# OPTIMIZING BASIS FUNCTIONS

# OPTIMIZING BASIS FUNCTIONS

## GOALS FOR TODAY

A step towards deep learning and neural networks

How to optimize the basis function $\phi(x)$

General formulization

# OPTIMIZING BASIS FUNCTIONS

## OVERVIEW

Choose heuristic basis functions and do pretty good

If the fit is not good enough, we generate more features

— create more bins, etc

# OPTIMIZING BASIS FUNCTIONS

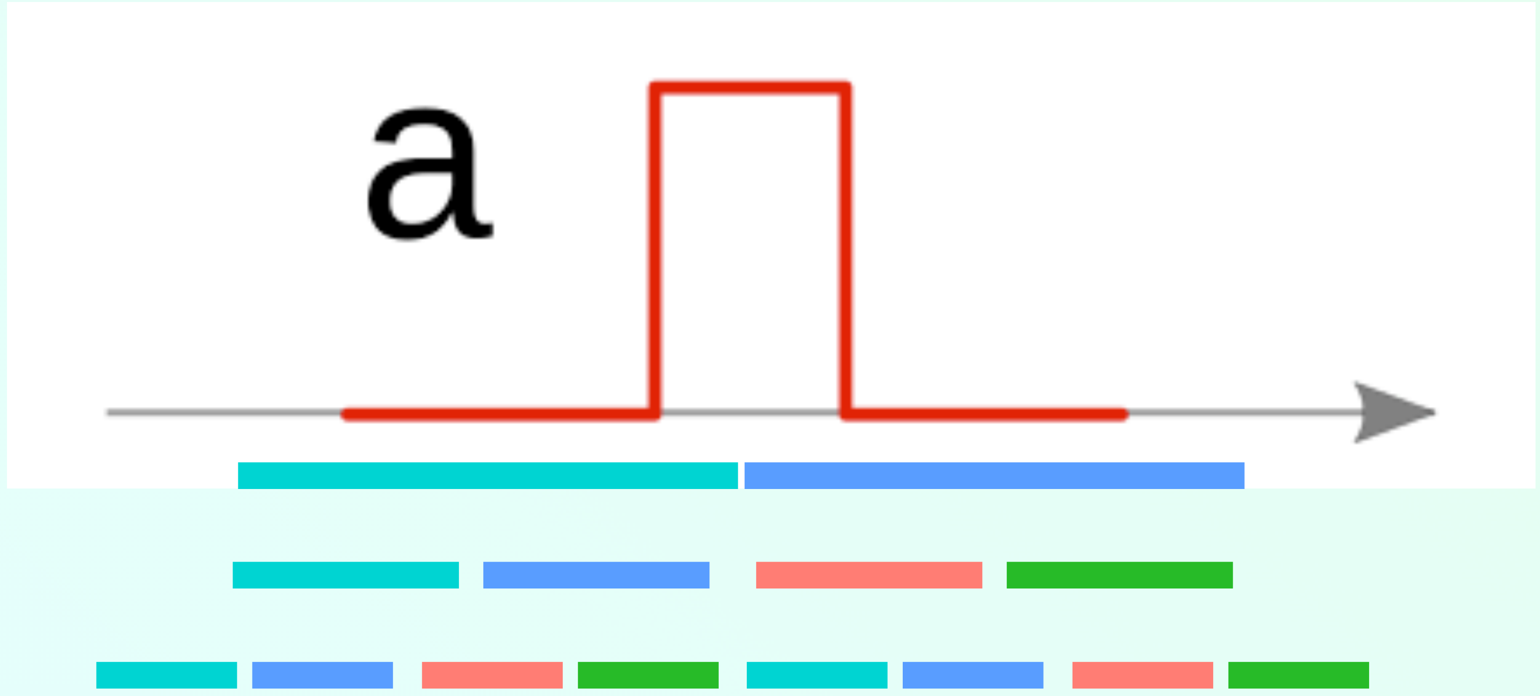
OVERVIEW

Choose heuristic basis functions and do pretty good

If the fit is not good enough, we generate more features

— create more bins, etc

Can lead to a huge number of features to get small error



# OPTIMIZING BASIS FUNCTIONS

OVERVIEW

Choose heuristic basis functions and do pretty good

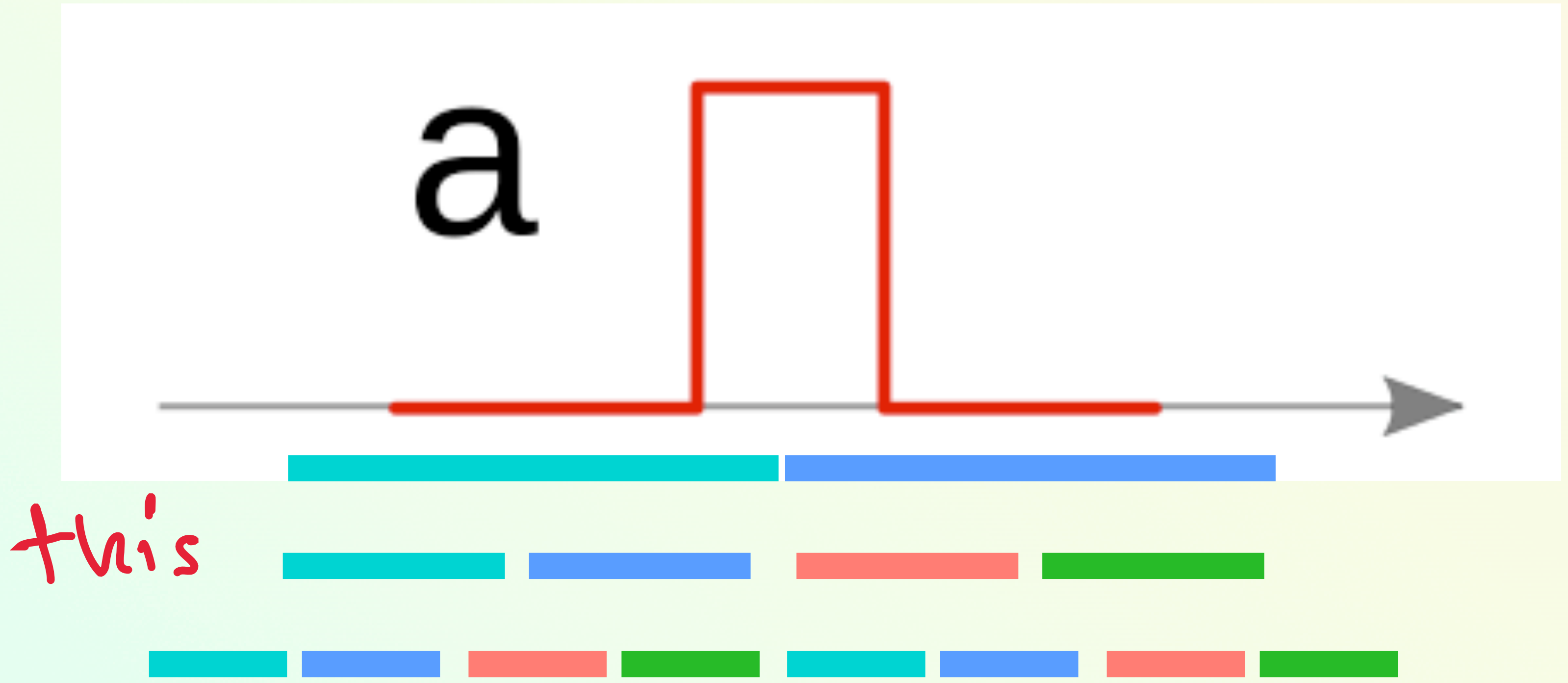
If the fit is not good enough, we generate more features

— create more bins, etc

Can lead to a huge number of features to get small error

Features might just need a little bit of adjustment

what does this mean ?




Solution: Optimize a fixed set of features so they can best approximate the function.

— move the center or scale a little bit to decrease the loss

# PARAMETERIZING BASIS FUNCTIONS

MECHANICS

$$f(x, w) = w\phi(x)$$

$$\phi(x) \doteq \sin(\tau x + \kappa)$$ ← check how varying $\tau$ and $\kappa$ change the sin function

$\tau$ is a parameter for the frequency

$\kappa$ is a parameter for the offset

# PARAMETERIZING BASIS FUNCTIONS

MECHANICS

$$f(x, w) = w\phi(x, \tau, \kappa)$$

$$\phi(x, \tau, \kappa) \doteq \sin(\tau x + \kappa)$$

# PARAMETERIZING BASIS FUNCTIONS

MECHANICS

$$f(x, w, \tau, \kappa) = w\phi(x, \tau, \kappa)$$

$$\phi(x, \tau, \kappa) \doteq \sin(\tau x + \kappa)$$

# PARAMETERIZING BASIS FUNCTIONS

MECHANICS

$$\beta = \begin{bmatrix} \tau \\ \kappa \end{bmatrix}$$

$$f(x, w, \beta) = w\phi(x, \beta)$$

$$\phi(x, \beta) \doteq \sin(\beta_1 x + \beta_2)$$

# PARAMETERIZING BASIS FUNCTIONS

MECHANICS

$$\beta = \begin{bmatrix} \tau \\ \kappa \end{bmatrix} \begin{array}{l} = \text{Frequency } (\beta_1) \\ = \text{offset } (\beta_0) \text{ or } \beta_2 \end{array}$$

$$f(x, w, \beta) = w\phi(x, \beta)$$

$$\phi(x, \beta) \doteq \sin(\beta_1 x + \beta_2)$$

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

# QUIZ

# OPTIMIZING BASIS FUNCTIONS

MECHANICS

$$
\begin{aligned}
\nabla\, l(w,\beta) &= \left[\frac{\partial l(w,\beta)}{\partial w}, \frac{\partial l(w,\beta)}{\partial \beta}\right] \\
&= \mathbf{E}\left[\big(f(X,w,\beta) - Y\big)\left[\frac{\partial f(X,w,\beta)}{\partial w}, \frac{\partial f(X,w,\beta)}{\partial \beta}\right]\right] \\
&= \mathbf{E}\left[\big(f(X,w,\beta) - Y\big)\left[\phi(X,\beta), \frac{\partial f(X,w,\beta)}{\partial \beta}\right]\right]
\end{aligned}
$$

# OPTIMIZING BASIS

MECHANICS

$$\frac{\partial f(X, w, \beta)}{\partial \beta} = \frac{\partial}{\partial \beta} w \phi(X, \beta)$$

$$= \boxed{w \frac{\partial}{\partial \beta} \phi(X, \beta)}$$

# OPTIMIZING BASIS

MECHANICS

$$\phi(x, \beta) = \sin(\beta_1 x + \beta_2)$$

$$\begin{aligned}
\frac{\partial \phi(x, \beta)}{\partial \beta} &= \frac{\partial}{\partial \beta} \sin(\beta_1 x + \beta_2) \\
&= \cos(\beta_1 x + \beta_2) \frac{\partial}{\partial \beta} \left(\beta_1 x + \beta_2\right) \\
&= \cos(\beta_1 x + \beta_2) \begin{bmatrix} x \\ 1 \end{bmatrix}
\end{aligned}$$

# OPTIMIZING BASIS

MECHANICS

$$\frac{\partial f(X, w, \beta)}{\partial \beta} = w \frac{\partial}{\partial \beta} \phi(X, \beta)$$

$$\frac{\partial \phi(x, \beta)}{\partial \beta} = \cos(\beta_1 x + \beta_2) \begin{bmatrix} x \\ 1 \end{bmatrix}$$

$$\frac{\partial f(x, w, \beta)}{\partial \beta} = w \cos(\beta_1 x + \beta_2) \begin{bmatrix} x \\ 1 \end{bmatrix}$$

# OPTIMIZING BASIS

MECHANICS

$$
\begin{aligned}
\nabla l(w, \beta) = &\ \mathbf{E}\left[\Big(f(X, w, \beta) - Y\Big)\left[\phi(X, \beta), \frac{\partial f(X, w, \beta)}{\partial \beta}\right]\right] \\
= &\ \mathbf{E}\left[\Big(f(X, w, \beta) - Y\Big)\left[\phi(X, \beta), w\cos(\beta_1 X + \beta_2)\begin{bmatrix} x \\ 1 \end{bmatrix}\right]\right]
\end{aligned}
$$

# CHAIN RULE DERIVATIVES

MECHANICS

$$\frac{\partial l(X, Y, w, \beta)}{\partial w} = \frac{\partial l(X, Y, w, \beta)}{\partial f(X, w, \beta)} \frac{\partial f(X, w, \beta)}{\partial w}$$

$$\frac{\partial l(X, Y, w, \beta)}{\partial \beta} = \frac{\partial l(X, Y, w, \beta)}{\partial f(X, w, \beta)} \frac{\partial f(X, w, \beta)}{\partial \phi(X, \beta)} \frac{\partial \phi(X, \beta)}{\partial \beta}$$

# OPTIMIZING BASIS FUNCTIONS

EXAMPLE

$$f(x, w, \beta) = w\phi(x, \beta)$$

$$\phi(x, \beta) \doteq \sin(\beta_1 x + \beta_2)$$

$$Y = w_* \sin\left(2X + \frac{1}{2}\right) + \xi$$

# OPTIMIZING BASIS FUNCTIONS

EXAMPLE

$$f(x, w, \beta) = w\phi(x, \beta)$$

$$\phi(x, \beta) \doteq \sin(\beta_1 x + \beta_2)$$

How did we get those values?

$$Y = w_* \sin\left(2X + \frac{1}{2}\right) + \xi$$

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

$$\arg\min_{w,\beta} l(w, \beta)$$

Use gradient descent to optimize $l(w, \beta)$

# OPTIMIZING BASIS FUNCTIONS

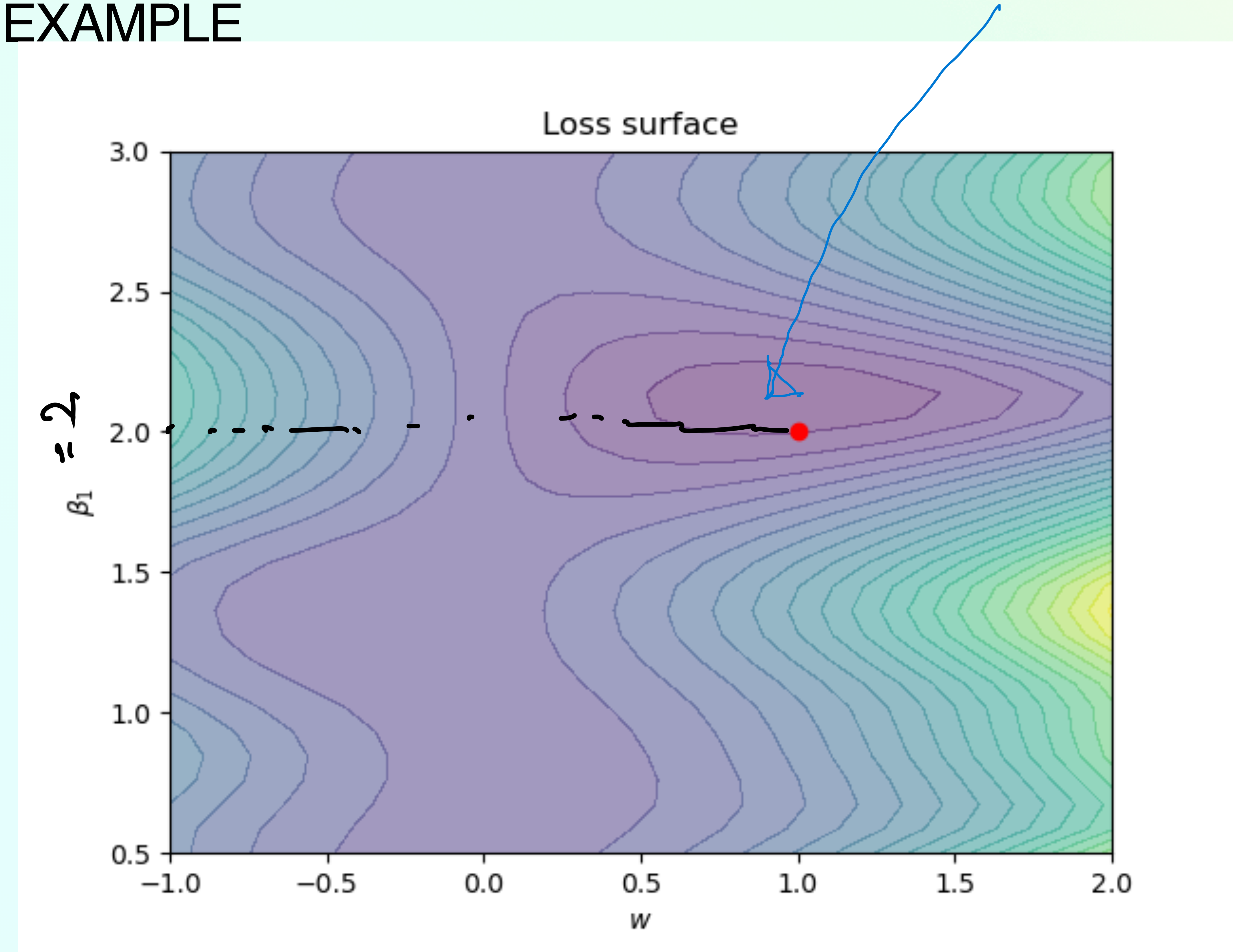
EXAMPLE

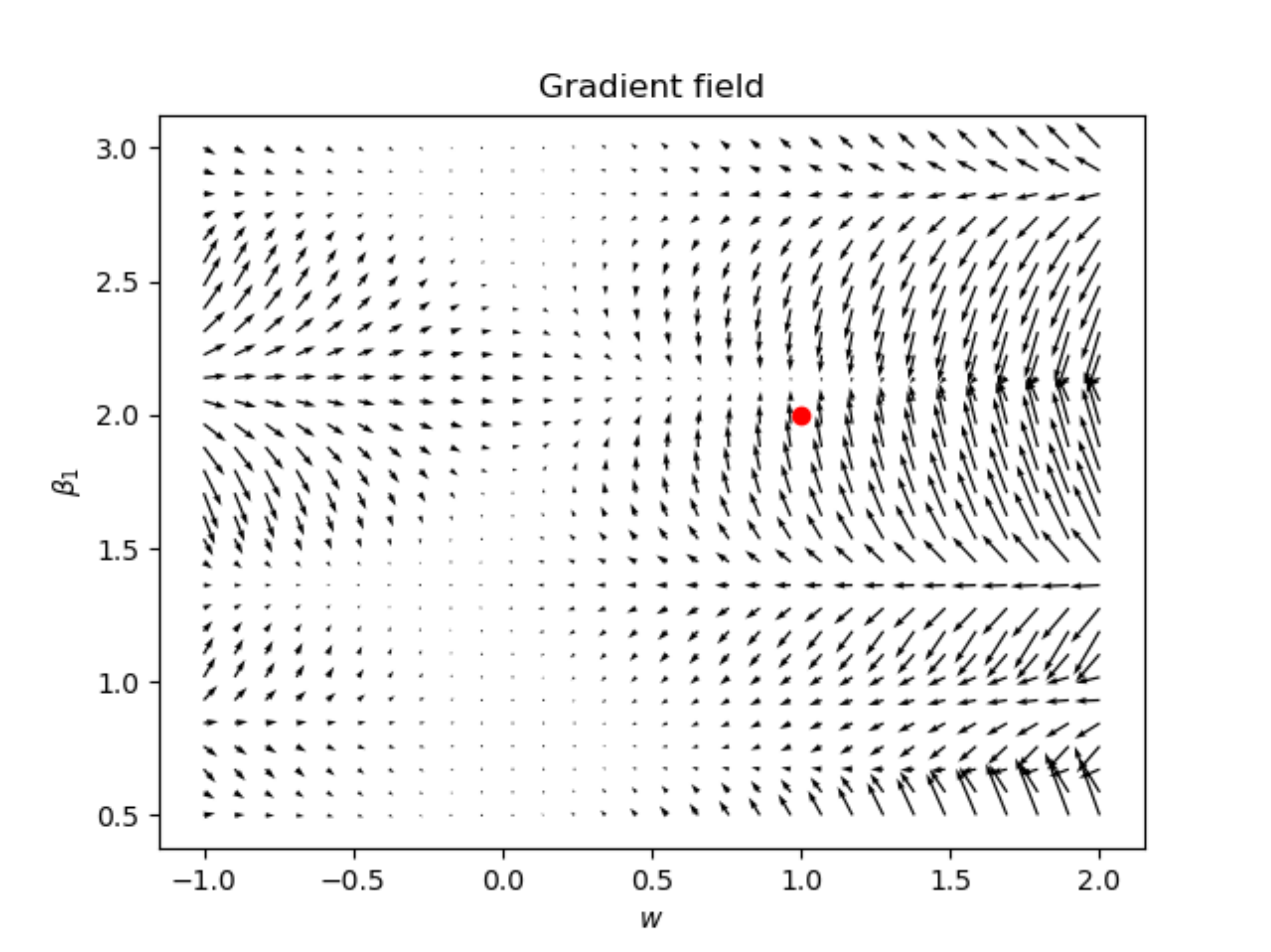
$$\begin{bmatrix} w^{k+1} \\ \beta^{k+1} \end{bmatrix} = \begin{bmatrix} w^{k} \\ \beta^{k} \end{bmatrix} - \eta \nabla l(w, \beta)$$

Is this going to be successful?

What does $l(w, \beta)$ and $\nabla l(w, \beta)$ look like?

EXAMPLE

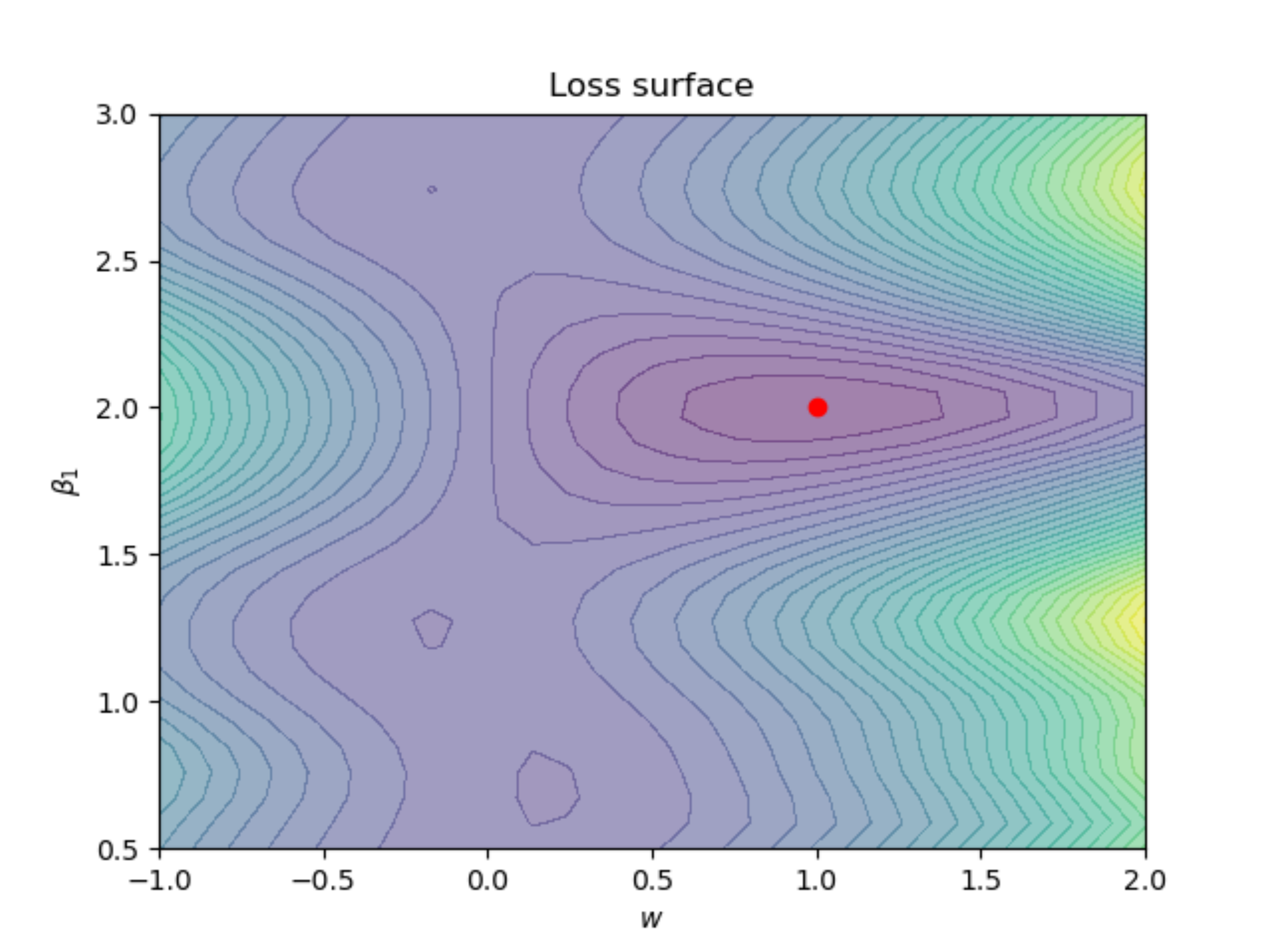


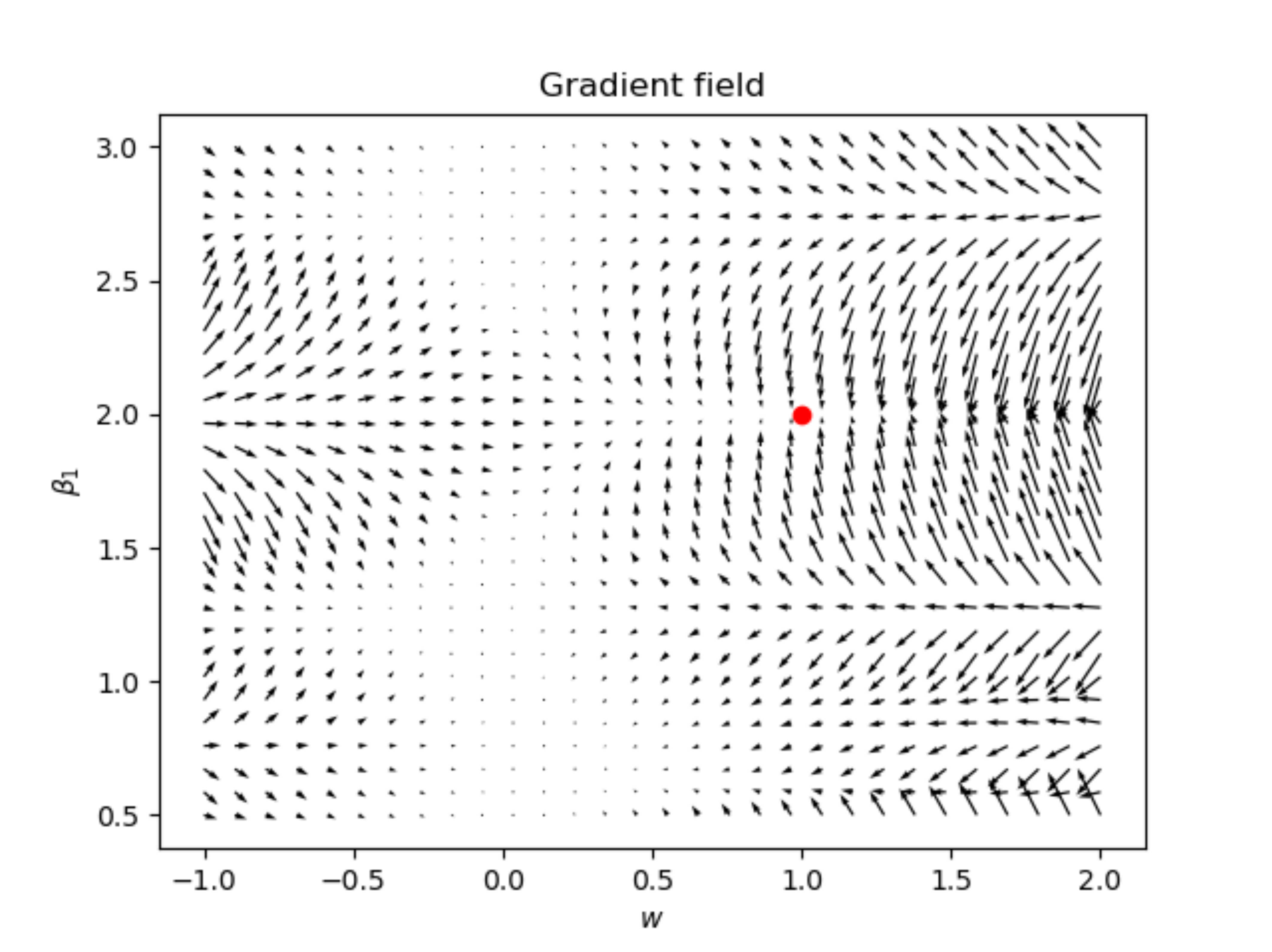




$$\boxed{\beta_2 = 0}$$

# OPTIMIZING BASIS FUNCTIONS

EXAMPLE

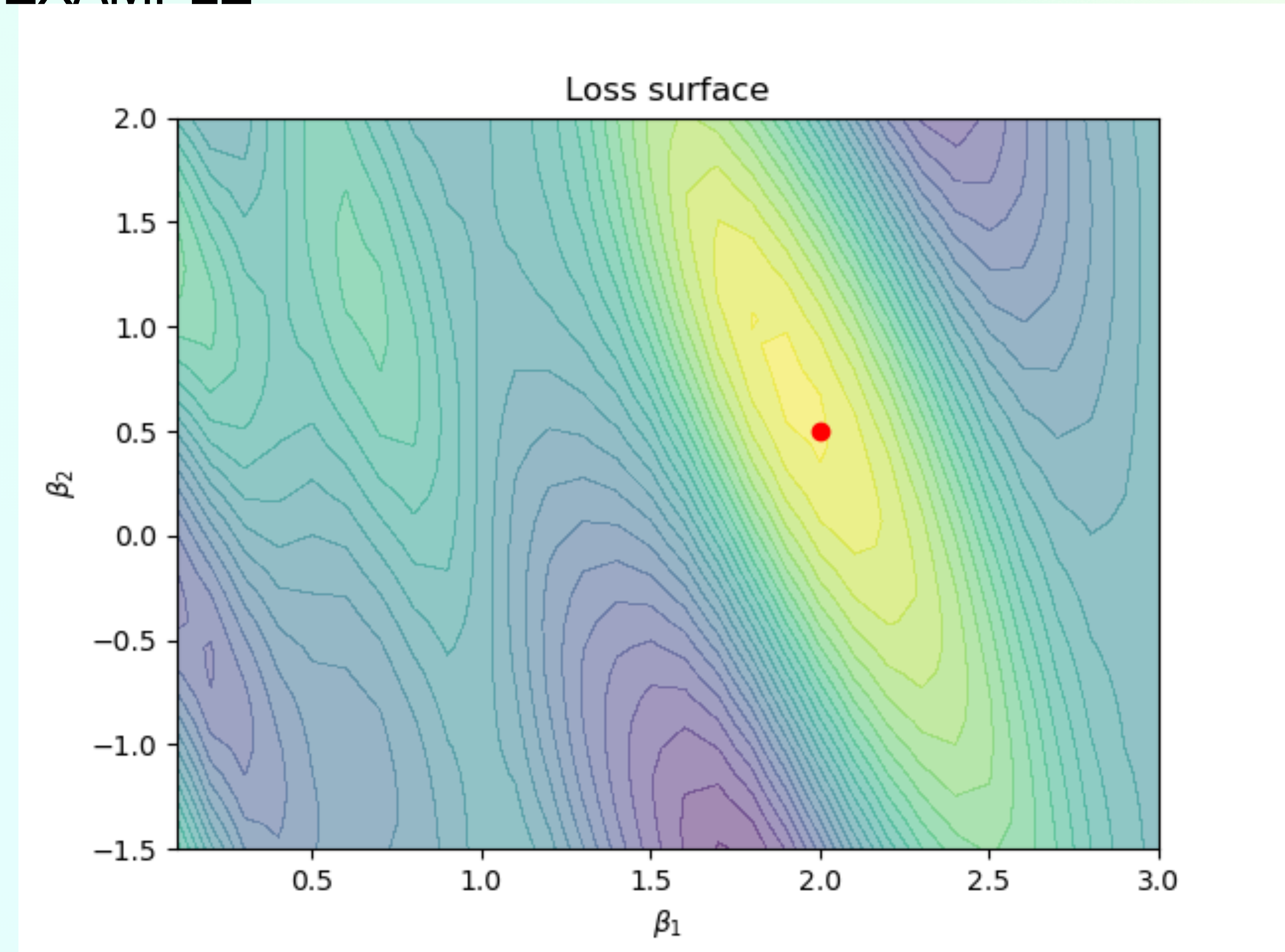


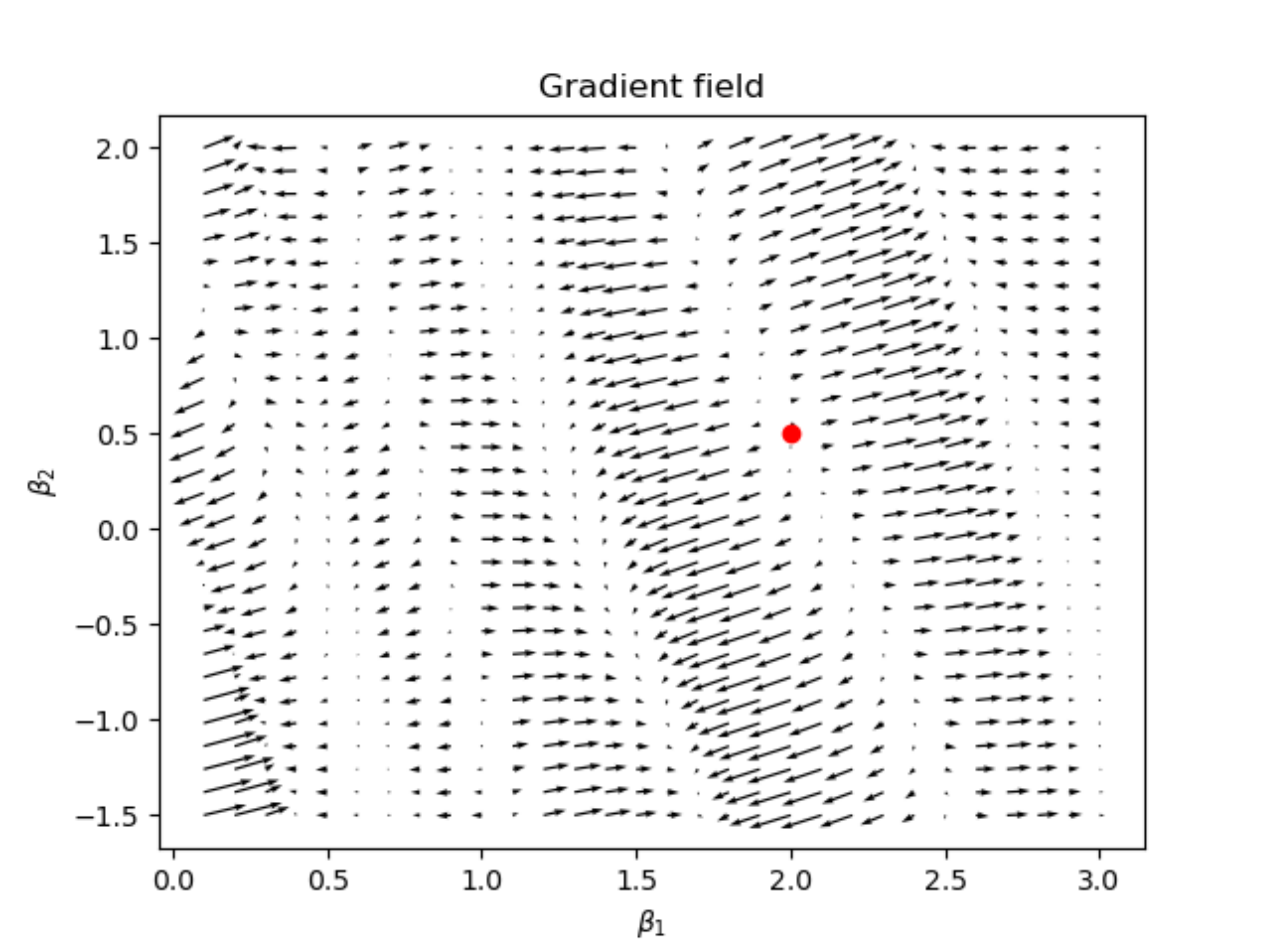




$$\boxed{\beta_2 = \frac{1}{2}}$$

# OPTIMIZING BASIS FUNCTIONS

EXAMPLE

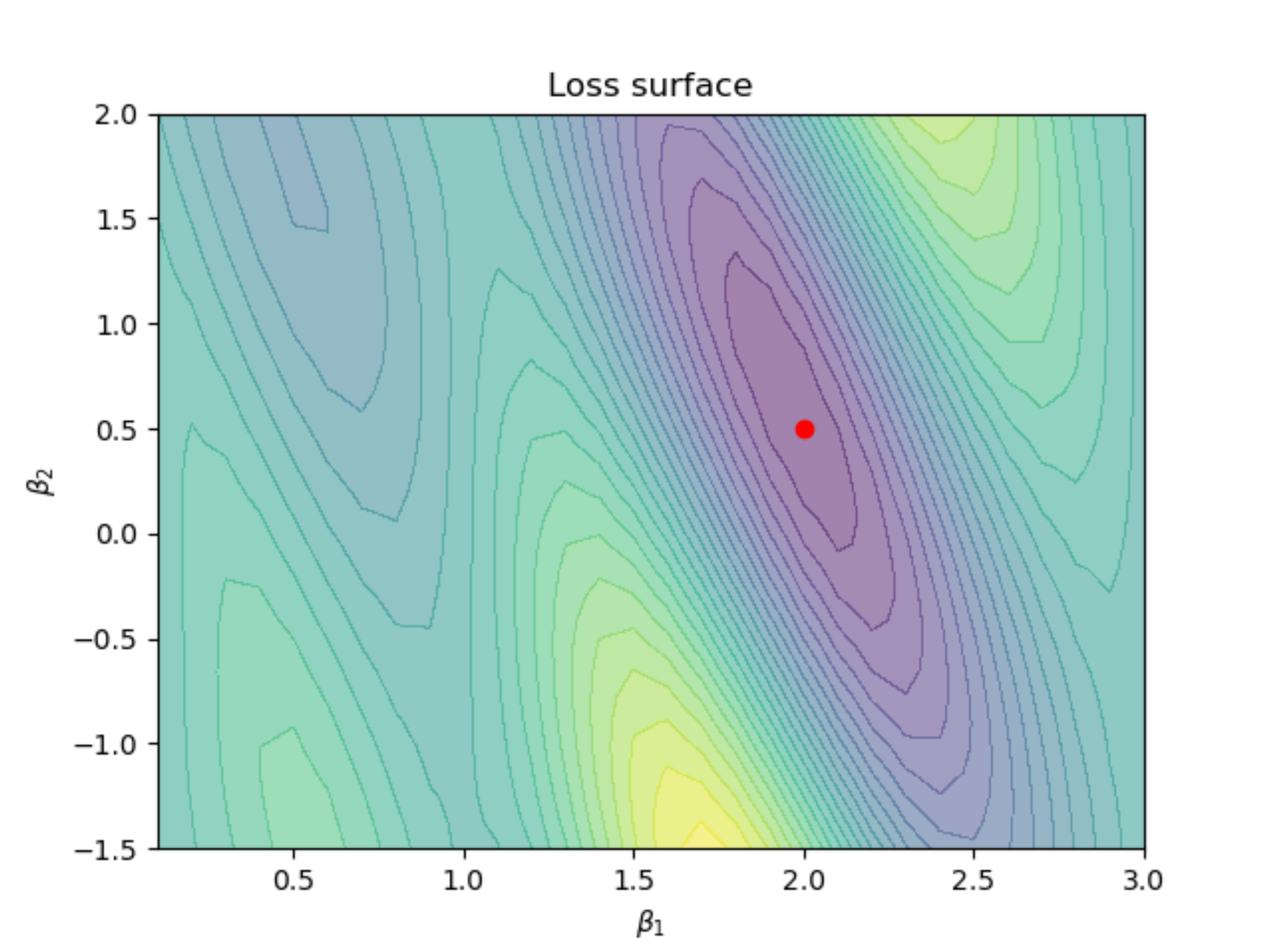


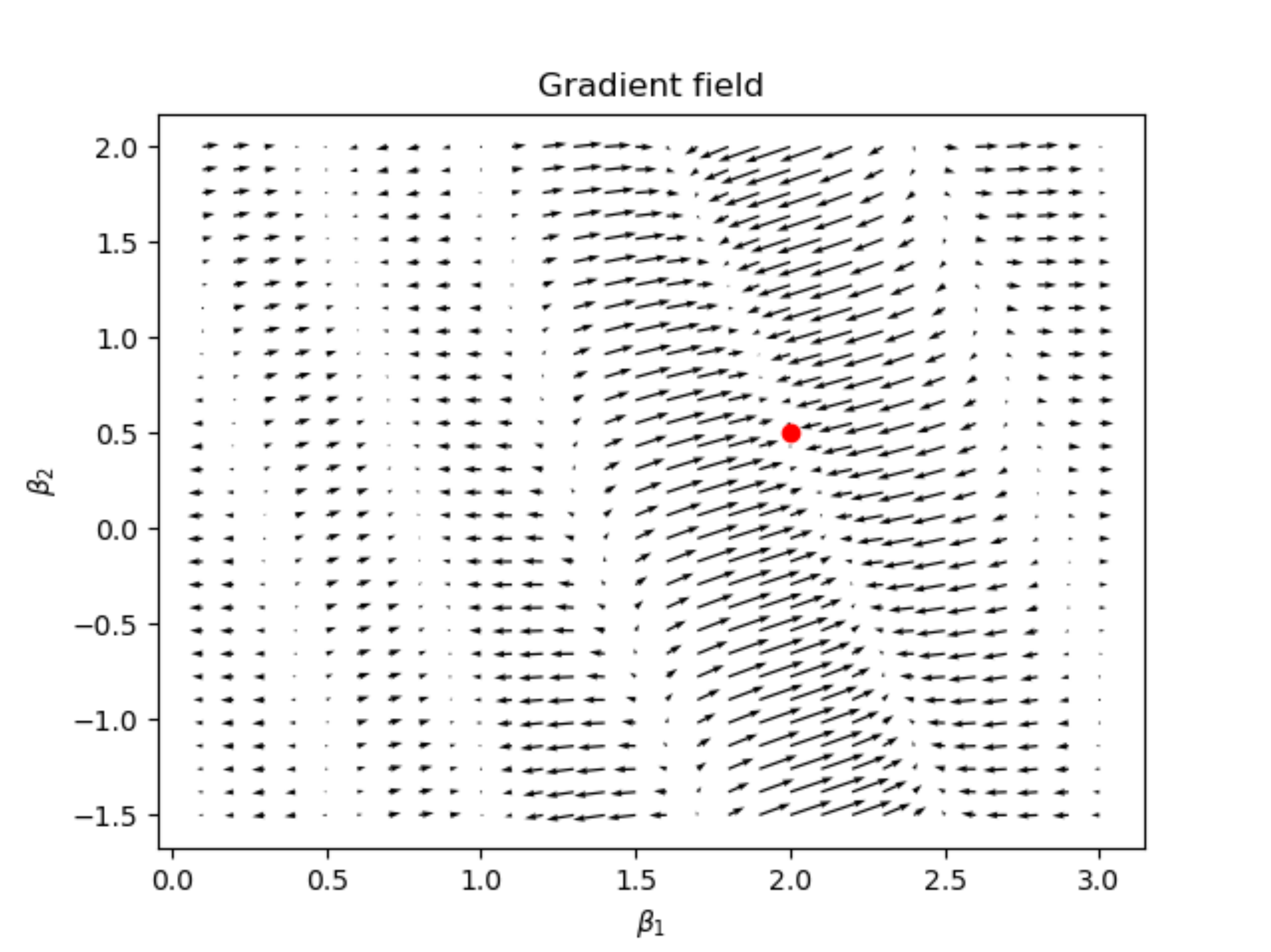




$$\boxed{w = -1}$$

# OPTIMIZING BASIS FUNCTIONS

EXAMPLE




$$\boxed{w = 0.1}$$

# OPTIMIZING BASIS FUNCTIONS

## EXAMPLE

What does $l(w, \beta)$ and $\nabla l(w, \beta)$ look like?

- Many local minima
- Many plateaus/saddle points where the gradient is near zero in all or most dimensions
- Wrong value of $w$ can cause $\beta$ to move away from the optimum
- A small range of values to initialize $w^0, \beta^0$ that leads to correct values
  - With $\xi = 0$ there is no approximation error (perfect fit to the data)

Disappointing – gradient descent is not a good optimization method for this function.

# OPTIMIZING BASIS FUNCTIONS

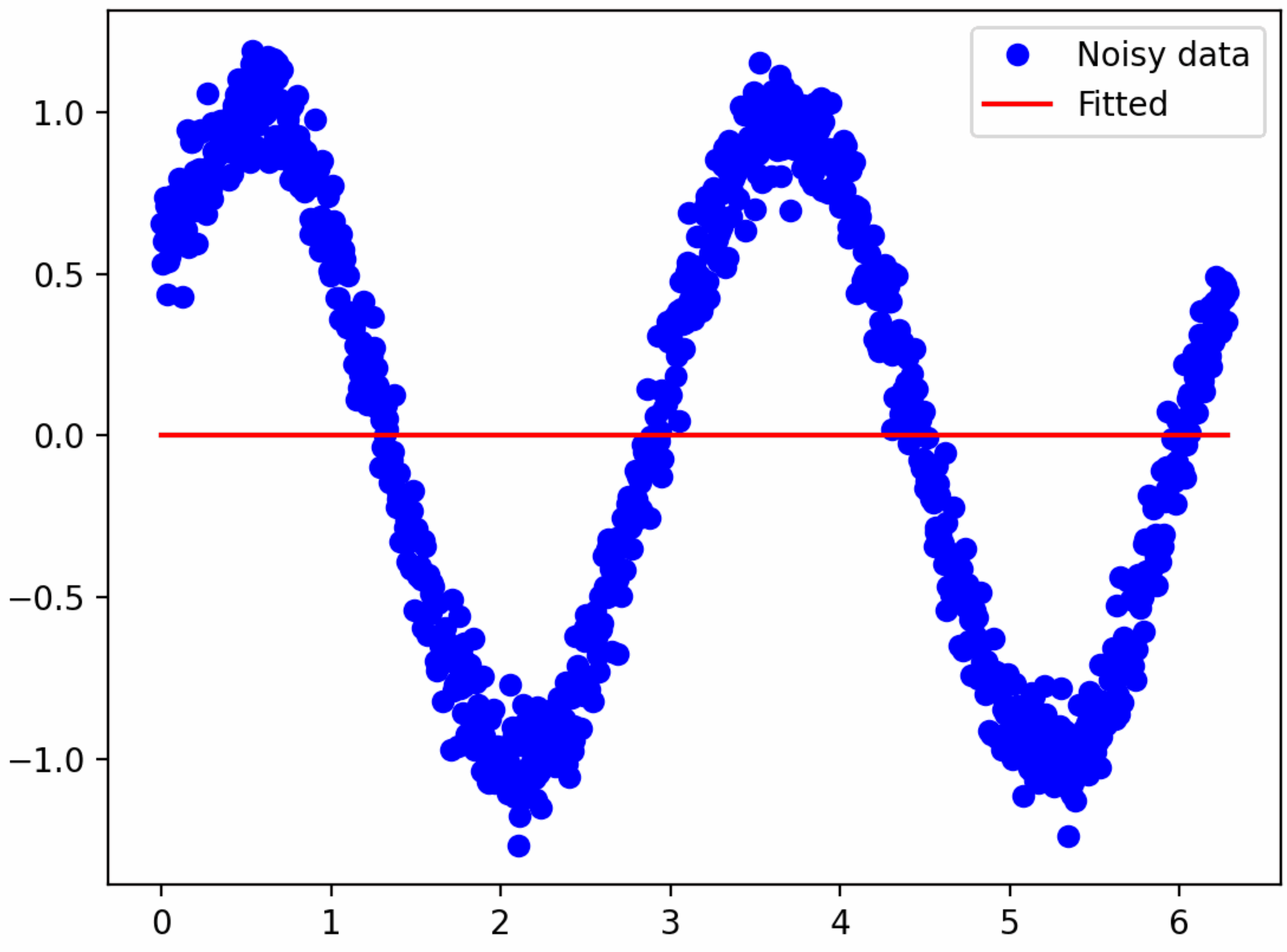
WHAT IF THERE ARE MANY FEATURES?

$$\phi(x, \beta) = \begin{array}{cc} \begin{bmatrix} \sin(\beta_{1,1}x + \beta_{1,2}) \\ \sin(\beta_{2,1}x + \beta_{2,2}) \\ \sin(\beta_{3,1}x + \beta_{3,2}) \\ \sin(\beta_{4,1}x + \beta_{4,2}) \\ \sin(\beta_{5,1}x + \beta_{5,2}) \end{bmatrix} & \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \end{array}$$

$$f(x, w, \beta) = w^\top \phi(x, \beta)$$

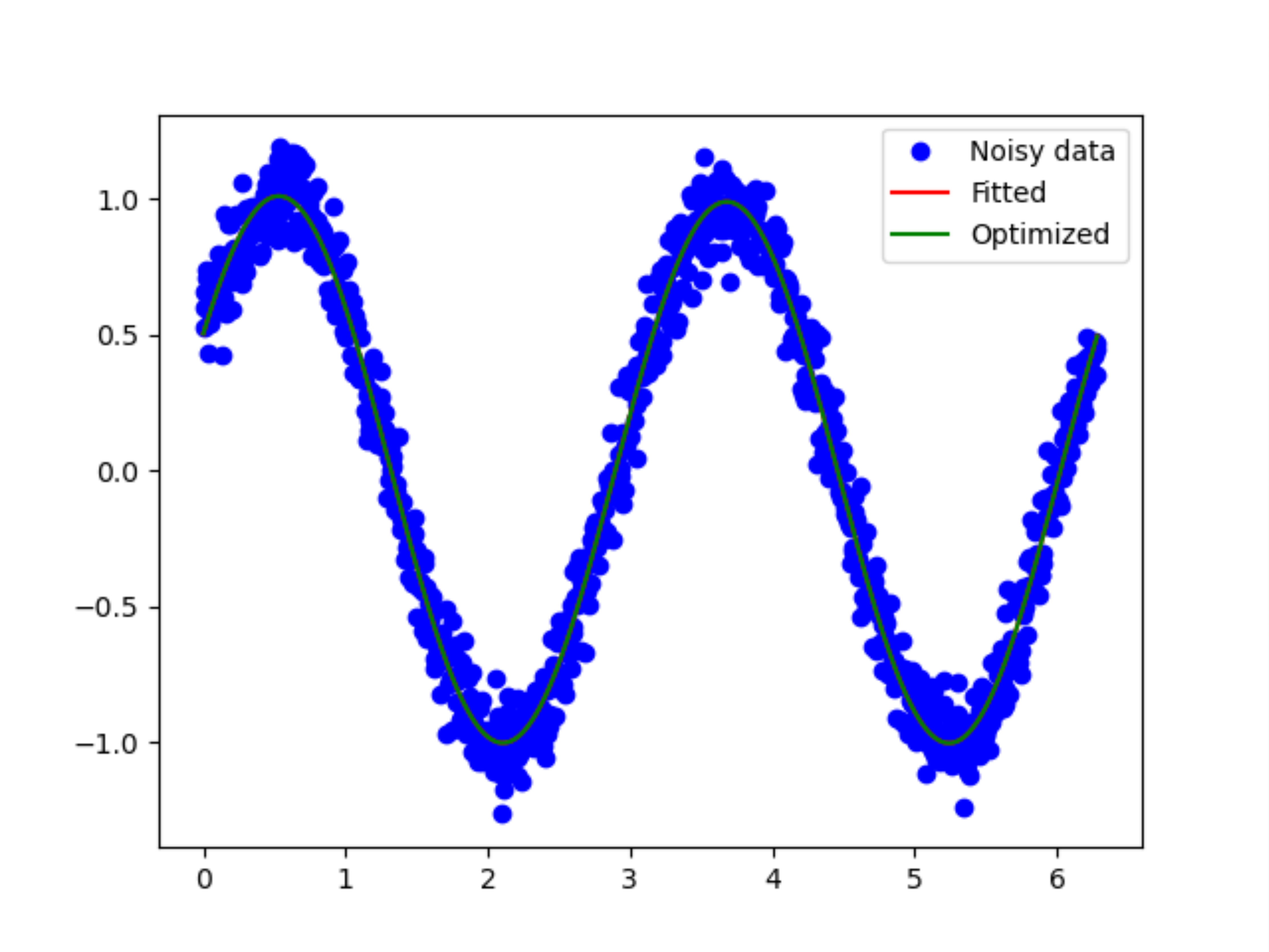
# OPT

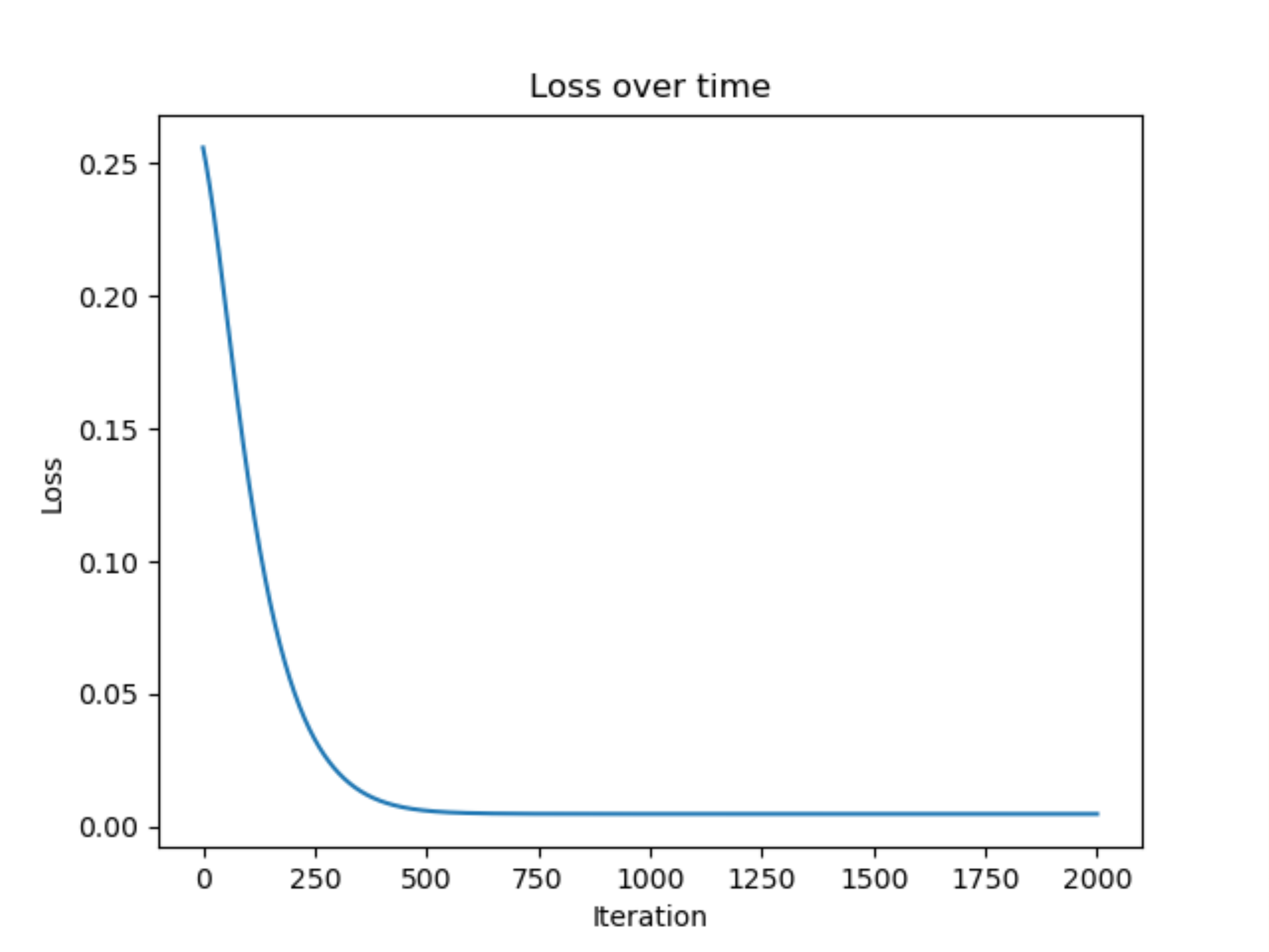
WHAT IF THE



# OPTIMIZING BASIS FUNCTIONS

WHAT IF THERE ARE MANY FEATURES?



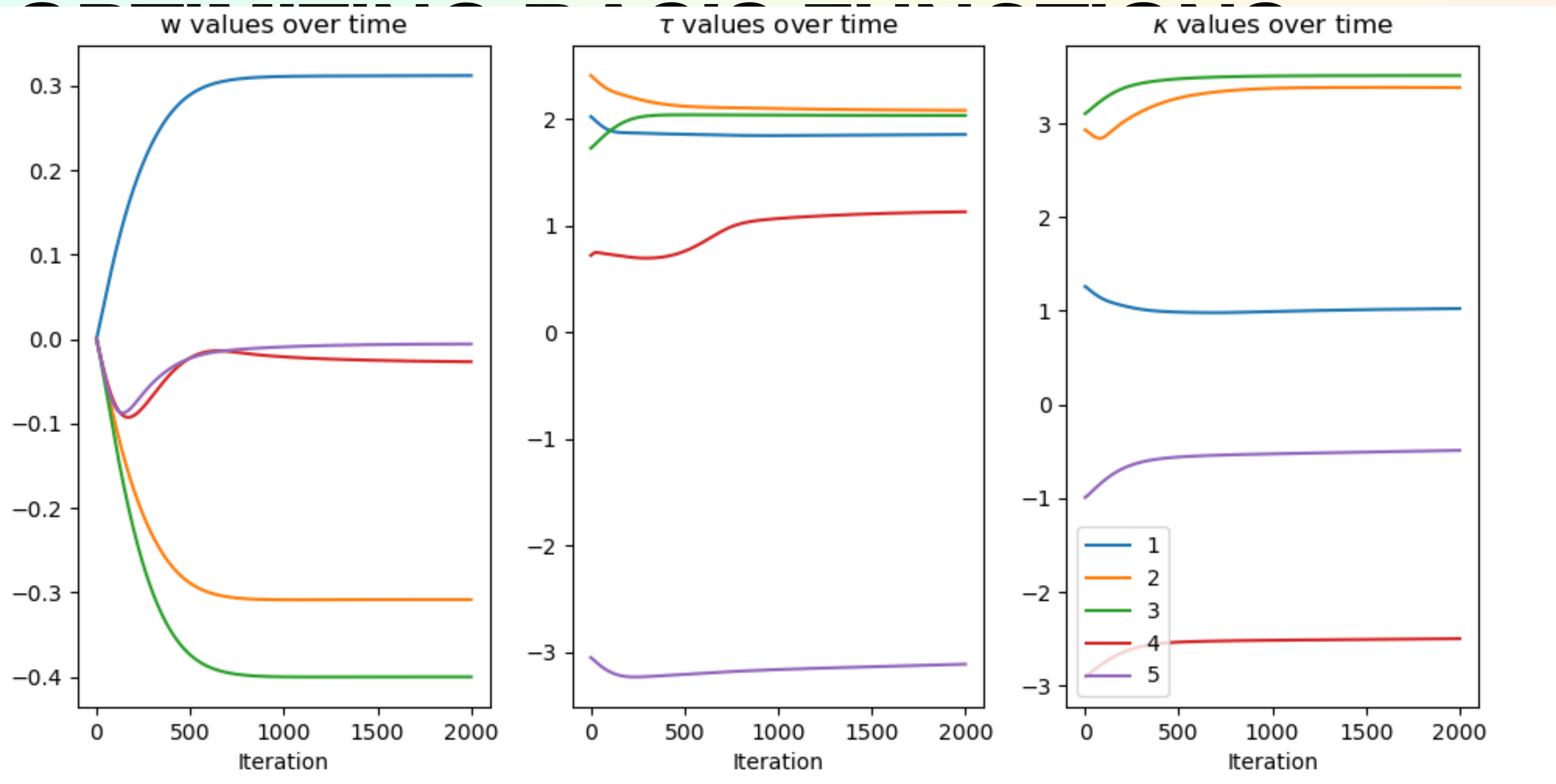



# OPTIMIZING BASIS FUNCTIONS

WHAT IF THERE ARE MANY FEATURES?

Does it identify the correct sinewave?

$$f_*(x) = \sin(2x + 0.5)$$

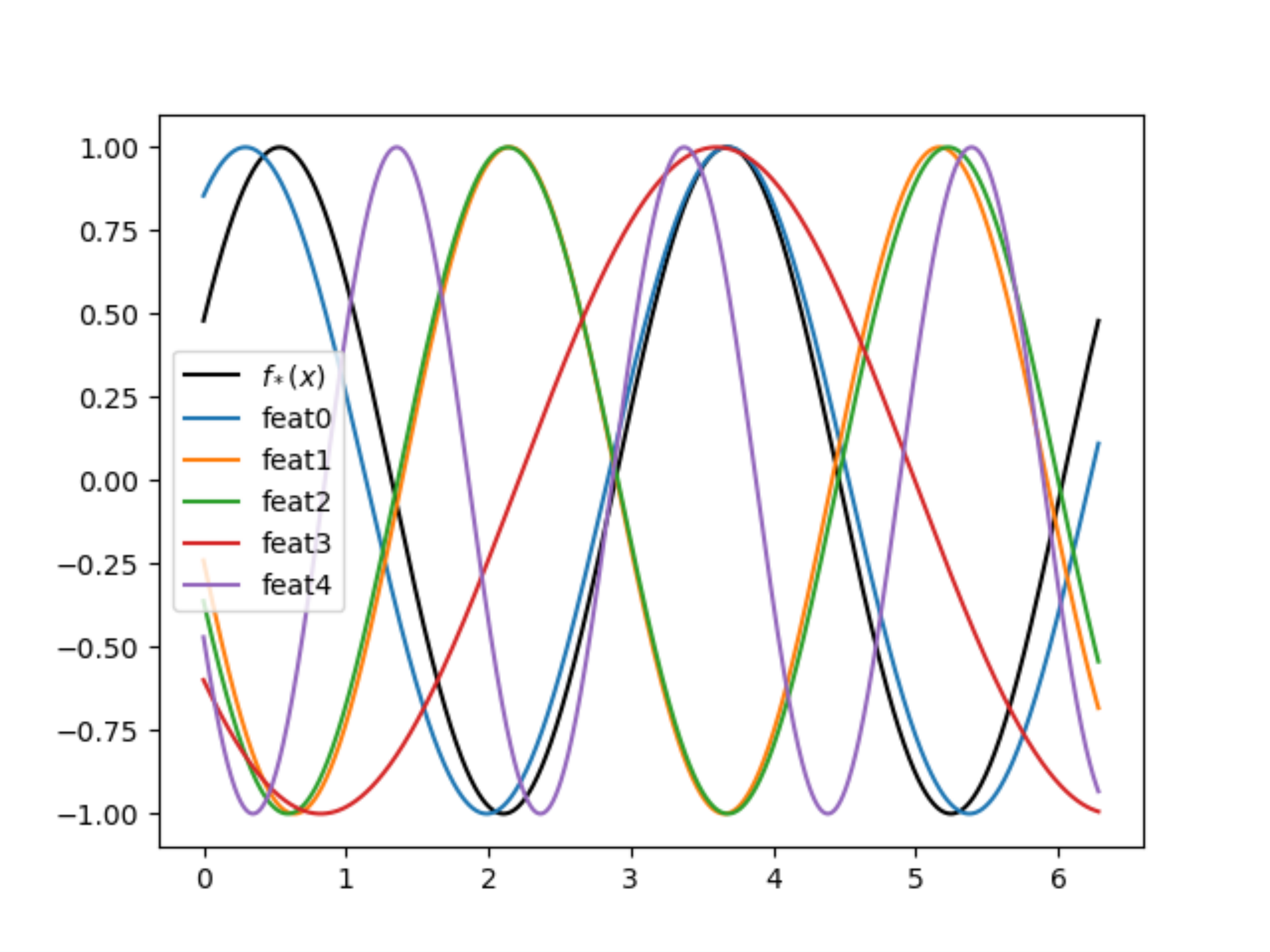

w values over time
τ values over time
κ values over time
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
−0.4
2
1
0
−1
−2
−3
3
0
500
1000
1500
2000
Iteration
1
2
3
4
5

# OPTI

WHAT IF THER

Does it identi

$f_*(x) = \sin$



# OPTIMIZING BASIS FUNCTIONS

## WHAT IF THERE ARE MANY FEATURES?

Does it identify the correct sinewave?

$$f_*(x) = \sin(2x + 0.5)$$

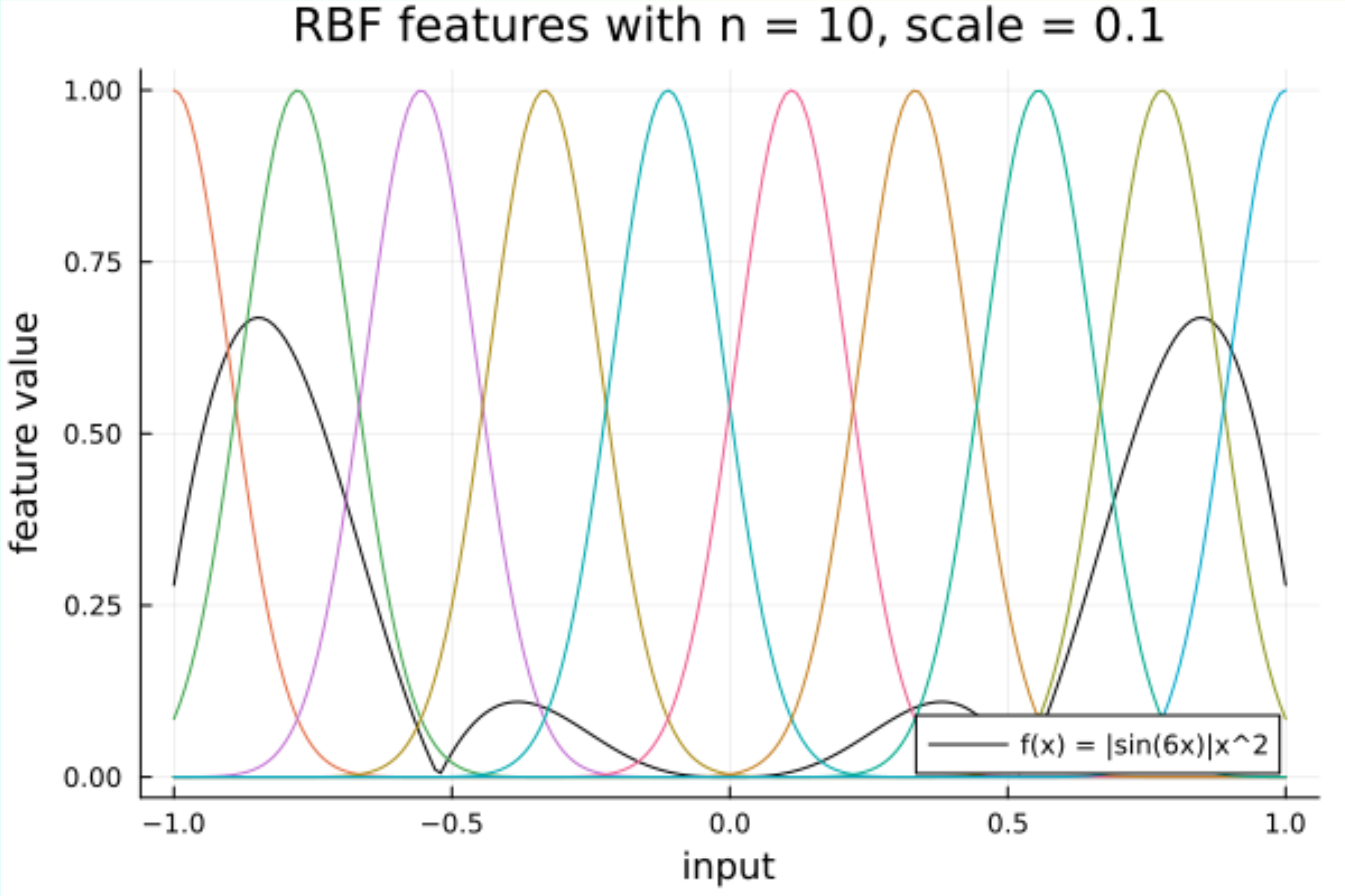
No, but it got close.

The gradient is close to zero

The model does not need to find the exact function, only a close one.

No guarantee that $f \to f_*$ in the same parameterization

# DIFFERENTIABLE BINNING



# RADIAL BASIS FUNCTIONS

$$\phi_i(x) = e^{-\frac{1}{2}\frac{(a-\mu_i)^2}{\sigma_i^2}}$$

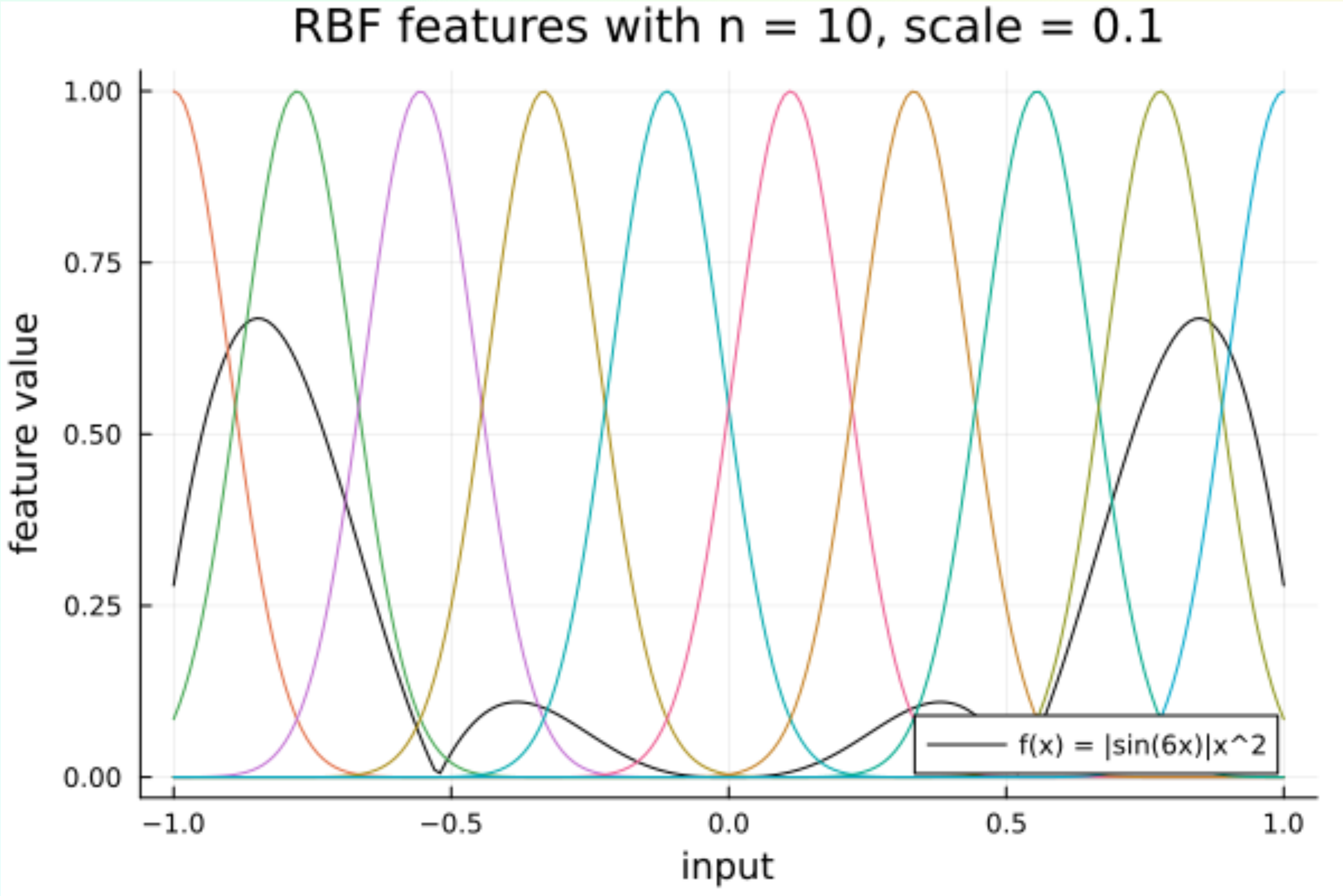
$\mu_i$ is the center

$\sigma_i$ is the scale and controls the “width” of the feature

$\phi_i(\mu) = 1$

$\phi_i(\mu + 3\sigma) \approx 0$

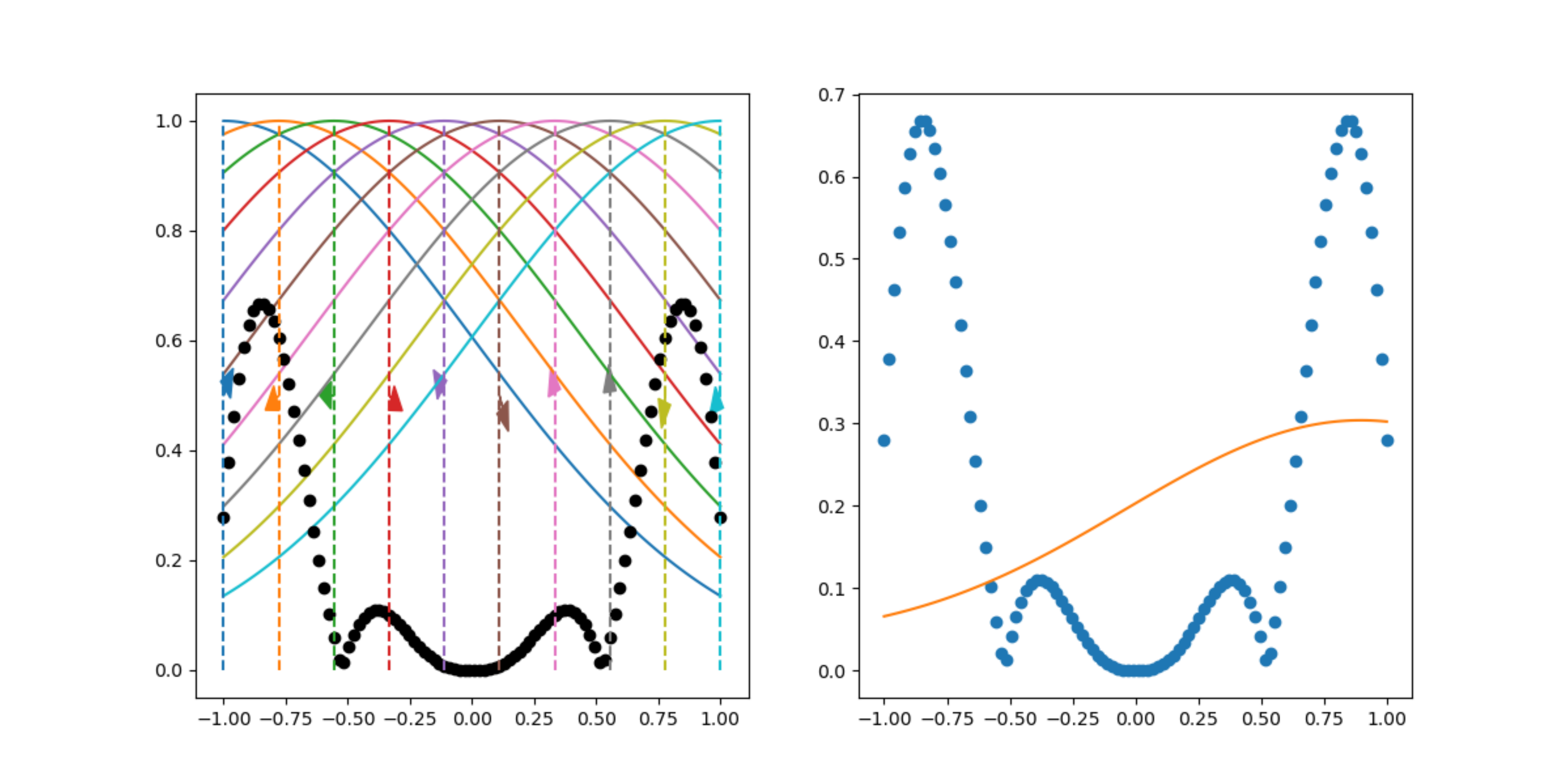
# OPTIMIZING RBFS




Unlikely local peaks will align on a nice grid.

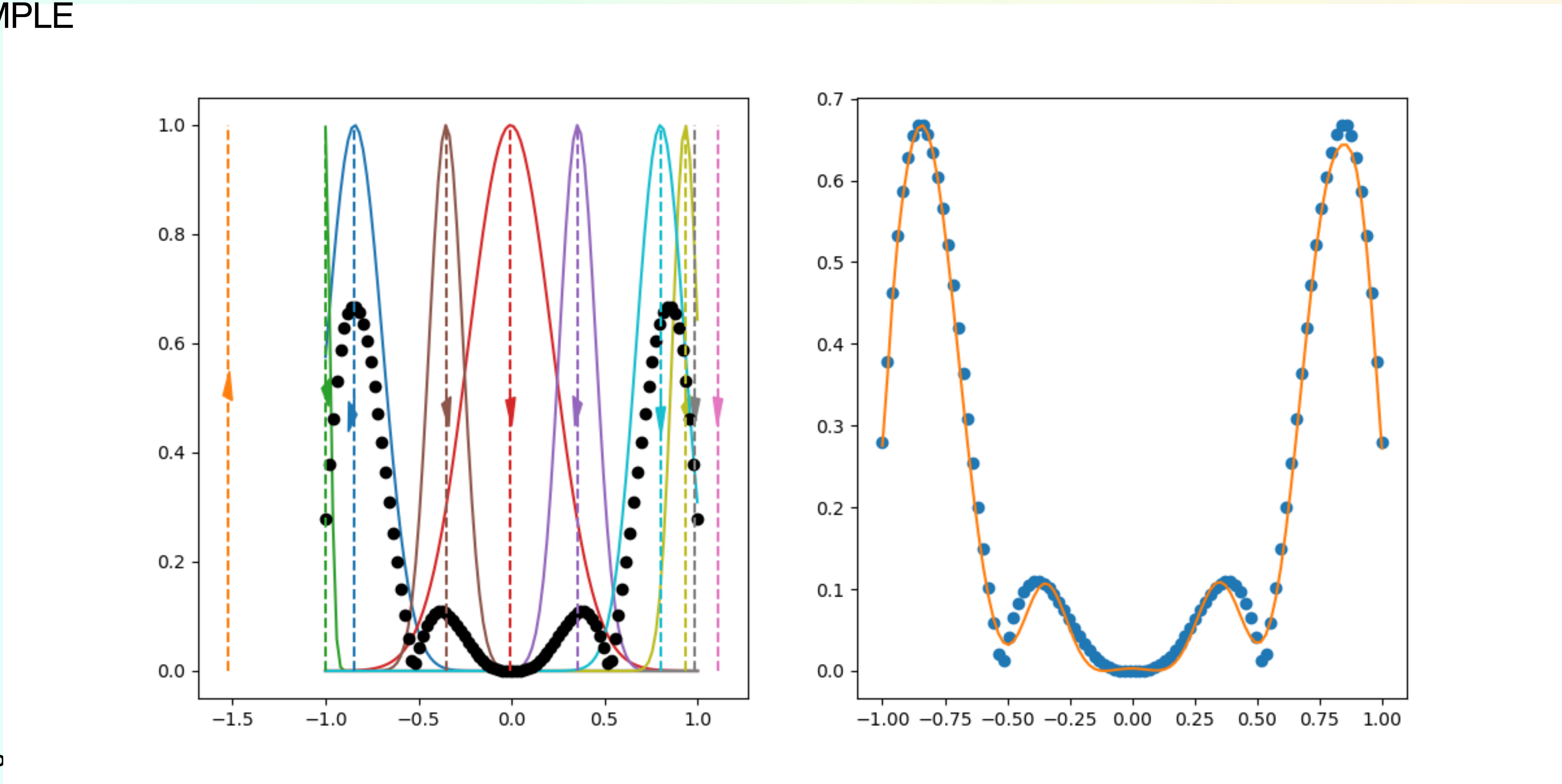
# OPTIMIZING RBFS

EXAM



# OPTIMIZING RBFS

EXAMPLE



# OPTIMIZING RBFS

EXAMPLE

- Centers and scales adapted to fit the function’s peaks and valleys
- Useless features were removed from the representation (no guarantee)

# SCALING TO HIGHER DIMENSIONS

CURSE OF DIMENSIONALITY

In low-dimension feature vectors $x$ (1 to 5)

- Basis functions work great!

What about higher dimensions?

- The number of features generated grows exponentially with the size of the input.
- Can't consider all possible combinations of features

# COMPOSING BASIS FUNCTIONS

## EXAMPLE

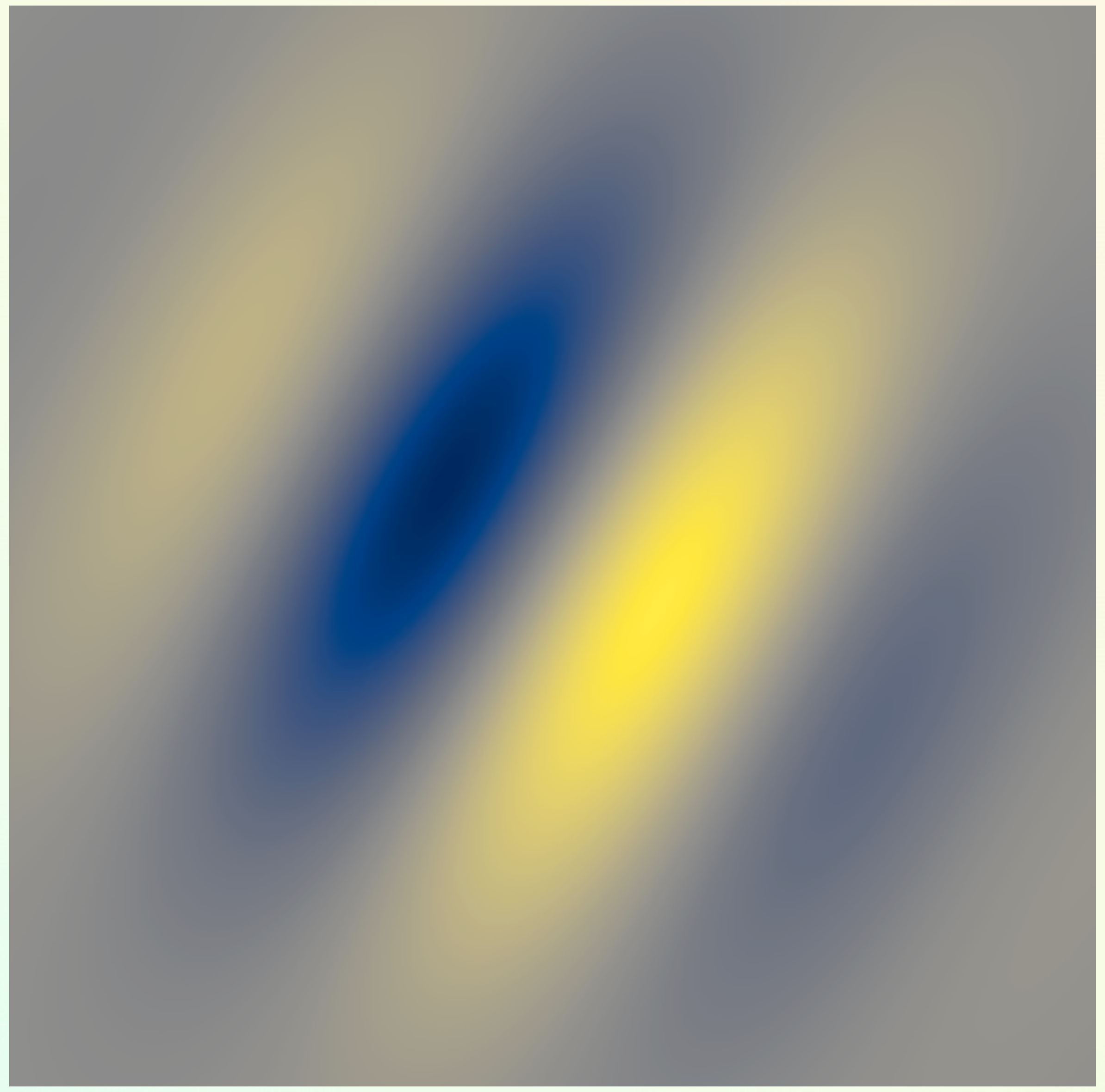
- Need to represent images to classify cat/dog
- Images can represented by lines, then groups of lines, groups of groups

# COMPOSING BASIS FUNCTIONS

## EXAMPLE

Line filter (a feature that activates if there is a line in a specific orientation)

Evidence that this is how neurons in early layers of the visual cortex represent images



# COMPOSING BASIS FUNCTIONS

## GENERAL PROCESS

$$f(x, w, \beta^1, \beta^2) = w^\top \phi^2\left(\phi^1(x, \beta^1), \beta^2\right)$$

$$h^1 = \phi^1(x, \beta^1)$$

$$h^2 = \phi^2(h^1, \beta^2)$$

$$\hat{y} = w^\top h^2$$

# COMPOSING BASIS FUNCTIONS

GENERAL PROCESS

$$f(x, w, \beta^1, \beta^2) = w^\top \phi^2 \left(\phi^1(x, \beta^1), \beta^2\right)$$

$h^1 = \phi^1(x, \beta^1)$

$h^2 = \phi^2(h^1, \beta^2)$

$\hat{y} = w^\top h^2$

# NEXT CLASS

Next Class — Neural Networks!