

# **CLASSIFICATION & GRADIENT DESCENT**

# GOALS FOR TODAY

AND WHAT YOU SHOULD LEARN

1. Know the classification problem
2. What is gradient gradient descent
3. What is logistic regression

# FUNCTION APPROXIMATION

## RECAP

For any function  $f_*$ , e.g.,  $f_*(x) = ax^2 + bx + c$ , we want an approximation  $f(x)$

$$\forall x, |f(x) - f_*(x)| < \epsilon$$

# LINEAR FUNCTION APPROXIMATION

## RECAP

Represent as a linear function with a basis function

$\phi: \mathbb{R} \rightarrow \mathbb{R}^n$  is the basis function

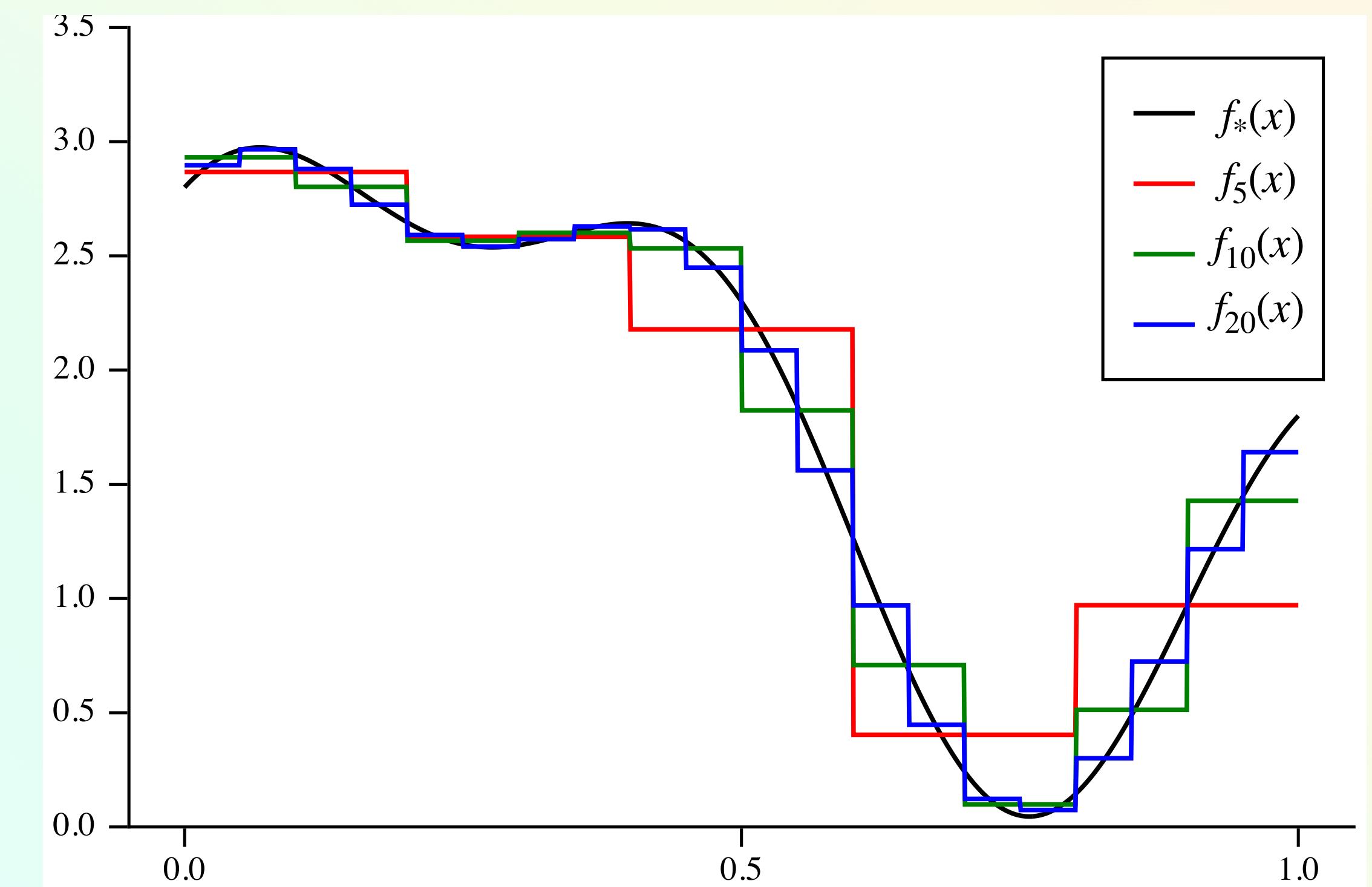
$\phi_i(x)$  is the output of  $i^{\text{th}}$  the basis function

For discrete approximation

$$\phi_i(x) = \begin{cases} 1 & x \in \text{bin}_i \\ 0 & \text{otherwise} \end{cases}$$

$\phi(x) = [0,0,1,0,0]^T$  – one-hot vector

$$f(x, w) = w^\top \phi(x) = \sum_{i=1}^n w_i \phi_i(x)$$



# LINEAR FUNCTION APPROXIMATION

RECAP

$$\phi(x) = [1, x, x^2, \cos(3\pi x), x \in [0.5, 0.6], \dots]^\top$$

# REGRESSION

## RECAP

Let  $X$  be a random variable representing an a draw of  $x$  from  $\mathcal{X}$

Let  $Y = f_*(X) + \xi$

Assume:  $\mathbf{E}[Y | X = x] = \underbrace{f_*(x) + \mathbf{E}[\xi | X = x]}_{=0} = f_*(x)$

# REGRESSION

## RECAP

Loss function (mean squared error):

$$l(w) \doteq \mathbf{E} \left[ (f(X, w) - Y)^2 \right]$$

“On average, how far away is the estimate from the samples  $Y$ ”

# LEAST SQUARES

RECAP

Find optimal weights:

$$w^* \in \arg \min_w l(w)$$

Solution:

$$\begin{aligned} w^* &= \underbrace{\mathbf{E} [\phi(X)\phi(X)^\top]^{-1}}_{=A} \underbrace{\mathbf{E} [Y\phi(X)]}_{=b} \\ &= A^{-1}b \end{aligned}$$

# FINDING THE BEST FIT

## GRADIENT DESCENT

Assume we have a data set of inputs and outputs:

$$x_1, x_2, \dots, x_m, y_1 = f(x_1), y_2 = f(x_2), \dots, y_m = f(x_m)$$

Loss function for weights  $w$  (how bad the approximation is)

$$l(w) = \frac{1}{m} \sum_{i=1}^m (f(x_i, w) - y_i)^2$$

# GRADIENT DESCENT ON $l(w)$

STOCHASTIC GRADIENT DESCENT

$$l(w) = \frac{1}{2} \frac{1}{m} \sum_{i=1}^m (f(x_i, w) - y_i)^2$$

$$\begin{aligned}\nabla l(w) &= \frac{\partial}{\partial w} \frac{1}{2} \frac{1}{m} \sum_{i=1}^m (y_i - f(x_i, w))^2 \\ &= \frac{1}{2} \frac{1}{m} \sum_{i=1}^m \frac{\partial}{\partial w} (f(x_i, w) - y_i)^2 = \frac{1}{m} \sum_{i=1}^m (f(x_i, w) - y_i) \frac{\partial f(x_i, w)}{\partial w}\end{aligned}$$

# GRADIENT DESCENT ON $l(w)$

STOCHASTIC GRADIENT DESCENT

$$l(w) = \frac{1}{2} \frac{1}{m} \sum_{i=1}^m (f(x_i, w) - y_i)^2$$

$$\begin{aligned}\nabla l(w) &= \frac{\partial}{\partial w} \frac{1}{2} \frac{1}{m} \sum_{i=1}^m (y_i - f(x_i, w))^2 \\ &= \frac{1}{2} \frac{1}{m} \sum_{i=1}^m \frac{\partial}{\partial w} (f(x_i, w) - y_i)^2 = \frac{1}{m} \sum_{i=1}^m (f(x_i, w) - y_i) \frac{\partial f(x_i, w)}{\partial w}\end{aligned}$$

Idea: move in the direction of the negative gradient

$$w \leftarrow w - \eta \nabla l(w)$$

# GRADIENT DESCENT ON $l(w)$

STOCHASTIC GRADIENT DESCENT

$$l(w) = \frac{1}{2} \frac{1}{m} \sum_{i=1}^m (f(x_i, w) - y_i)^2$$

$$\begin{aligned}\nabla l(w) &= \frac{\partial}{\partial w} \frac{1}{2} \frac{1}{m} \sum_{i=1}^m (y_i - f(x_i, w))^2 \\ &= \frac{1}{2} \frac{1}{m} \sum_{i=1}^m \frac{\partial}{\partial w} (f(x_i, w) - y_i)^2 = \frac{1}{m} \sum_{i=1}^m (f(x_i, w) - y_i) \frac{\partial f(x_i, w)}{\partial w}\end{aligned}$$

Idea: move in the direction of the negative gradient

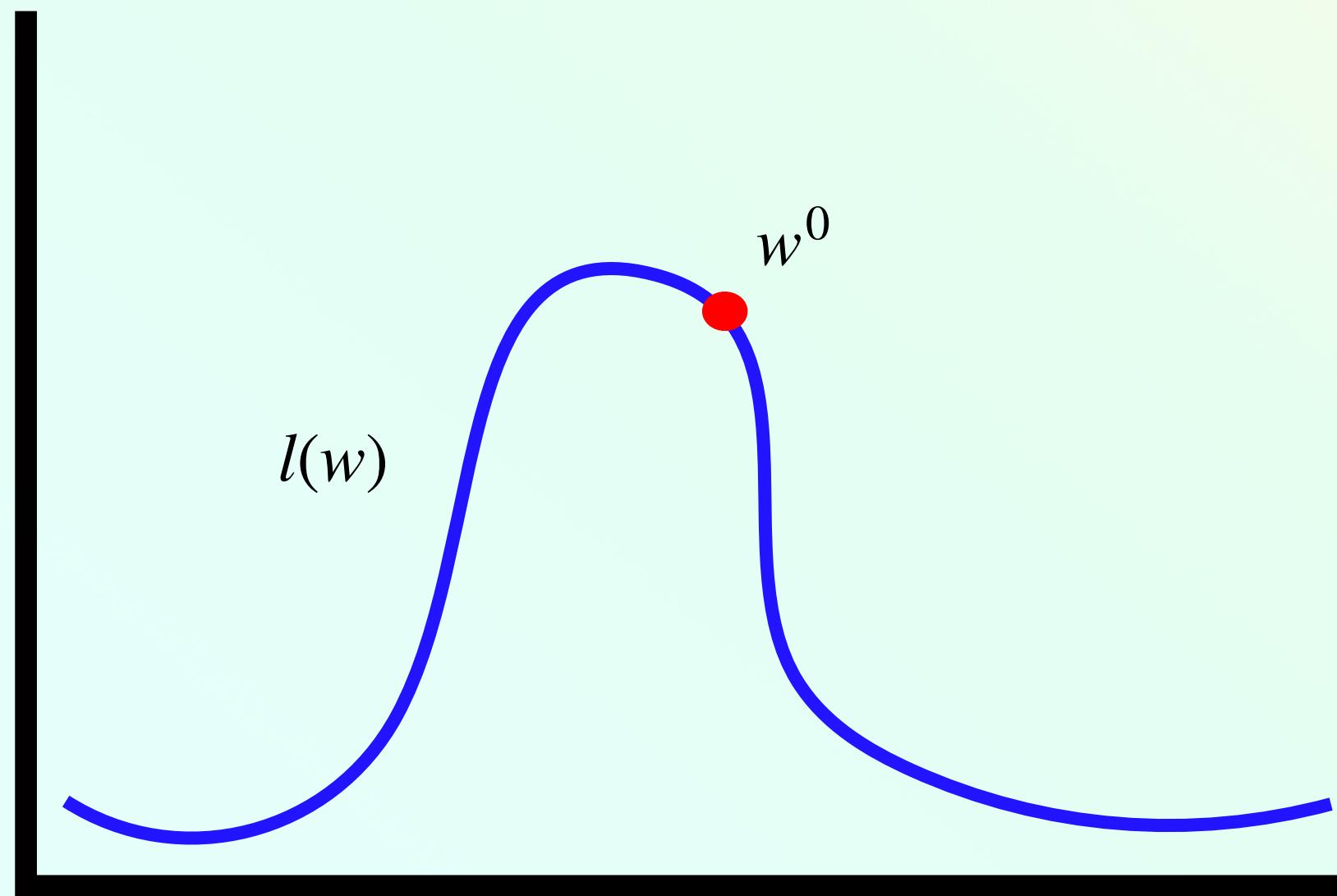
$$w \leftarrow w - \eta \nabla l(w)$$

$\eta$  is called the *step size*  
Some people refer to this as a  
*learning rate*

# GRADIENT DESCENT

PROCESS

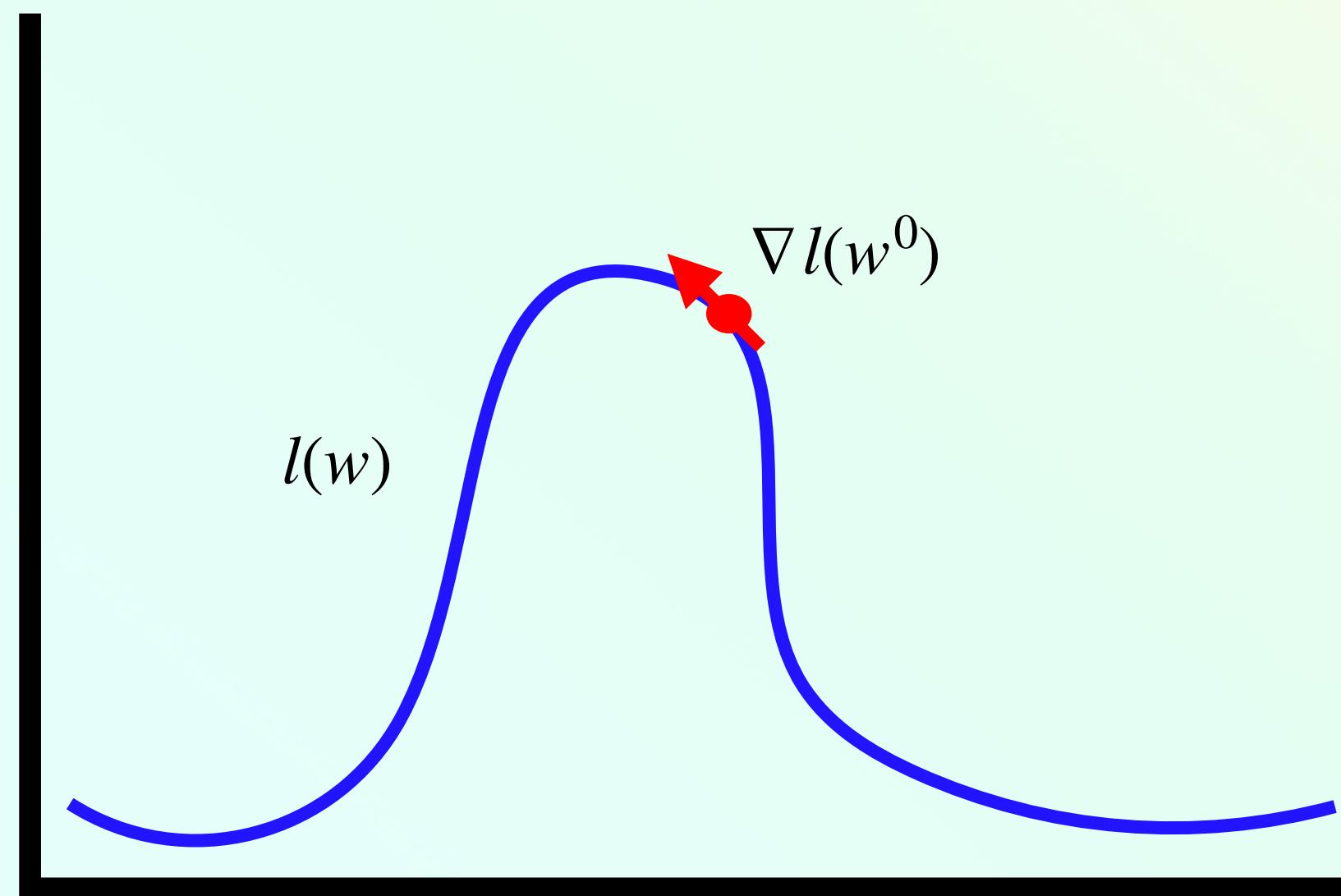
$$w^{k+1} = w^k - \eta \nabla l(w^k)$$



# GRADIENT DESCENT

PROCESS

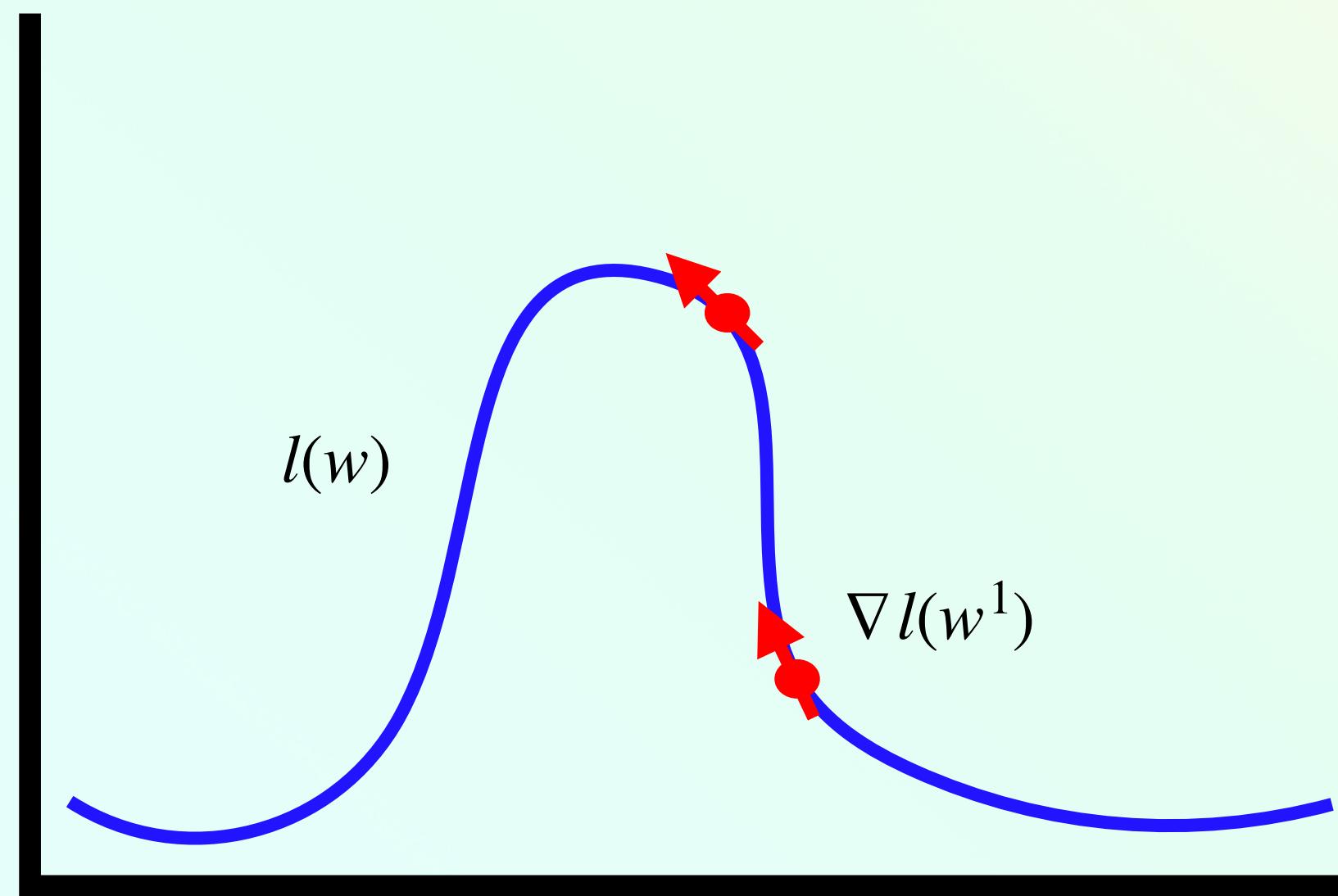
$$w^{k+1} = w^k - \eta \nabla l(w^k)$$



# GRADIENT DESCENT

PROCESS

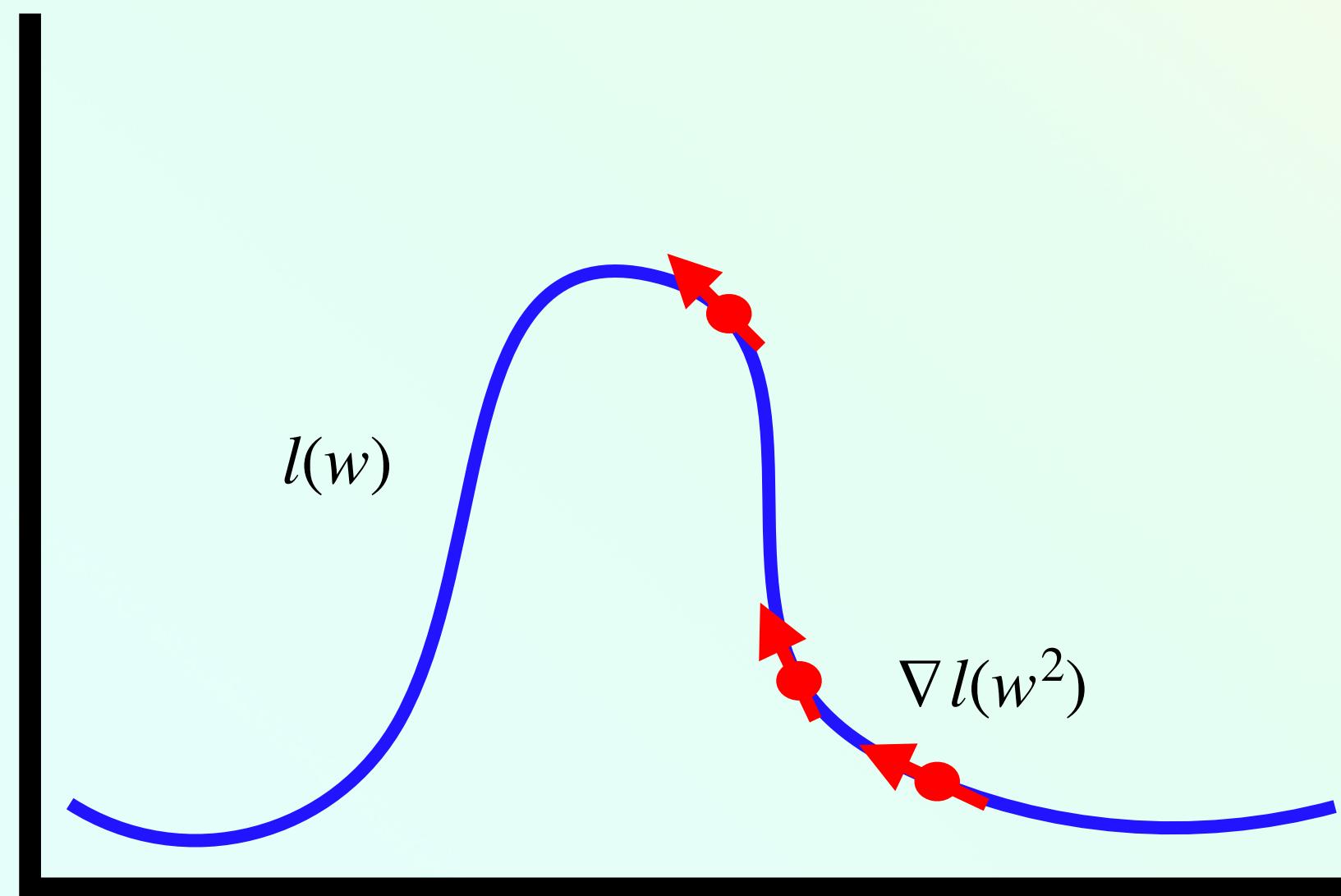
$$w^{k+1} = w^k - \eta \nabla l(w^k)$$



# GRADIENT DESCENT

PROCESS

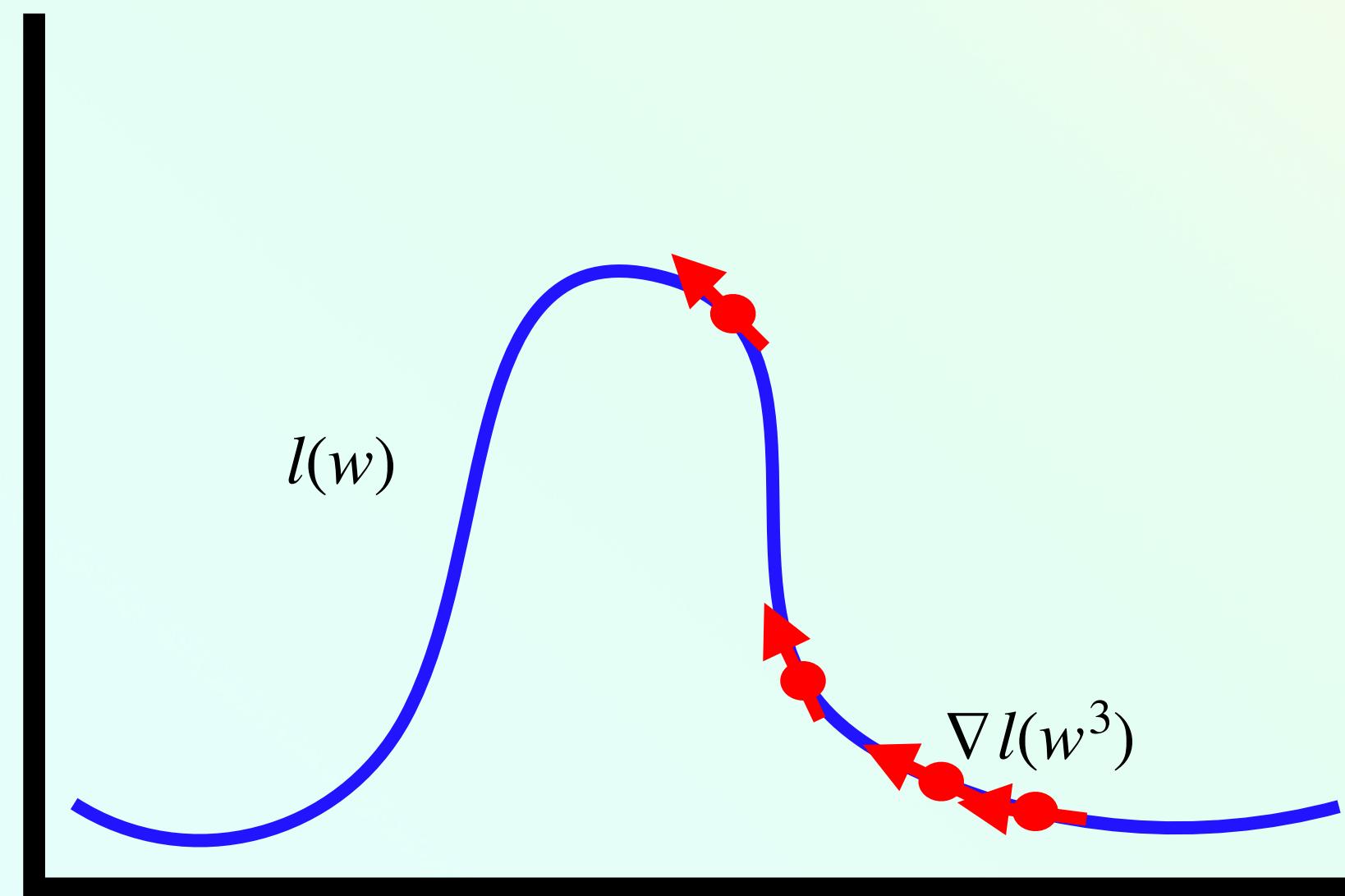
$$w^{k+1} = w^k - \eta \nabla l(w^k)$$



# GRADIENT DESCENT

PROCESS

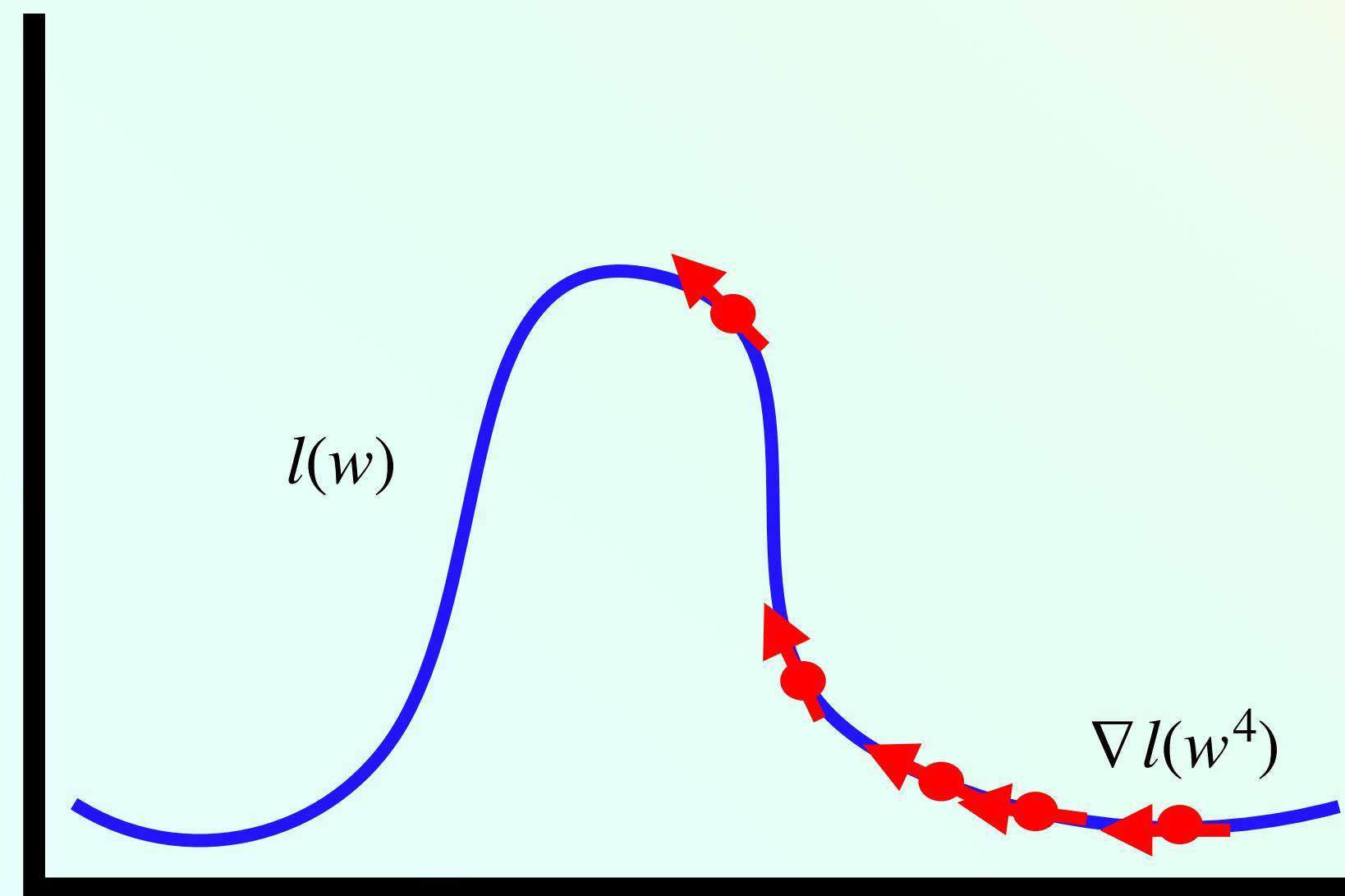
$$w^{k+1} = w^k - \eta \nabla l(w^k)$$



# GRADIENT DESCENT

PROCESS

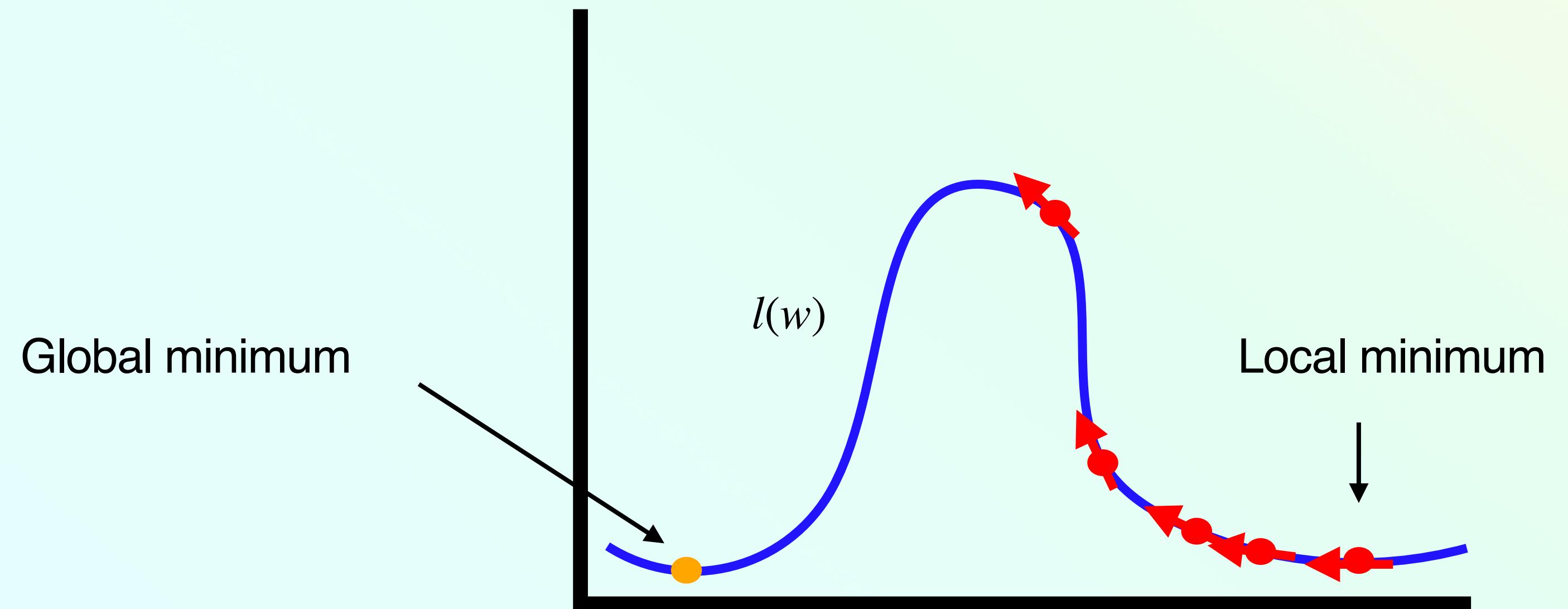
$$w^{k+1} = w^k - \eta \nabla l(w^k)$$



# GRADIENT DESCENT

PROCESS

$$w^{k+1} = w^k - \eta \nabla l(w^k)$$



# GRADIENT DESCENT

## PROPERTIES

- Only guaranteed to find a **local minimum** ← *Important*
- Some  $l$  only have one minimum (e.g.,  $l(w)$  is convex for all  $w$ ) → *If convex, it might work*

# GRADIENT DESCENT

## PROPERTIES

- Only guaranteed to find a **local minimum**
  - Some  $l$  only have one minimum (e.g.,  $l(w)$  is convex for all  $w$ )
- $l$  must be differentiable at least once
- Works best when  $\nabla l(w)$  is smooth

$$\bullet \quad \forall w, w' \frac{\|\nabla l(w) - \nabla l(w')\|_2}{\|w - w'\|_2} \leq L$$

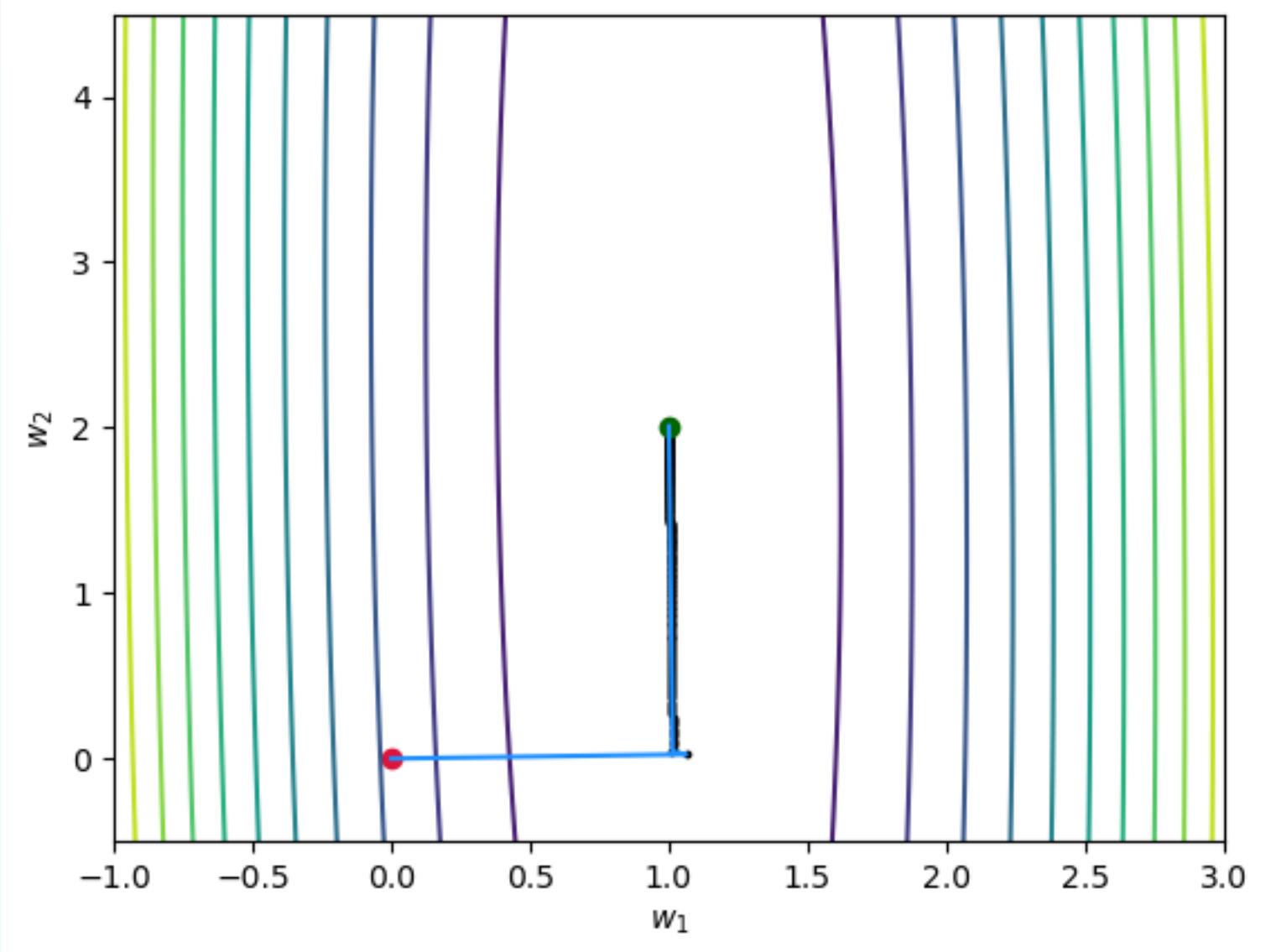
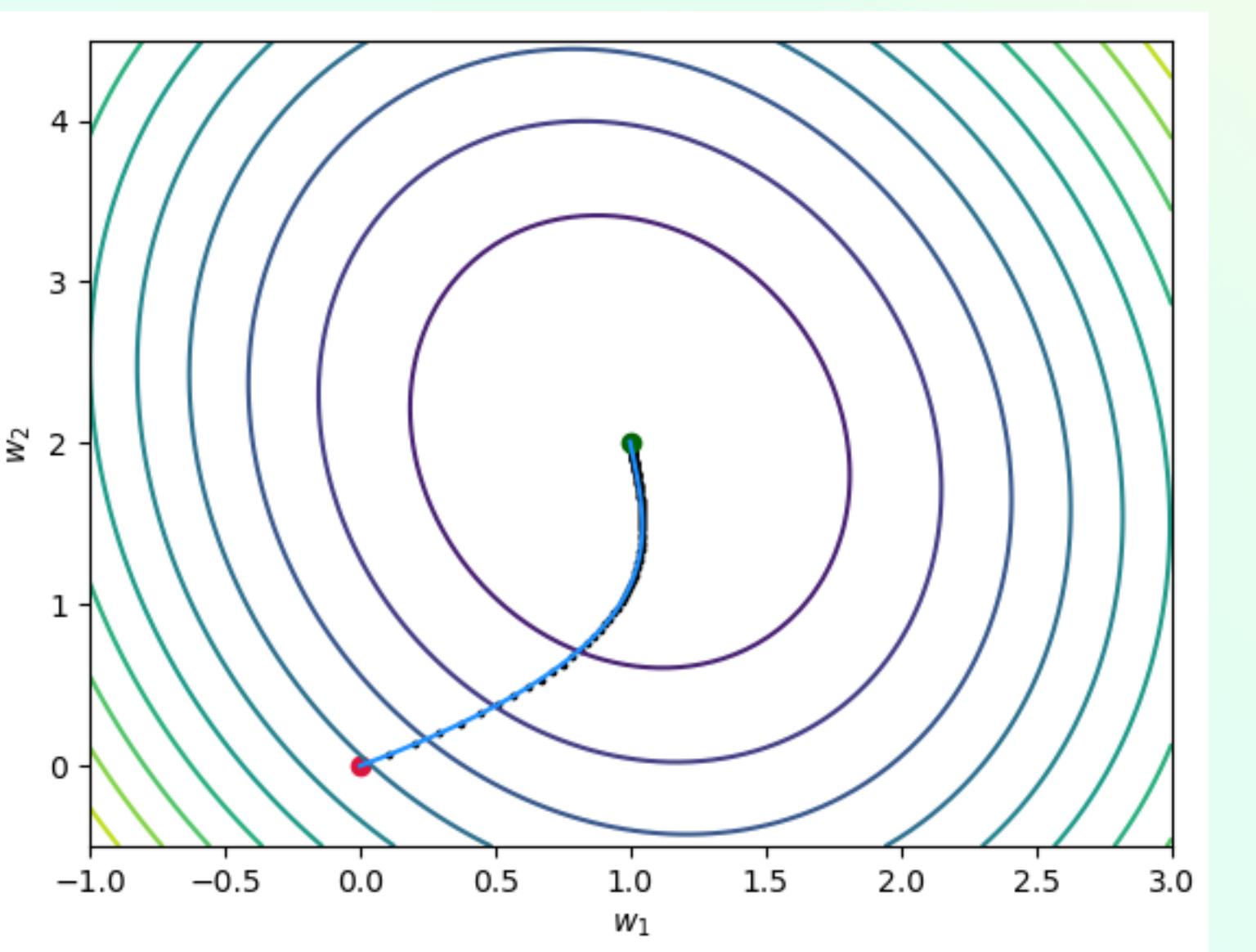
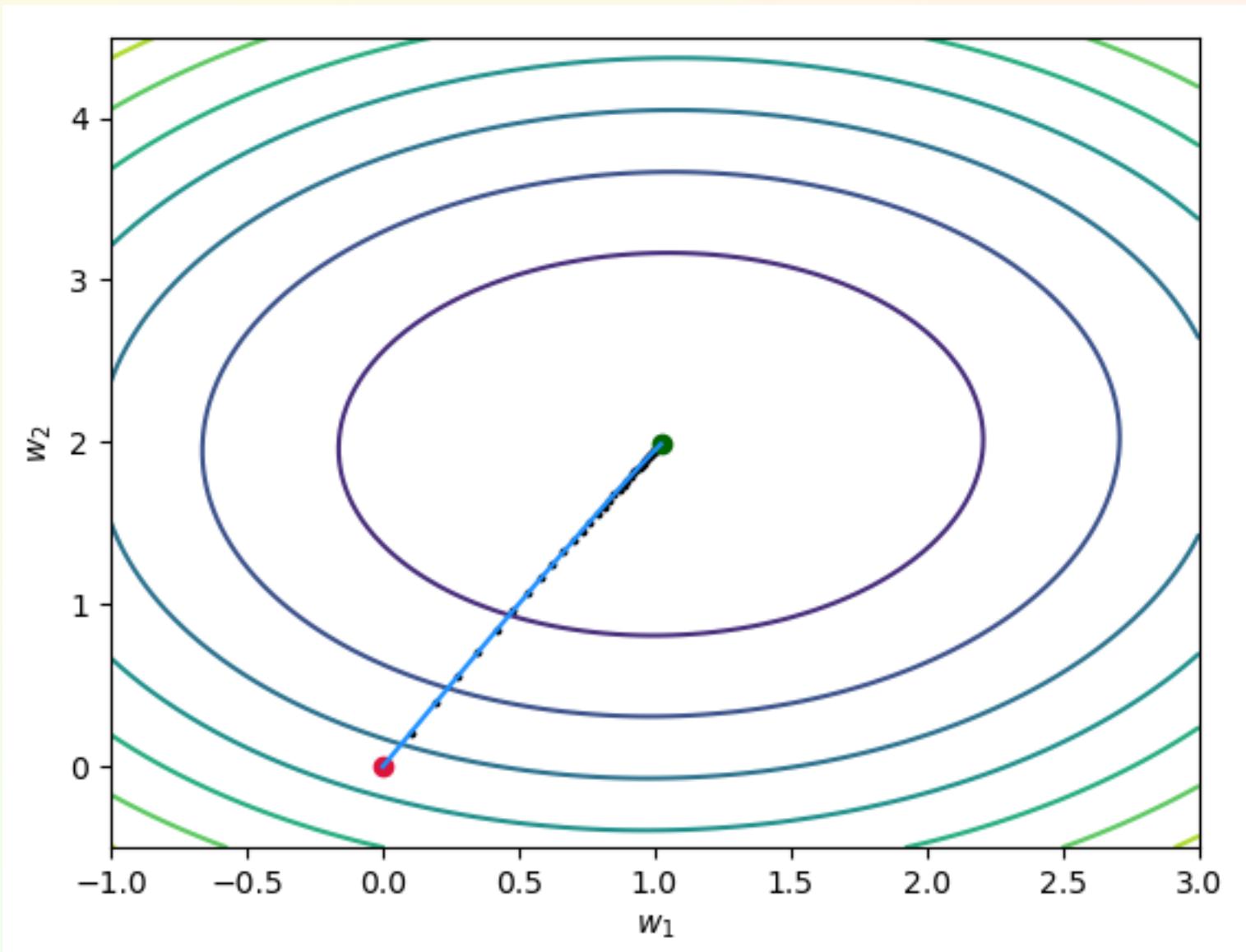
$$\|x\|_2 \doteq \sqrt{\sum_{i=1}^n x_i^2}$$

- Small change in  $w$  implies small change in  $\nabla l(w)$
- No discrete jumps in  $l(w)$

# LINEAR REGRESSION

LOSS FUNCTION HAS ONE MINIMIZER

- The loss function looks like a bowl
- It can be squashed or stretched in different directions

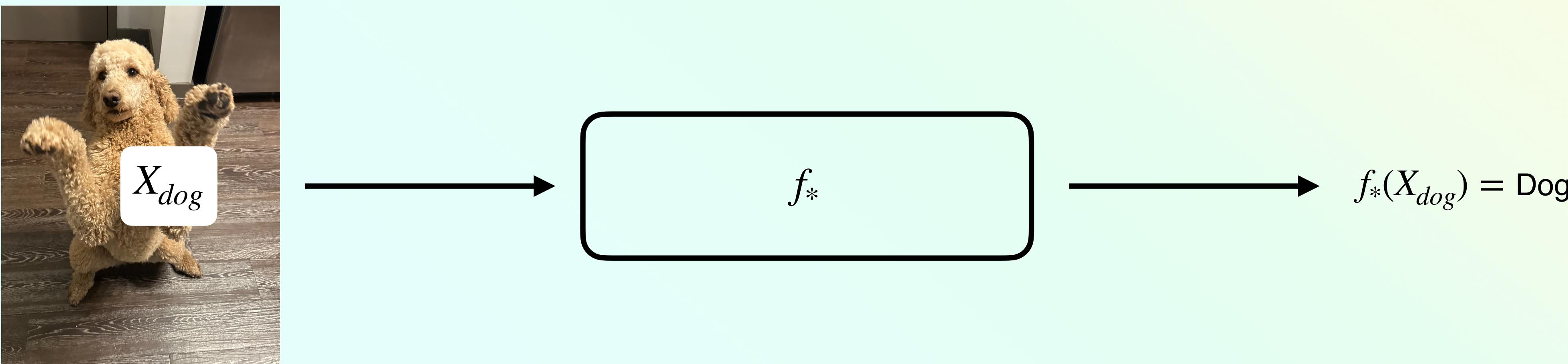
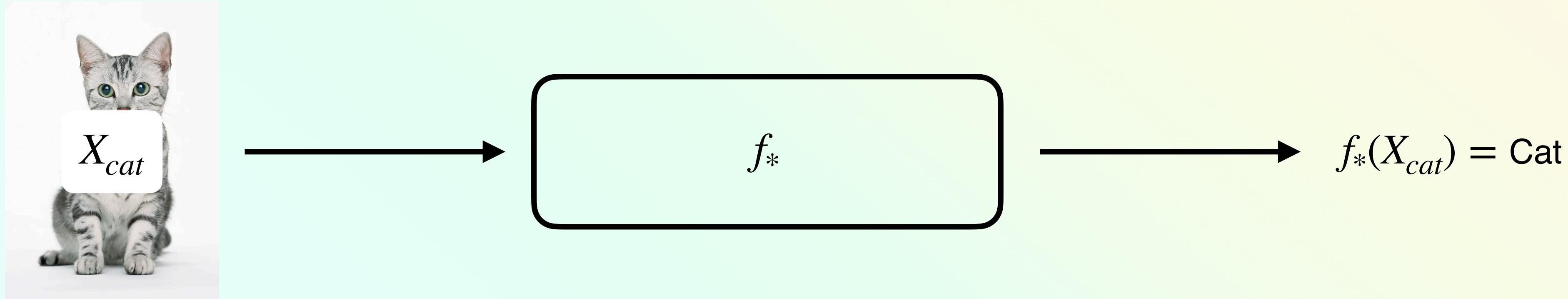


# QUIZ

- o A couple of questions here:
  - how the gradient descent approach is different from the least squares method

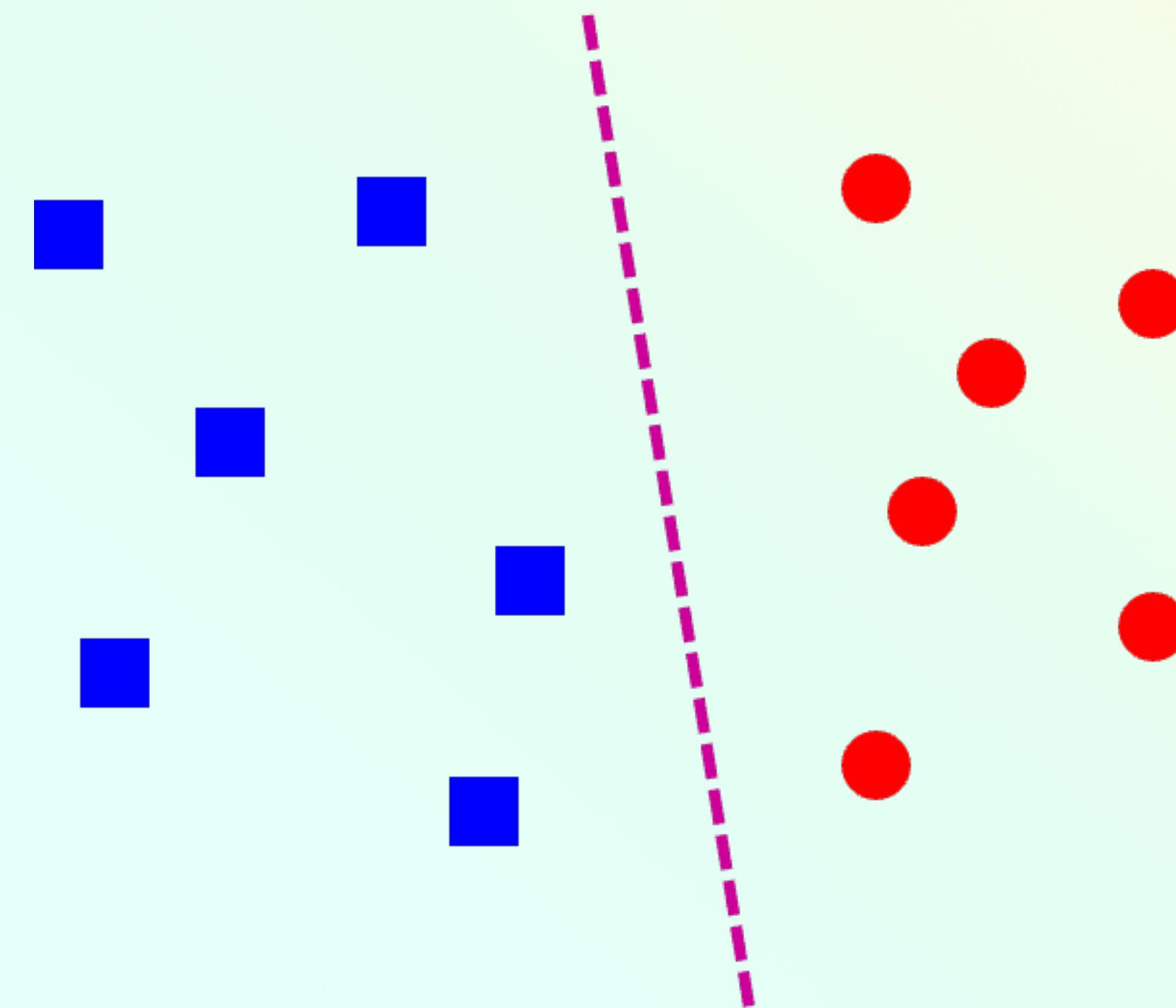
# CLASSIFICATION

## OVERVIEW



# LINEAR CLASSIFICATION

## OVERVIEW



# LINEAR CLASSIFICATION

## OVERVIEW

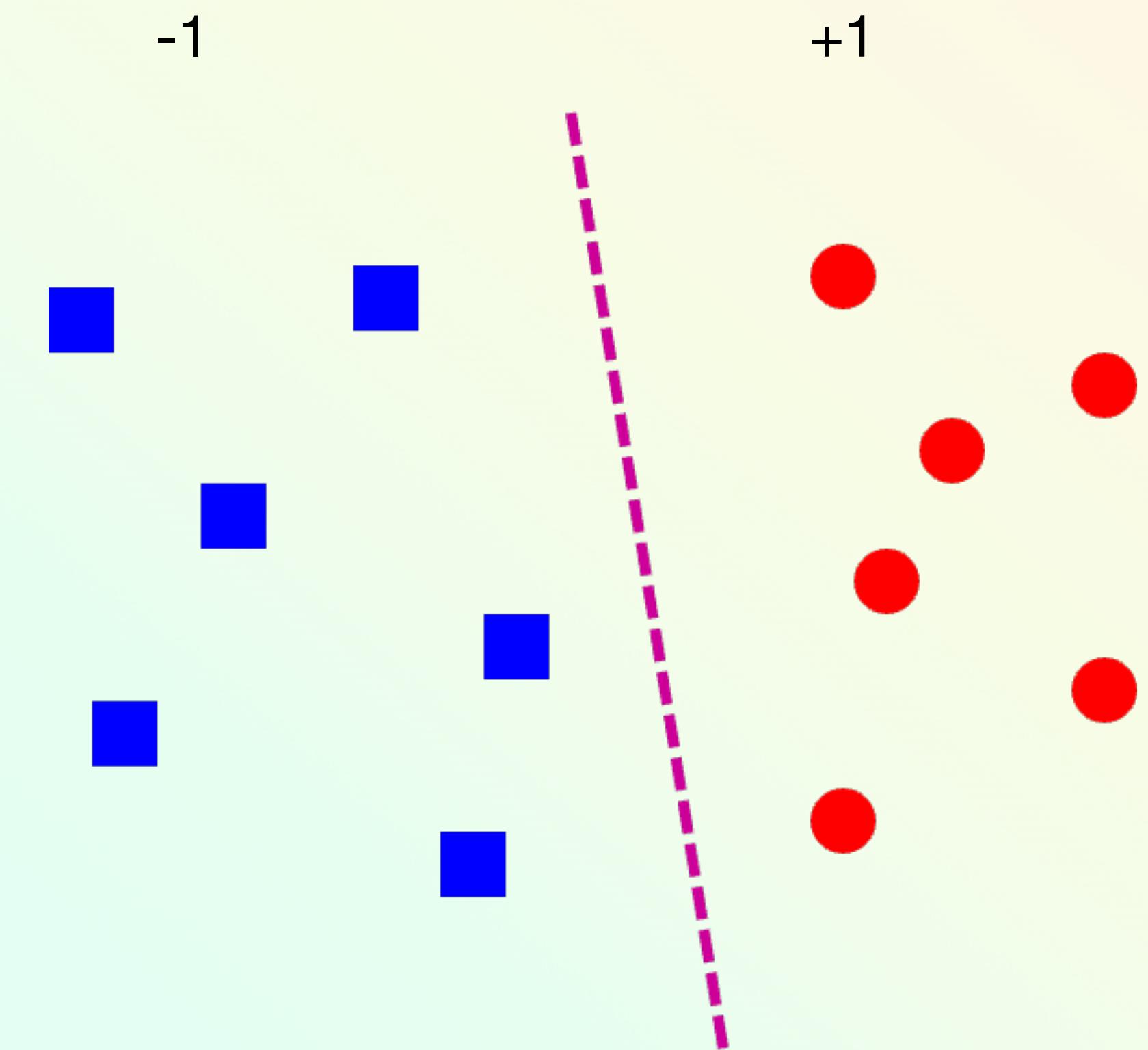
$$\mathcal{X} = \mathbb{R}^2$$

$$\mathcal{Y} = \{-1, +1\}$$

$$f: \mathcal{X} \times \mathbb{R}^2 \rightarrow \mathcal{Y}$$

what if we use the sgn as  
an approximate function

~~triangle~~

$$f(x, w) = \text{sgn}(w^\top x)$$


# LINEAR CLASSIFICATION

## OVERVIEW

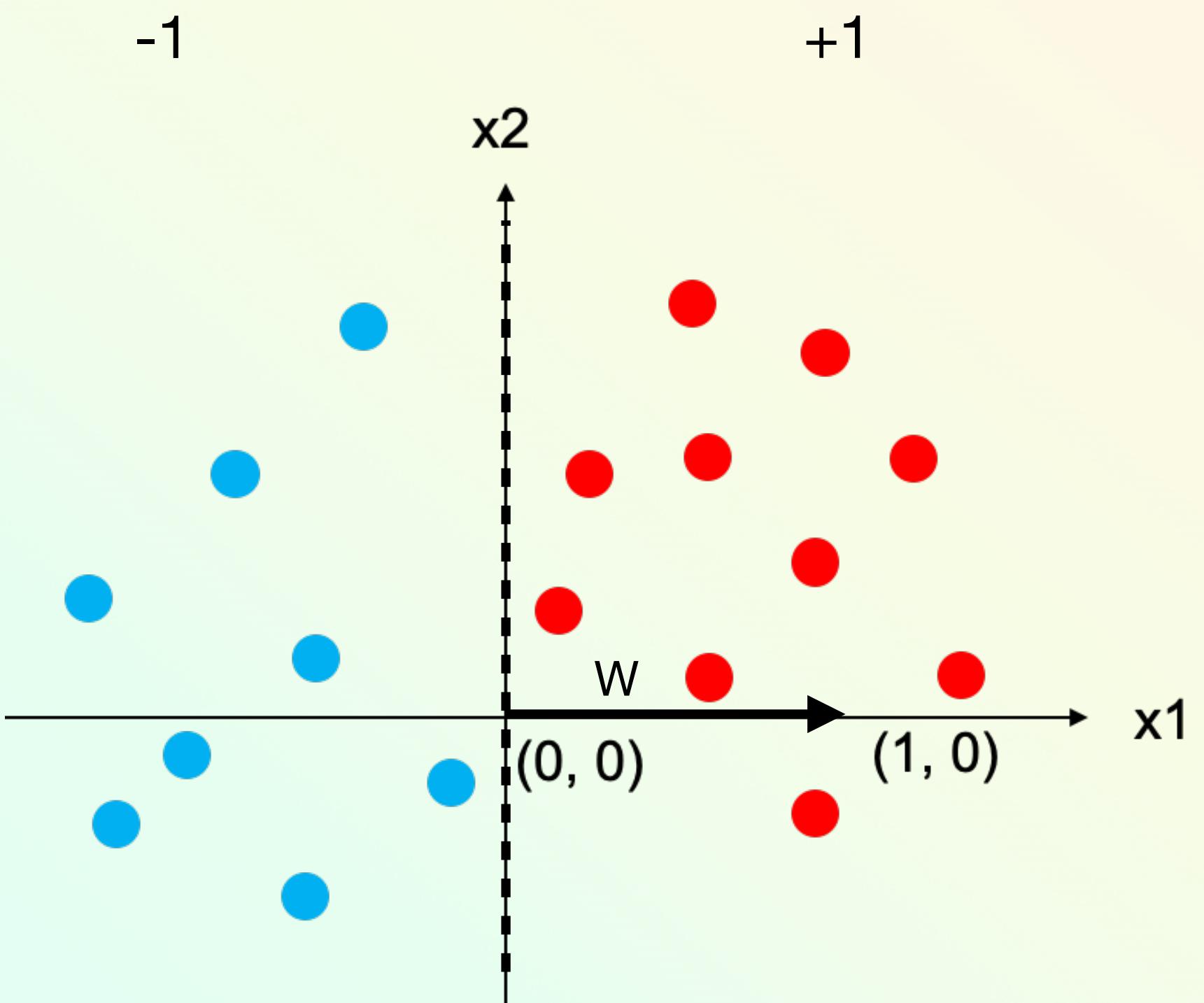
$$\mathcal{X} = \mathbb{R}^2$$

$$\mathcal{Y} = \{-1, +1\}$$

$$f: \mathcal{X} \times \mathbb{R}^2 \rightarrow \mathcal{Y}$$

$$f(x, w) = \text{sgn}(w^\top x) = \text{sgn}(w_1 x_1 + w_2 x_2)$$

What should the weights  $w$  be?



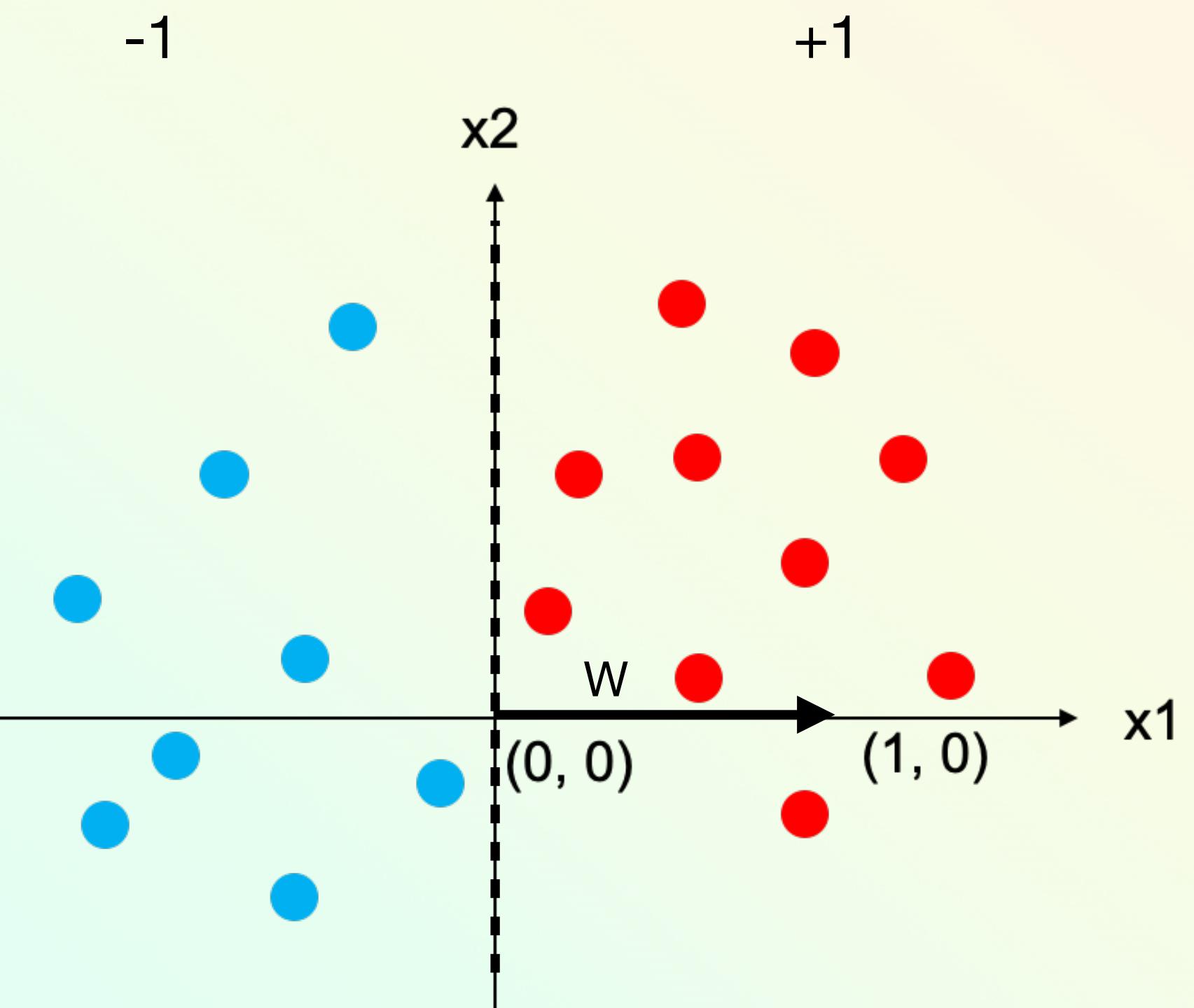
# LINEAR CLASSIFICATION

## OVERVIEW

$$f(x, w) = \text{sgn}(w^\top x) = \text{sgn}(w_1 x_1 + w_2 x_2)$$

What should the weights  $w$  be?

$$w = \begin{bmatrix} \alpha \\ 0 \end{bmatrix} \text{ for any } \alpha > 0$$



# LINEAR CLASSIFICATION

## OVERVIEW

$$f(x, w) = \text{sgn}(w^\top x) = \text{sgn}(w_1 x_1 + w_2 x_2)$$

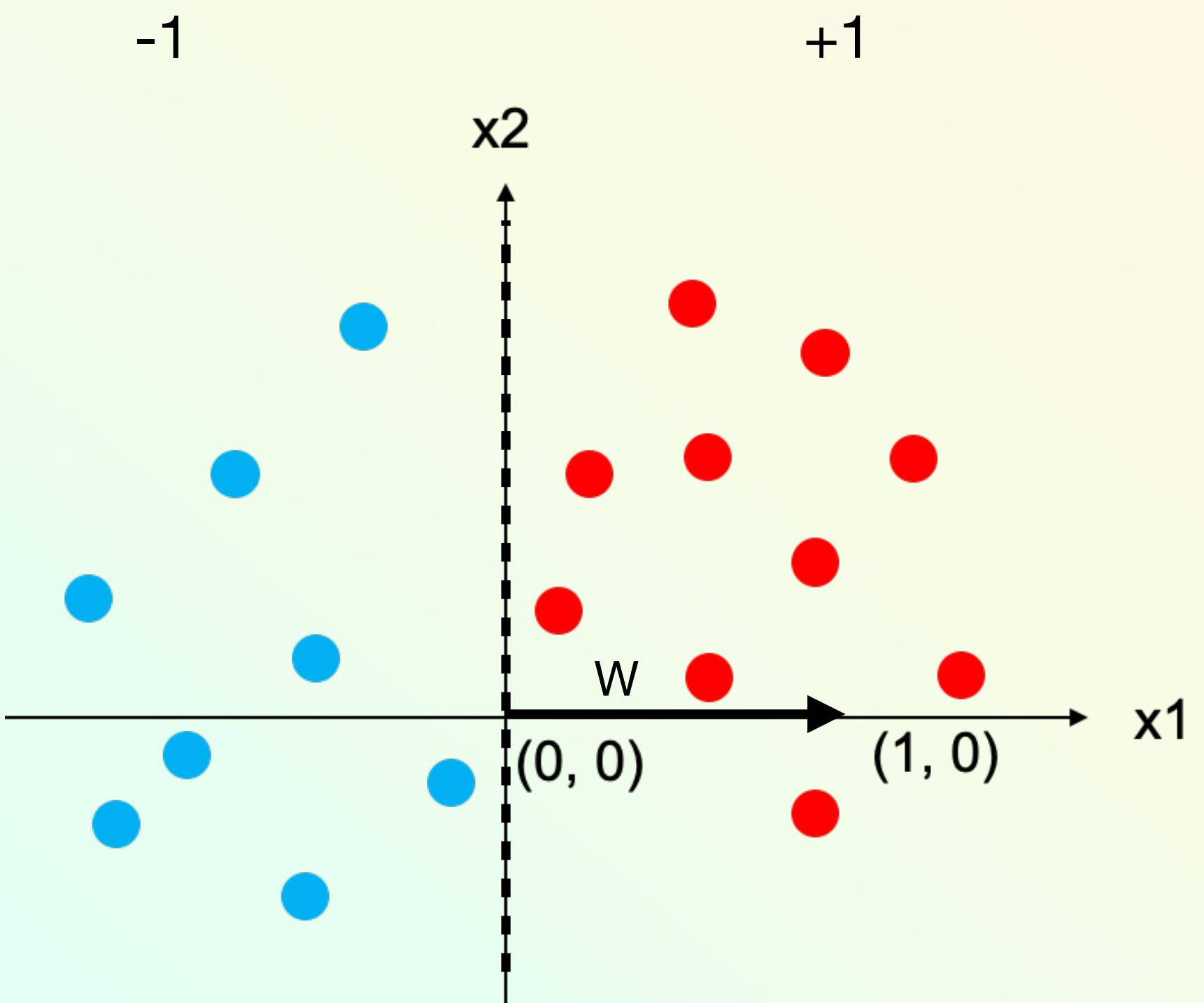
What should the weights  $w$  be?

$$w = \begin{bmatrix} \alpha \\ 0 \end{bmatrix} \text{ for any } \alpha > 0$$

$$w^\top \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \alpha > 0$$

+1 class

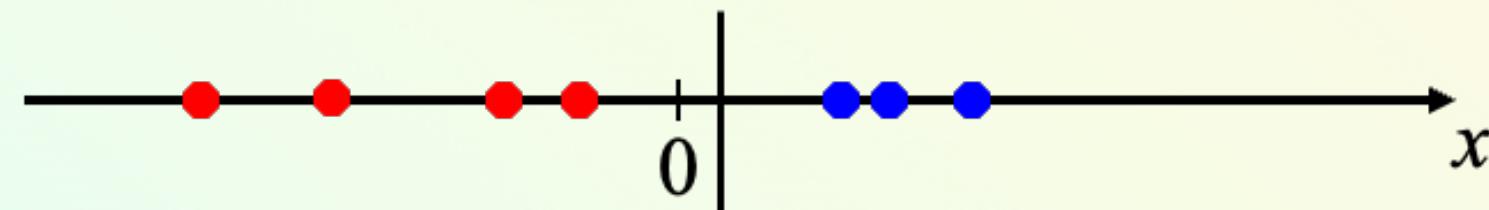
$$w^\top \begin{bmatrix} -0.1 \\ 100 \end{bmatrix} = -0.1\alpha < 0 \quad -1 \text{ class}$$



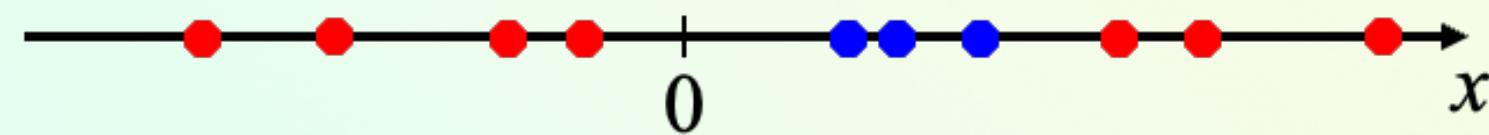
# THE IMPORTANCE OF BASIS FUNCTIONS

BASIS EXPANSION (MAPPING TO HIGHER DIMENSIONS)

Some data is linearly separable:

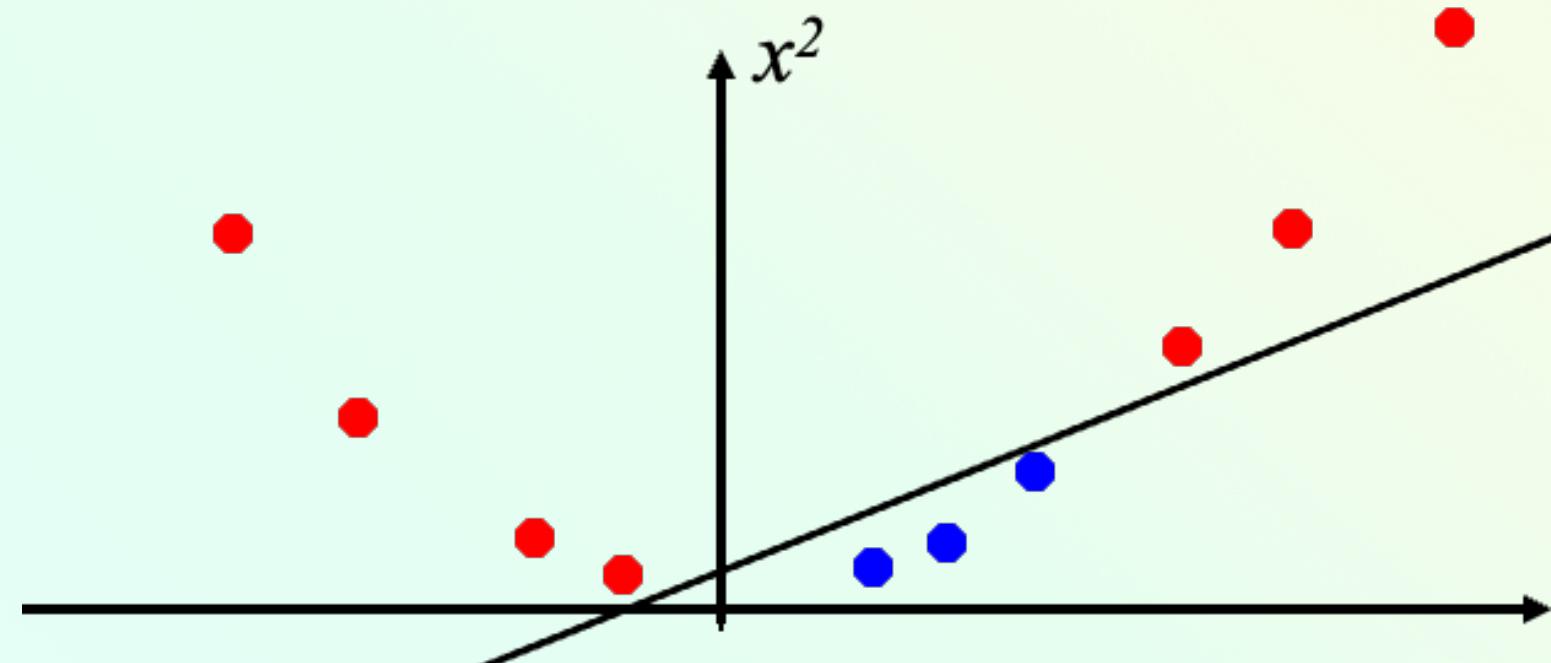


Some is not:



Data becomes separable in higher dimensions

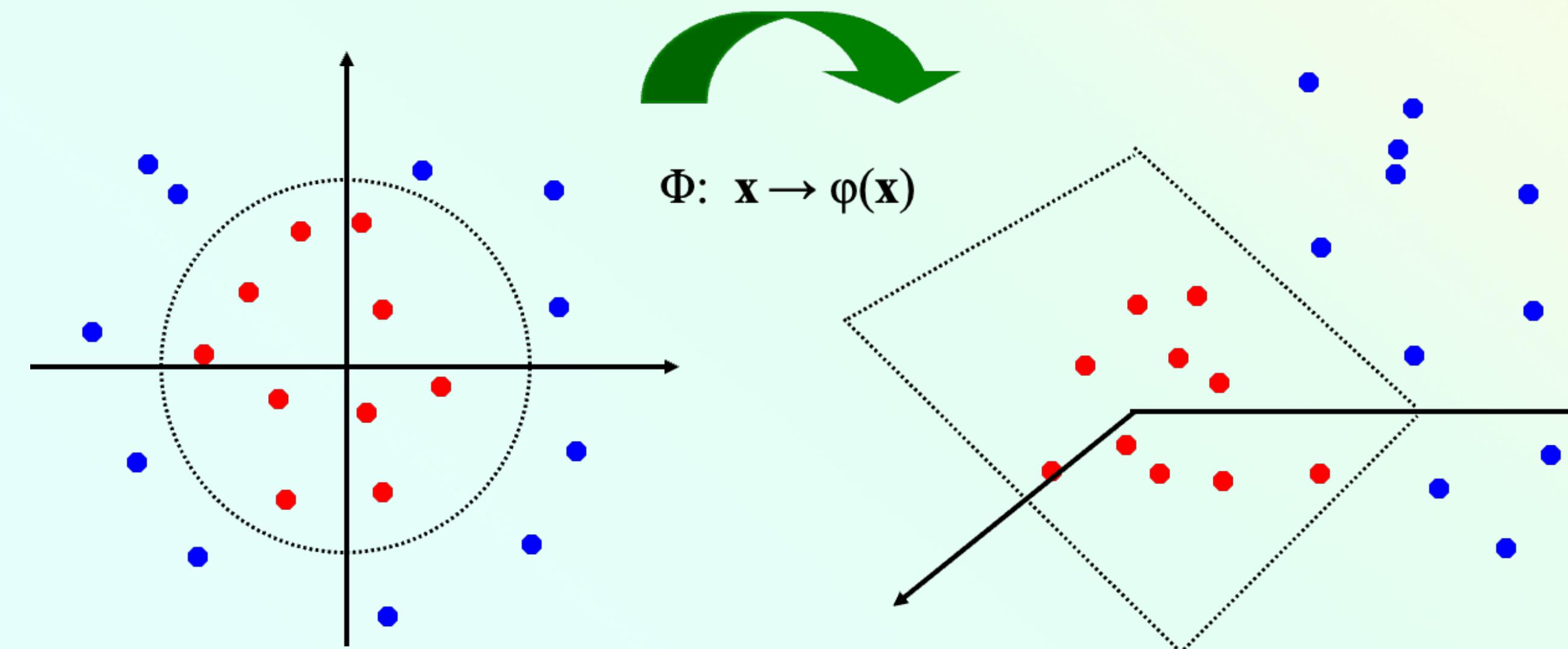
$$\phi(x) = \begin{bmatrix} x \\ x^2 \end{bmatrix}$$



# THE IMPORTANCE OF BASIS FUNCTIONS

## BASIS EXPANSION (MAPPING TO HIGHER DIMENSIONS)

General idea: the original input space can always be mapped to some higher-dimensional feature space where the training set is separable.

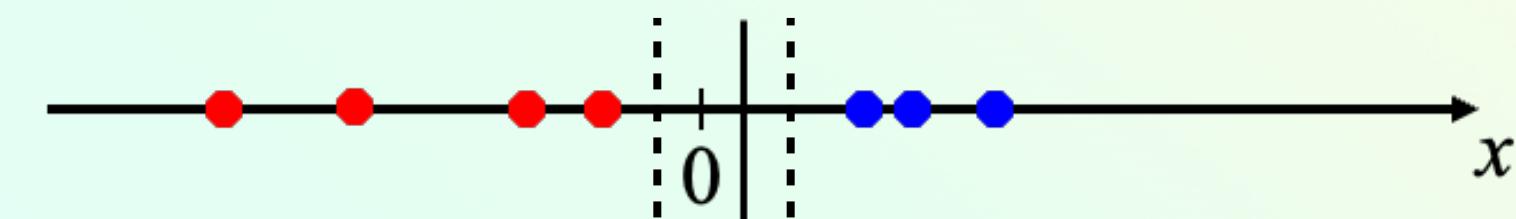


# WHICH LINES

## MAXIMUM MARGIN

Many lines that separate the data

Which one do we want?



Heuristic: One that separates the data with the most distance between classes.

This line is called the maximum margin line

# LINEAR CLASSIFICATION

## FINDING THE WEIGHTS

$$D = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$$

$\mathbf{x}_i$  are the features for the  $i^{\text{th}}$  data point

$y_i \in \{-1, 1\}$  is the class label for the  $i^{\text{th}}$  data point

$$\text{Minimize } \frac{1}{2} \mathbf{w}^\top \mathbf{w}$$

$$\text{Subject to } \forall i, y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1$$

Classifier is:  $\text{sgn}(\mathbf{w}^\top \mathbf{x} + b - 1)$

# LINEAR CLASSIFICATION

## FINDING THE WEIGHTS

$$D = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$$

$\mathbf{x}_i$  are the features for the  $i^{\text{th}}$  data point

$y_i \in \{-1, 1\}$  is the class label for the  $i^{\text{th}}$  data point

$$\text{Minimize } \frac{1}{2} \mathbf{w}^\top \mathbf{w}$$

$$\text{Subject to } \forall i, y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1$$

Classifier is:  $\text{sgn}(\mathbf{w}^\top \mathbf{x} + b - 1)$

Only consider the  $w$  that correctly classify all data points

# LINEAR CLASSIFICATION

## FINDING THE WEIGHTS

$$D = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$$

$\mathbf{x}_i$  are the features for the  $i^{\text{th}}$  data point

$y_i \in \{-1, 1\}$  is the class label for the  $i^{\text{th}}$  data point

$$\text{Minimize } \frac{1}{2} \mathbf{w}^\top \mathbf{w}$$

$$\text{Subject to } \forall i, y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1$$

Classifier is:  $\text{sgn}(\mathbf{w}^\top \mathbf{x} + b - 1)$

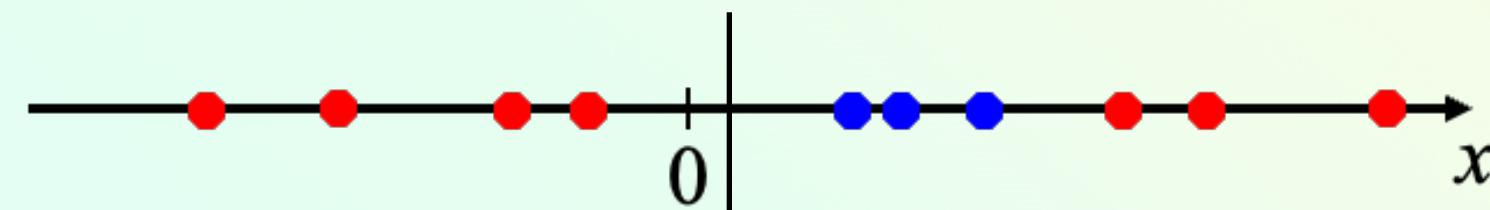
Finds the line with the smallest weights  $\rightarrow$  most distance between classes

Only consider the  $w$  that correctly classify all data points

# LINEAR CLASSIFICATION

## FINDING THE WEIGHTS

Sometimes it is not possible to separate all data points



Need to define a penalty for being misclassified

Soft-margin Support Vector Machines — not covered in this course

Create optimization problem to maximize margin and minimize classification penalty

# CLASSIFICATION

## DIFFERENT APPROACH

Deep learning approach:

1. Define a loss function  $l(w)$  to calculate the error in classification
2. Find a local minimum  $w^*$  of  $l(w)$  using (stochastic) gradient descent

# CLASSIFICATION

## CLASSIFICATION LOSS

Idea 1: +1 error for each miss classified data point

$$l(w) = - \sum_{i=1}^m \text{sgn}(w^\top \mathbf{x}_i) y_i$$

# CLASSIFICATION

## CLASSIFICATION LOSS

Idea 1: +1 error for each miss classified data point

$$l(w) = - \sum_{i=1}^m \text{sgn}(w^\top \mathbf{x}_i) y_i$$

What is  $\nabla l(w)$ ?

What is  $\frac{d}{dx} \text{sgn}(x)$ ?

# CLASSIFICATION

## CLASSIFICATION LOSS

Idea 1: +1 error for each miss classified data point

$$l(w) = - \sum_{i=1}^m \text{sgn}(w^\top \mathbf{x}_i) y_i$$

What is  $\nabla l(w)$ ?

$$\frac{d}{dx} \text{sgn}(x) = 0 \text{ for } x \neq 0$$

Not differentiable everywhere and the derivative has no direction to maximize function

# CLASSIFICATION

## CLASSIFICATION LOSS

Idea #2:

- treat labels as random variables,  $Y \in \{-1, +1\}$
- $\hat{Y}$  random variable representing the prediction from the model  $f$
- we want to know  $\Pr(\hat{Y} = +1 | X = x)$  for each  $x$

$$\Pr(\hat{Y} = -1 | X = x) = 1 - \Pr(\hat{Y} = +1 | X = x)$$

- Classification error is based on how unlikely the label is
  - $\Pr(\hat{Y} = y_i | X = x_i)$  If the probability is low under our model estimate, then that is bad

# MODELING PROBABILITIES

FUNCTION REPRESENTATION

$$f: \mathbb{R}^n \times \mathbb{R}^n \rightarrow [0,1]$$

$$f(x, w) \in [0,1]$$

$$f(x, w) = \sigma(w^\top x)$$

$$w^\top x \rightarrow \mathbb{R}$$

$$\sigma: \mathbb{R} \rightarrow [0,1]$$

# MODELING PROBABILITIES

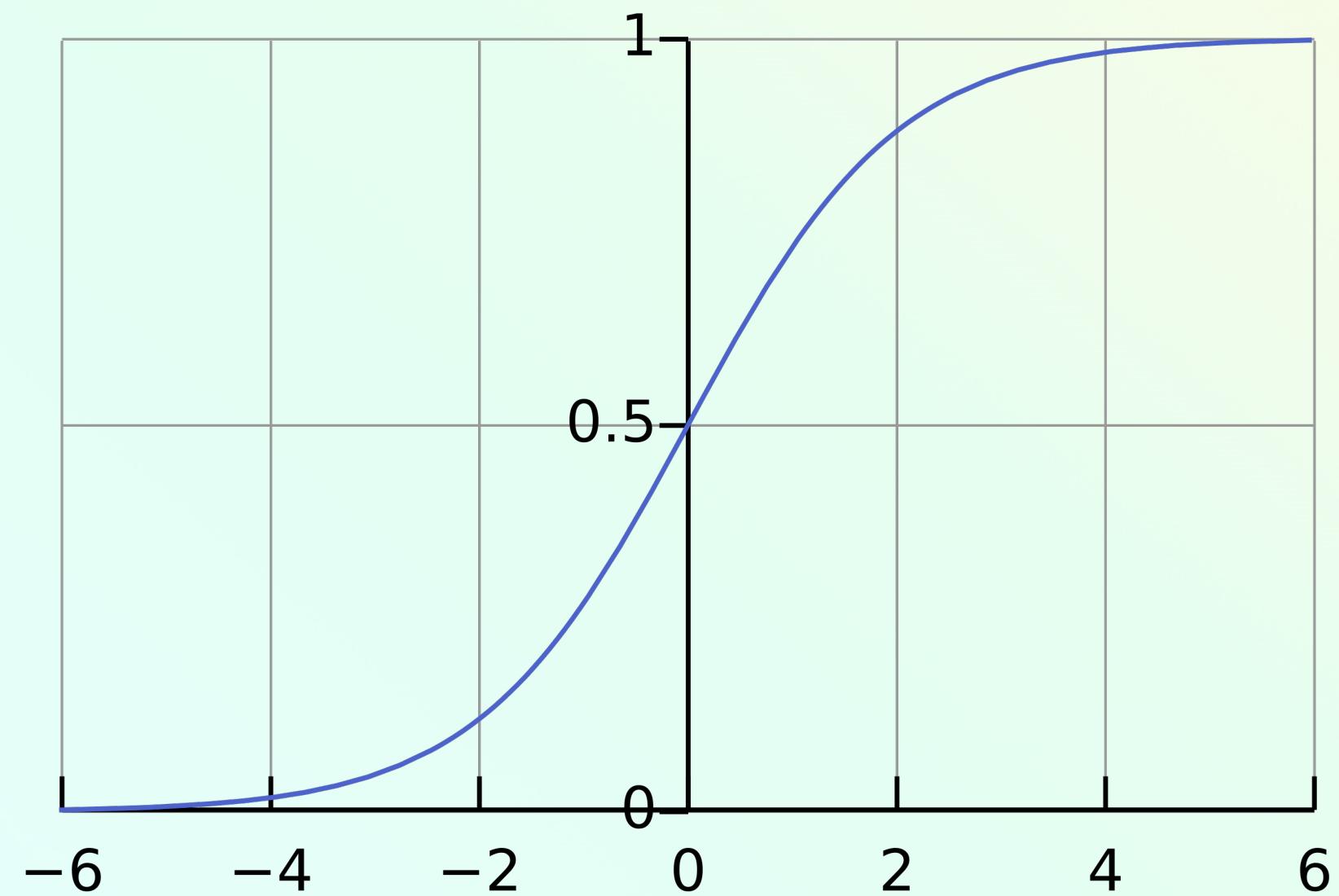
FUNCTION REPRESENTATION

$$f(x, w) = \sigma(w^\top x)$$

Sigmoid function:

$$\sigma(x) \doteq \frac{1}{1 + e^{-x}}$$

$$\sigma(x) \in (0,1)$$



# MODELING PROBABILITIES

FUNCTION REPRESENTATION

$$f(x, w) = \sigma(w^\top x) = \frac{1}{1 - e^{-w^\top x}}$$

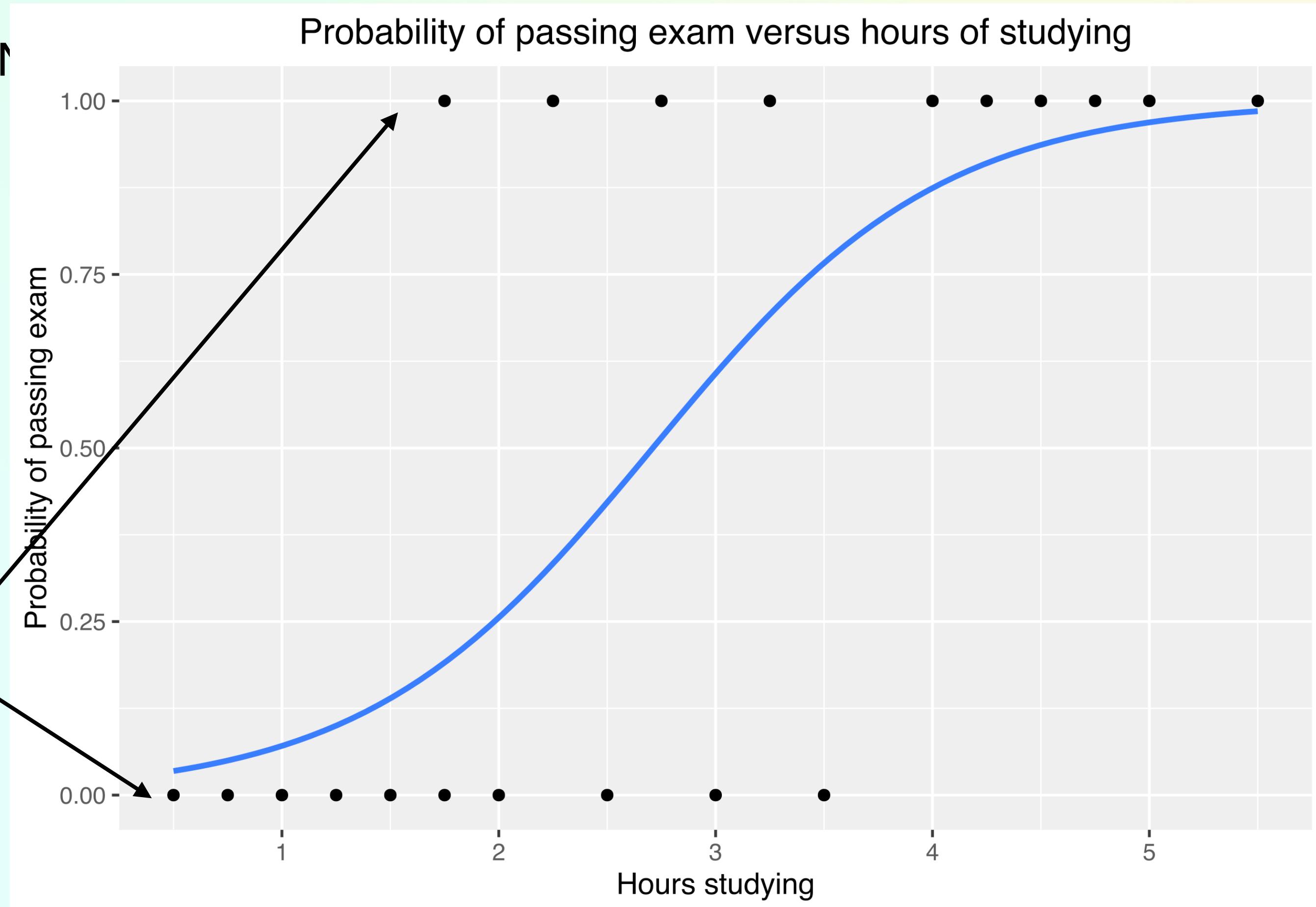
$$\Pr(\hat{Y} = +1 | X = x) \doteq f(x, w)$$

This is the probability under the model, not the actual probability  $\Pr(Y = +1 | X = x)$

# MODELING PROBABILITIES

FUNCTION REPRESENTATION

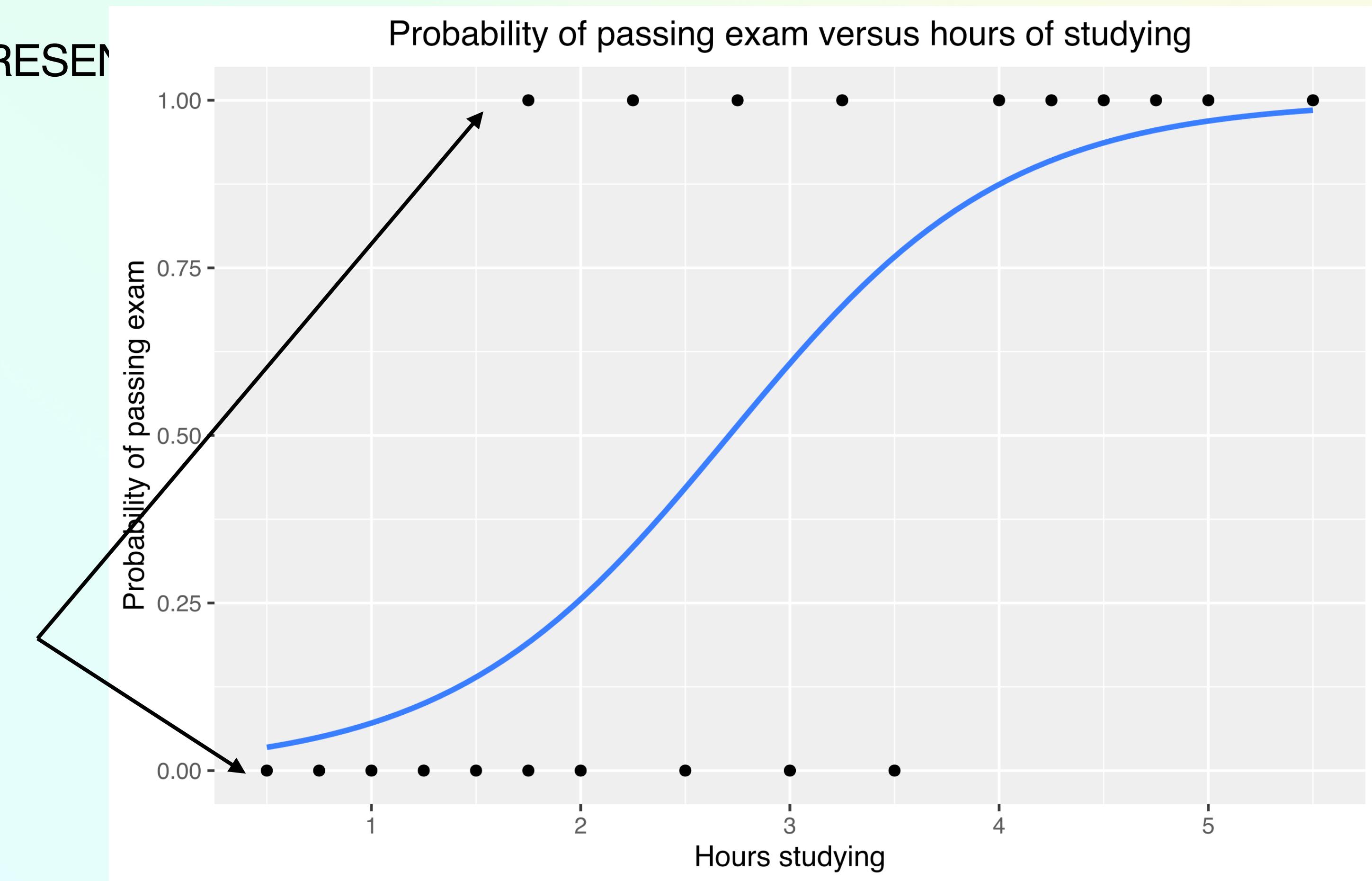
Y labels are  $\{0, 1\}$   
Instead of  $\{-1, 1\}$



# MODELING PROBABILITIES

FUNCTION REPRESENTATION

Y labels are  $\{0, 1\}$   
Instead of  $\{-1, 1\}$



Can view  $f(x, w)$  as a probability

More accurate to call  $f(x, w)$  model confidence

$f(x, w) \approx 1$  is very confident that  $Y = +1$

$f(x, w) \approx 0$  is very confident that  $Y = 0$

# CLASSIFICATION

## LOSS FUNCTION

Want to maximize the probability of all data being correctly classified

$$\Pr(\hat{Y} = y_1 | X = x_1) \Pr(\hat{Y} = y_2 | X = x_2) \dots \Pr(\hat{Y} = y_m | X = x_m) = \prod_{i=1}^m \Pr(\hat{Y} = y_i | X = x_i)$$

The product of all probabilities is in (0,1)

Maximum is reached when all probabilities are 1

$$l(w) \doteq - \prod_{i=1}^m \Pr(\hat{Y} = y_i | X = x_i)$$

# CLASSIFICATION

## LOSS FUNCTION

Minimize a loss function:

$$l(w) \doteq - \prod_{i=1}^m \Pr(\hat{Y} = y_i \mid X = x_i)$$

Minimize the negative product of probabilities

Equivalent to maximizing product of probabilities

# CLASSIFICATION

## LOSS FUNCTION

Gradient descent:

We need to compute  $\nabla l(w)$

Calculus — Product rule

$$\nabla l(w) = -\frac{\partial}{\partial w} \prod_{i=1}^m \Pr(\hat{Y} = y_i | X = x_i)$$

# CLASSIFICATION

## LOSS FUNCTION

$$\begin{aligned}\nabla l(w) &= -\frac{\partial}{\partial w} \prod_{i=1}^m \Pr(\hat{Y} = y_i | X = x_i) \\ &= -\left( \prod_{j=1, j \neq 1}^m \Pr(\hat{Y} = y_j | X = x_j) \right) \frac{\partial}{\partial w} \Pr(\hat{Y} = y_1 | X = x_1) - \left( \prod_{j=1, j \neq 2}^m \Pr(\hat{Y} = y_j | X = x_j) \right) \frac{\partial}{\partial w} \Pr(\hat{Y} = y_2 | X = x_2) - \dots\end{aligned}$$

# CLASSIFICATION

## LOSS FUNCTION

$$\begin{aligned}\nabla l(w) &= -\frac{\partial}{\partial w} \prod_{i=1}^m \Pr(\hat{Y} = y_i | X = x_i) \\ &= -\left( \prod_{j=1, j \neq 1}^m \Pr(\hat{Y} = y_j | X = x_j) \right) \frac{\partial}{\partial w} \Pr(\hat{Y} = y_1 | X = x_1) - \left( \prod_{j=1, j \neq 2}^m \Pr(\hat{Y} = y_j | X = x_j) \right) \frac{\partial}{\partial w} \Pr(\hat{Y} = y_2 | X = x_2) - \dots \\ &= -\sum_{i=1}^m \left( \prod_{j=1, j \neq i}^m \Pr(\hat{Y} = y_j | X = x_j) \right) \frac{\partial}{\partial w} \Pr(\hat{Y} = y_i | X = x_i)\end{aligned}$$

# CLASSIFICATION

## LOSS FUNCTION

For some function  $p(x) \in (0,1)$

$$\frac{\partial}{\partial x} \ln p(x) = \frac{1}{p(x)} \frac{\partial}{\partial x} p(x)$$

logarithm rule  $\rightarrow \frac{\partial}{\partial x} \ln x = \frac{1}{x}$  plus the chain rule

Rewrite this as:

$$\frac{\partial}{\partial x} p(x) = p(x) \frac{\partial}{\partial x} \ln p(x)$$

# CLASSIFICATION

## LOSS FUNCTION

$$\begin{aligned}\nabla l(w) &= - \sum_{i=1}^m \left( \prod_{j=1, j \neq i}^m \Pr(\hat{Y} = y_j | X = x_j) \right) \frac{\partial}{\partial w} \Pr(\hat{Y} = y_i | X = x_i) \\ &= - \sum_{i=1}^m \left( \prod_{j=1, j \neq i}^m \Pr(\hat{Y} = y_j | X = x_j) \right) \Pr(\hat{Y} = y_i | X = x_i) \frac{\partial}{\partial w} \ln \Pr(\hat{Y} = y_i | X = x_i)\end{aligned}$$

# CLASSIFICATION

## LOSS FUNCTION

$$\begin{aligned}\nabla l(w) &= - \sum_{i=1}^m \left( \prod_{j=1, j \neq i}^m \Pr(\hat{Y} = y_j | X = x_j) \right) \frac{\partial}{\partial w} \Pr(\hat{Y} = y_i | X = x_i) \\ &= - \sum_{i=1}^m \left( \prod_{j=1, j \neq i}^m \Pr(\hat{Y} = y_j | X = x_j) \right) \Pr(\hat{Y} = y_i | X = x_i) \frac{\partial}{\partial w} \ln \Pr(\hat{Y} = y_i | X = x_i) \\ &= - \sum_{i=1}^m \left( \prod_{j=1}^m \Pr(\hat{Y} = y_j | X = x_j) \right) \frac{\partial}{\partial w} \ln \Pr(\hat{Y} = y_i | X = x_i)\end{aligned}$$

# CLASSIFICATION

## LOSS FUNCTION

$$\begin{aligned}\nabla l(w) &= - \sum_{i=1}^m \left( \prod_{j=1, j \neq i}^m \Pr(\hat{Y} = y_j | X = x_j) \right) \frac{\partial}{\partial w} \Pr(\hat{Y} = y_i | X = x_i) \\ &= - \sum_{i=1}^m \left( \prod_{j=1, j \neq i}^m \Pr(\hat{Y} = y_j | X = x_j) \right) \Pr(\hat{Y} = y_i | X = x_i) \frac{\partial}{\partial w} \ln \Pr(\hat{Y} = y_i | X = x_i) \\ &= - \sum_{i=1}^m \left( \prod_{j=1}^m \Pr(\hat{Y} = y_j | X = x_j) \right) \frac{\partial}{\partial w} \ln \Pr(\hat{Y} = y_i | X = x_i) \\ &= - \left( \prod_{j=1}^m \Pr(\hat{Y} = y_j | X = x_j) \right) \sum_{i=1}^m \frac{\partial}{\partial w} \ln \Pr(\hat{Y} = y_i | X = x_i)\end{aligned}$$

# QUIZ

# CLASSIFICATION

## LOSS FUNCTION

$$\nabla l(w) = \underbrace{- \left( \prod_{j=1}^m \Pr(\hat{Y} = y_j | X = x_j) \right)}_{(a)} \sum_{i=1}^m \frac{\partial}{\partial w} \ln \Pr(\hat{Y} = y_i | X = x_i)$$

For any  $w$ ,  $(a) > 0$ . We can interpret this as rescaling the gradient, but not changing the direction, i.e., it does not really matter.

# CLASSIFICATION

## LOSS FUNCTION

Could instead define a different loss function  $l'$  that measure the log-likelihood

$$l'(w) \doteq -\ln \prod_{i=1}^m \Pr(\hat{Y} = y_i | X = x_i)$$

This is called the **negative log-likelihood** (NLL) and is a common loss function in machine learning

We do not change the optimal point because we are just rescaling the loss  $l$ , e.g.,

$$l'(w) = -\ln (-l(w))$$

# CLASSIFICATION

## LOSS FUNCTION

Logs turns products into summations!

$$\begin{aligned} l'(w) &= -\ln \prod_{i=1}^m \Pr(\hat{Y} = y_i | X = x_i) \\ &= -\sum_{i=1}^m \ln \Pr(\hat{Y} = y_i | X = x_i) \end{aligned}$$

# CLASSIFICATION

## LOSS FUNCTION

Simpler derivative

$$\begin{aligned}\nabla l'(w) &= -\frac{\partial}{\partial w} \sum_{i=1}^m \ln \Pr(\hat{Y} = y_i | X = x_i) \\ &= -\sum_{i=1}^m \frac{\partial}{\partial w} \ln \Pr(\hat{Y} = y_i | X = x_i)\end{aligned}$$

# CLASSIFICATION

## LOSS FUNCTION

Same direction as  $\nabla l(w)$

$$\begin{aligned}\nabla l(w) &= -(a) \sum_{i=1}^m \frac{\partial}{\partial w} \ln \Pr(\hat{Y} = y_i | X = x_i) \\ &= (a) \nabla l'(w)\end{aligned}$$

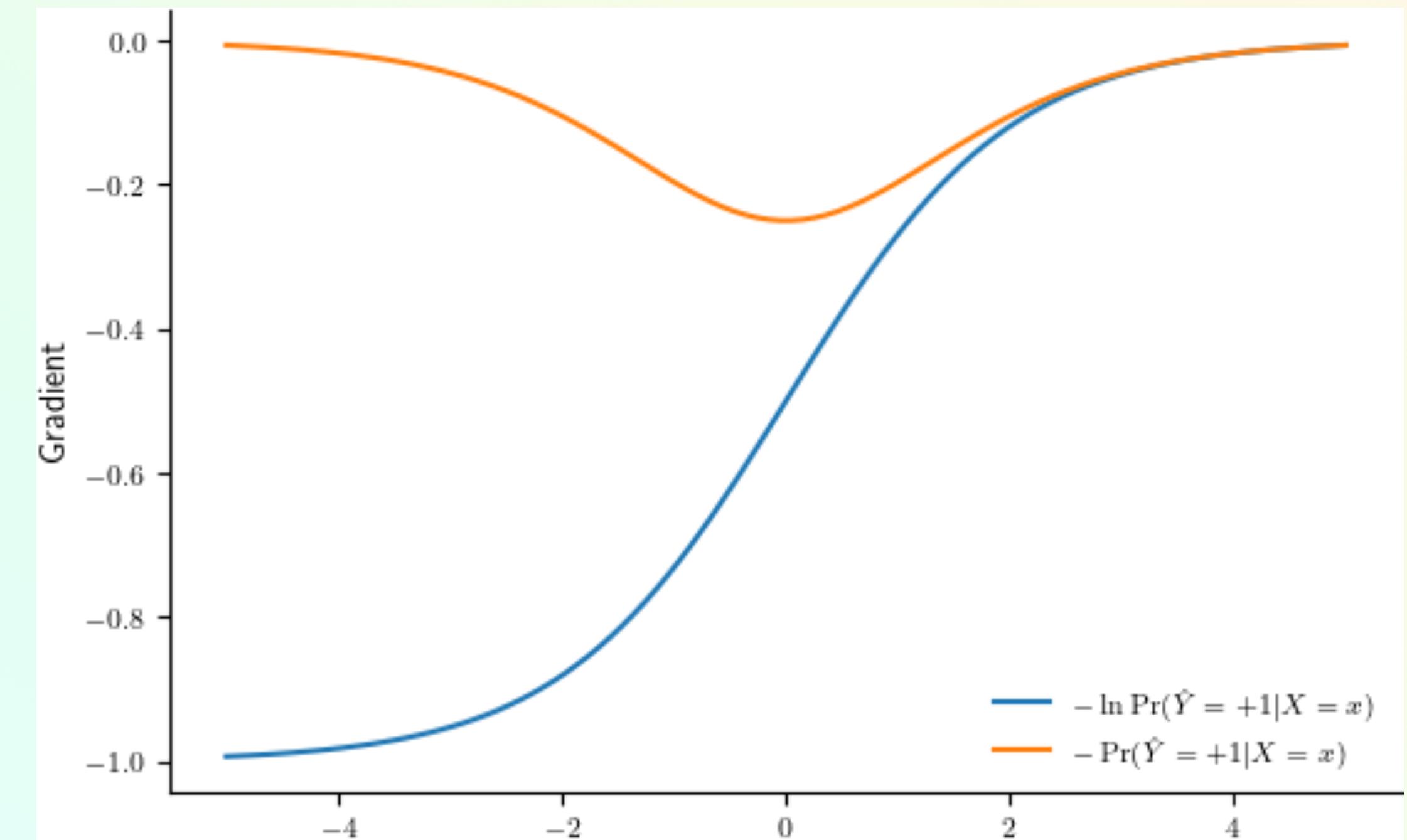
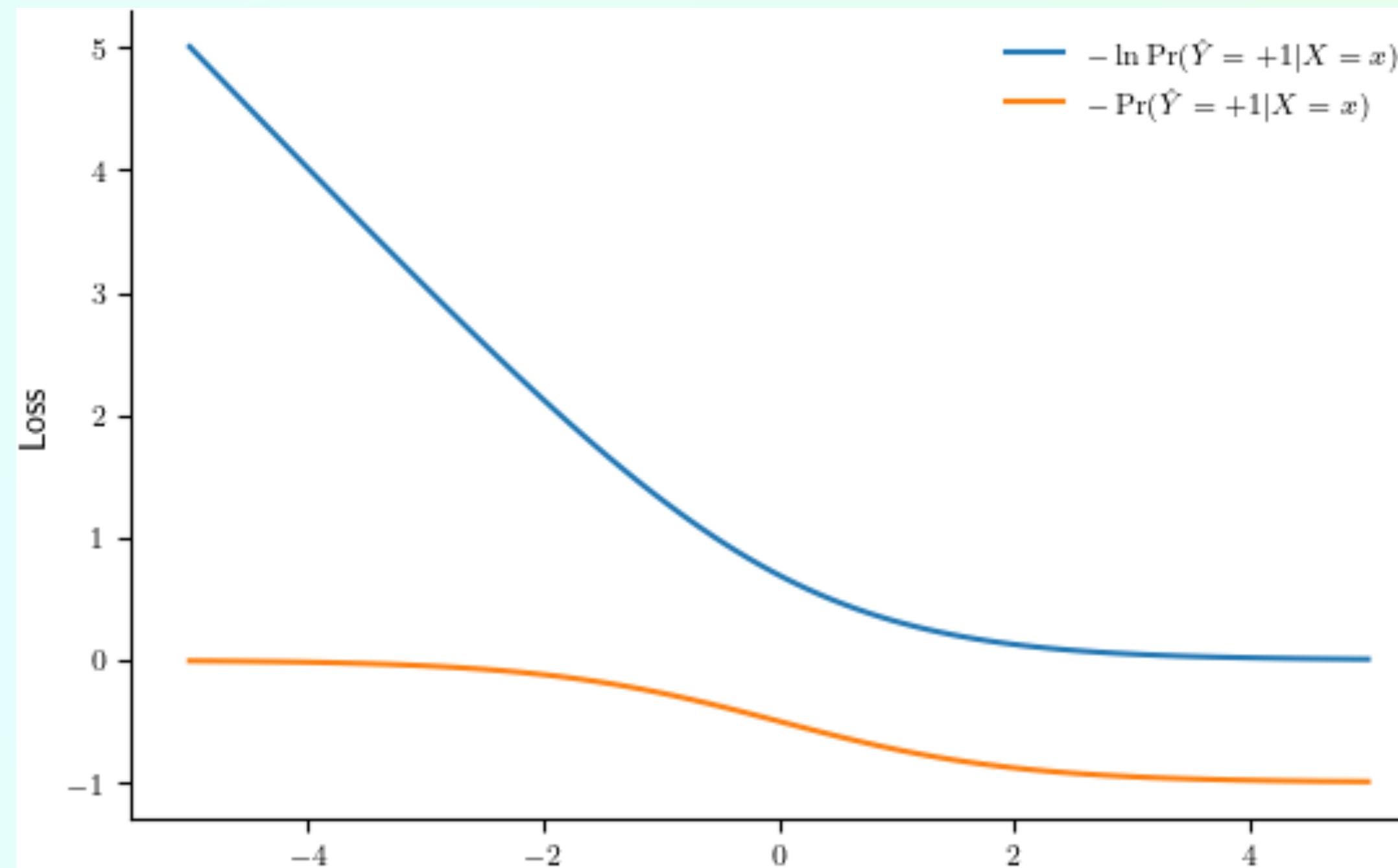
$\nabla l(w)$  points in the exact same direction as  $\nabla l'(w)$ , they just have different lengths.

This means we can use NLL objectives, which are often easier to work with.

NLL also has a better gradient

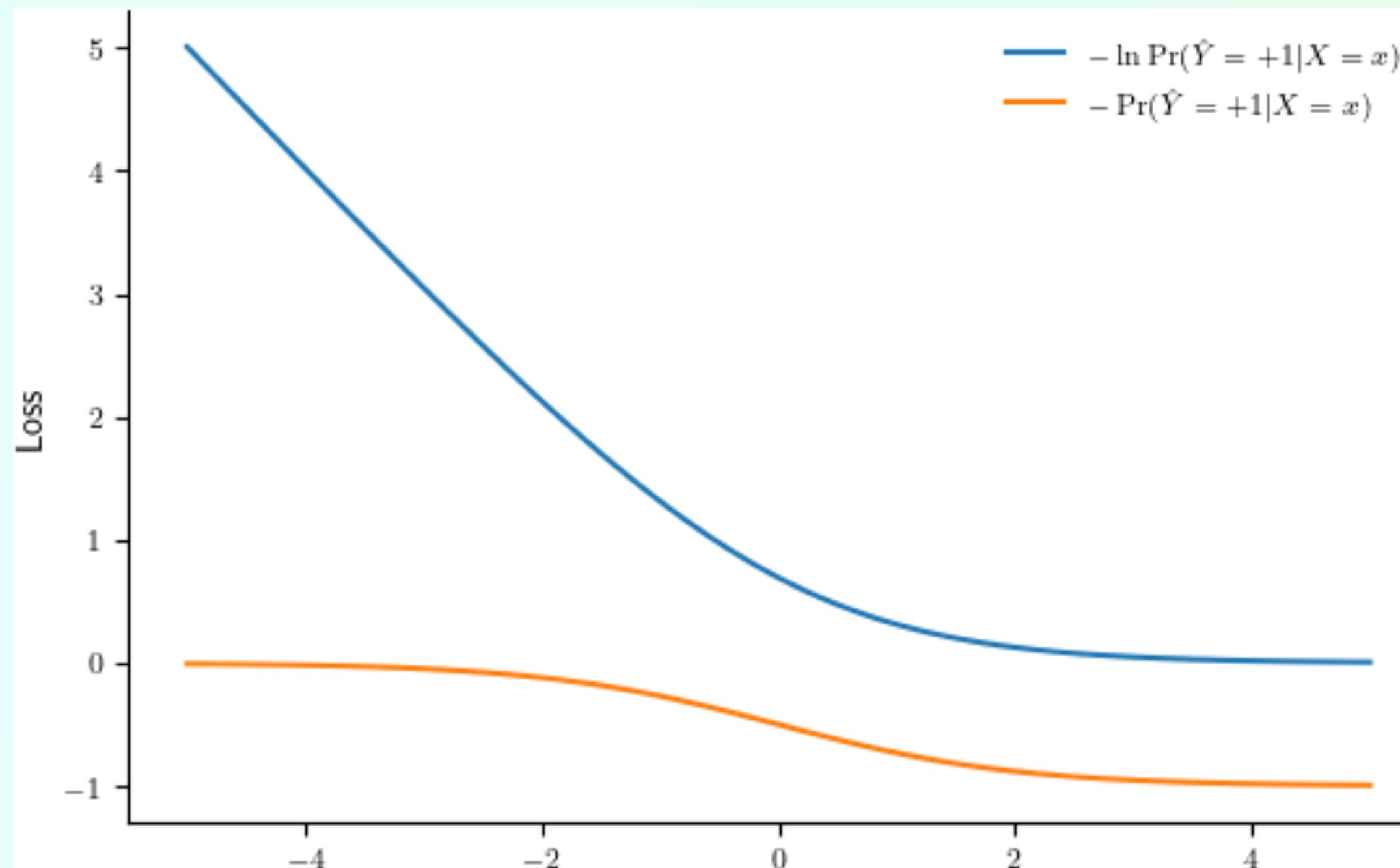
# CLASSIFICATION

## LOSS FUNCTION



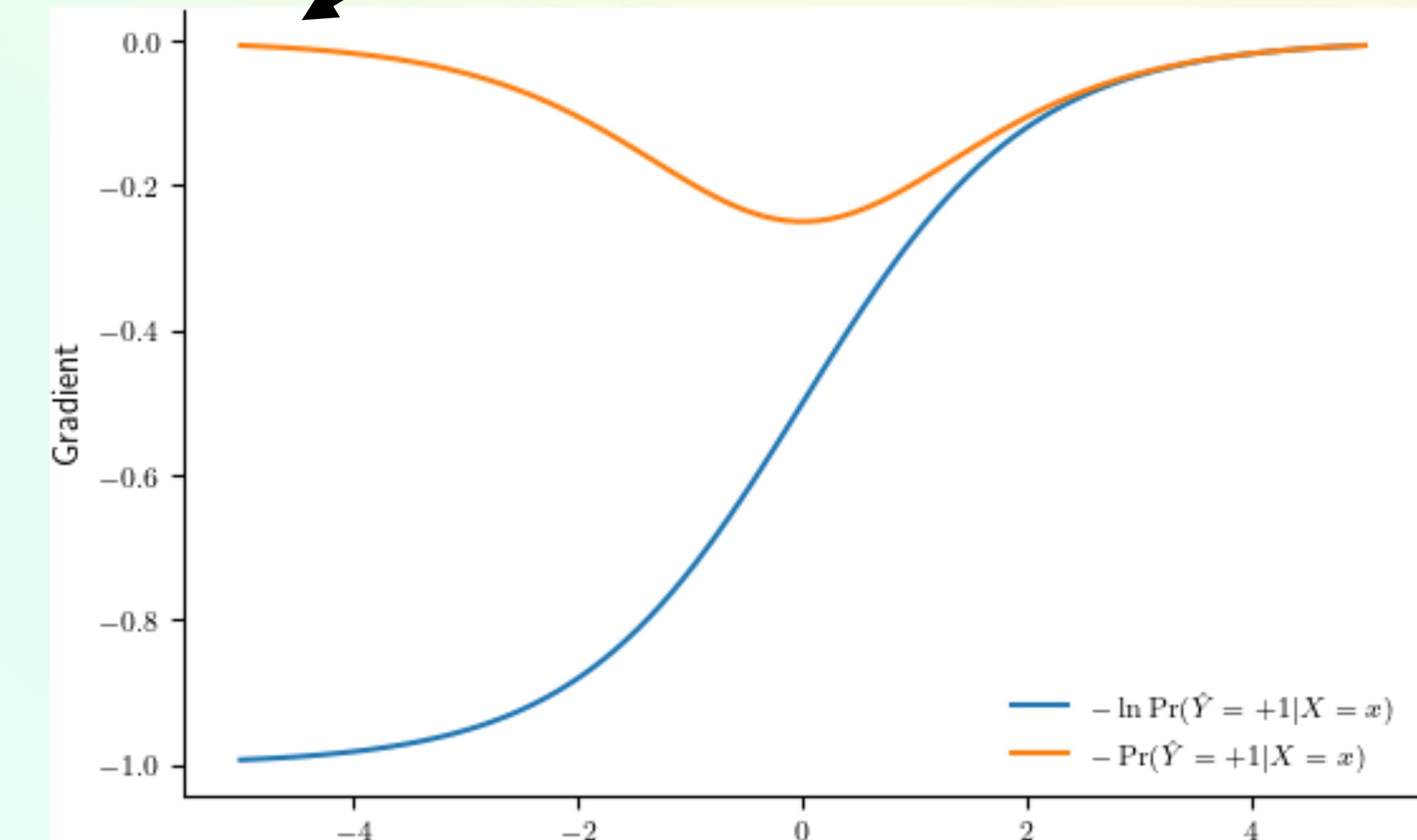
# CLASSIFICATION

## LOSS FUNCTION



A flat gradient (near zero) means parameters should not change much.

But predictions were really bad and we want them to change a lot



# CLASSIFICATION

## LOSS FUNCTION

We need an expression for  $\Pr(\hat{Y} = y_i | X = x_i)$  in terms  $f(x_i, w)$  so that we can differentiate

$$\Pr(\hat{Y} = +1 | X = x_i) = f(x_i, w)$$

$$\Pr(\hat{Y} = 0 | X = x_i) = 1 - \Pr(\hat{Y} = +1 | X = x_i) = 1 - f(x_i, w)$$

# CLASSIFICATION

## LOSS FUNCTION

We need an expression for  $\Pr(\hat{Y} = y_i | X = x_i)$  in terms  $f(x_i, w)$  so that we can differentiate

$$\Pr(\hat{Y} = +1 | X = x_i) = f(x_i, w)$$

$$\Pr(\hat{Y} = 0 | X = x_i) = 1 - \Pr(\hat{Y} = +1 | X = x_i) = 1 - f(x_i, w)$$

$$\Pr(\hat{Y} = y_i | X = x_i) = \mathbf{1}_{y_i=+1} \Pr(\hat{Y} = +1 | X = x_i) + \mathbf{1}_{y_i=0} \Pr(Y = 0 | X = x_i)$$

$\mathbf{1}_A = 1$  if  $A$  is true and  $\mathbf{1}_A = 0$  otherwise (called an indicator function)

# CLASSIFICATION

## LOSS FUNCTION

$$\begin{aligned}\Pr(\hat{Y} = y_i | X = x_i) &= \mathbf{1}_{y_i=+1} \Pr(\hat{Y} = +1 | X = x_i) + \mathbf{1}_{y_i=0} \Pr(\hat{Y} = 0 | X = x_i) \\ &= \mathbf{1}_{y_i=+1} f(x_i, w) + \mathbf{1}_{y_i=0} (1 - f(x_i, w))\end{aligned}$$

Simplify the computation by replacing  $\mathbf{1}_A$

$$\Pr(\hat{Y} = y_i | X = x_i) = y_i f(x_i, w) + (1 - y_i)(1 - f(x_i, w))$$

# CLASSIFICATION

## LOSS FUNCTION

$$\begin{aligned}\Pr(\hat{Y} = y_i | X = x_i) &= \mathbf{1}_{y_i=+1} \Pr(\hat{Y} = +1 | X = x_i) + \mathbf{1}_{y_i=0} \Pr(\hat{Y} = 0 | X = x_i) \\ &= \mathbf{1}_{y_i=+1} f(x_i, w) + \mathbf{1}_{y_i=0} (1 - f(x_i, w))\end{aligned}$$

Simplify the computation by replacing  $\mathbf{1}_A$

$$\Pr(\hat{Y} = y_i | X = x_i) = y_i f(x_i, w) + (1 - y_i) (1 - f(x_i, w))$$

Similarly, we have

$$\ln \Pr(\hat{Y} = y_i | X = x_i) = y_i \ln f(x_i, w) + (1 - y_i) \ln (1 - f(x_i, w))$$

# CLASSIFICATION

## LOSS FUNCTION

NLL loss function for classification with  $y_i \in \{0,1\}$

$$\begin{aligned} l(w) &= - \sum_{i=1}^m \ln \Pr(\hat{Y} = y_i | X = x_i) \\ &= - \sum_{i=1}^m y_i \ln(f(x_i, w)) + (1 - y_i) \ln(1 - f(x_i, w)) \end{aligned}$$

# CLASSIFICATION

## LOSS FUNCTION

Take-home questions:

1.  $f(x, w) = \frac{1}{1 + e^{-w^\top x}}$ , what is  $\frac{\partial f(x, w)}{\partial w}$ ?
2. What is  $\nabla l(w)$  for the NLL loss?

We will go over these answers next time.

# NEXT CLASS

Next Class — Stochastic Gradient Descent