mathematical programming is the selection of a best element (with regard to some criterion) from some set of available alternatives.

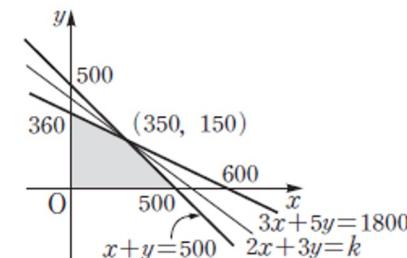


$$\mathcal{L}(\hat{y}, y)$$

$$\begin{cases} 3x+5y \leq 1800 \\ x+y \leq 500 \\ x \geq 0 \\ y \geq 0 \end{cases}$$

따라서 연립부등식을 만족시
키는 영역은 오른쪽 그림의 어
두운 부분(경계선 포함)과 같
다.

이때 포만감은 $2x+3y$ 이므로
 $2x+3y=k$ (k 는 상수)로 놓
으면 이 직선이 두 직선
 $3x+5y=1800$, $x+y=500$ 의 교점 $(350, 150)$ 을 지날 때,
 k 의 값은 최대가 된다.
따라서 식품 A를 350 g 섭취하는 것이 적당하다.



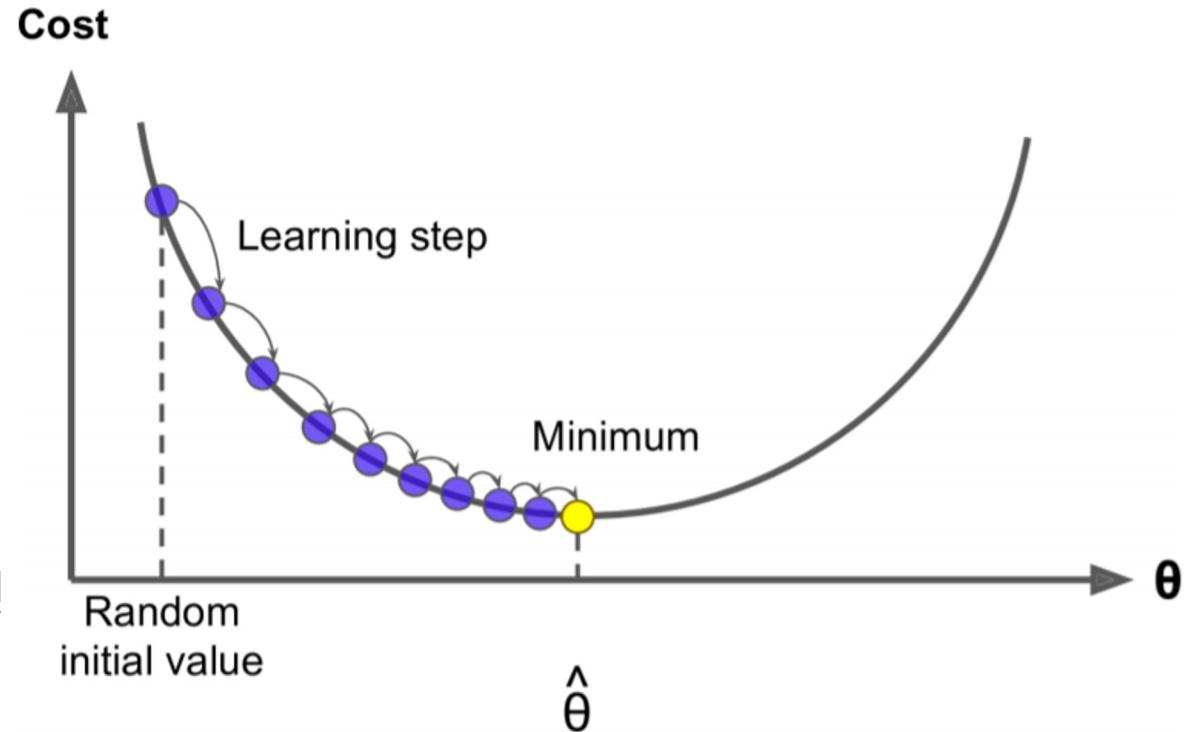
4.3 Gradient Descent

최적화

최소화하고자 하는 비용 함수 $J(\theta)$ 가 있습니다.

아이디어:

- θ 의 현재 값에 대해 $J(\theta)$ 의 **기울기**를 계산한 다음 음의 기울기 방향으로 작은 단계를 밟고, 이를 반복합니다.
 - 한 지점에서 함수의 기울기(또는 미분)는 해당 지점에서 함수에 대한 **가장 좋은 선형 근사치**라는 것을 기억하세요.



4.3 Gradient Descent

최적화

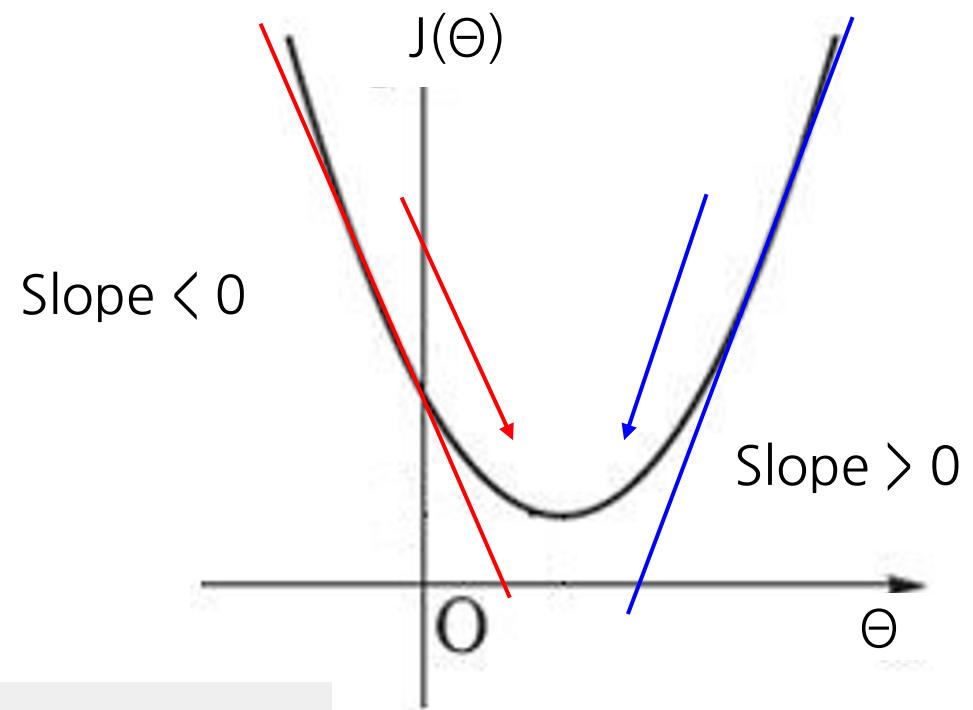
- 업데이트 방정식(벡터 표기법에서) :

$$\Theta^{new} = \Theta^{old} - \alpha \nabla_{\Theta} J(\Theta)$$

- 업데이트 방정식(단일 매개변수에서) :

$$\theta_i^{new} = \theta_i^{old} - \alpha \frac{\partial}{\partial \theta_i^{old}} J(\Theta)$$

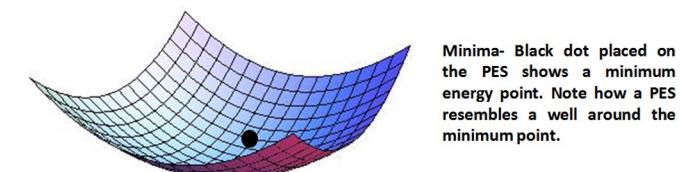
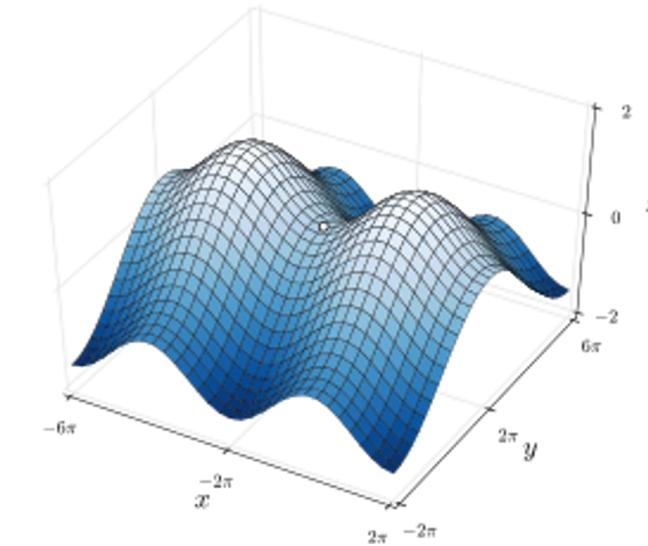
```
theta = rand(vector)
while true:
    theta_grad = evaluate_gradient(J, data,
theta)
    theta = theta - alpha * theta_grad
    if (norm(theta_grad) <= beta) break
```



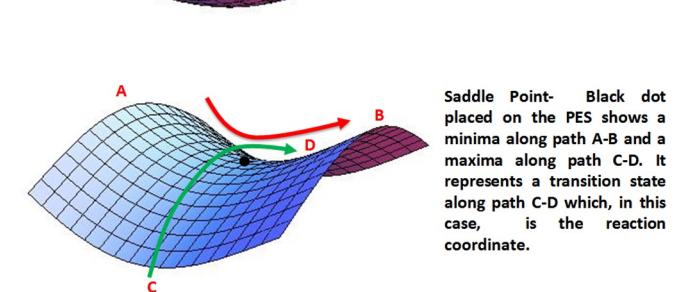
4.3 Gradient Descent : 포텐셜 문제들

최적화

- 표면이 **볼록하지 않을 수 있습니다.**
 - 즉, 로컬 최적점 또는 새들 포인트가 많을 수 있습니다.
 - 여기서 그라데이션은 전역 최소값이 아니더라도 0이 됩니다.
- 비용 함수가 **미분 가능한 경우에만** 작동합니다.
 - 일부 손실 함수(예: 0/1 손실)는 항상 미분 가능하지 않습니다.
 - 다른 부드러운 대리 함수가 채택됩니다.
- 국부 최소값으로 수렴하는 속도가 상당히 느릴 수 있습니다.
 - 기울기는 모든 데이터 샘플에서 계산되므로 그 평균을 사용하여 매개변수를 업데이트합니다.
 - 데이터 셋이 큰 경우 모든 데이터 포인트에서 그래디언트를 계산하는 데 시간이 오래 걸립니다.



Minima- Black dot placed on the PES shows a minimum energy point. Note how a PES resembles a well around the minimum point.



Saddle Point- Black dot placed on the PES shows a minima along path A-B and a maxima along path C-D. It represents a transition state along path C-D which, in this case, is the reaction coordinate.

4.4 Stochastic Gradient Descent

최적화

- 모든 훈련 예제에 대해 그라데이션을 계산하는 대신 **무작위로 샘플링된 하위 집합**에 대해서만 수행합니다.
- 가장 극단적인 경우는 모든 샘플에 대해 파라미터를 업데이트하는 것입니다.
- 가장 일반적인 방법 : 32, 64, 128, 256, ..., 8192 크기의 **미니 배치**로 수행하는 것입니다.
 - 최적의 크기는 문제, 데이터 및 하드웨어(**메모리**)에 따라 다릅니다.
- 미니 배치가 작을 때, 두 배로 늘리면 기울기 추정이 훨씬 안정적입니다. 크기가 커지면 이득은 줄어들고 비용만 두 배로 증가합니다. (수익 감소 - **diminishing returns**)

4.5 머신 러닝 기반의 접근

최적화

이제 **W**(파라미터)의 값을 설정하는 방법을 포함한 모든 구성 요소를 갖추었습니다!

머신 러닝은 데이터 기반의 접근입니다.

- 우리는 모델의 형식(*e.g.*, $f(\mathbf{x}, \mathbf{W}) = \mathbf{W}\mathbf{x}$)을 디자인하고
- 랜덤하게 매개변수(**W**)의 값을 설정합니다.
- 그 후, 학습 데이터 **x**를 입력해 라벨 **y**을 예측합니다.
- 추정값 **ŷ**을 기준값 라벨 **y**와 비교하여 현재값이 얼마나 좋은지/나쁜지를 비교합니다.
- 손실값에 따라 매개변수(**W**)를 업데이트하고,
- $\hat{\mathbf{y}} \approx \mathbf{y}$ 가 될 때까지 이 과정을 반복합니다.

각 반복으로부터, (훈련)손실은 감소할 것으로 예상됩니다.

충분한 반복 이후에는

마무리 기준 ($\hat{\mathbf{y}} \approx \mathbf{y}$) 이 충족됩니다.

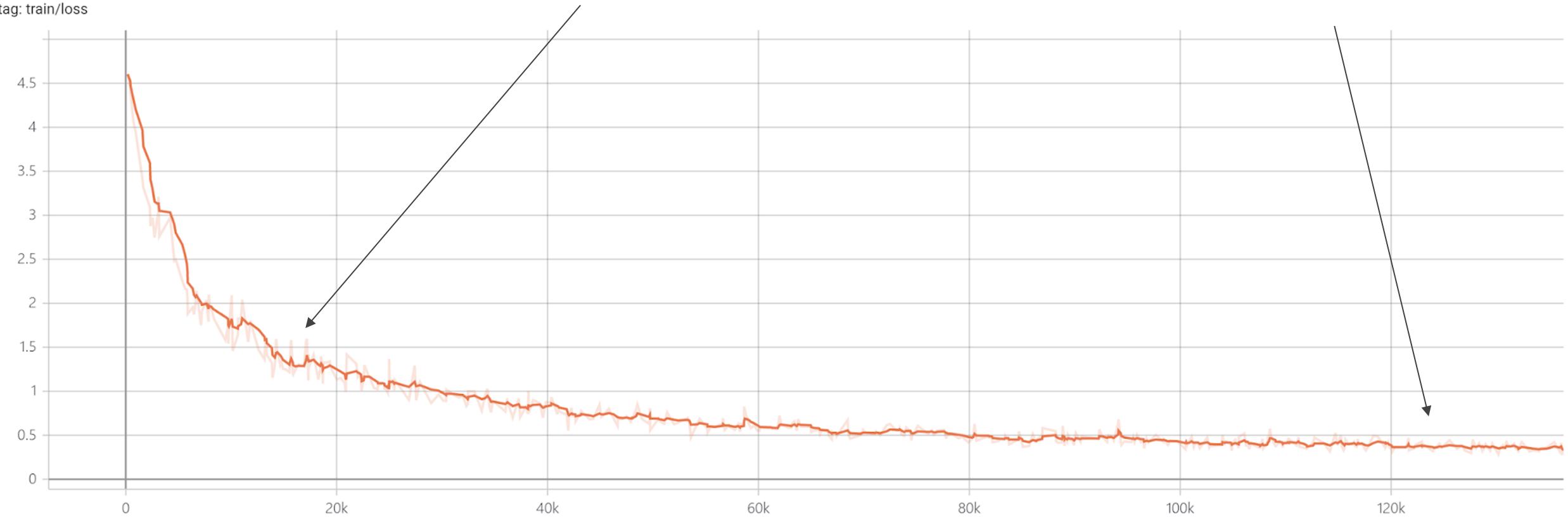
4.5 훈련 손실의 예시

최적화

train/loss
tag: train/loss

Q: 왜 노이즈가 많은 커브가 생기나요?

Q: 언제 훈련을 멈춰야 하나요?



5.

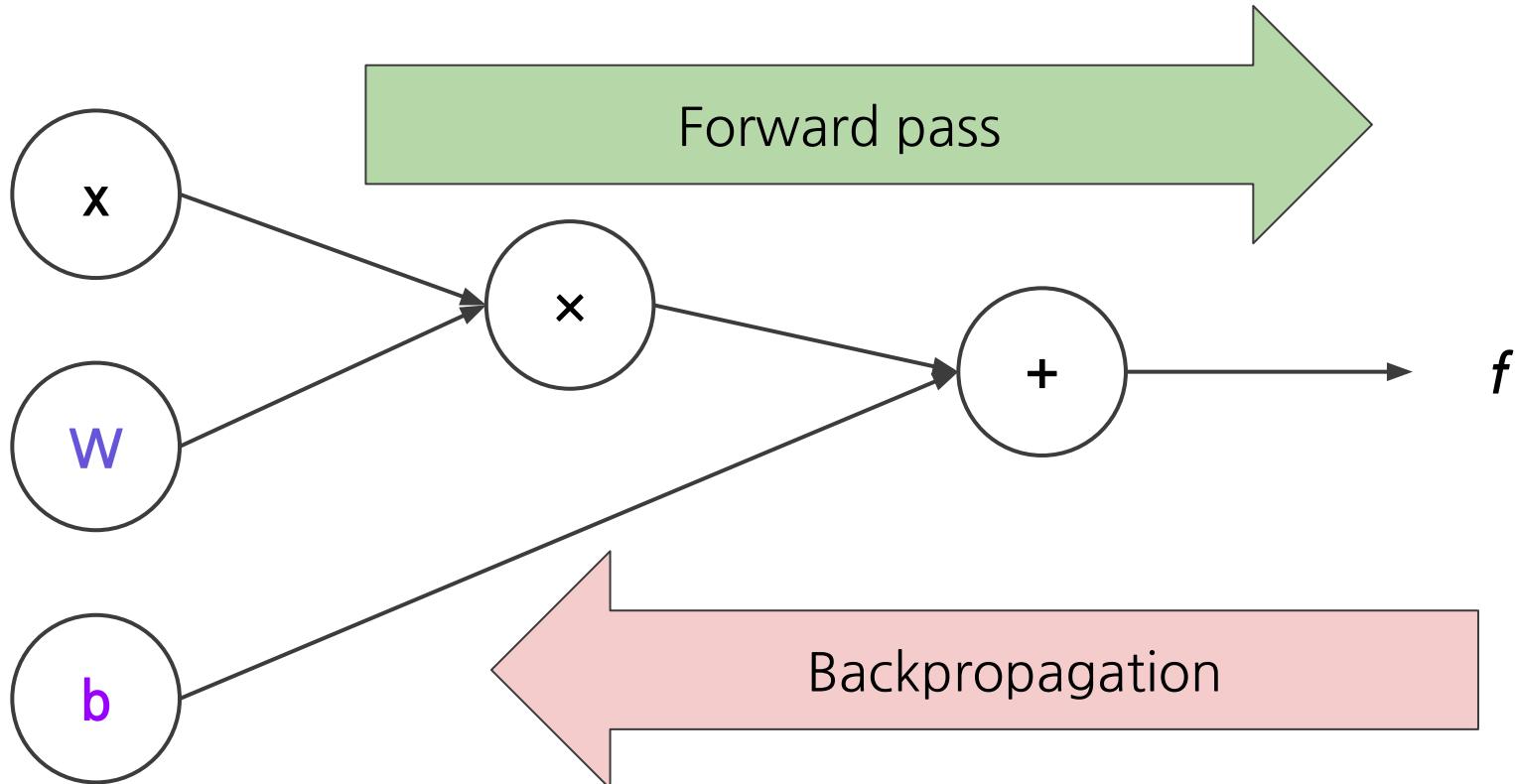
Backpropagation: Computing Gradients

Backpropagation에서 경사 하강법을 사용한 가중치 업데이트 방법에 대한 설명

Computational Graph

Backpropagation

$$f(\mathbf{x}, \mathbf{W}) = \mathbf{W}\mathbf{x} + \mathbf{b}$$



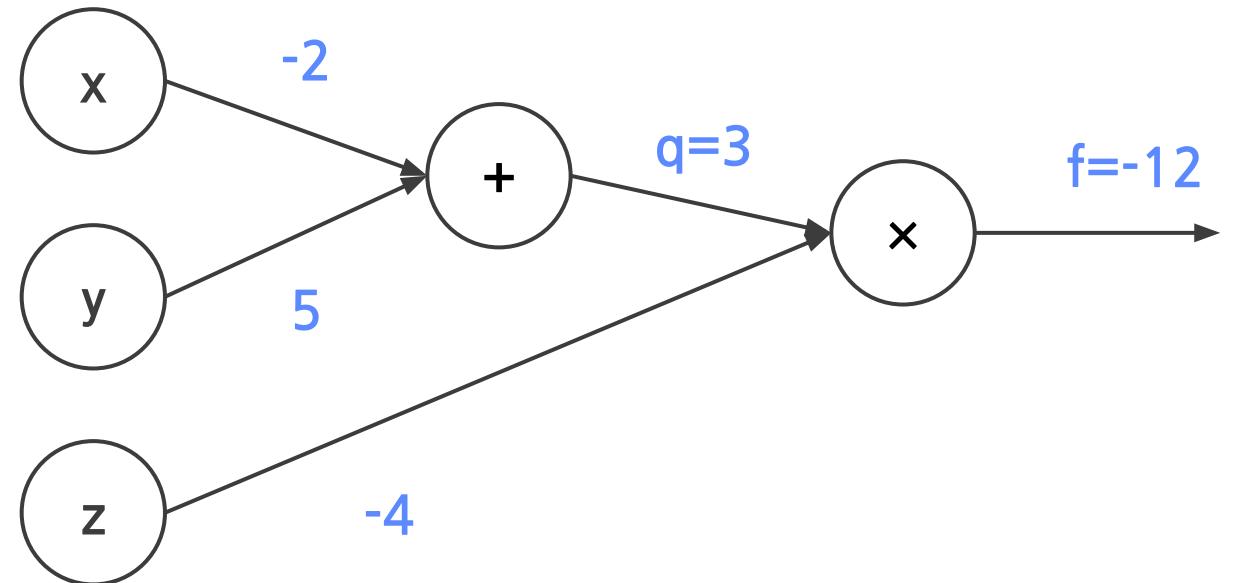
Backpropagation Example

Backpropagation

Forward pass

$$f(x, y, z) = (x+y)z \text{ 일 때},$$

$x = -2, y = 5, z = -4$ 라고 가정해보자.



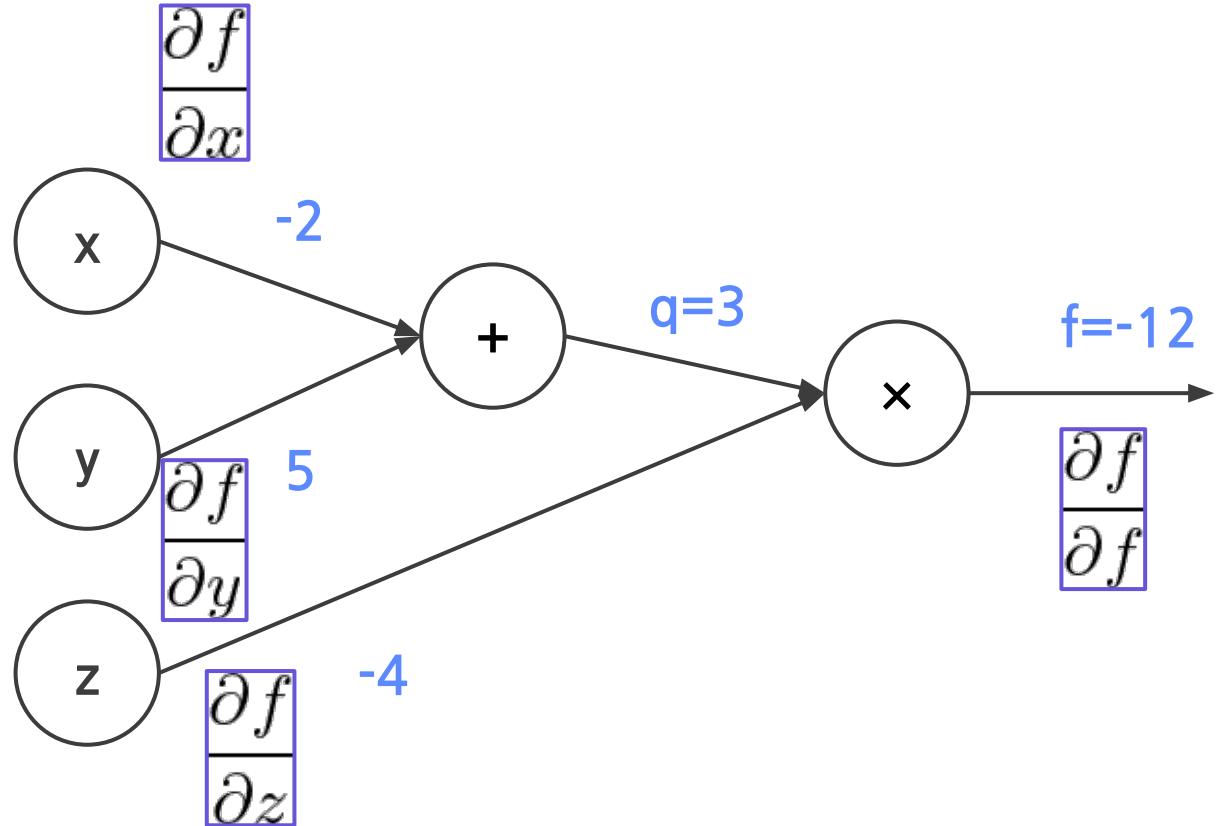
Backpropagation Example

Backpropagation

Backpropagation

$f(x, y, z) = (x+y)z$ 일 때,
 $x = -2, y = 5, z = -4$ 라고 가정해보자.

변수 (x, y, z) 에 대하여 편미분을 합니다.



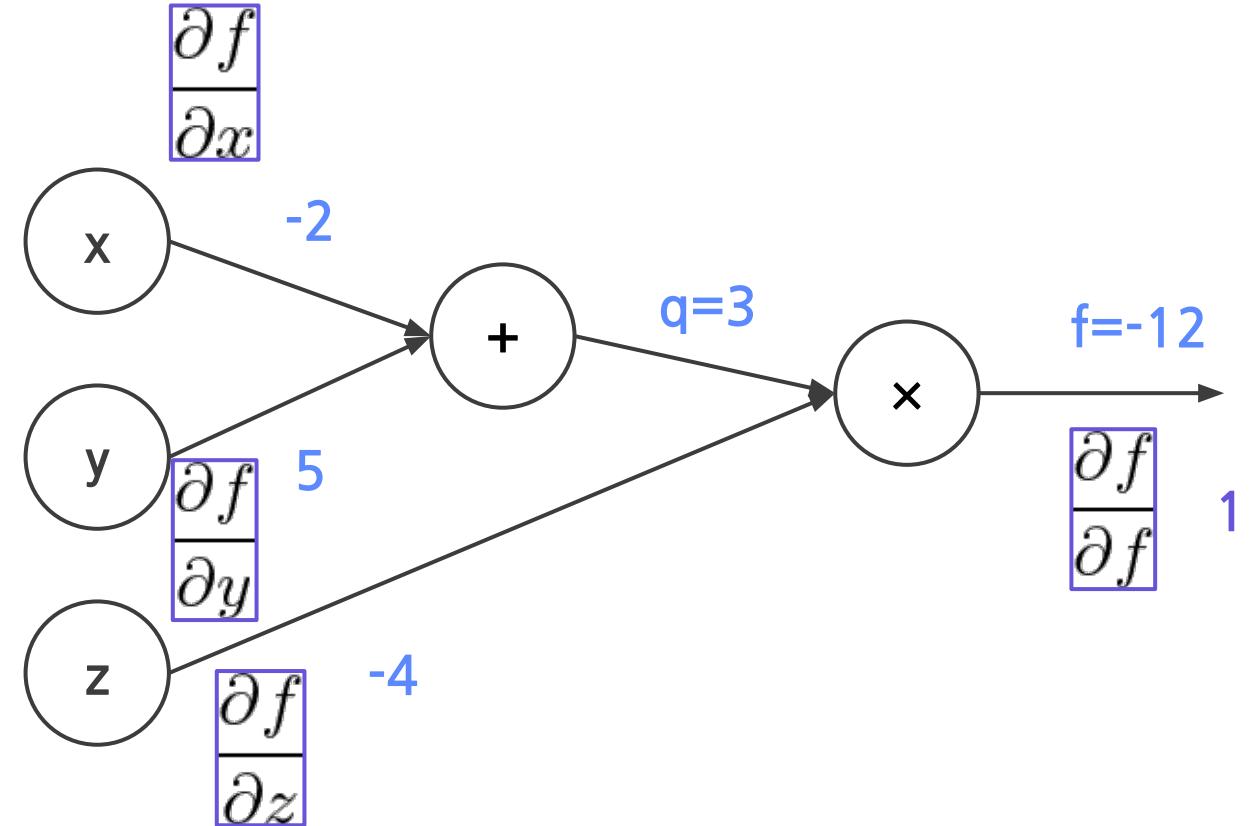
Backpropagation Example

Backpropagation

Backpropagation

$f(x, y, z) = (x+y)z$ 일 때,
 $x = -2, y = 5, z = -4$ 라고 가정해보자.

마지막은 항상 1입니다.



Backpropagation Example

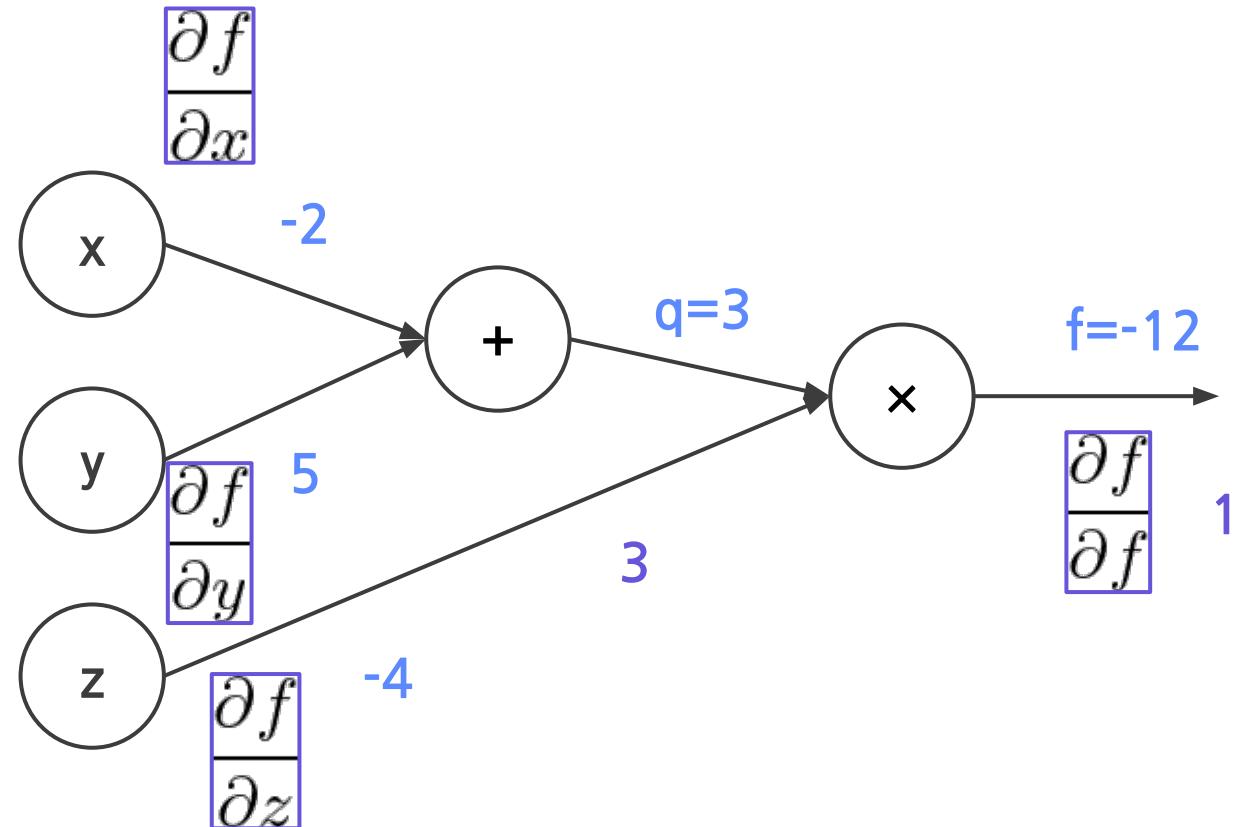
Backpropagation

Backpropagation

$f(x, y, z) = (x+y)z$ 일 때,
 $x = -2, y = 5, z = -4$ 라고 가정해보자.

f 에 대한 편미분

$$\text{wrt. } z: \frac{\partial f}{\partial z} = \frac{\partial(qz)}{\partial z} = q$$



Backpropagation Example

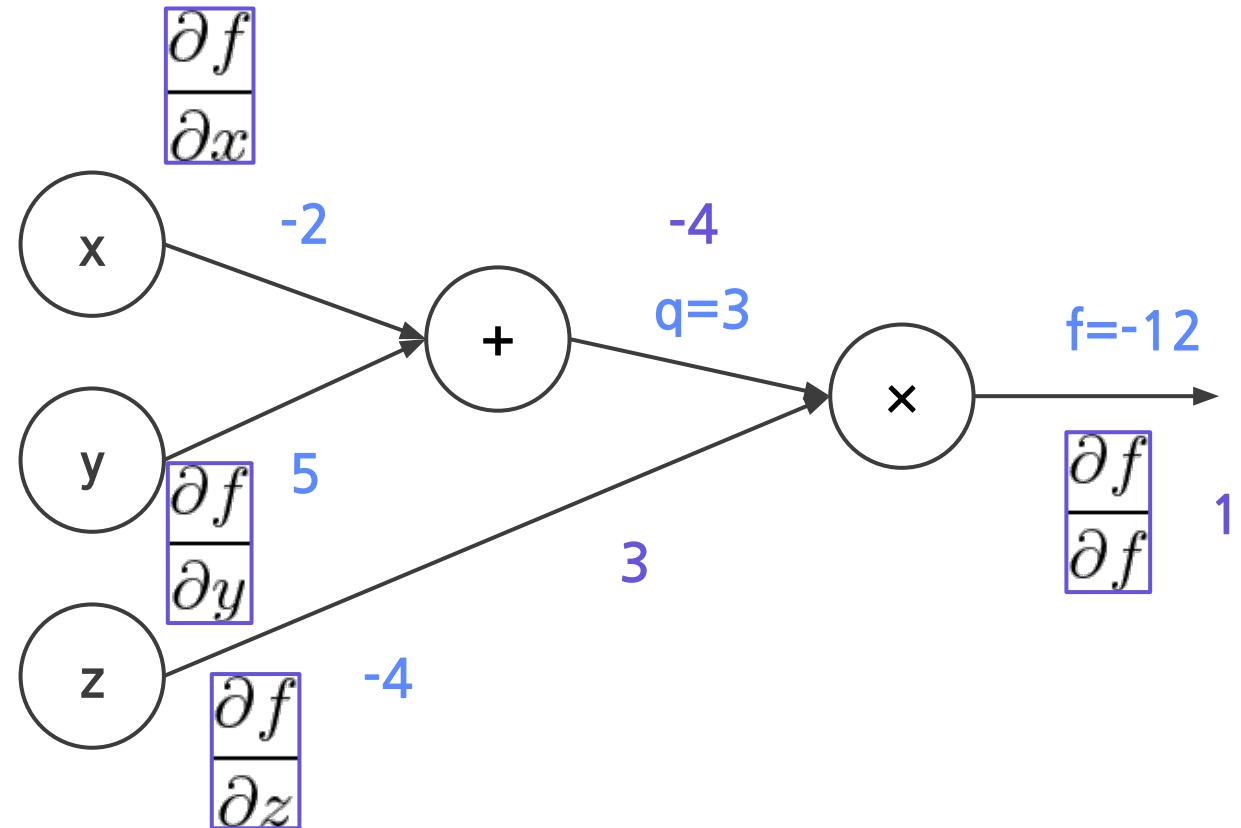
Backpropagation

Backpropagation

$f(x, y, z) = (x+y)z$ 일 때,
 $x = -2, y = 5, z = -4$ 라고 가정해보자.

f 에 대한 편미분

$$\text{wrt. } q: \frac{\partial f}{\partial q} = \frac{\partial(qz)}{\partial q} = z$$



Backpropagation Example

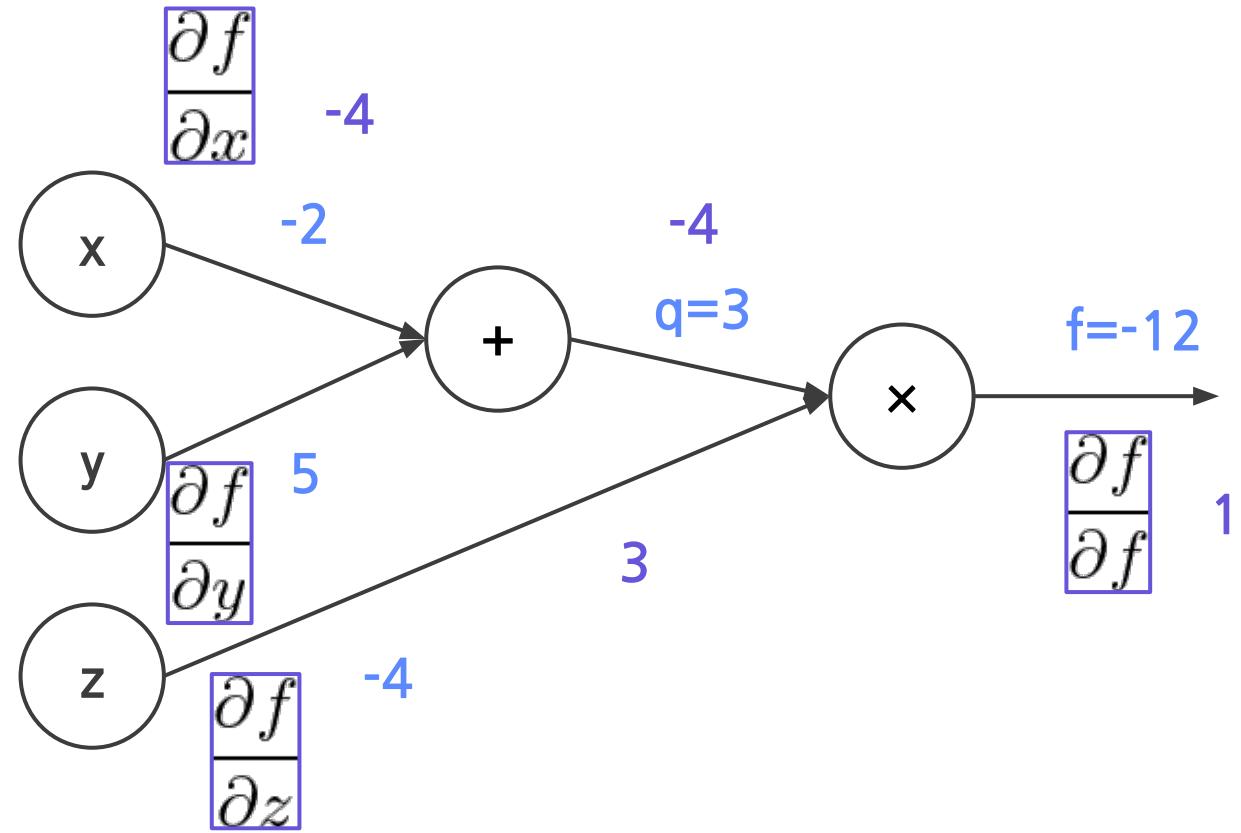
Backpropagation

Backpropagation

$f(x, y, z) = (x+y)z$ 일 때,
 $x = -2, y = 5, z = -4$ 라고 가정해보자.

f에 대한 편미분

$$\begin{aligned} \text{wrt. } x: \frac{\partial f}{\partial x} &= \frac{\partial f}{\partial q} \frac{\partial q}{\partial x} \\ &= \frac{\partial(qz)}{\partial q} \frac{\partial(x+y)}{\partial x} = z \cdot 1 \end{aligned}$$



Backpropagation Example

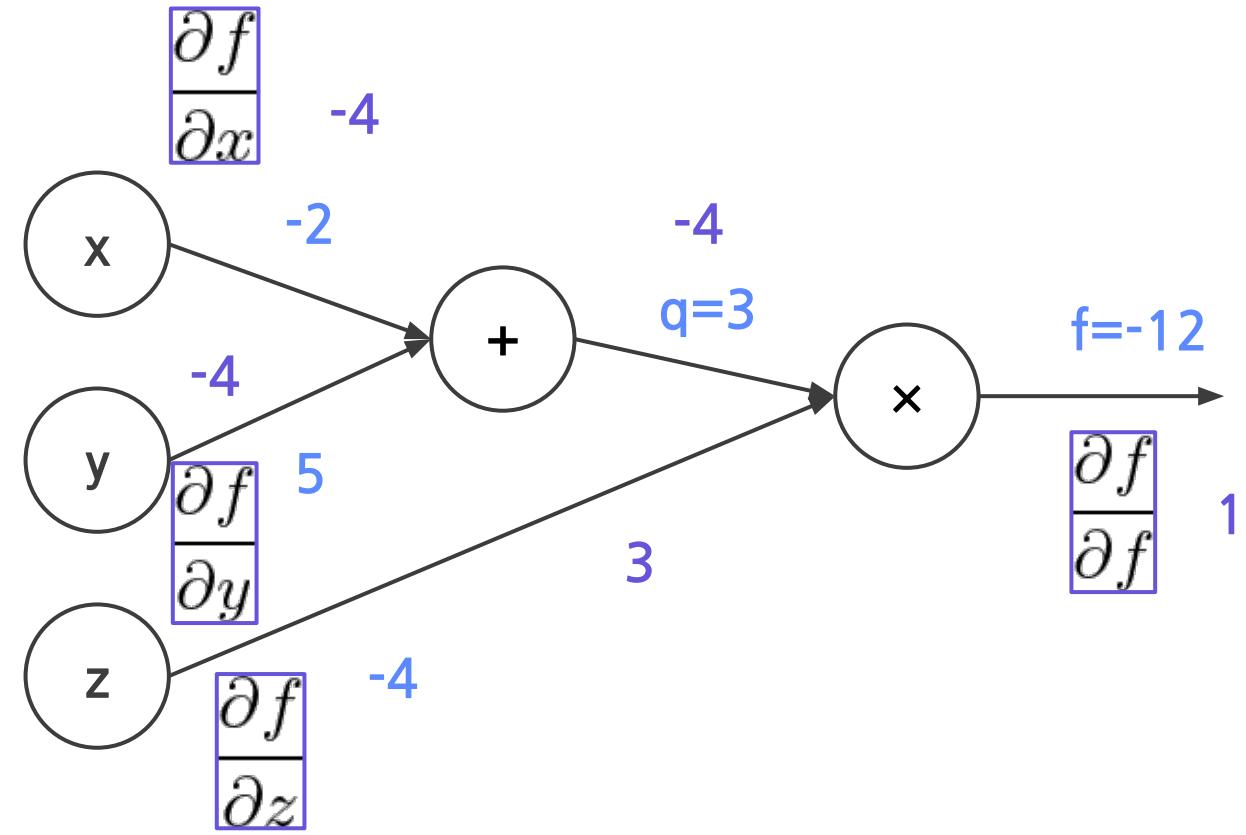
Backpropagation

Backpropagation

$f(x, y, z) = (x+y)z$ 일 때,
 $x = -2, y = 5, z = -4$ 라고 가정해보자.

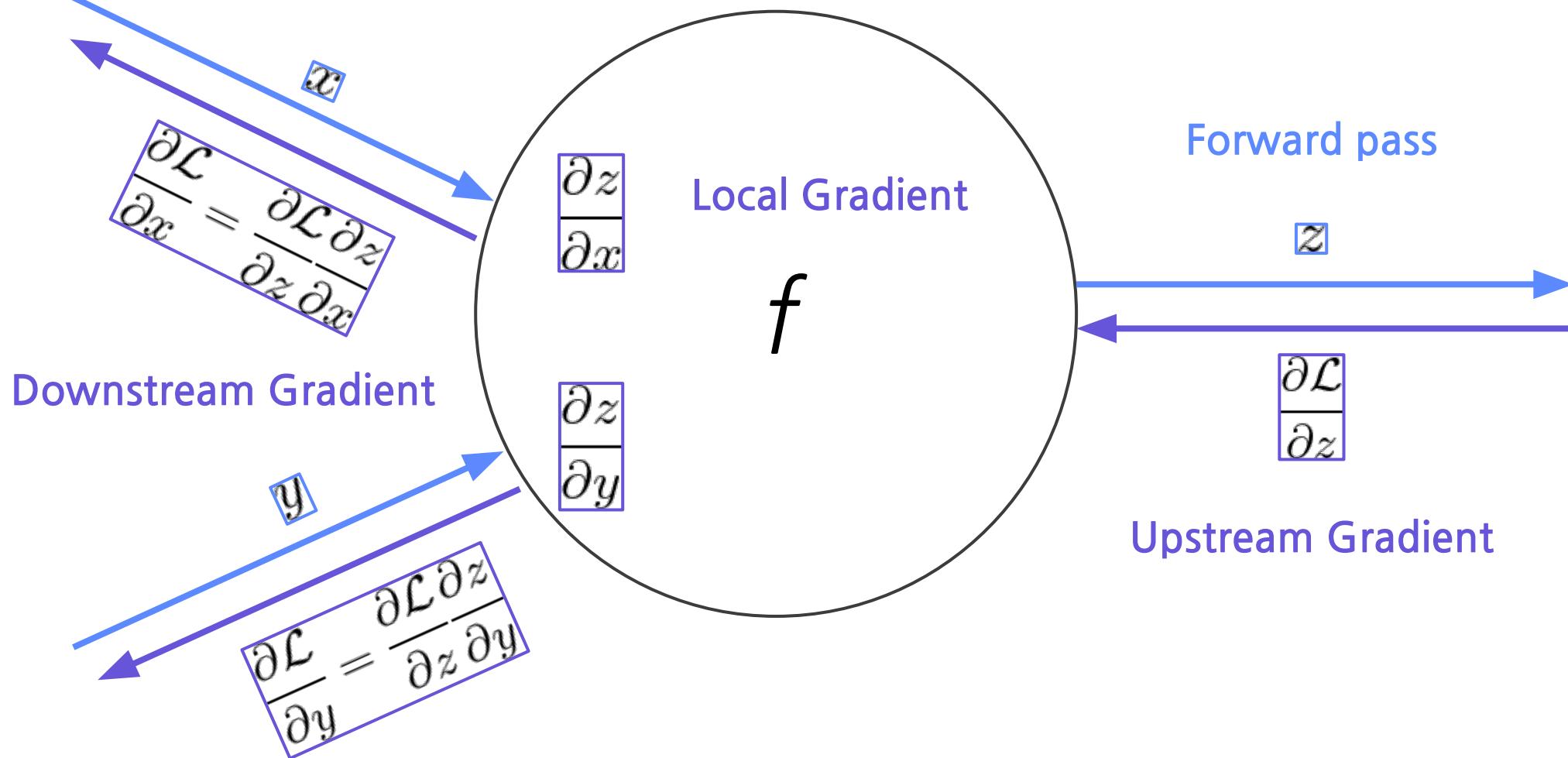
f 에 대한 편미분

$$\begin{aligned} \text{wrt. } y: \frac{\partial f}{\partial y} &= \frac{\partial f}{\partial q} \frac{\partial q}{\partial y} \\ &= \frac{\partial(qz)}{\partial q} \frac{\partial(x+y)}{\partial y} = z \cdot 1 \end{aligned}$$



Chain Rule

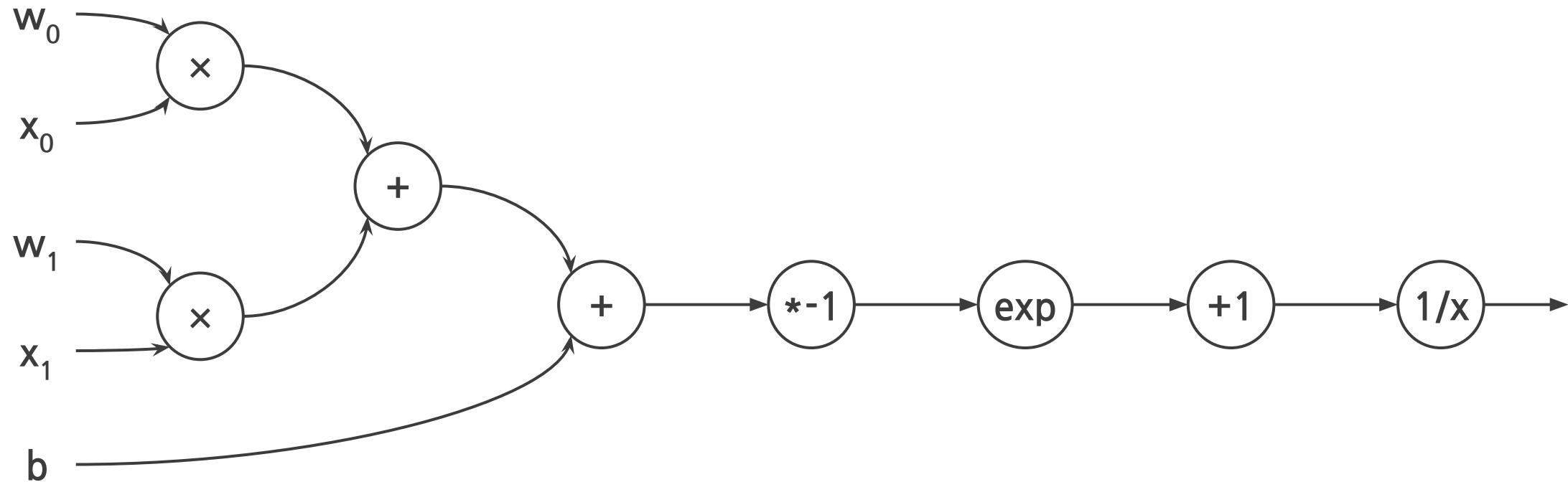
Backpropagation



Another Example: Linear classifier

Backpropagation

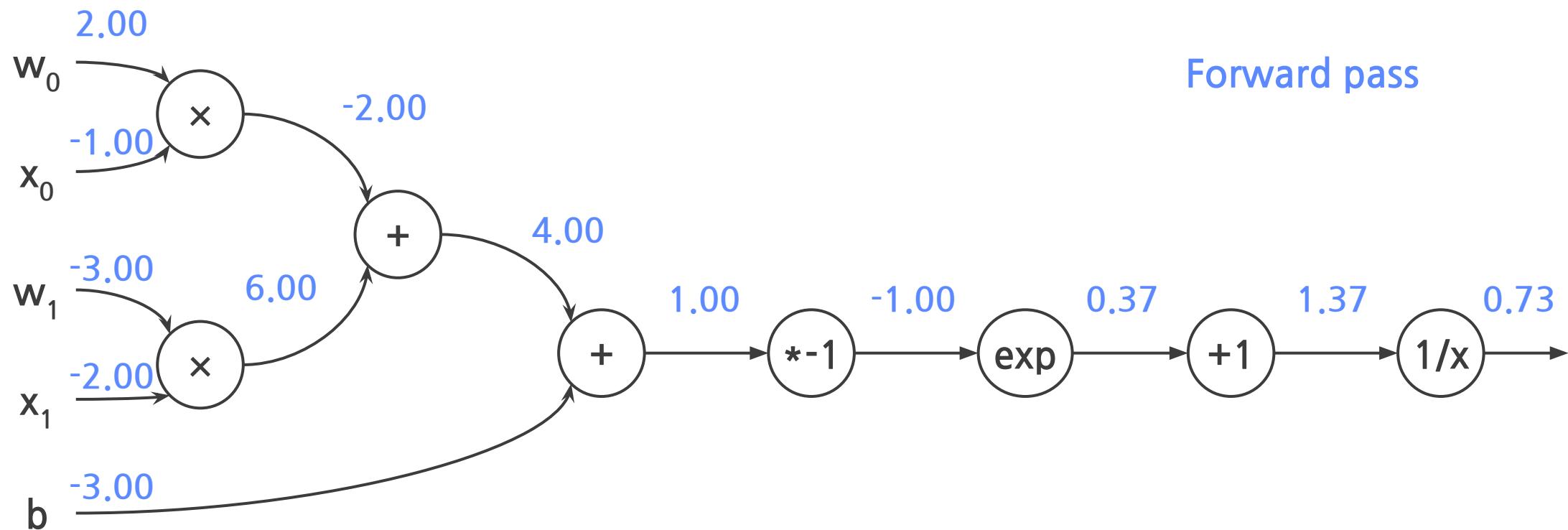
$$f(x, w) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + b)}}$$



Another Example: Linear classifier

Backpropagation

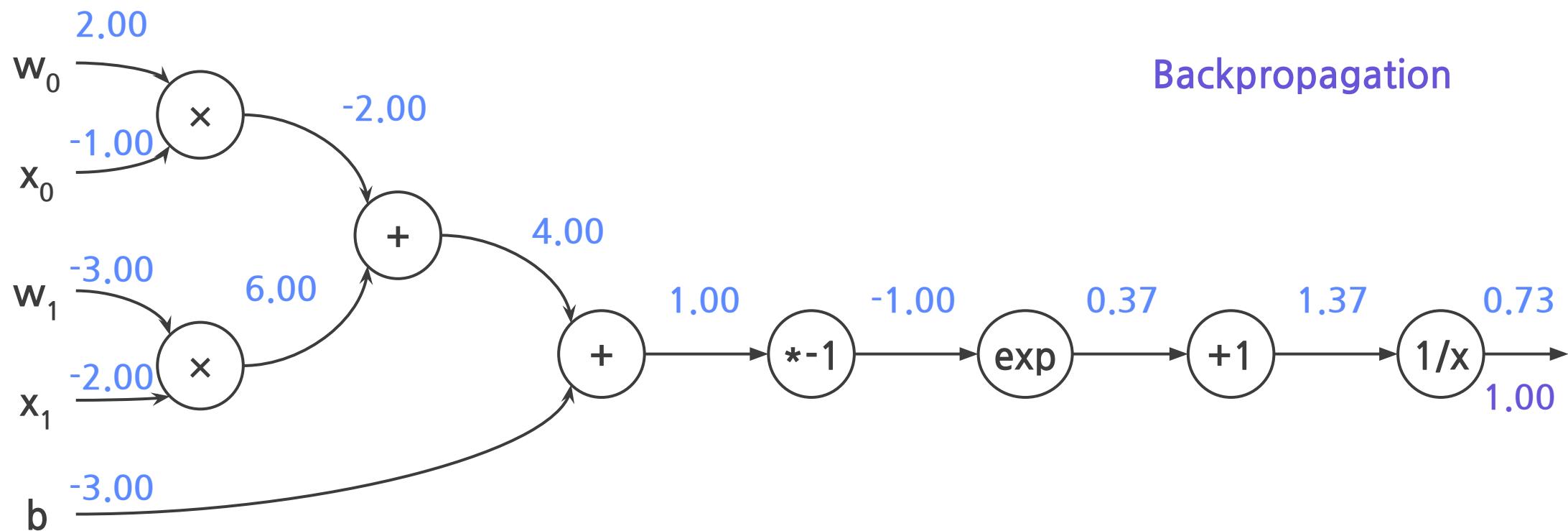
$$f(x, w) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + b)}}$$



Another Example: Linear classifier

Backpropagation

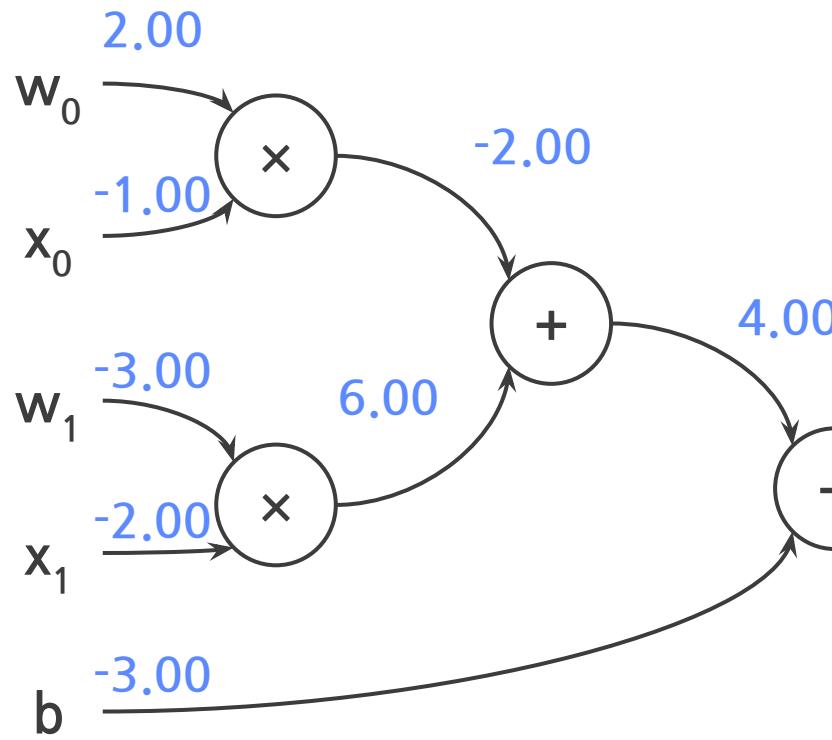
$$f(x, w) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + b)}}$$



Another Example: Linear classifier

Backpropagation

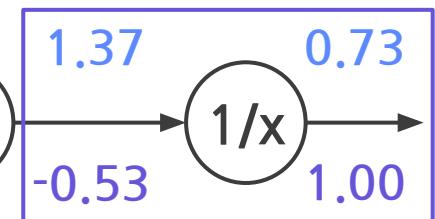
$$f(x, w) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + b)}}$$



Backpropagation

Upstream gradient Local gradient

$$\frac{(1.00)\left(\frac{-1}{1.37^2}\right)}{-0.53} = -0.53$$

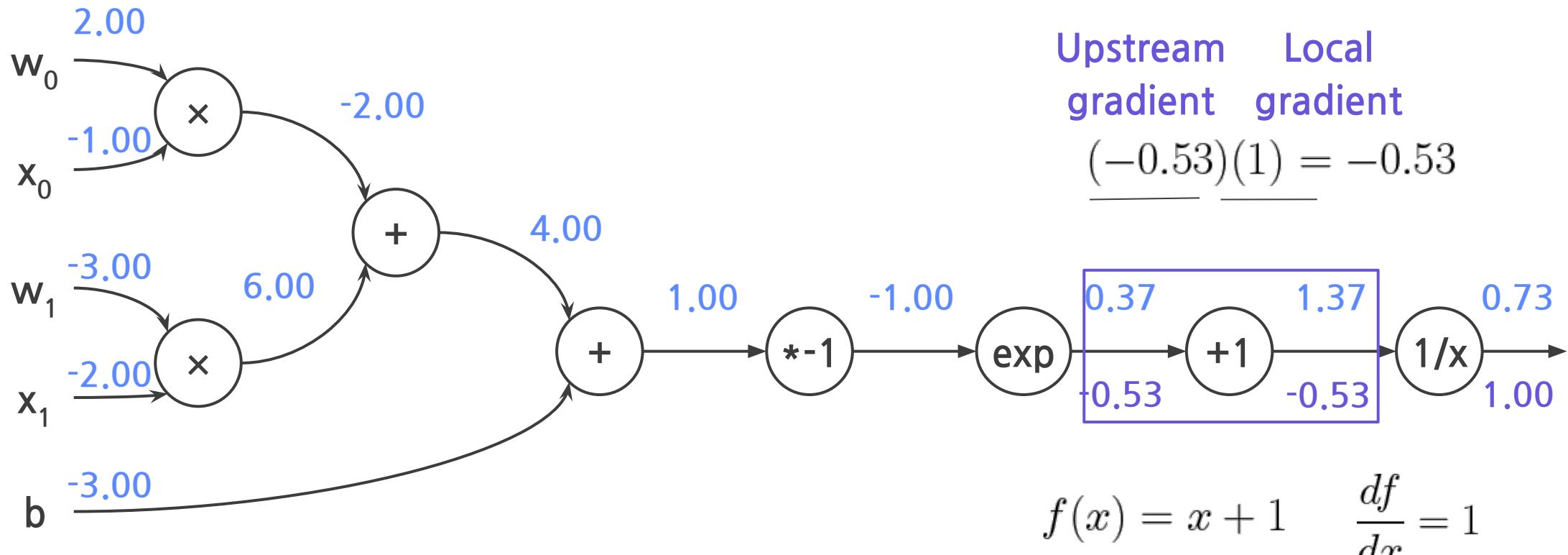


$$f(x) = \frac{1}{x} \quad \frac{df}{dx} = -\frac{1}{x^2}$$

Another Example: Linear classifier

Backpropagation

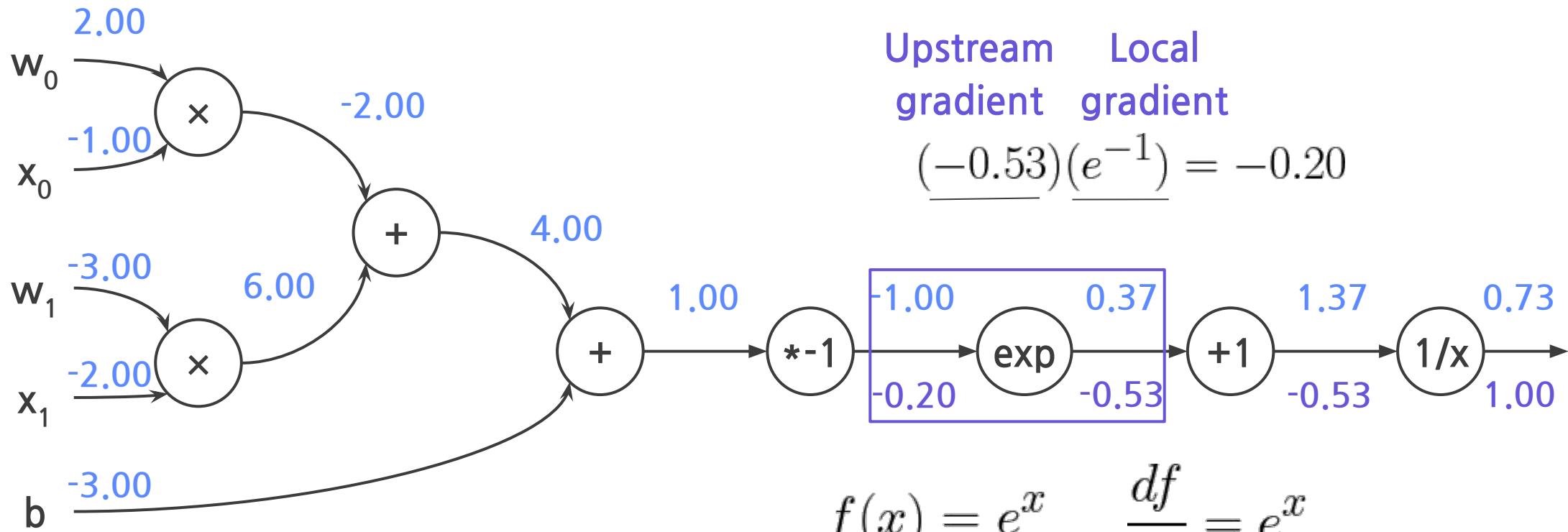
$$f(x, w) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + b)}}$$



Another Example: Linear classifier

Backpropagation

$$f(x, w) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + b)}}$$

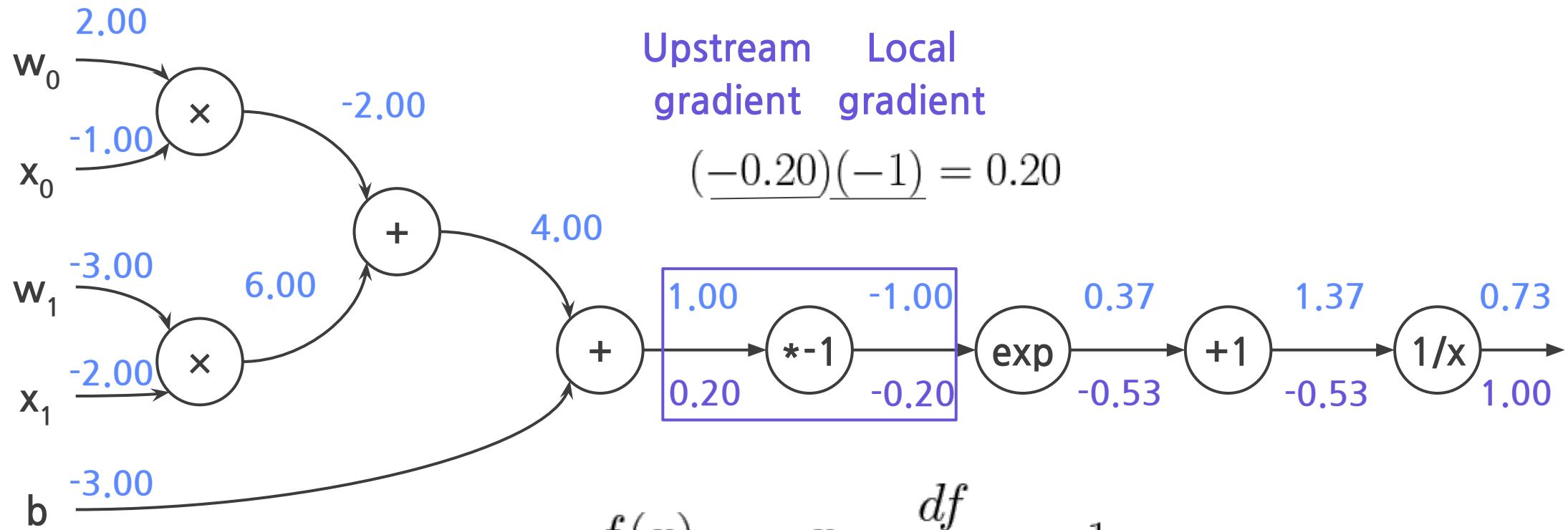


Another Example: Linear classifier

Backpropagation

$$f(x, w) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + b)}}$$

Backpropagation

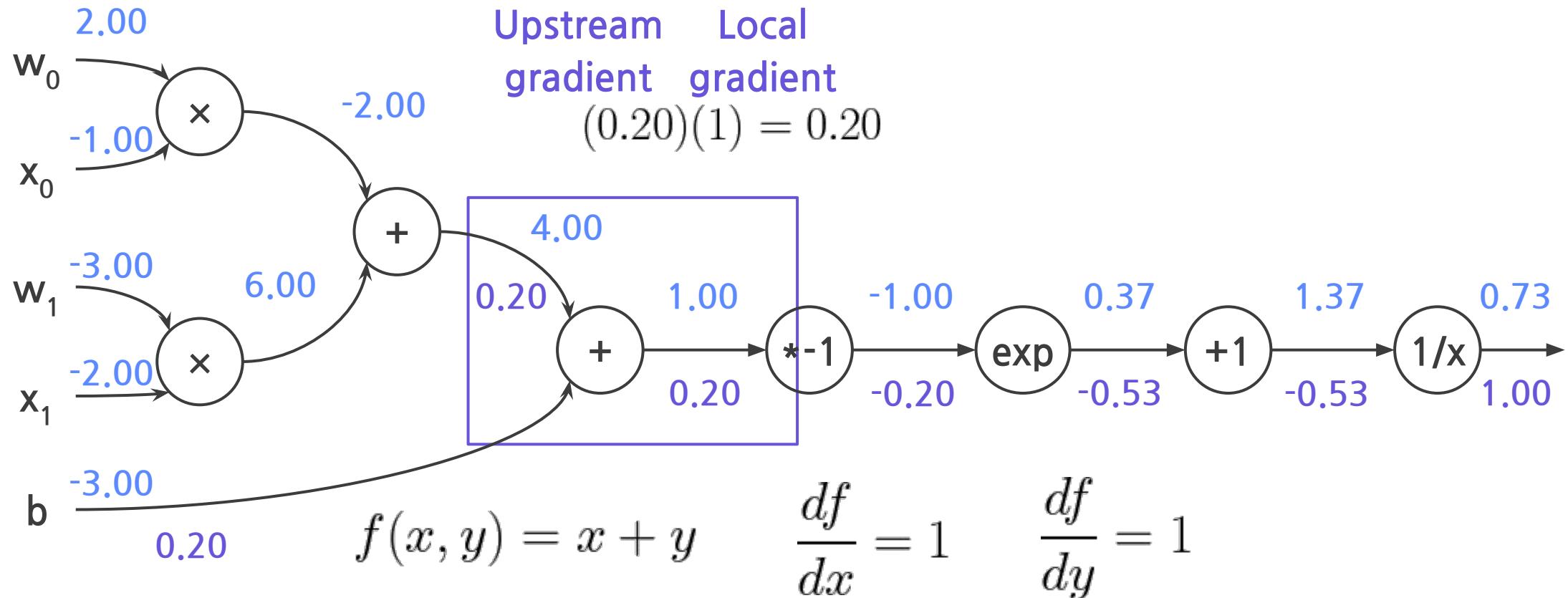


Another Example: Linear classifier

Backpropagation

$$f(x, w) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + b)}}$$

Backpropagation

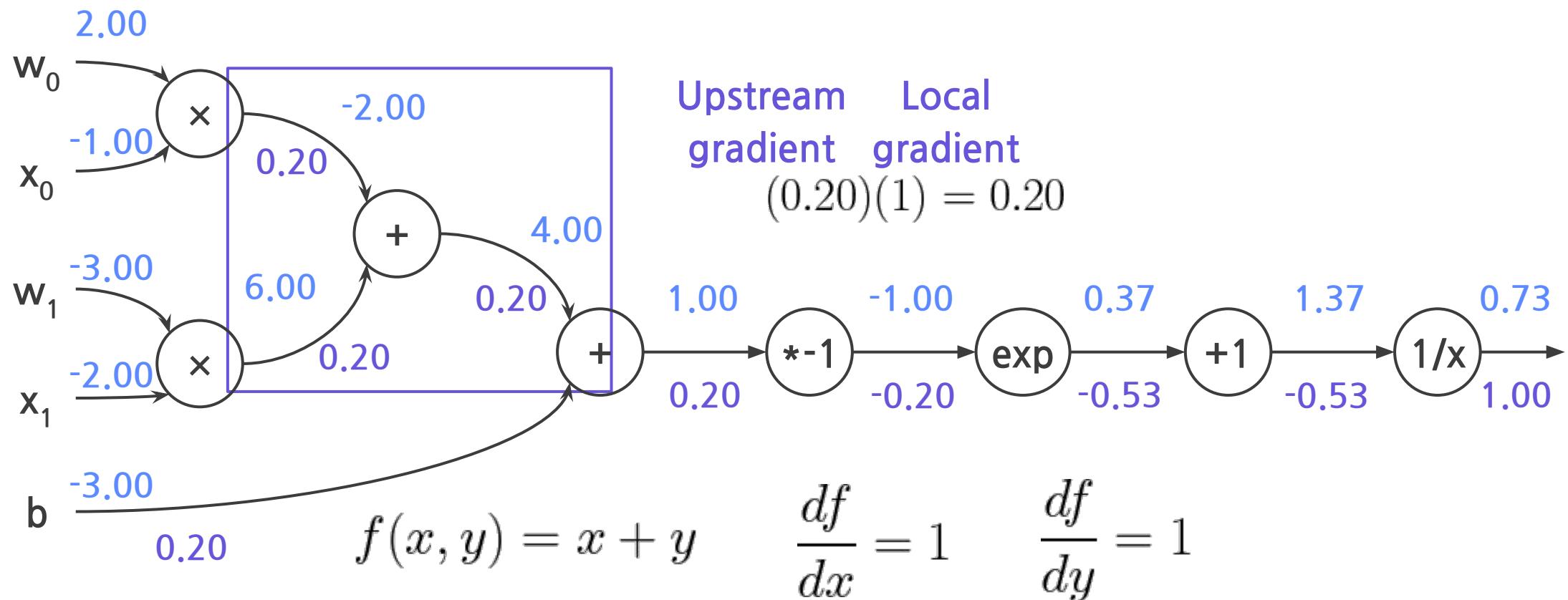


Another Example: Linear classifier

Backpropagation

$$f(x, w) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + b)}}$$

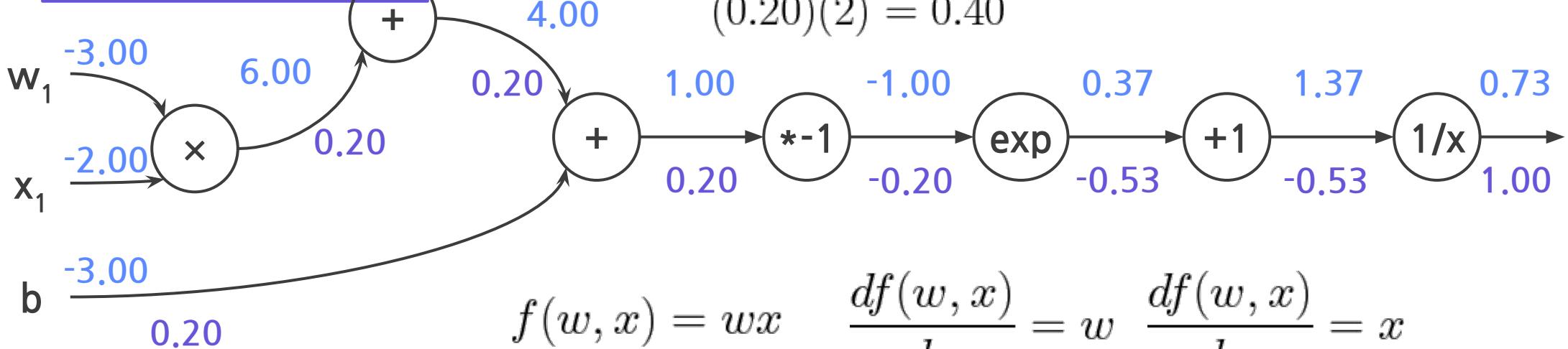
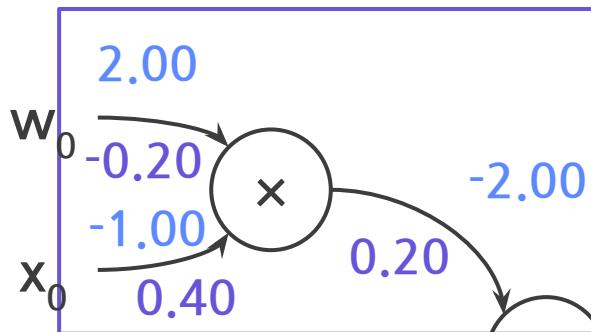
Backpropagation



Another Example: Linear classifier

Backpropagation

$$f(x, w) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + b)}}$$

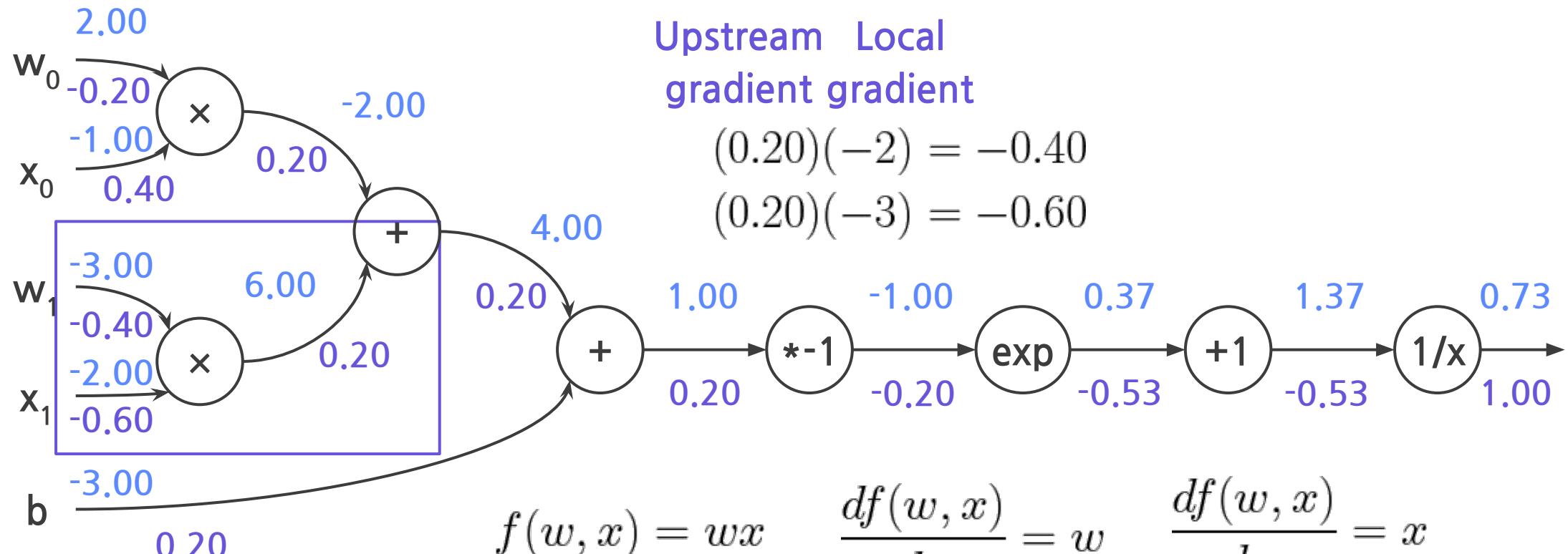


Another Example: Linear classifier

Backpropagation

$$f(x, w) = \frac{1}{1 + e^{-(w_0x_0 + w_1x_1 + b)}}$$

Backpropagation



Thank you

Prof. Jeon, Sangryul

Computer Vision Lab.

Pusan National University, Korea

Tel: +82-51-510-2423

Web: <http://sr-jeon.github.io/>

E-mail: srjeonn@pusan.ac.kr