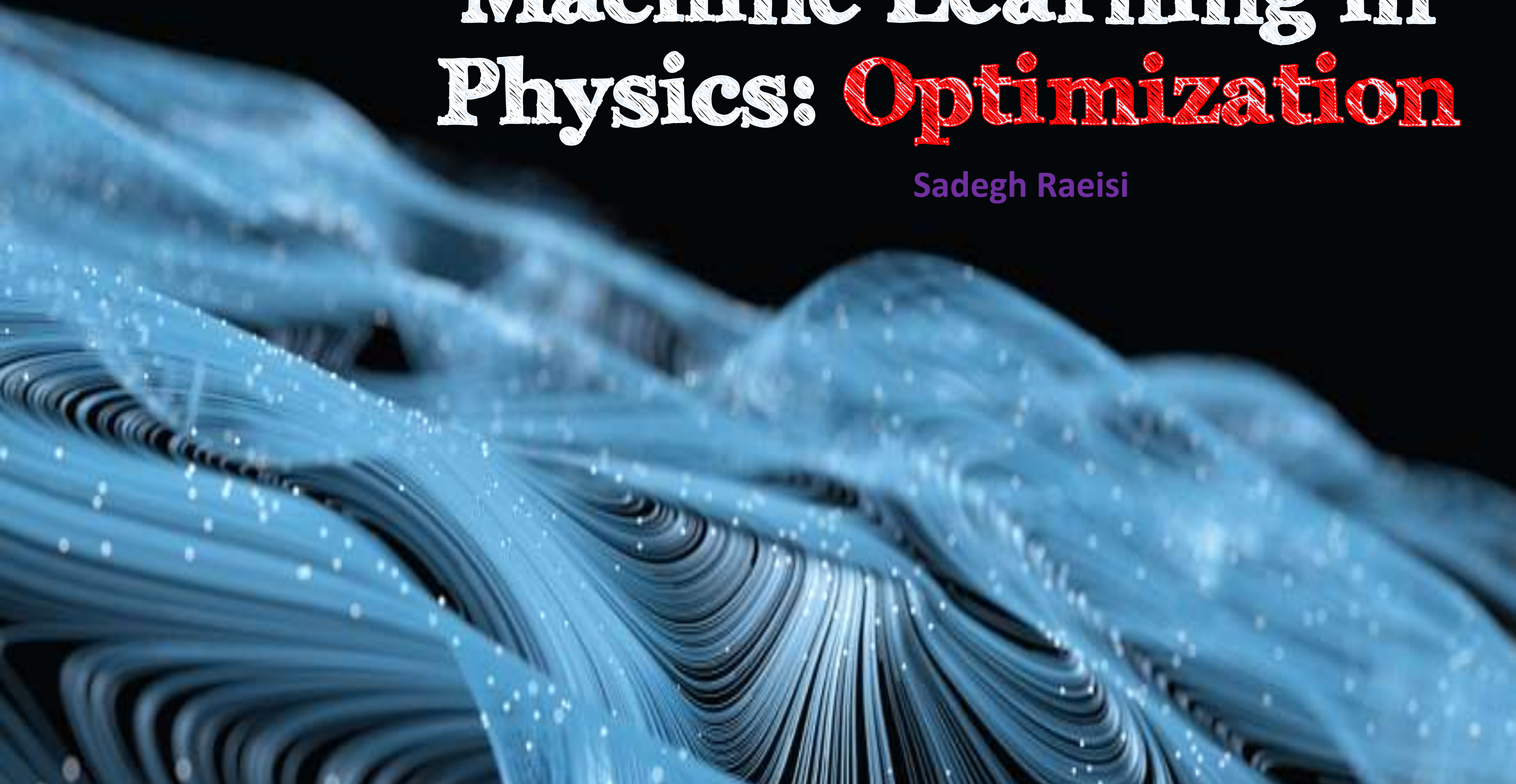
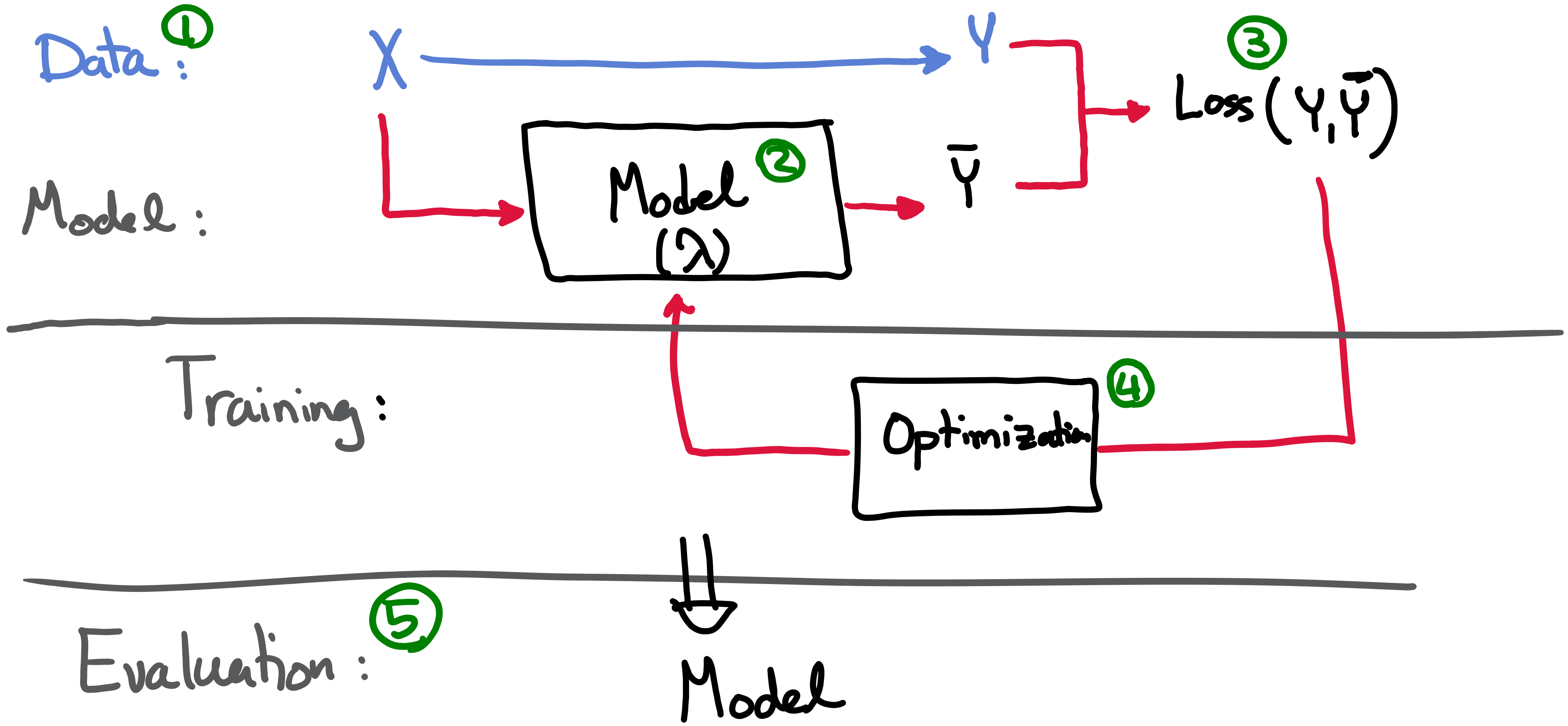
# Machine Learning in Physics: Optimization

Sadegh Raeisi



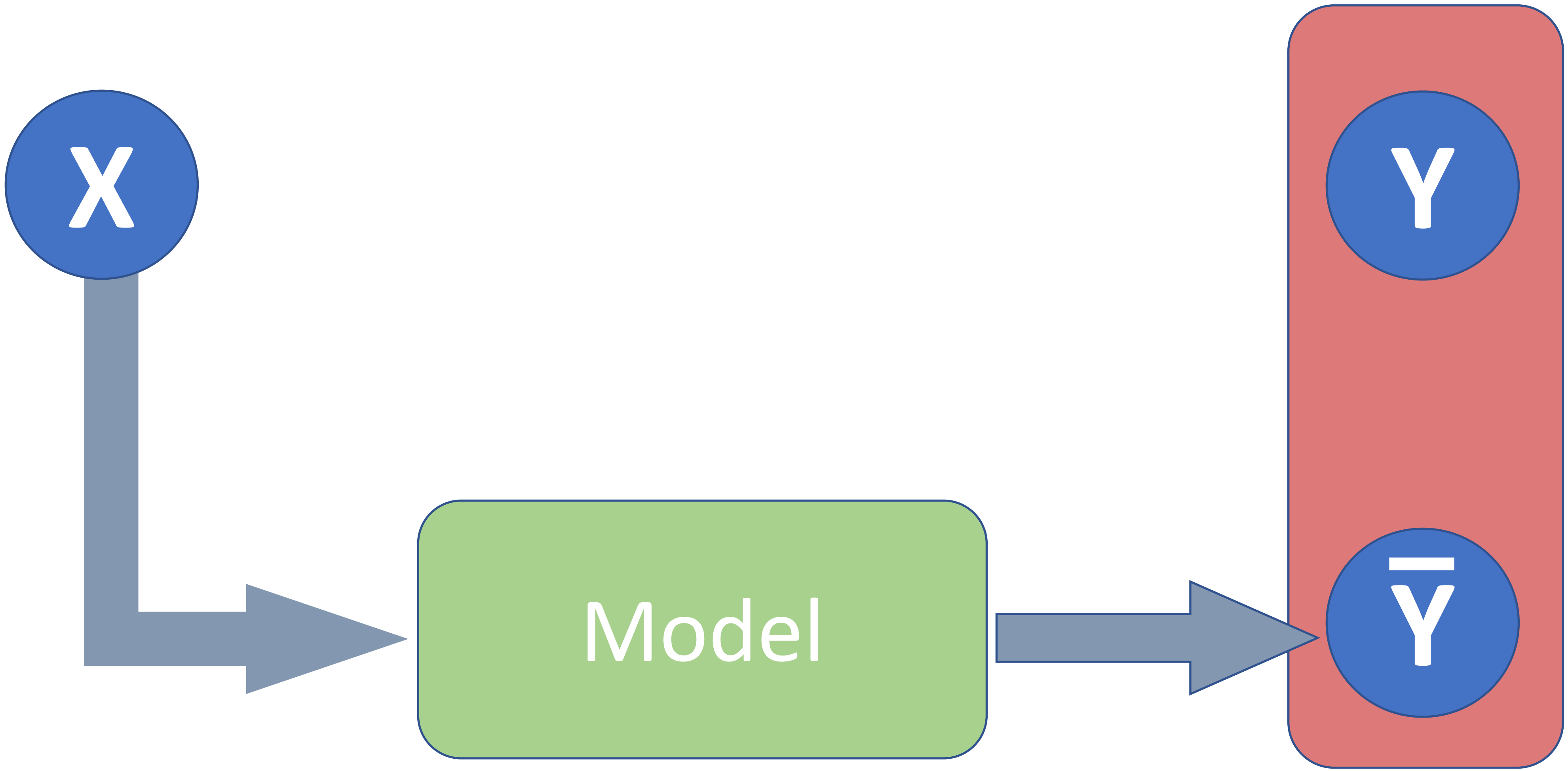
# Supervised: Ingredients



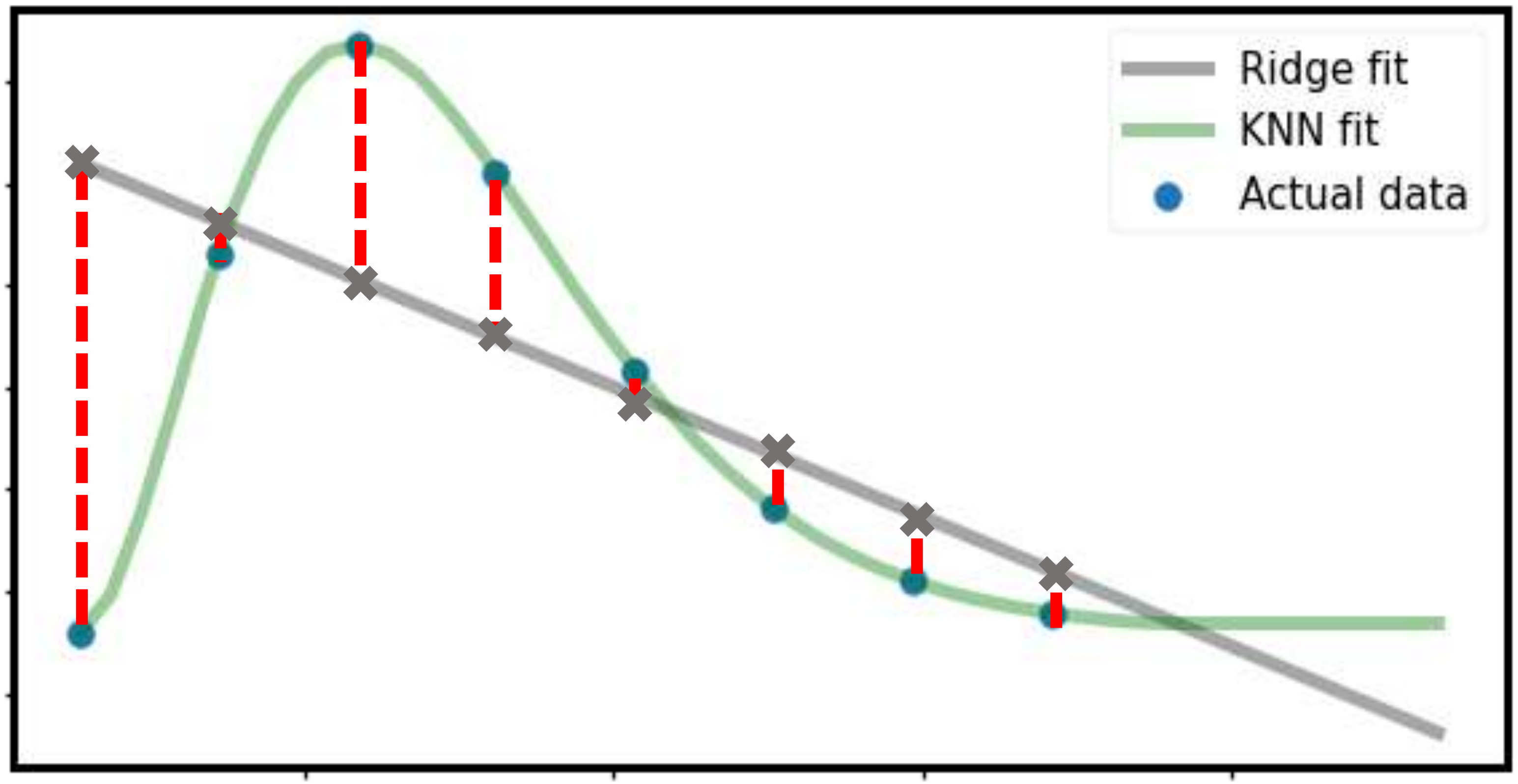
# Outline

- Concept
- Gradient descent
- Challenges
- Stochastic gradient descent
- Weight decay
- Adaptive learning rate

# Concept


X
Model
Y
Ȳ

# How close are the predictions?



# How can we quantify the difference?

$$\text{Dist}(Y, \bar{Y})$$

$$\boldsymbol{f_w(X)}$$

# How can we quantify the difference?

$$\mathcal{L}_{\boldsymbol{w}}\left( Y, \bar{Y} \right)$$

X Y

f

$$\mathcal{L}(\boldsymbol{w})$$

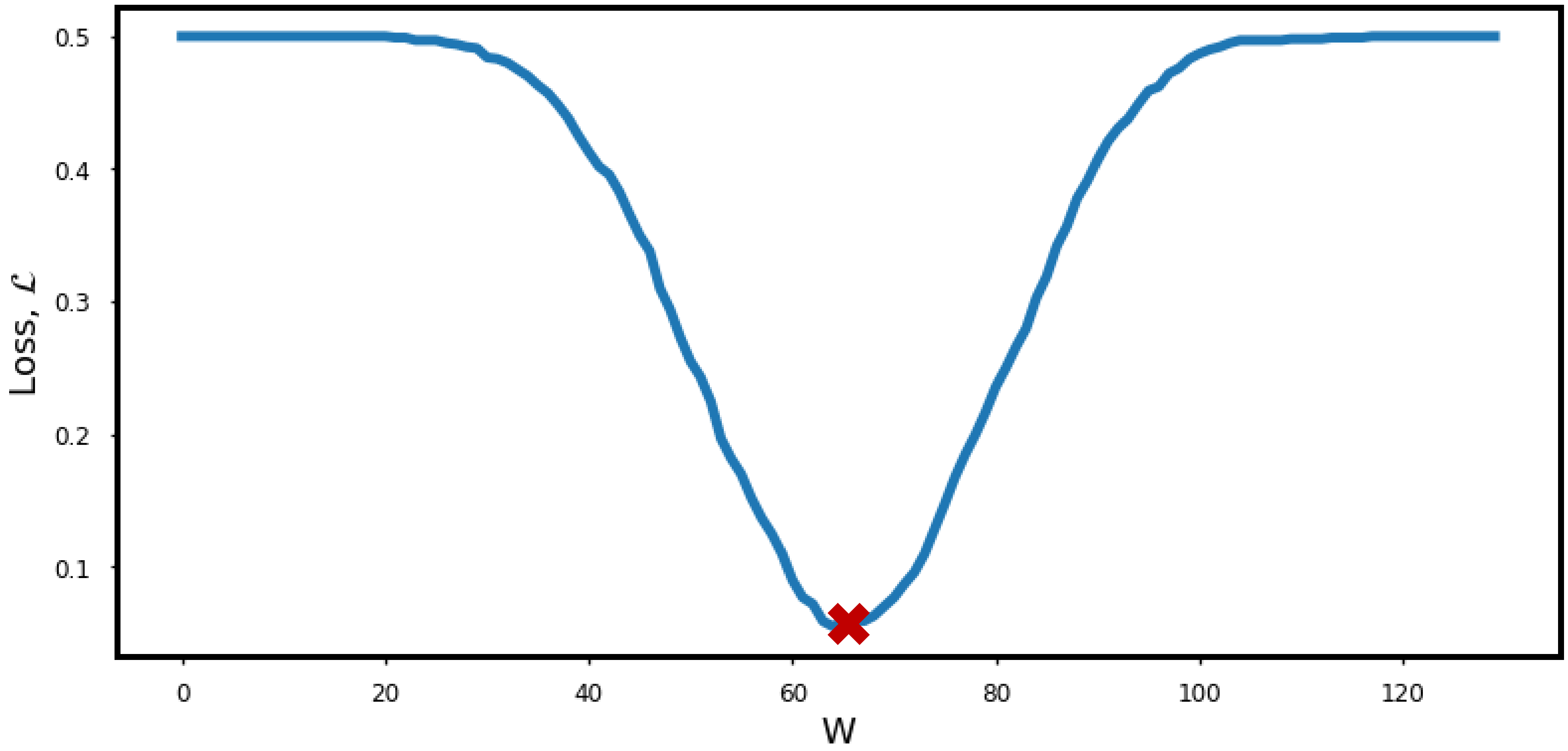
# Problem statement

X Y

f

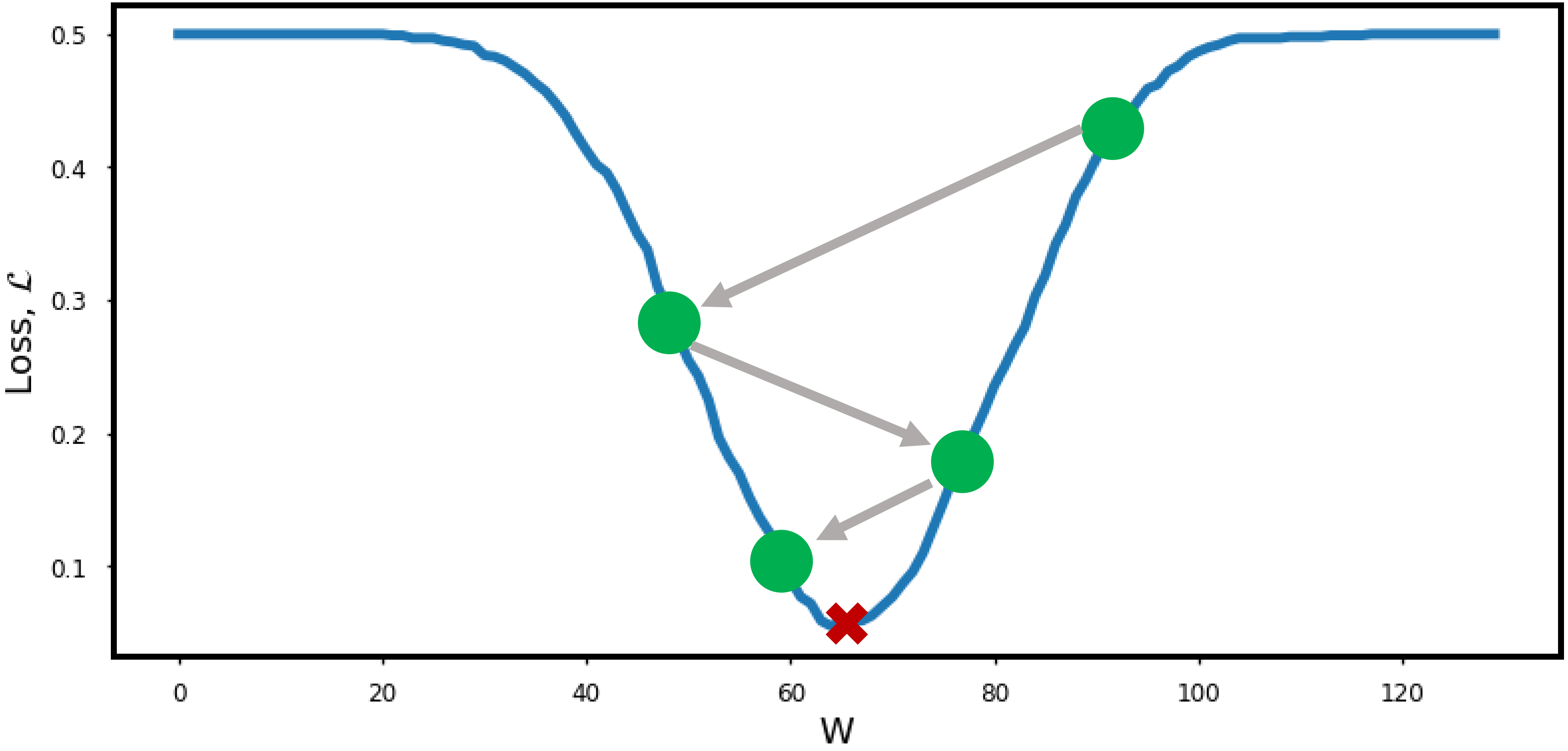
$$\underset{\mathbf{w}}{\textbf{Minimize}}\ \ \mathcal{L}(\boldsymbol{w})$$

# Loss landscape



# Gradient descent

# Idea



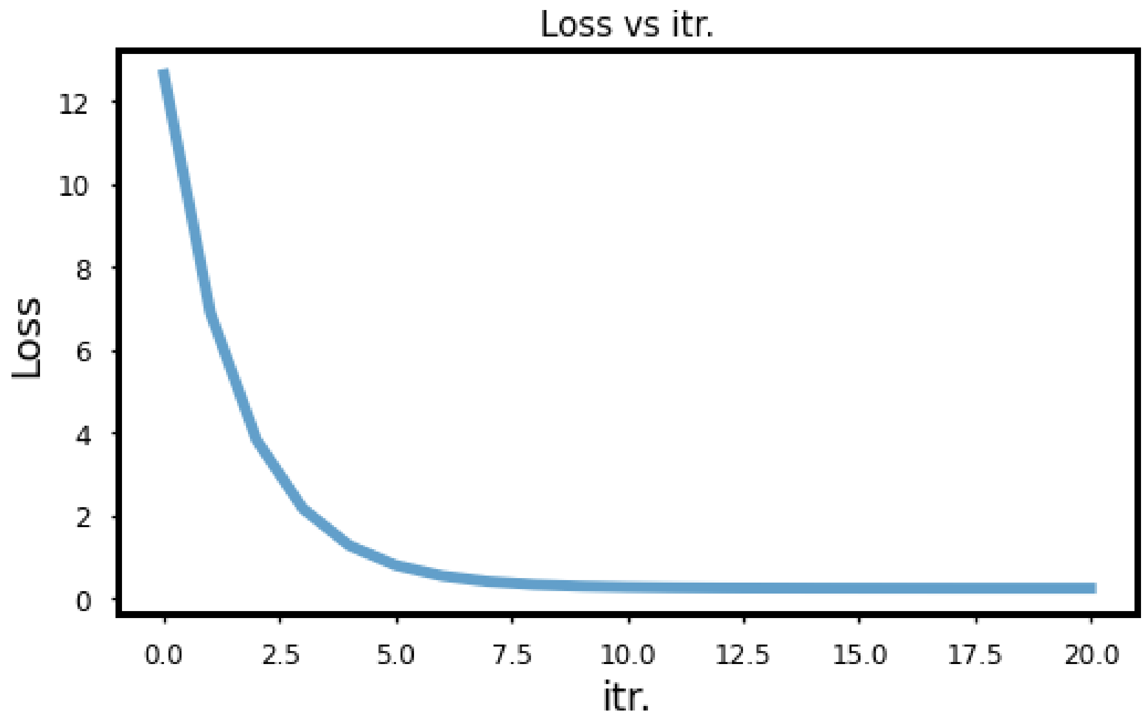
# Algorithm

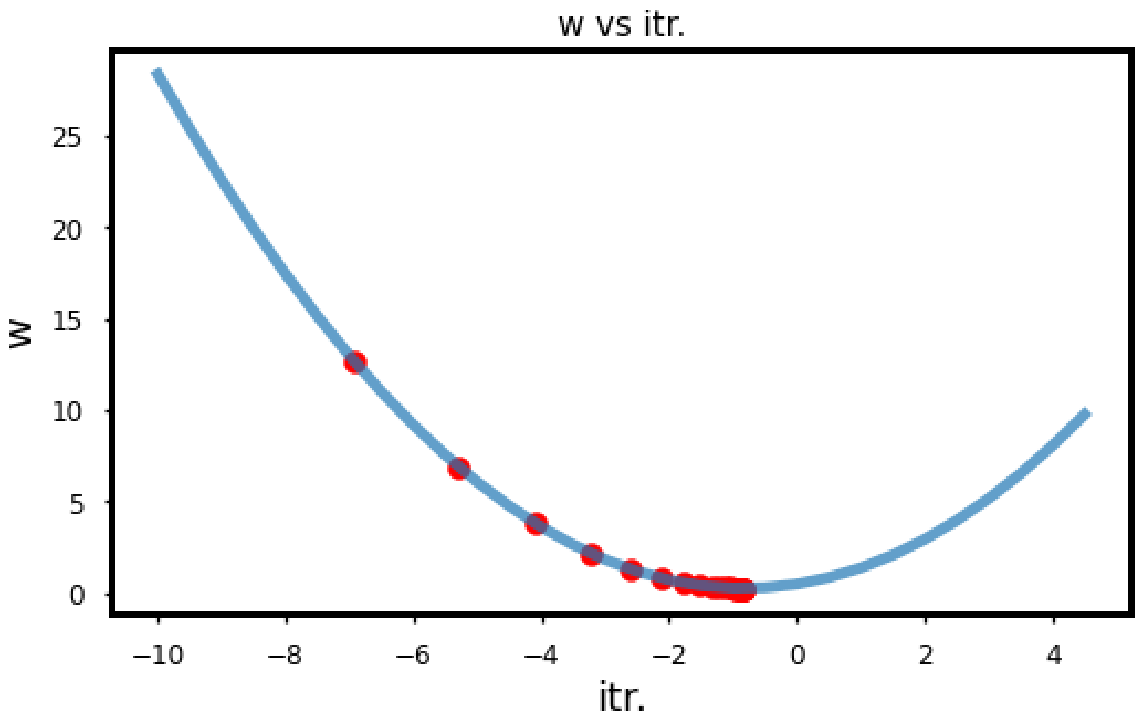
- Start with a random w.
- Calculate the derivative $\frac{\partial \mathcal{L}}{\partial w}$.
- Update $\mathrm{w} \rightarrow \mathrm{w} - \eta \frac{\partial \mathcal{L}}{\partial w}$.
- Repeat


- Start with a random w.
- Calculate the derivative $\frac{\partial \mathcal{L}}{\partial w}$.
- Update $\mathrm{w} \rightarrow \mathrm{w} - \eta \frac{\partial \mathcal{L}}{\partial w}$.
- Repeat

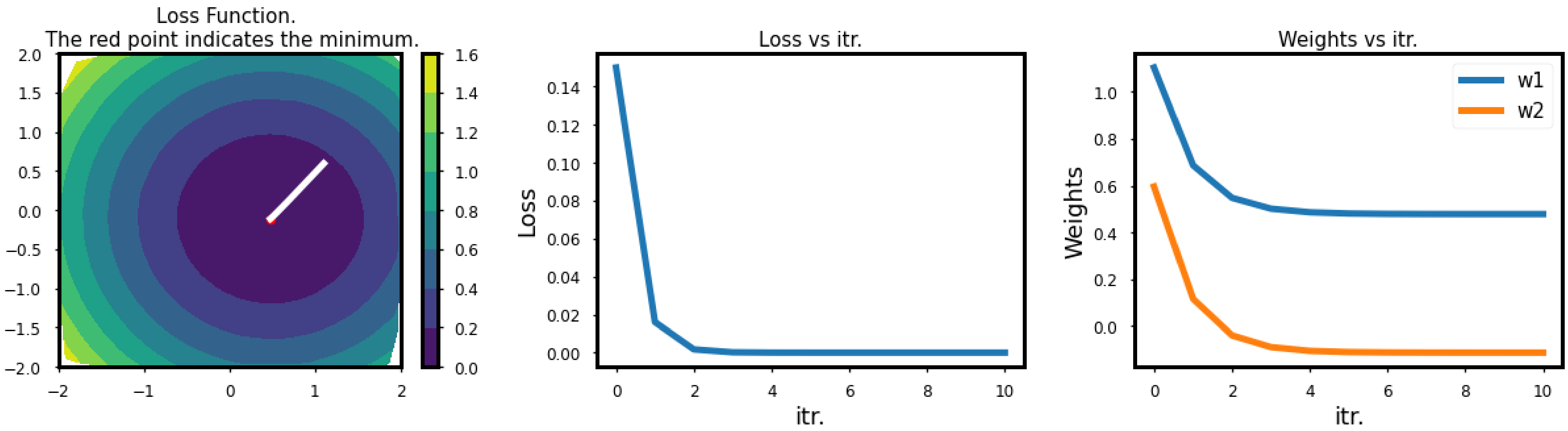

**$\boldsymbol{\eta}$ : Learning rate**

# How does it work?

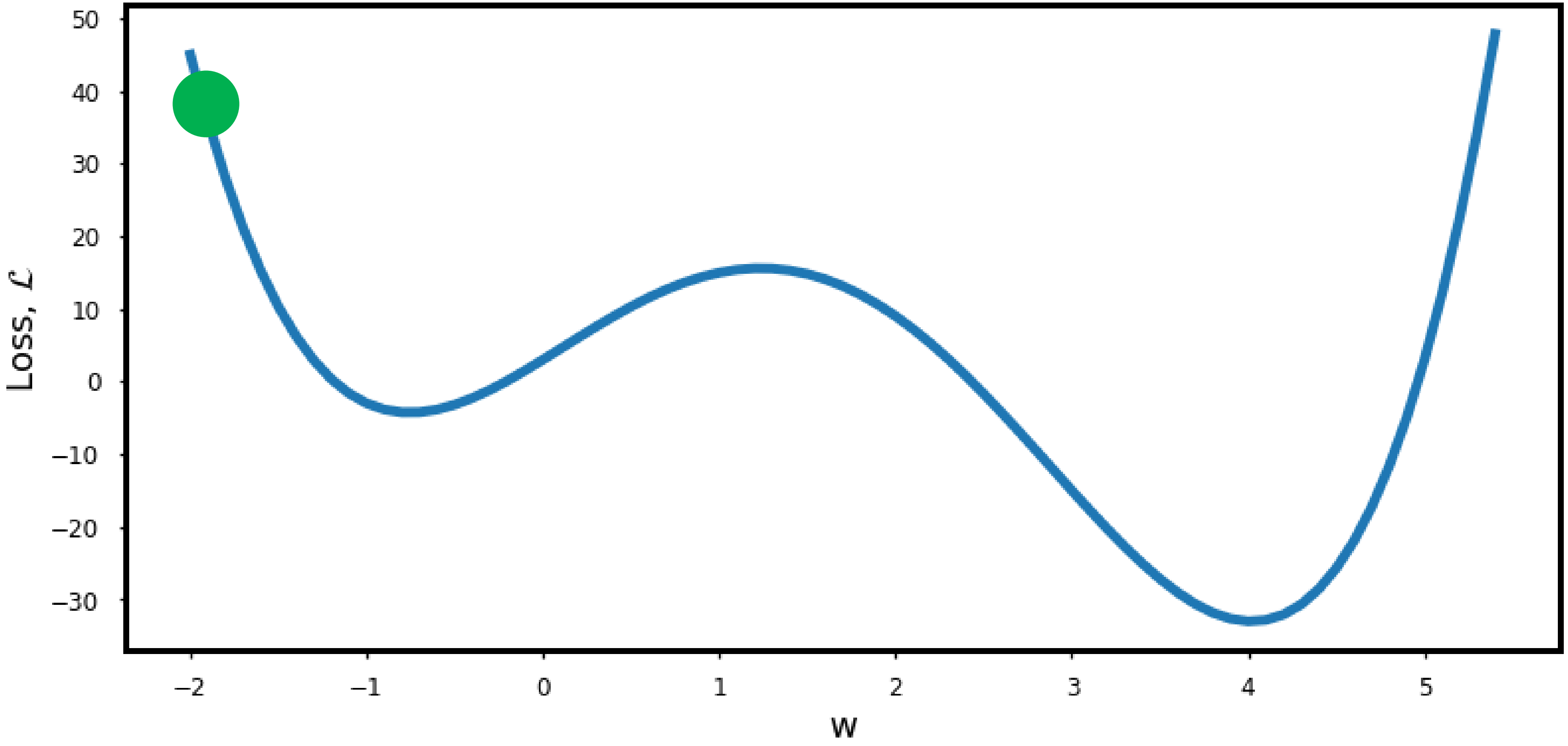


# How does it work?

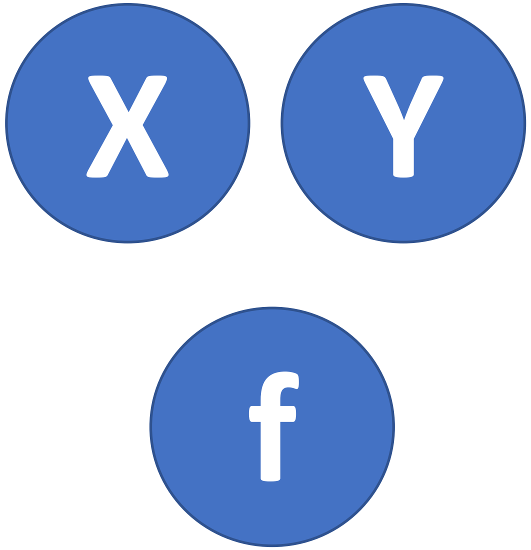


# What if there is a local minimum?



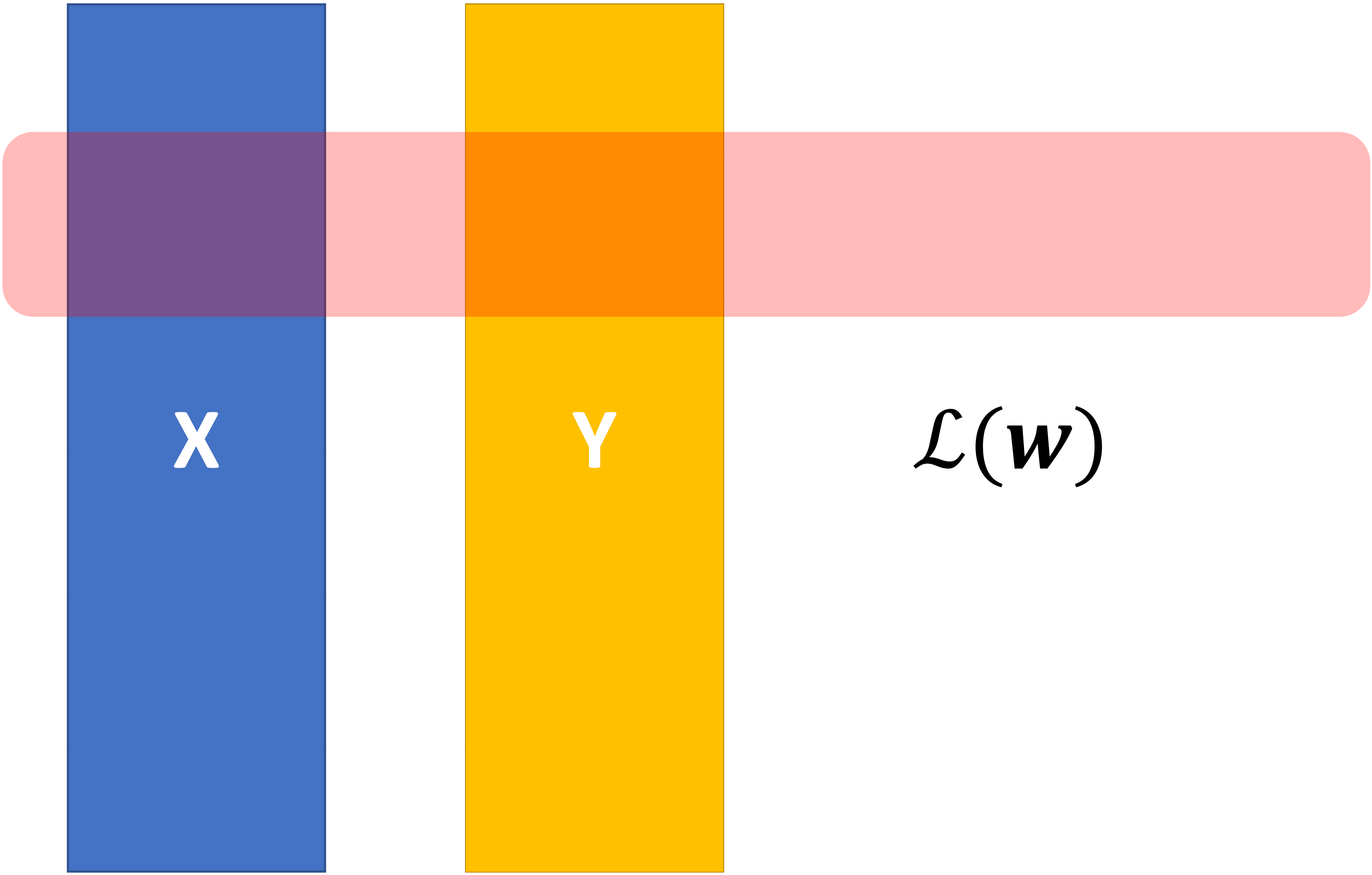
# Batch Optimization

# Reminder




$$\mathcal{L}(\boldsymbol{w})$$

# Batch Optimization



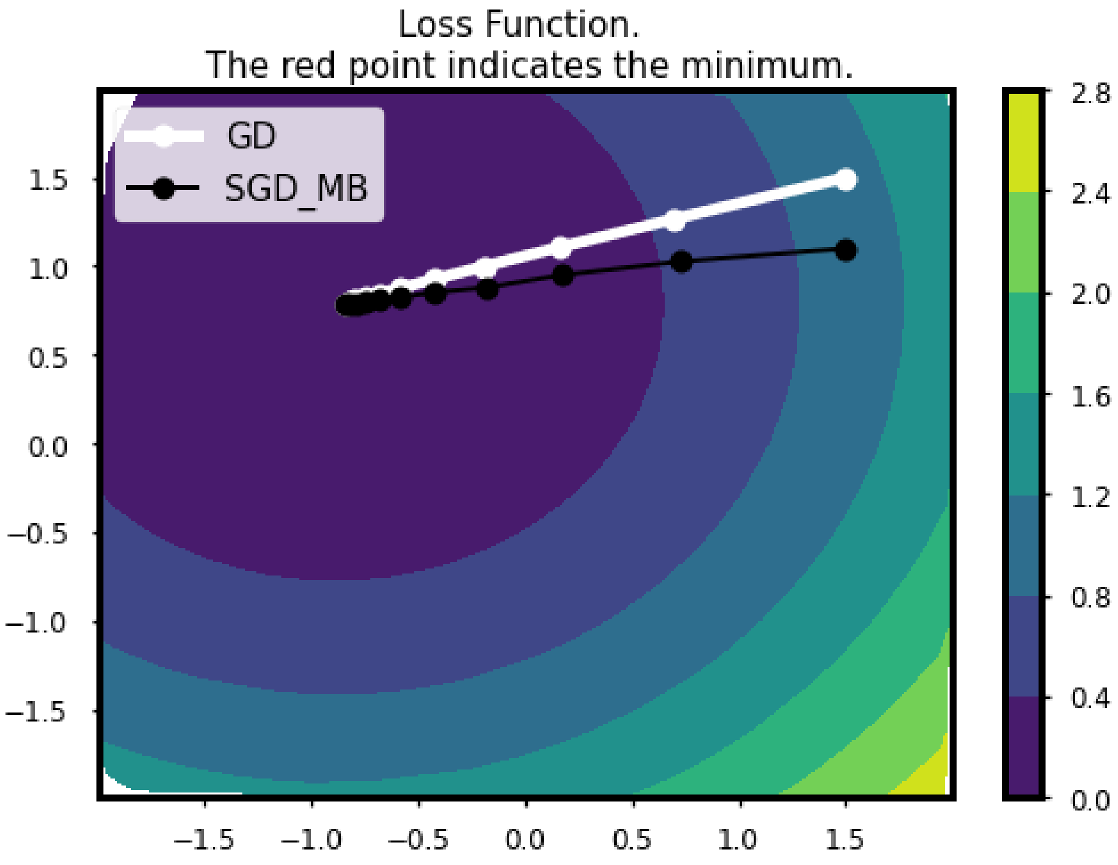

$\mathcal{L}(\boldsymbol{w})$

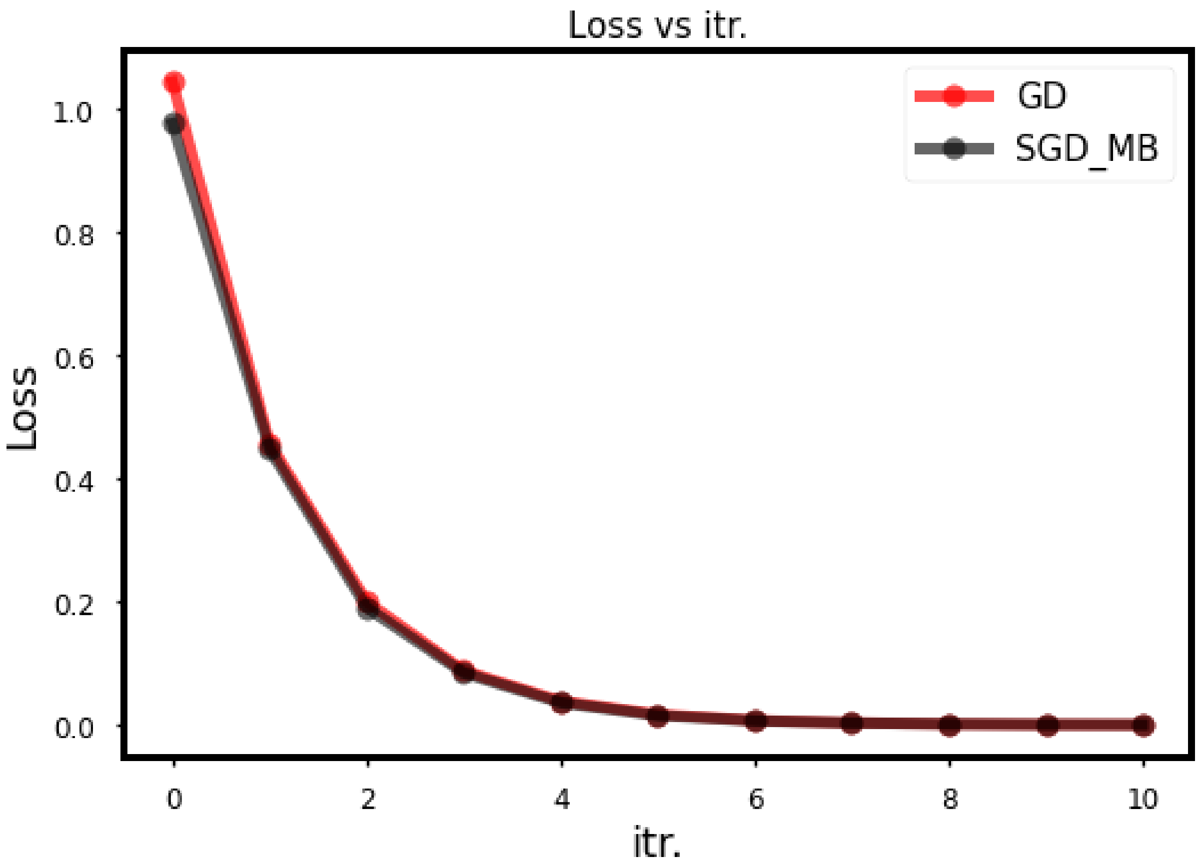
# Hyper-parameter

Batch-size

Learning rate

# How does it look like?

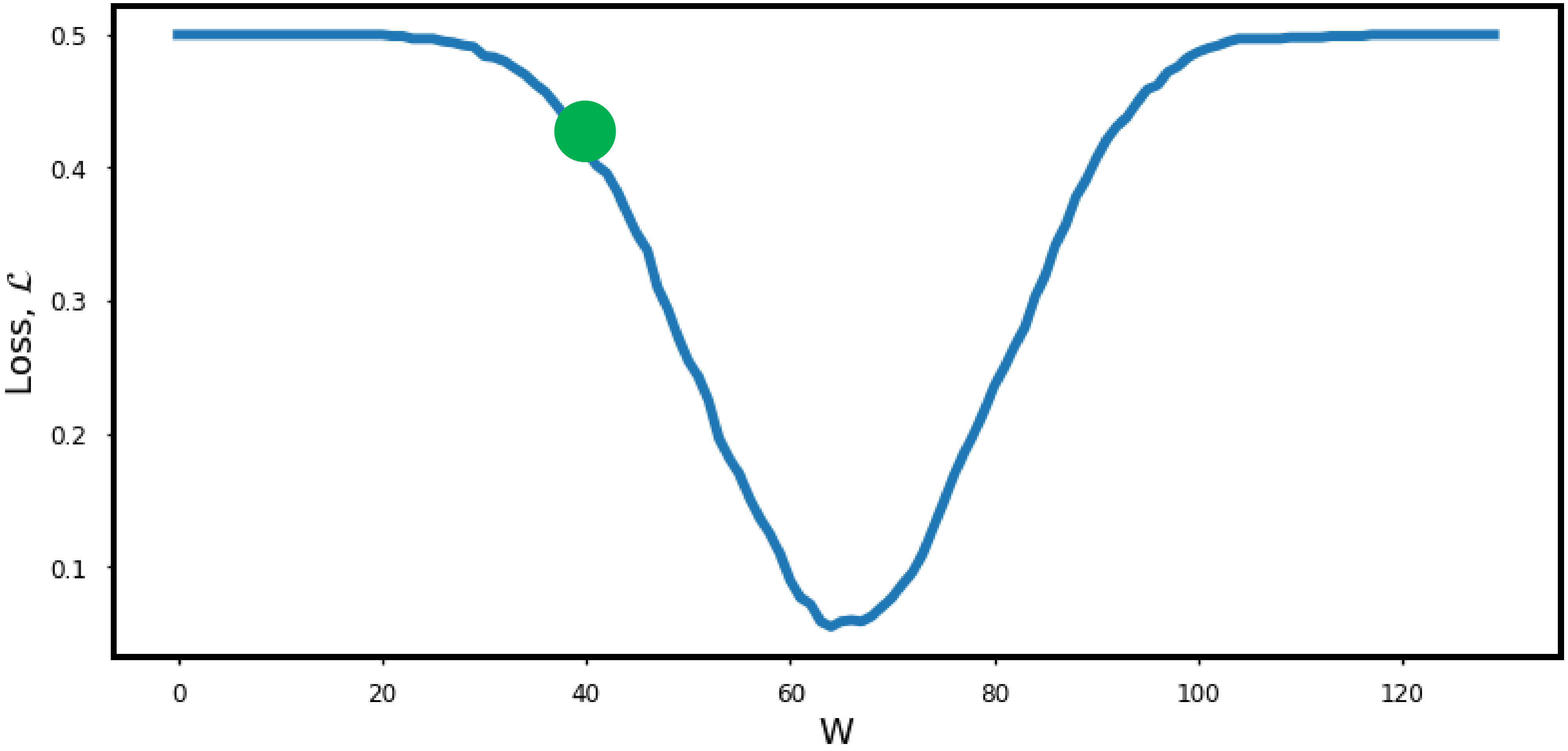


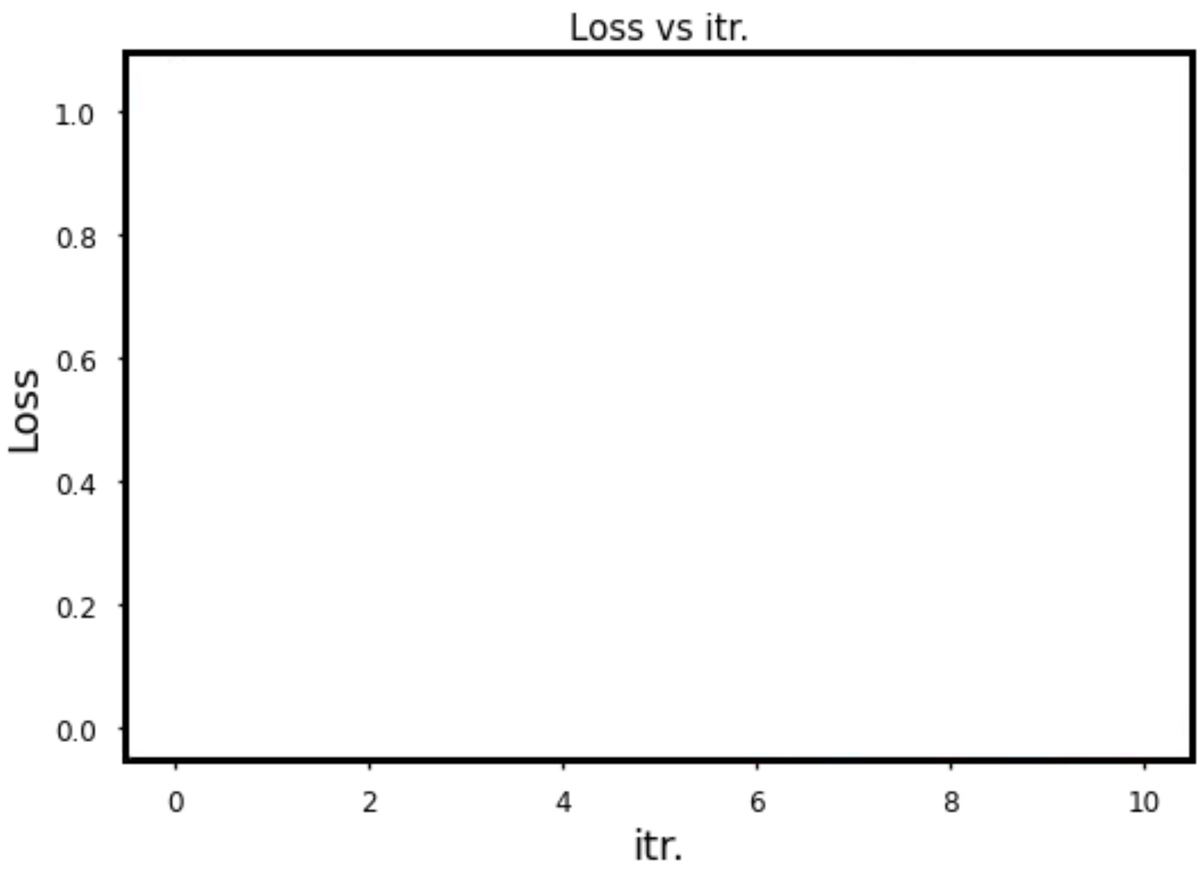
# How does batch optimization help?

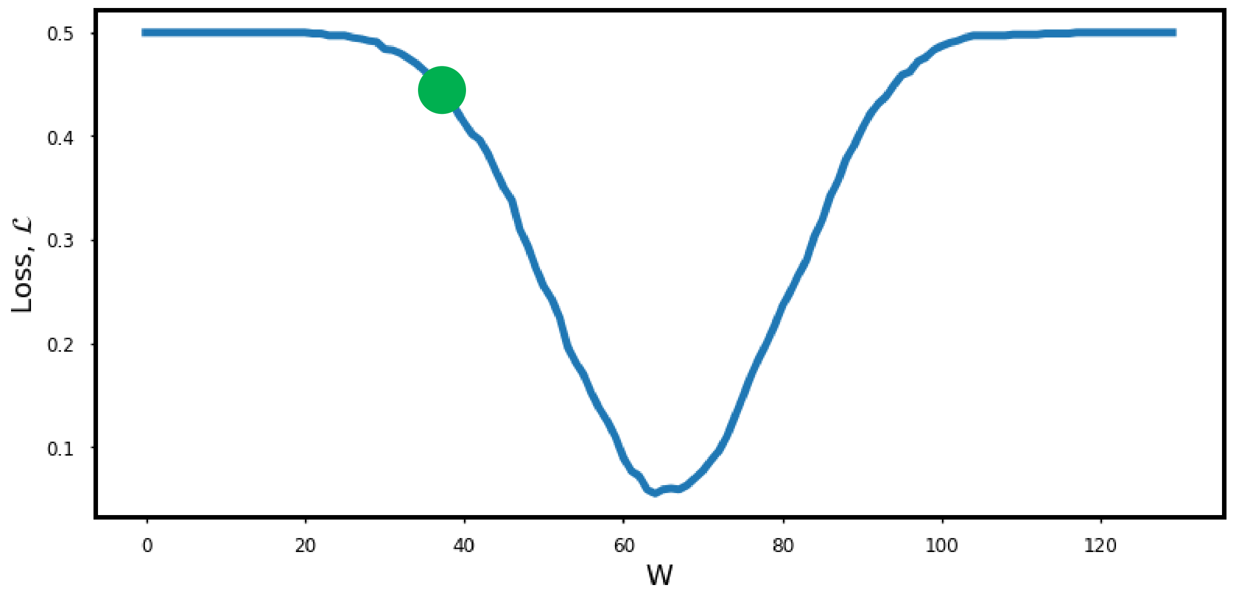
- Faster
- Can scape the local minimum

# What happens if lr is small?

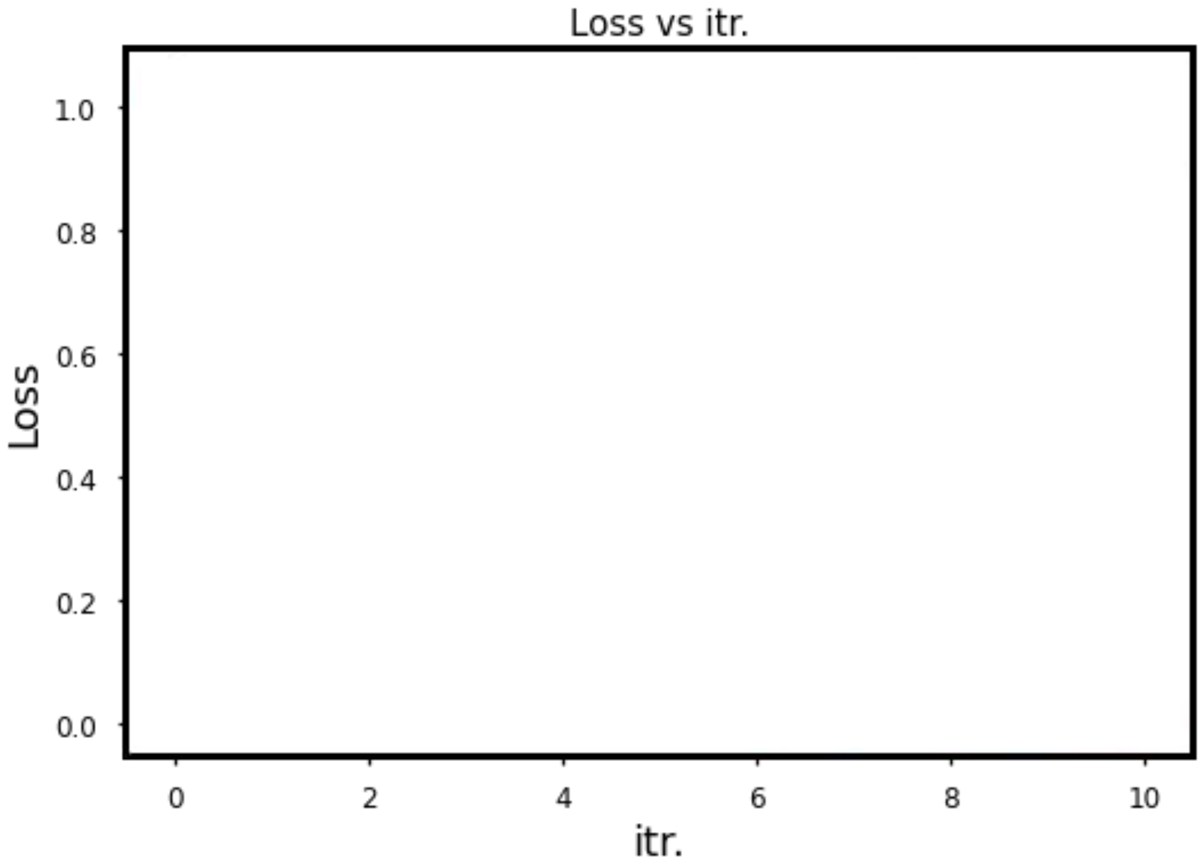


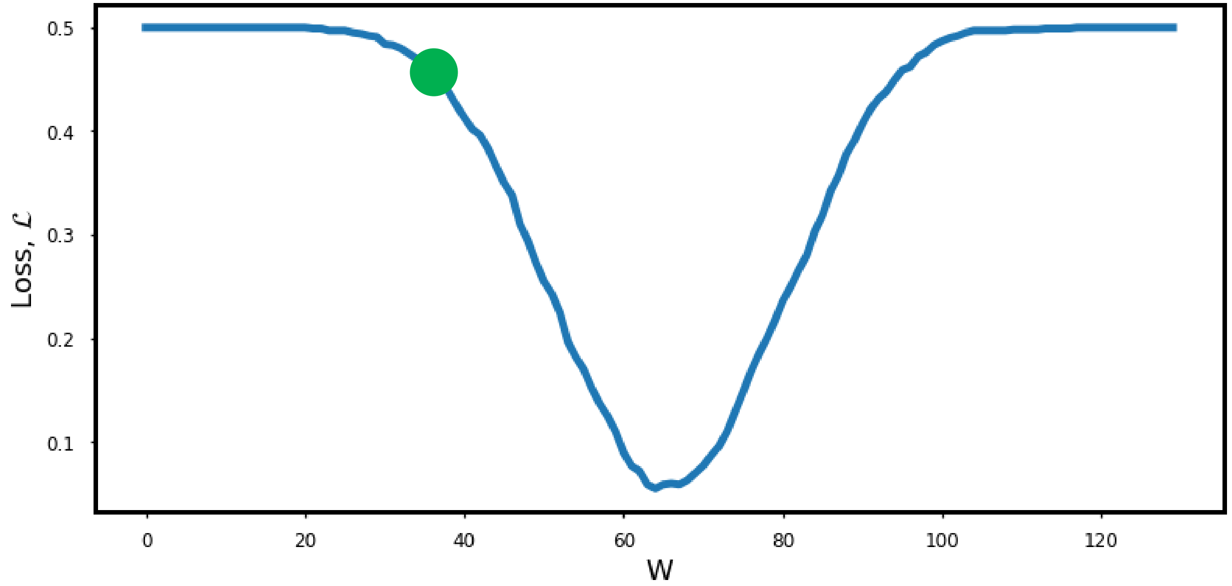
# What happens if lr is big?



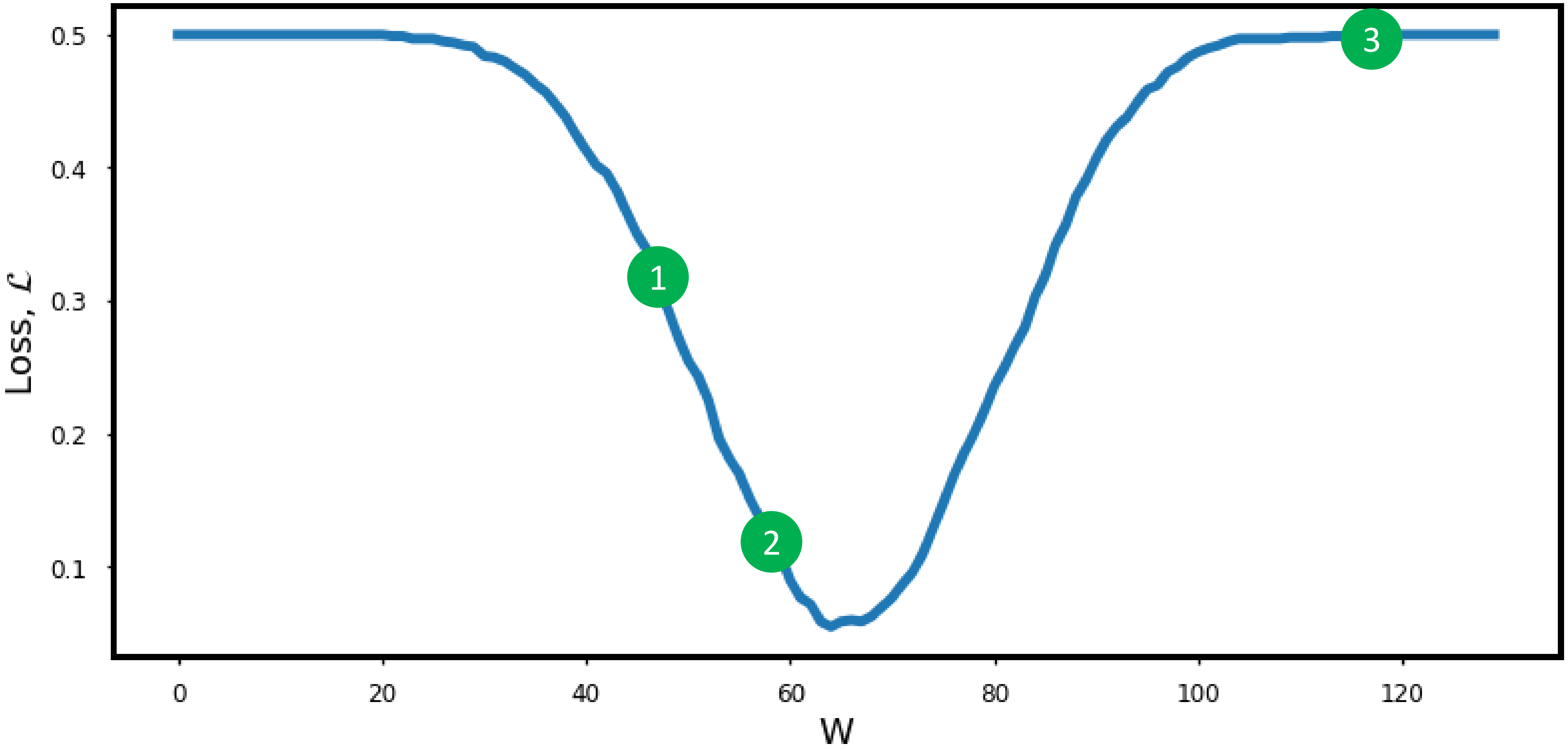



# What happens if lr is small?



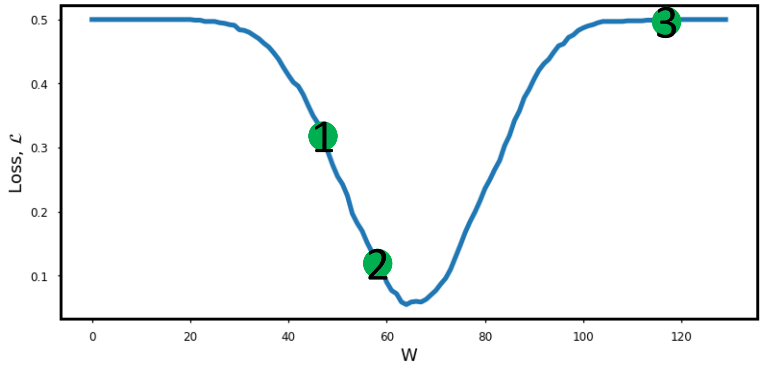



# What is a good choice for the lr?



# Variable learning rate

# Decaying learning rate



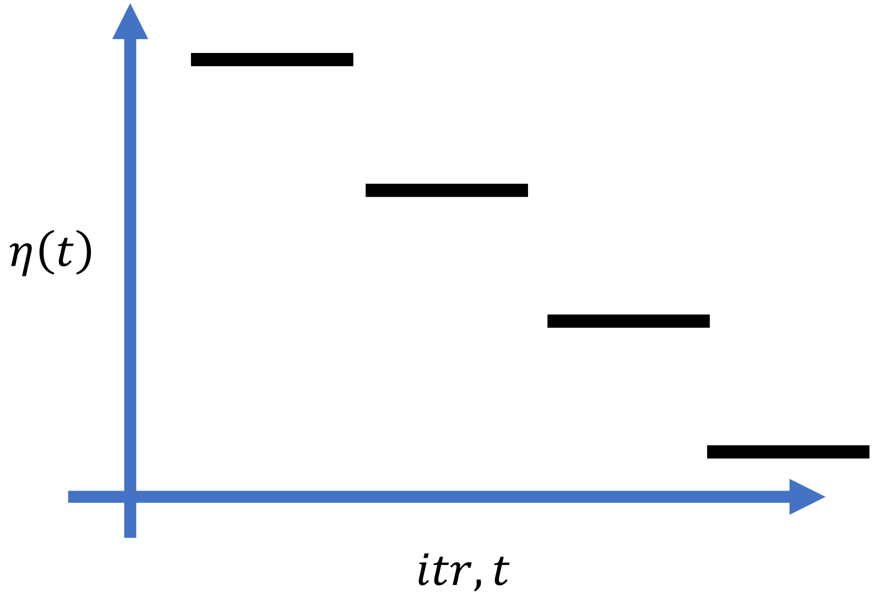

- $\eta(t) = \frac{1}{\sqrt{t}}\eta_0$

- $\eta(t) = e^{-kt}\eta_0$

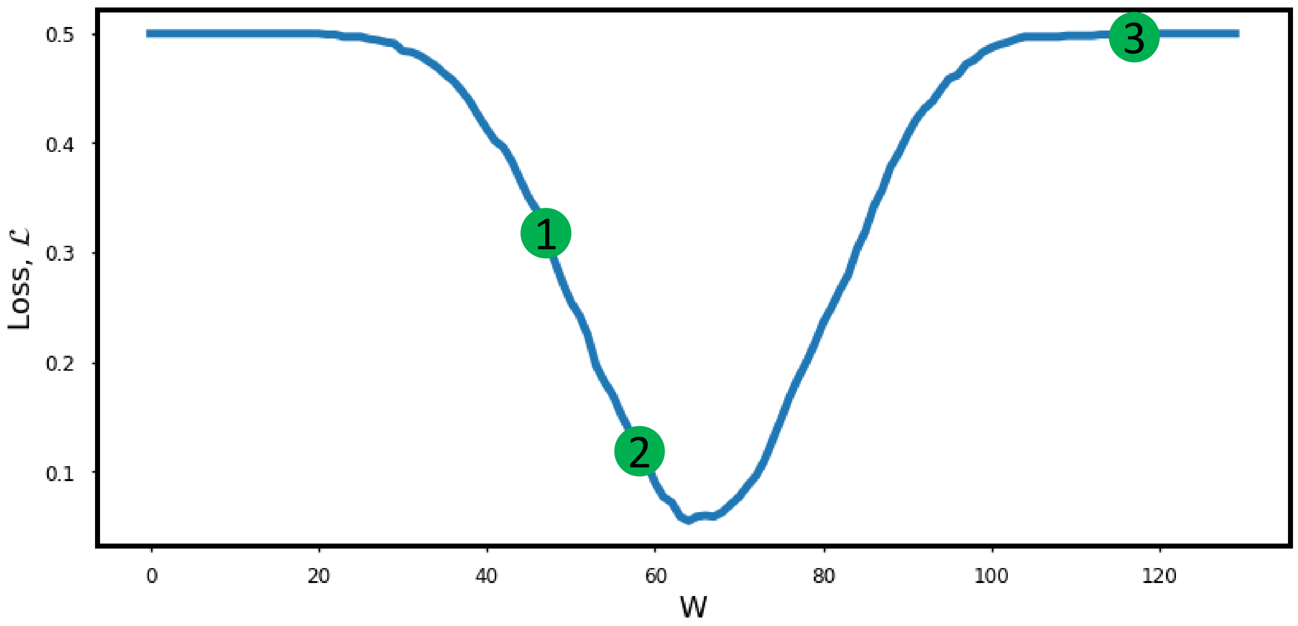
$t$ is the number of itr.

$\eta_0 \& k$ are a hyper-parameter.

- Staircase

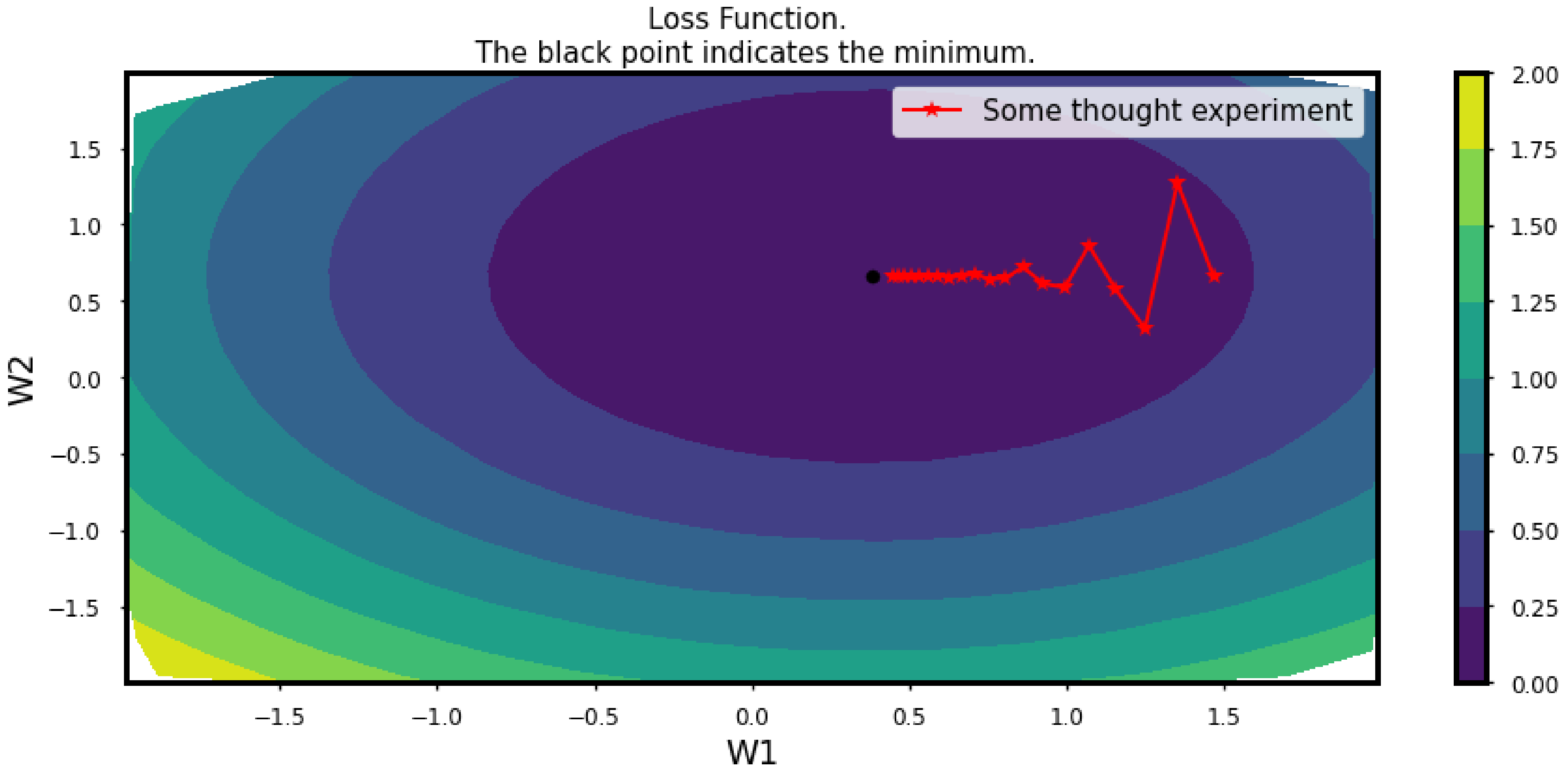


# Can we do better?



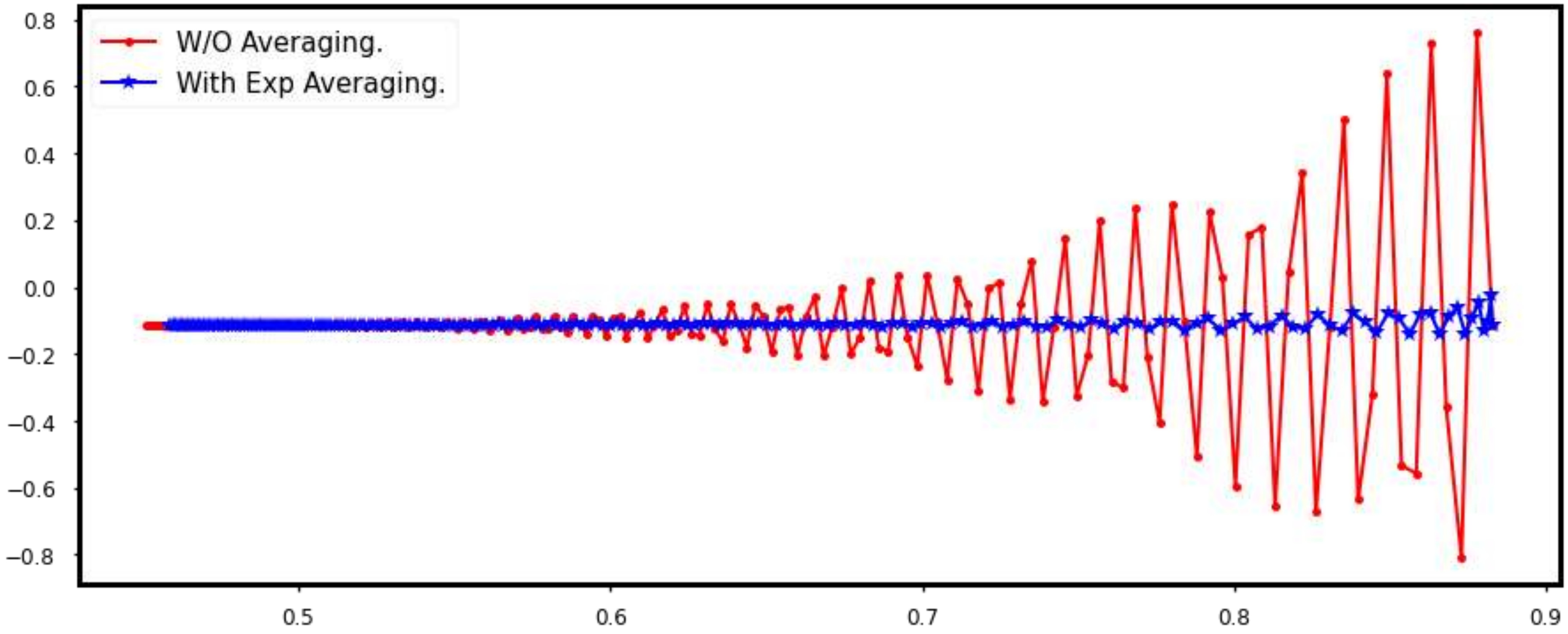
# Adaptive learning rate

# Imagine …



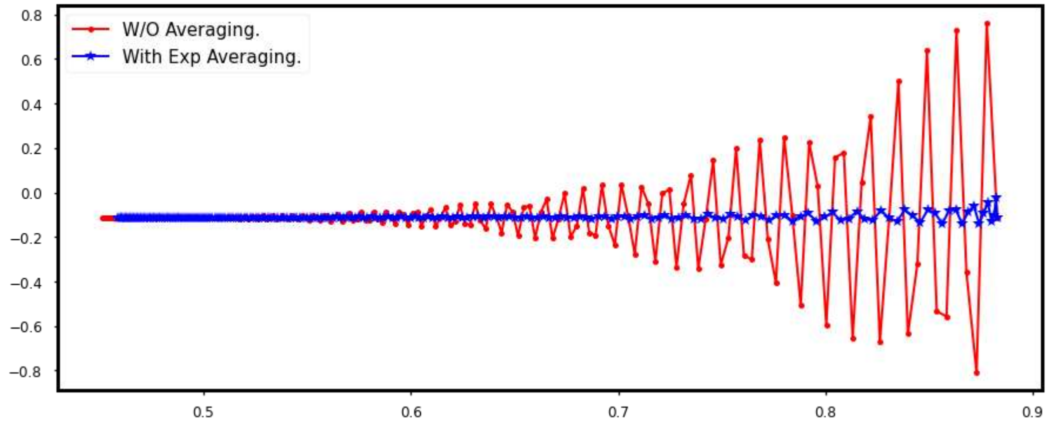
Loss Function.
The black point indicates the minimum.
Some thought experiment
W2
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
W1
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00

# Momentum



W/O Averaging.
With Exp Averaging.
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
0.5
0.6
0.7
0.8
0.9

# Momentum

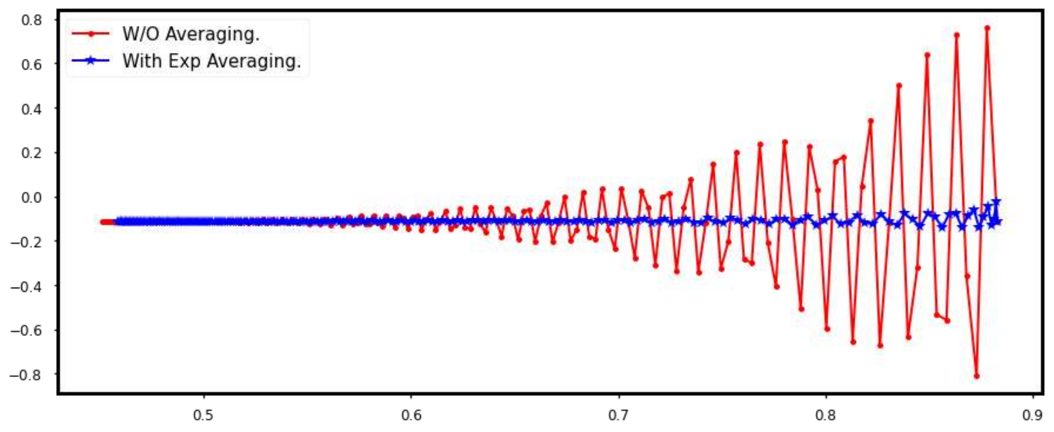



Update rule:

$$w \rightarrow w - \eta \frac{\partial \mathcal{L}}{\partial w}$$

$$w \rightarrow w - \eta \left( \frac{\partial \mathcal{L}}{\partial w} \right)_{avg}$$

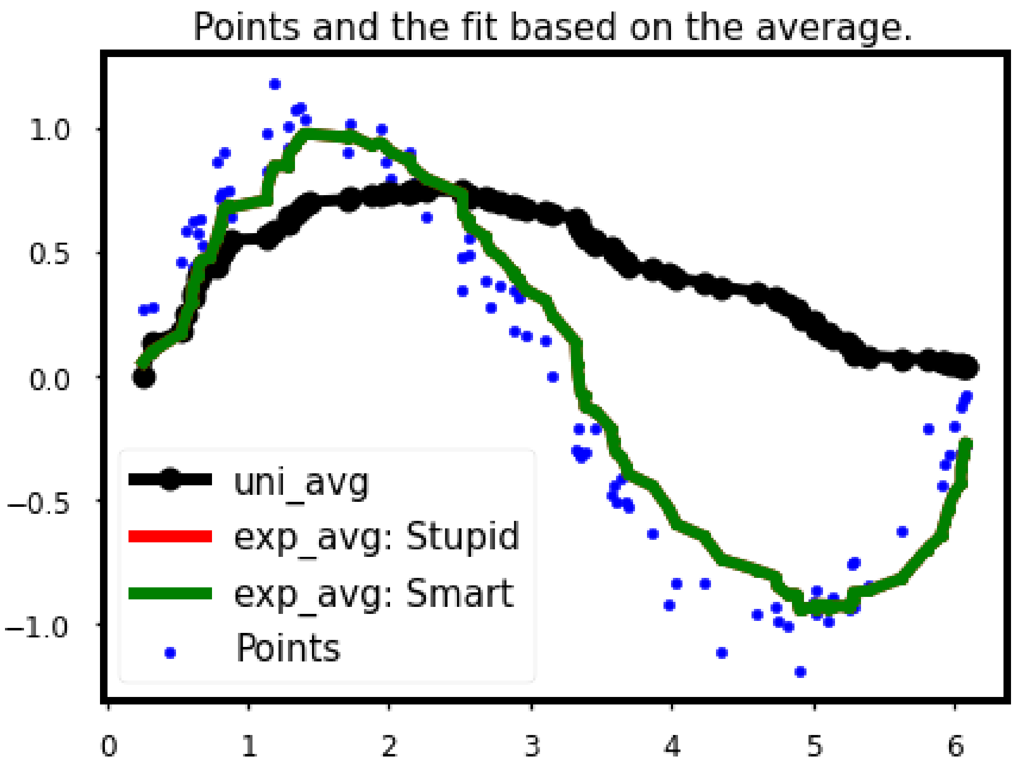
# What kind of average?

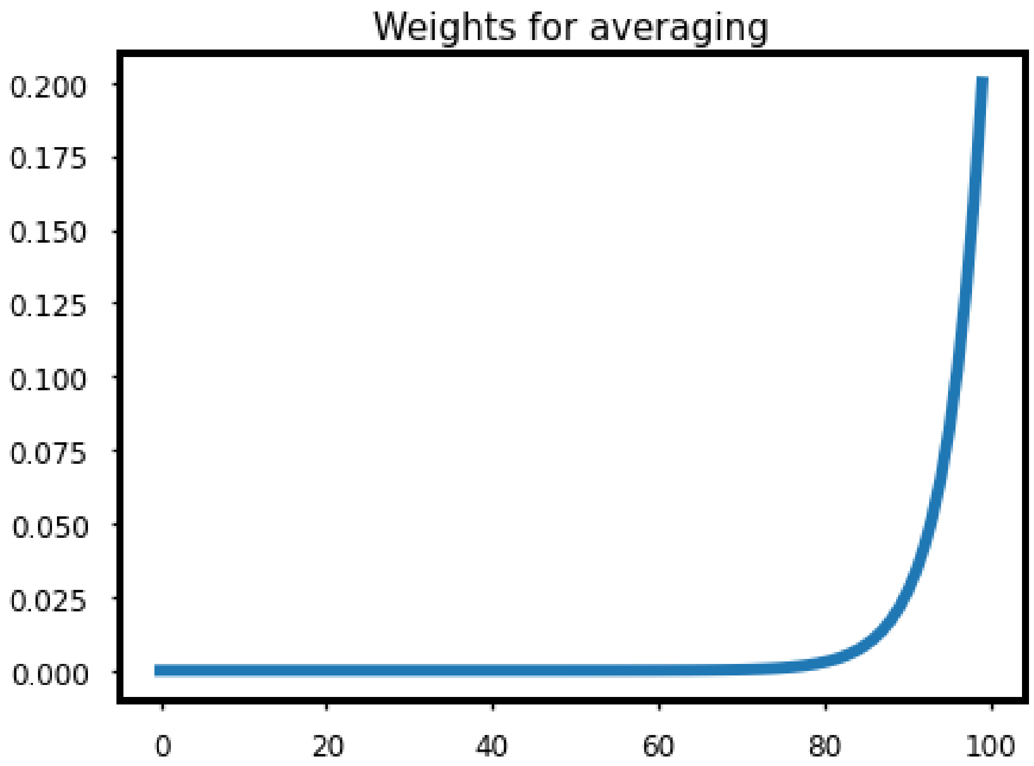



$$\left(\frac{\partial \mathcal{L}}{\partial w}\right)_{avg}^{t+1} = \beta \left(\frac{\partial \mathcal{L}}{\partial w}\right)_{avg}^{t} + (1-\beta)\left(\frac{\partial \mathcal{L}}{\partial w}\right)^{t+1}$$

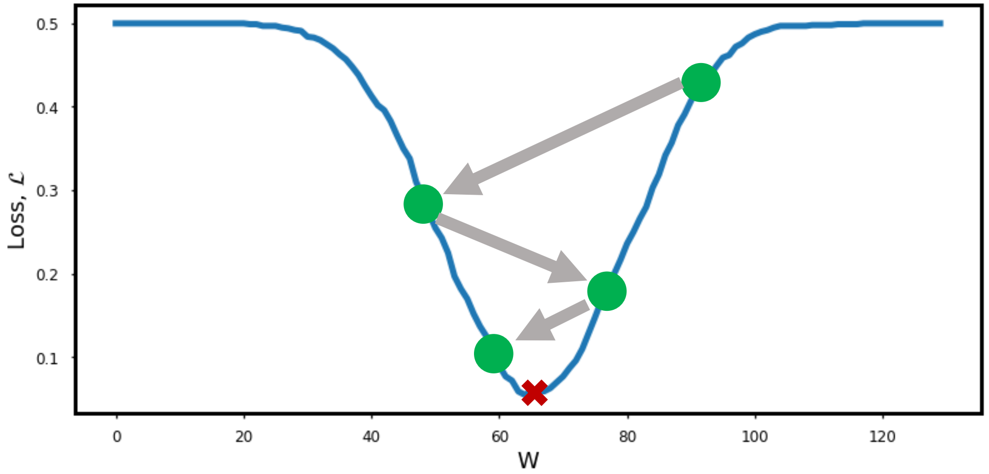
# Exponentially weighted average

$$\left(\frac{\partial \mathcal{L}}{\partial w}\right)_{avg}^{t+1} = \boldsymbol{\beta} \left(\frac{\partial \mathcal{L}}{\partial w}\right)_{avg}^{t} + (\mathbf{1} - \boldsymbol{\beta}) \left(\frac{\partial \mathcal{L}}{\partial w}\right)^{t+1}$$



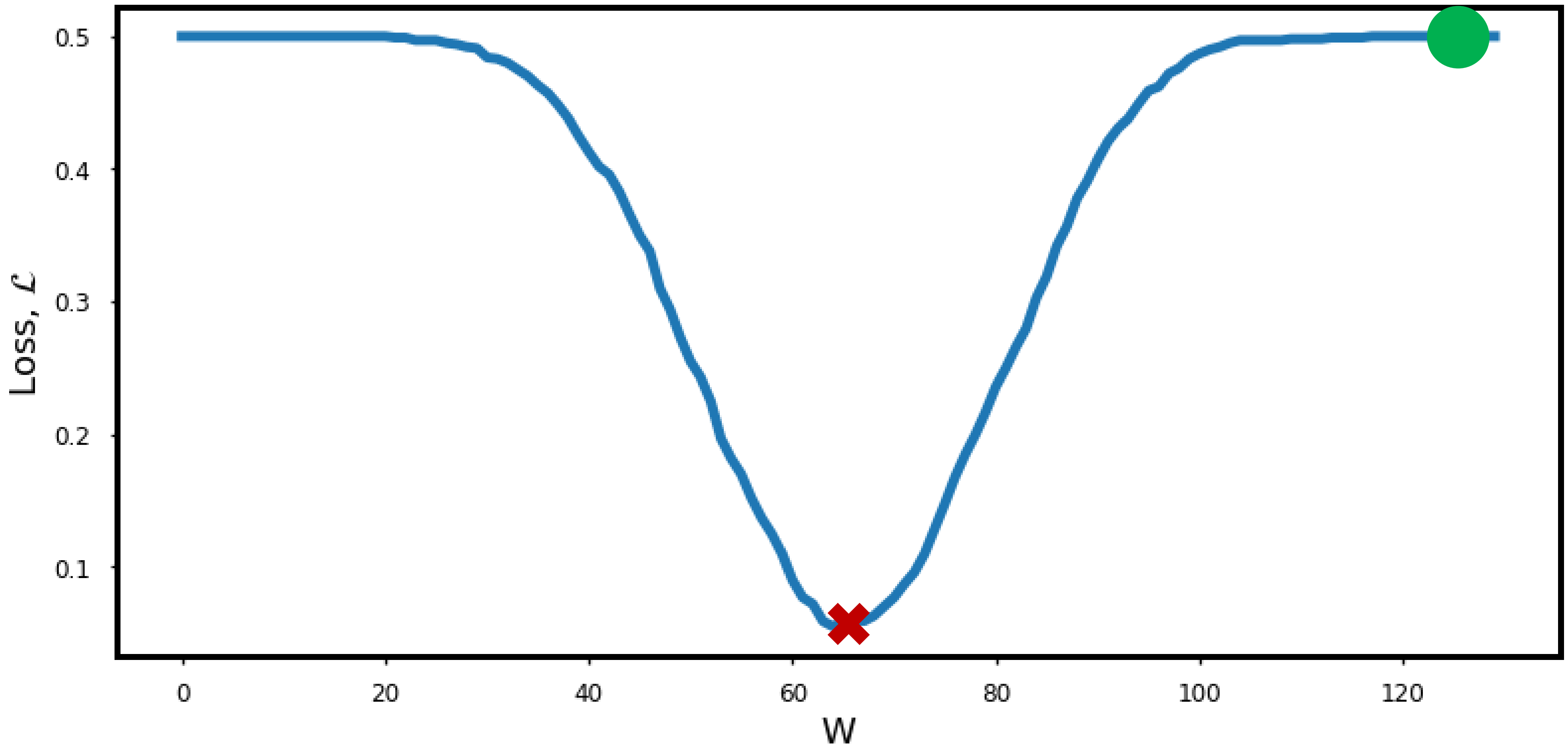
# Momentum



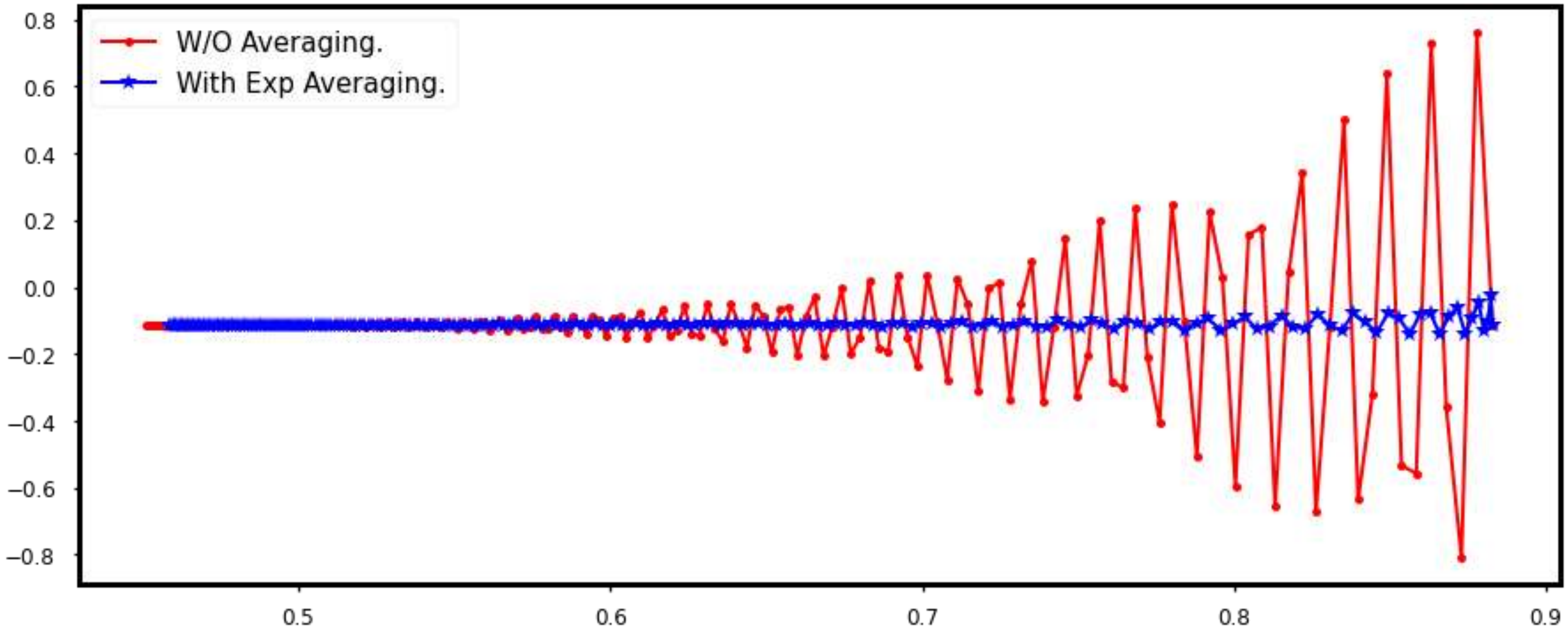

$$\mathrm{w} \rightarrow \mathrm{w} - \eta \left(\frac{\partial \mathcal{L}}{\partial w}\right)_{avg}$$

$$\left(\frac{\partial \mathcal{L}}{\partial w}\right)_{avg}^{t+1} = \beta \left(\frac{\partial \mathcal{L}}{\partial w}\right)_{avg}^{t} + (1-\beta)\left(\frac{\partial \mathcal{L}}{\partial w}\right)^{t+1}$$

# How could make it increase the lr?

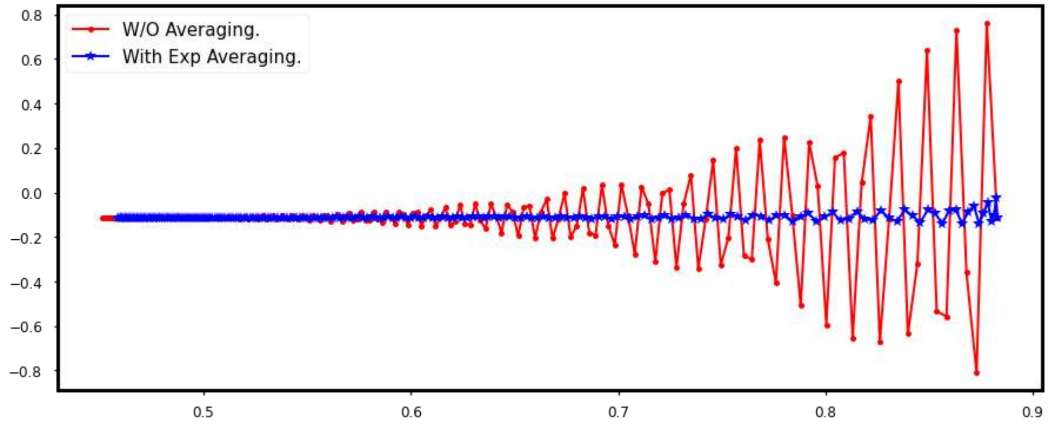


# Variance



W/O Averaging.
With Exp Averaging.
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
0.5
0.6
0.7
0.8
0.9

# Variance




Update rule:

$$w \rightarrow w - \eta \frac{\partial \mathcal{L}}{\partial w}$$

$$w \rightarrow w - \frac{\boldsymbol{\eta}}{\sqrt{\boldsymbol{ms}}} \left( \frac{\boldsymbol{\partial \mathcal{L}}}{\boldsymbol{\partial w}} \right)$$

ms: **Mean of squared of the gradients.**

# Average of variance

$$(\mathrm{ms})_{avg}^{t+1} = \boldsymbol{\beta}(\mathrm{ms})_{avg}^{t} + (\mathbf{1}-\boldsymbol{\beta})\left(\left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}}\right)^{2}\right)^{t+1}$$

# RMS prop

$$\mathrm{w} \rightarrow \mathrm{w} - \frac{\eta}{\sqrt{\mathrm{ms} + \boldsymbol{\epsilon}}} \left( \frac{\boldsymbol{\partial \mathcal{L}}}{\boldsymbol{\partial w}} \right)$$

$$(\mathrm{ms})_{\boldsymbol{avg}}^{\boldsymbol{t+1}} = \boldsymbol{\beta}'(\mathrm{ms})_{\boldsymbol{avg}}^{\boldsymbol{t}} + (\mathbf{1} - \boldsymbol{\beta}') \left( \left( \frac{\boldsymbol{\partial \mathcal{L}}}{\boldsymbol{\partial w}} \right)^{\mathbf{2}} \right)^{\boldsymbol{t+1}}$$

# Now let’s put both of them together …

# Adam

$$\mathrm{w} \to \mathrm{w} - \frac{\eta}{\sqrt{\mathrm{ms} + \boldsymbol{\epsilon}}} \left(\frac{\boldsymbol{\partial \mathcal{L}}}{\boldsymbol{\partial w}}\right)_{\boldsymbol{avg}}$$

$$\left(\frac{\boldsymbol{\partial \mathcal{L}}}{\boldsymbol{\partial w}}\right)_{\boldsymbol{avg}} = \boldsymbol{\beta_1} \left(\frac{\boldsymbol{\partial \mathcal{L}}}{\boldsymbol{\partial w}}\right)_{\boldsymbol{avg}}^{\boldsymbol{t}} + (\boldsymbol{1} - \boldsymbol{\beta_1}) \left(\frac{\boldsymbol{\partial \mathcal{L}}}{\boldsymbol{\partial w}}\right)^{\boldsymbol{t+1}}$$

$$\mathrm{ms} = \boldsymbol{\beta_2} (\mathrm{ms})_{\boldsymbol{avg}}^{\boldsymbol{t}} + (\boldsymbol{1} - \boldsymbol{\beta_2}) \left(\left(\frac{\boldsymbol{\partial \mathcal{L}}}{\boldsymbol{\partial w}}\right)^{\boldsymbol{2}}\right)^{\boldsymbol{t+1}}$$

# Hyper parameters

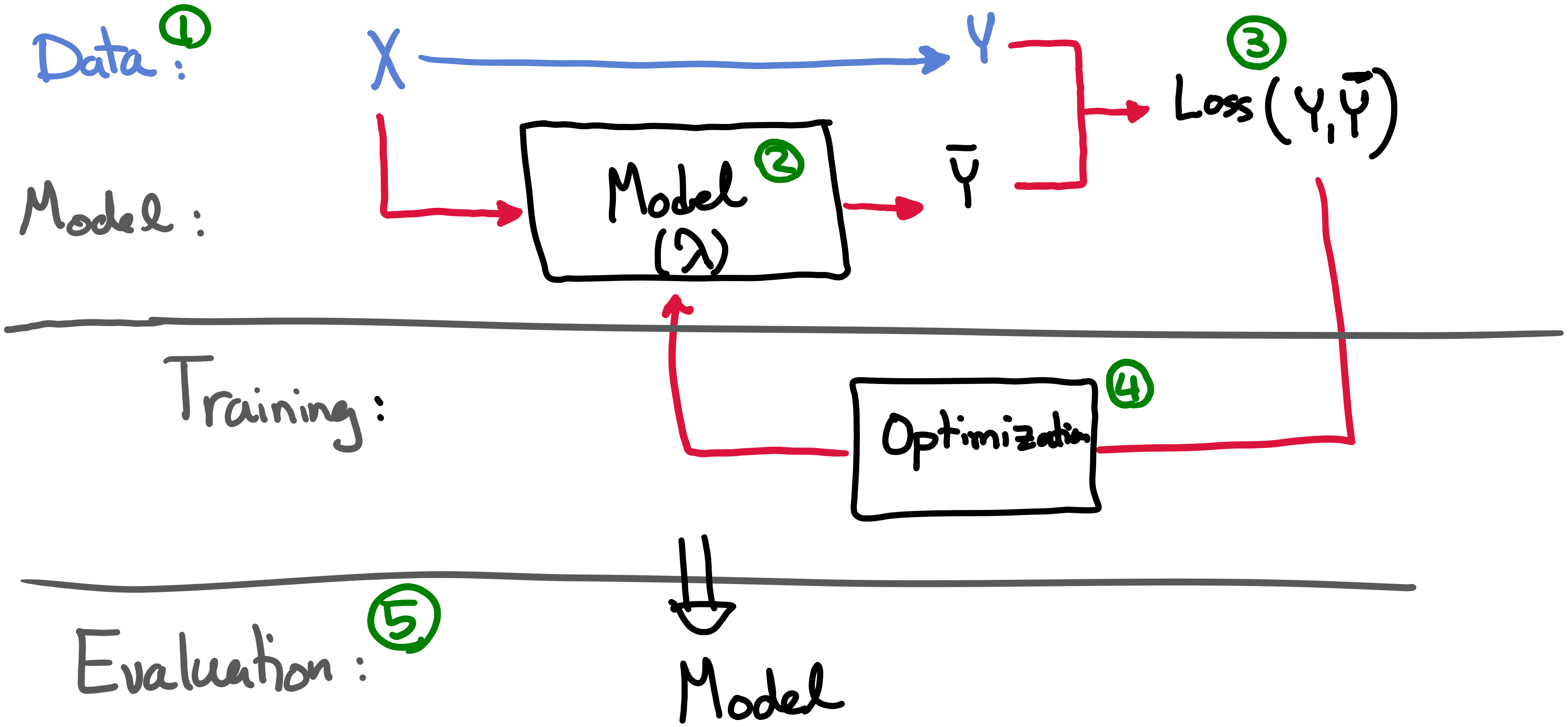
0.99

$$\eta, \beta_1, \beta_2, \epsilon$$

0.9

$10^{-8}$

# So far ...


Data: ①
X
Y
Model:
Model ② (λ)
Ȳ
③ Loss(Y, Ȳ)
Training:
Optimization ④
Evaluation: ⑤
Model