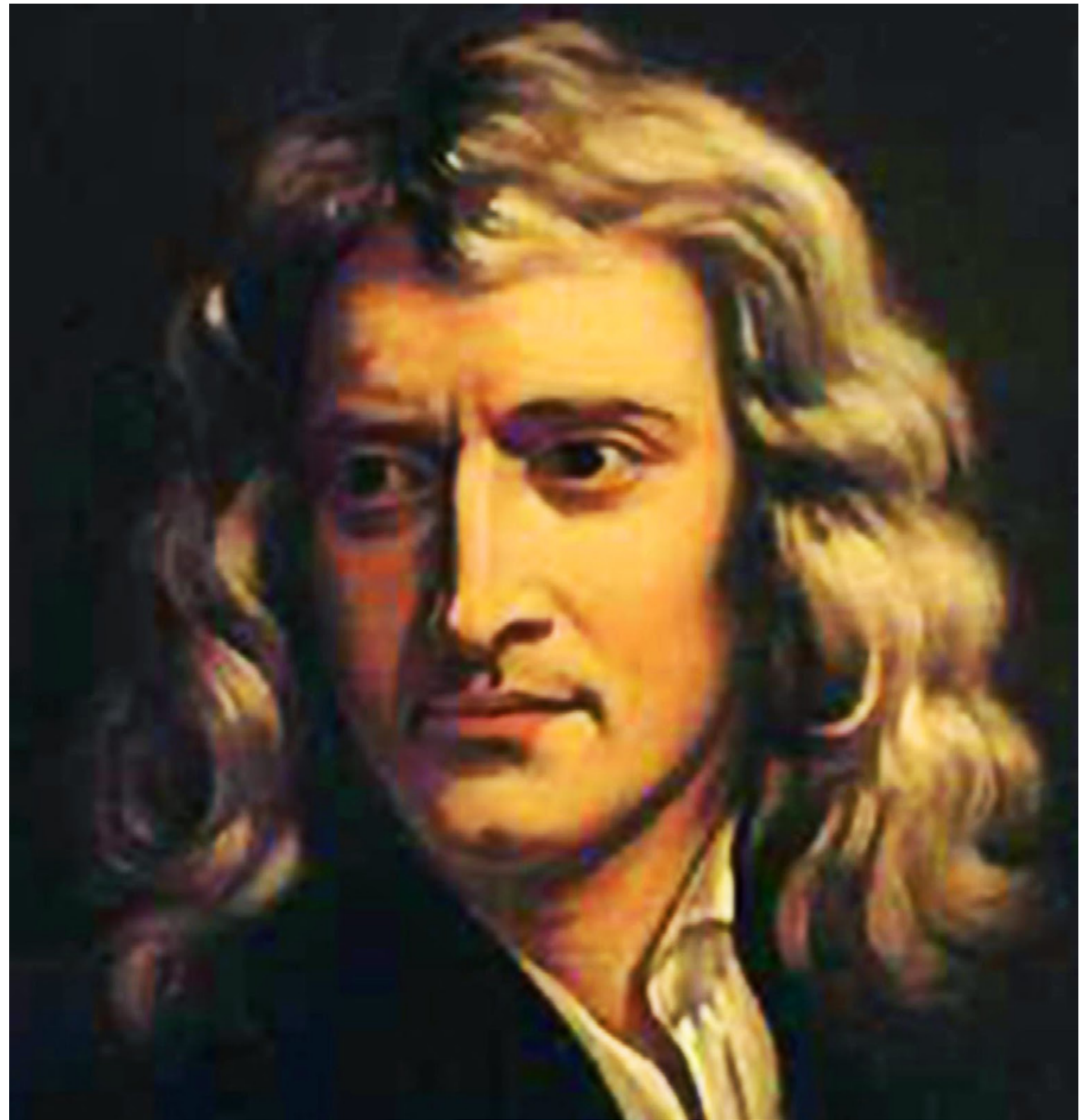


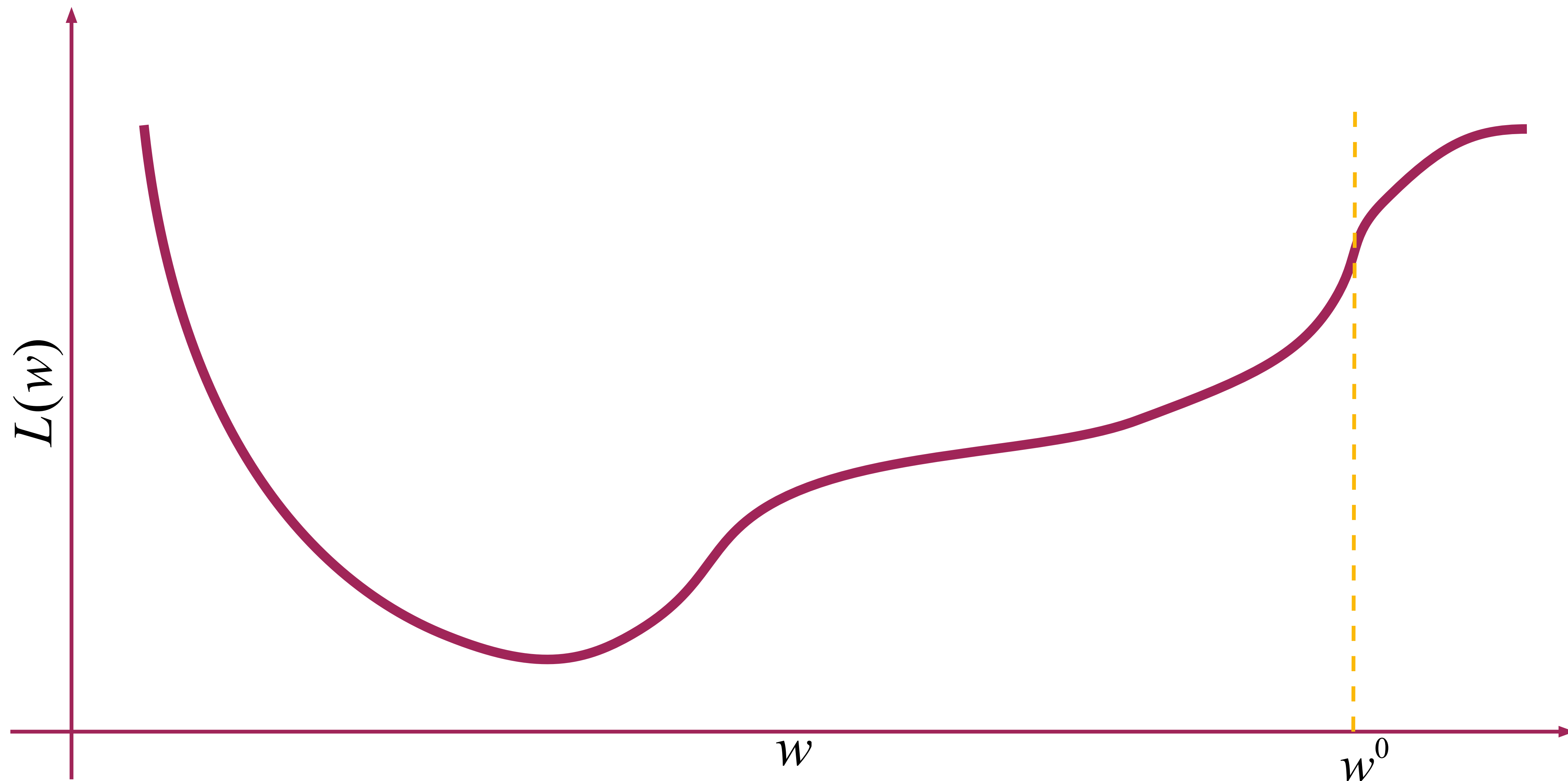
# Second order optimization

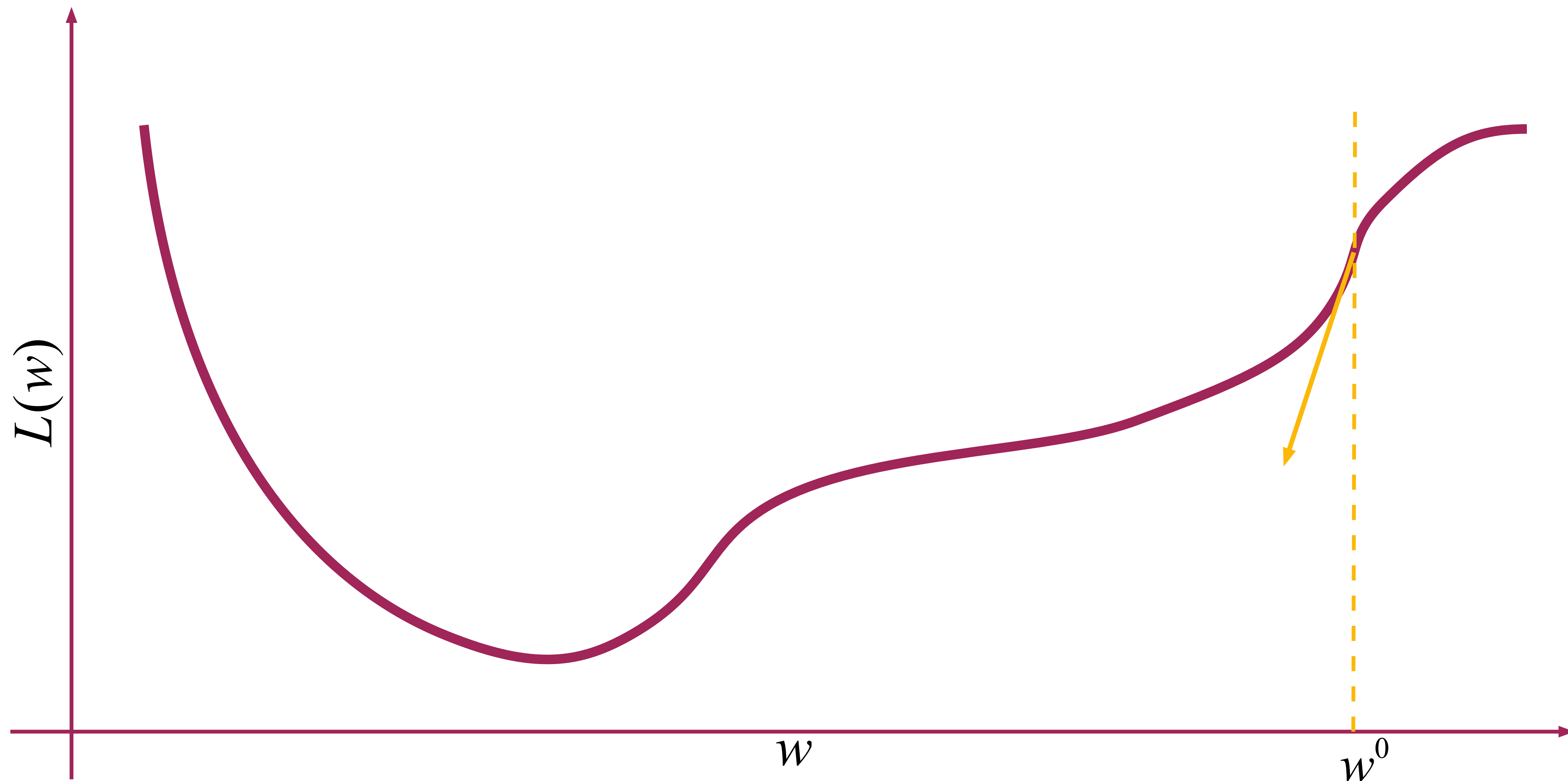


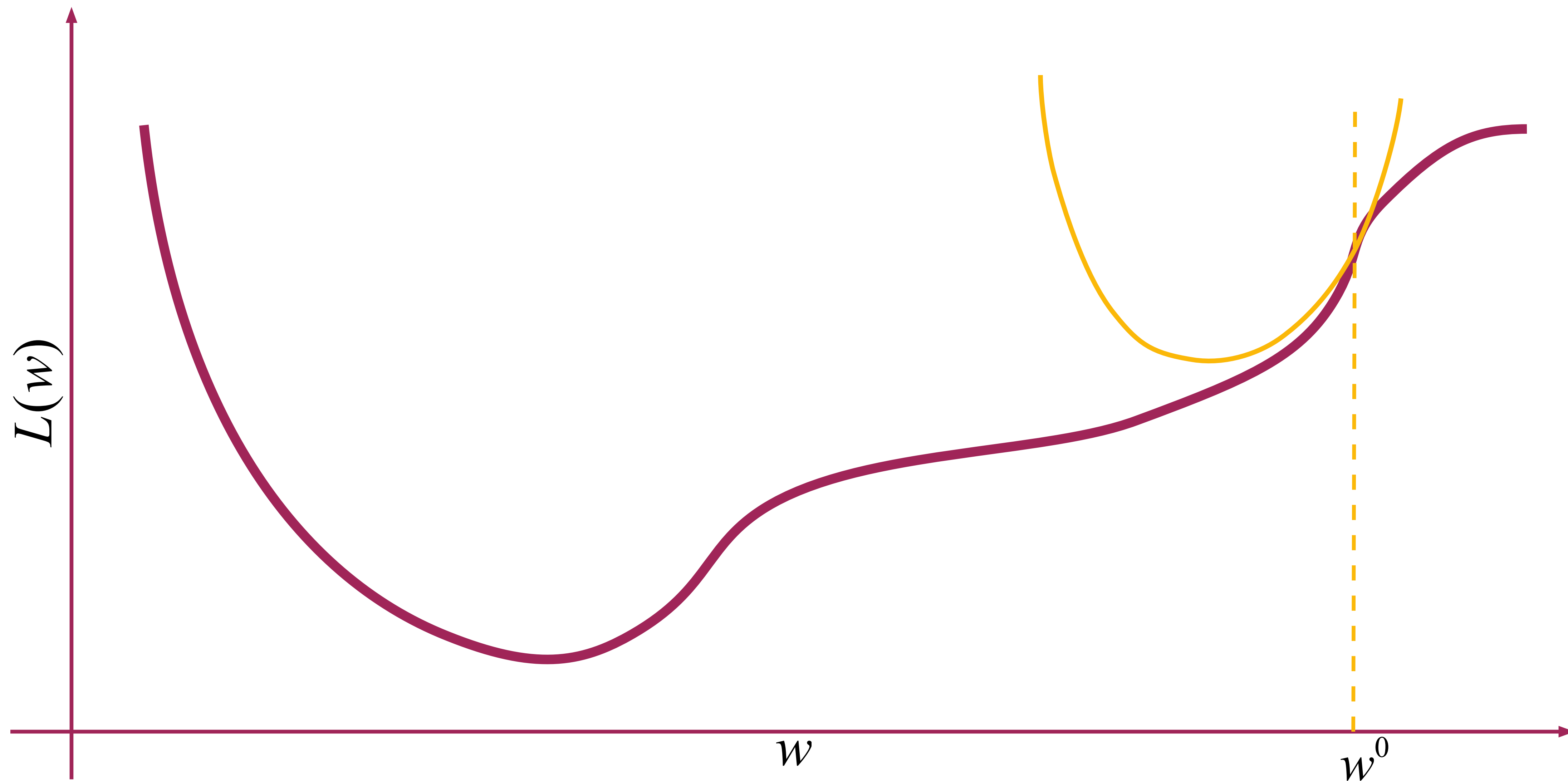
# Today you will learn about

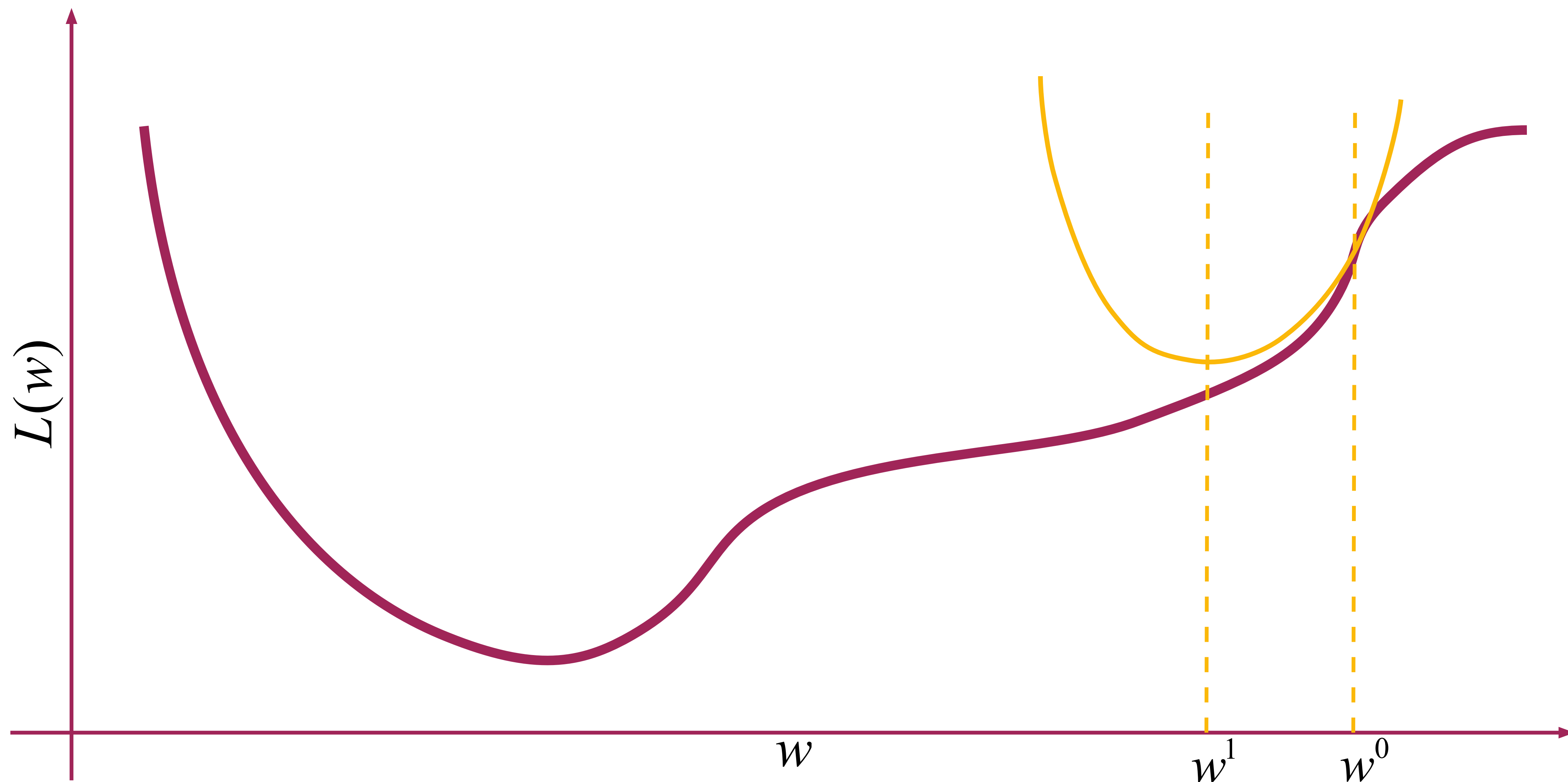
- The newton method
- The BFGS method
- The L-BFGS method
- The training of linear regression

# Newton method

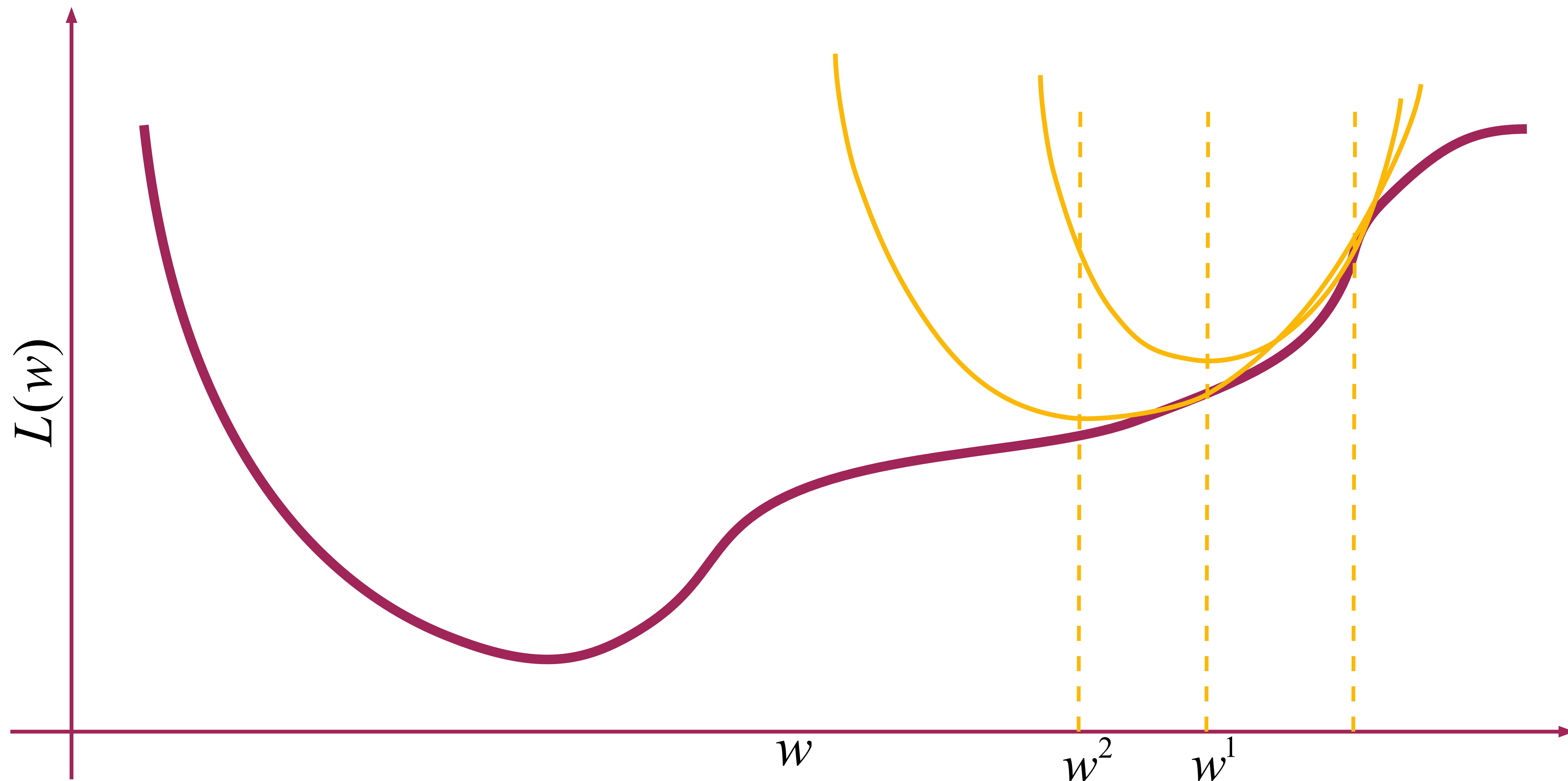


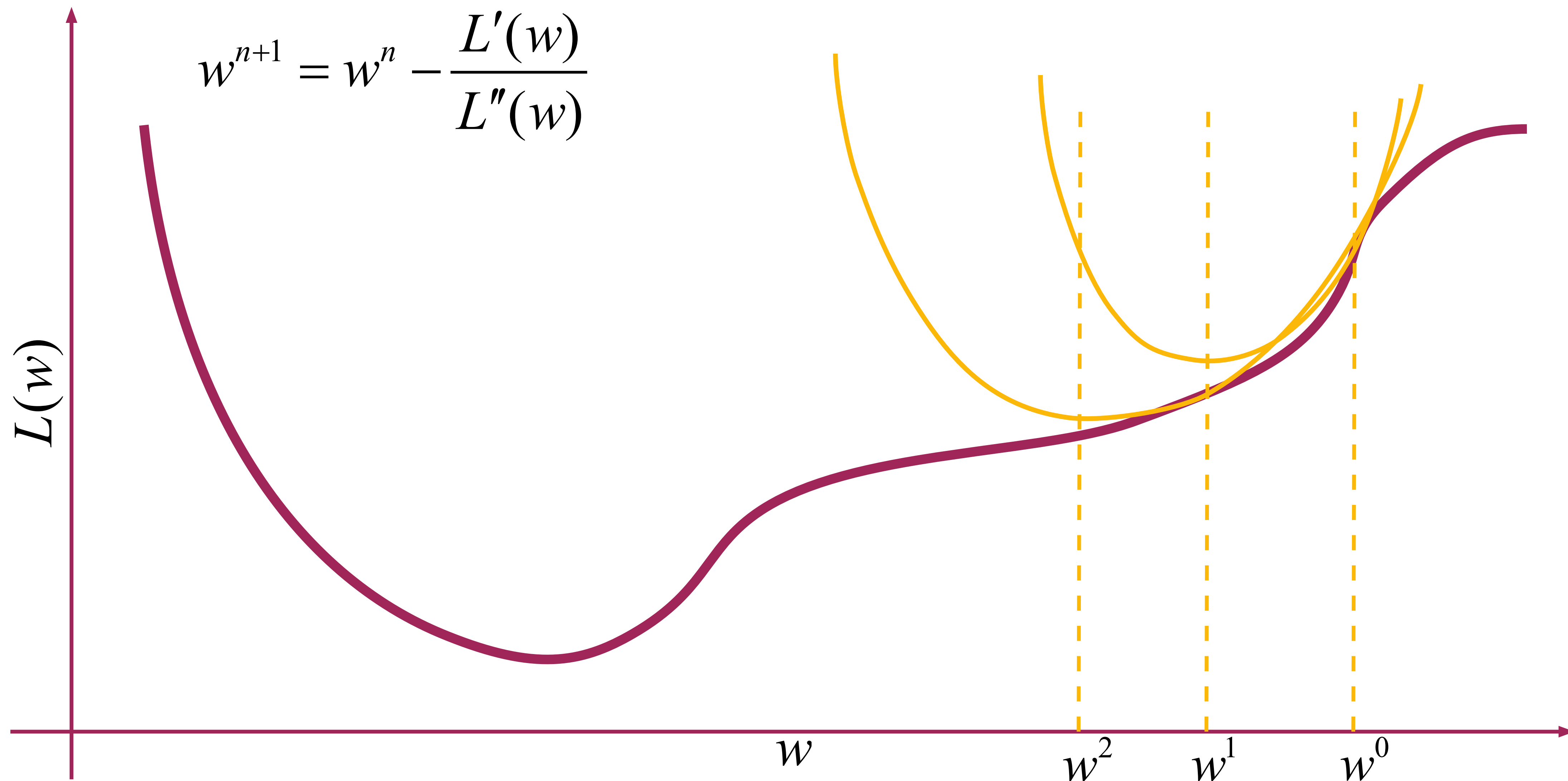


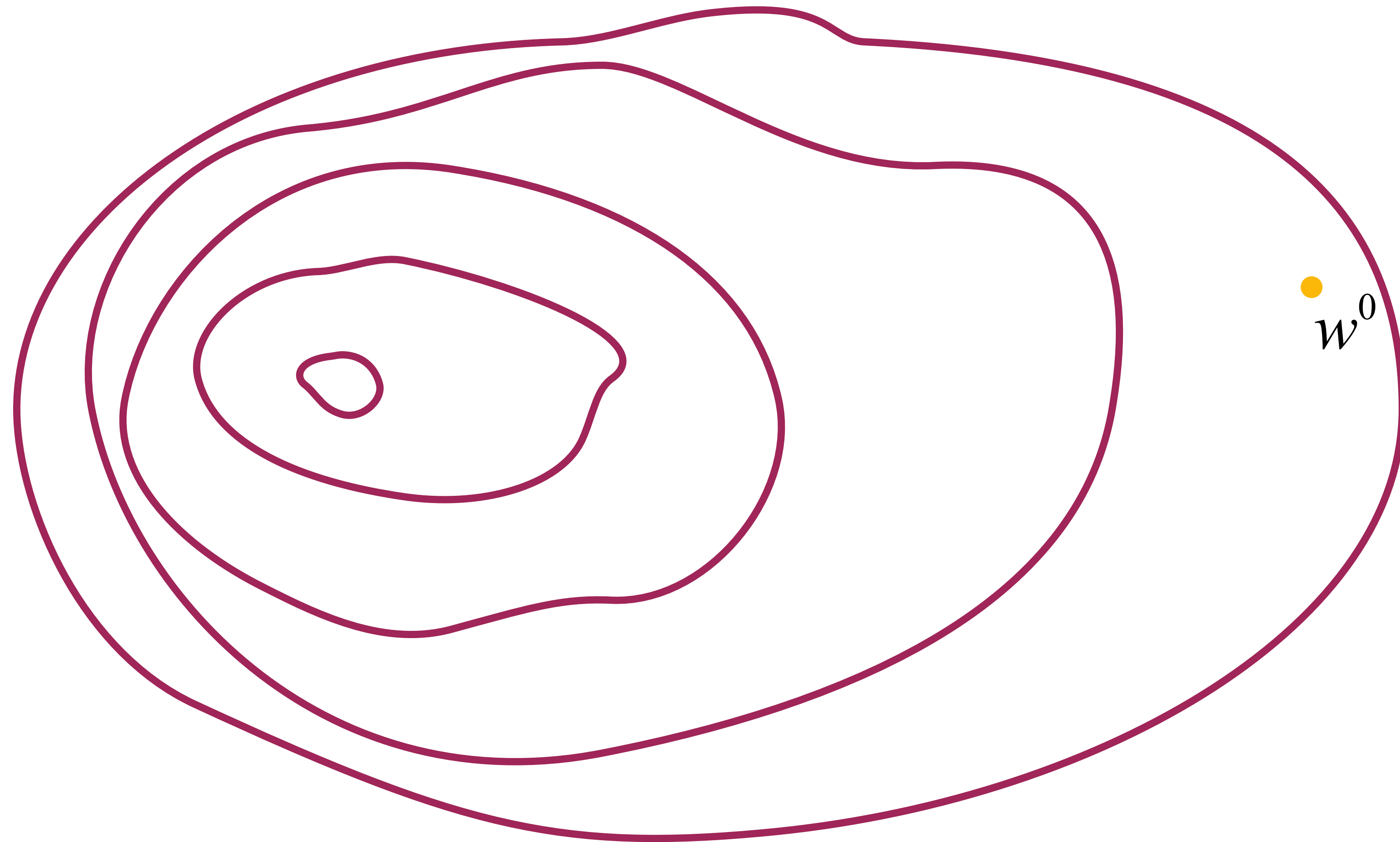


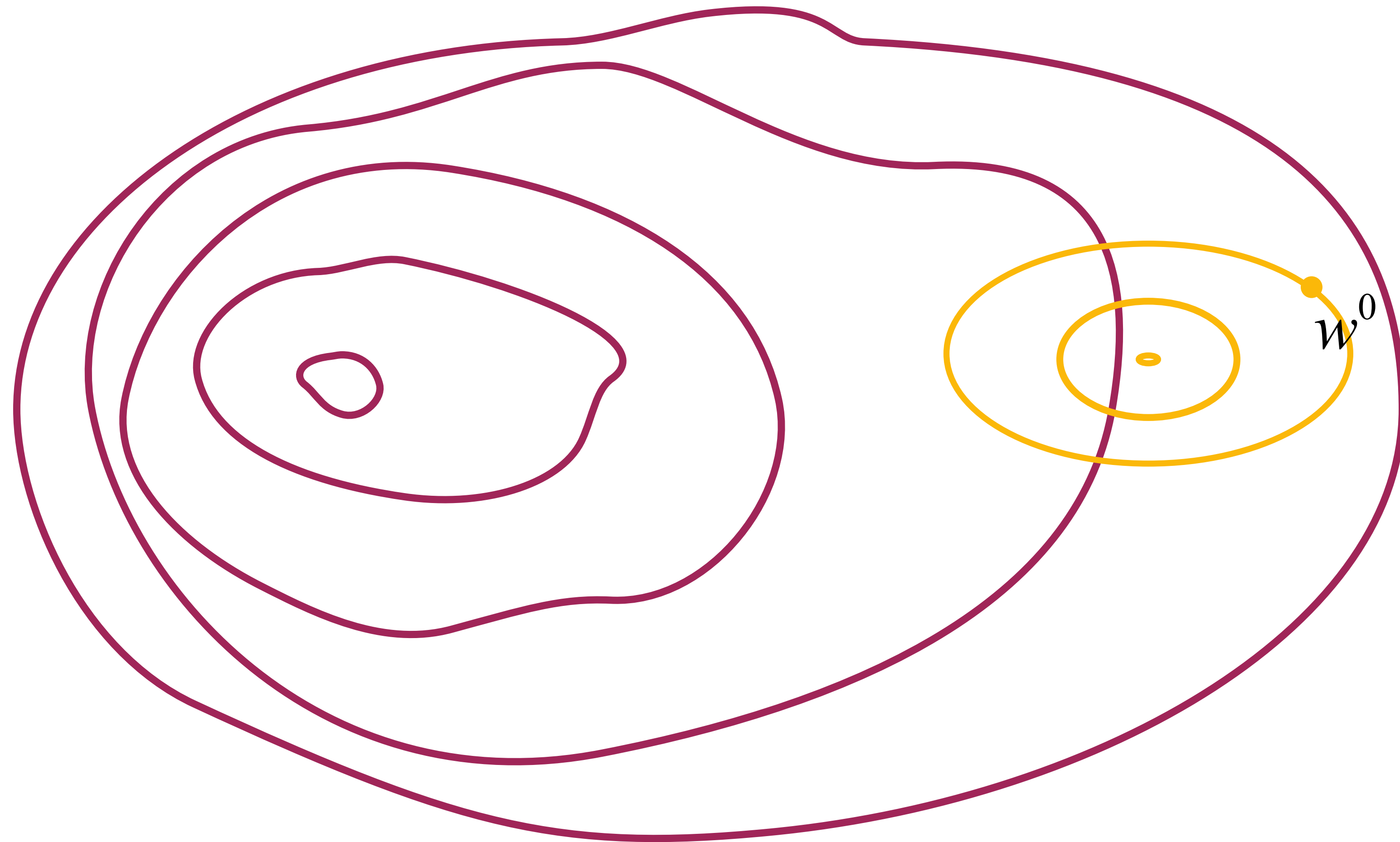


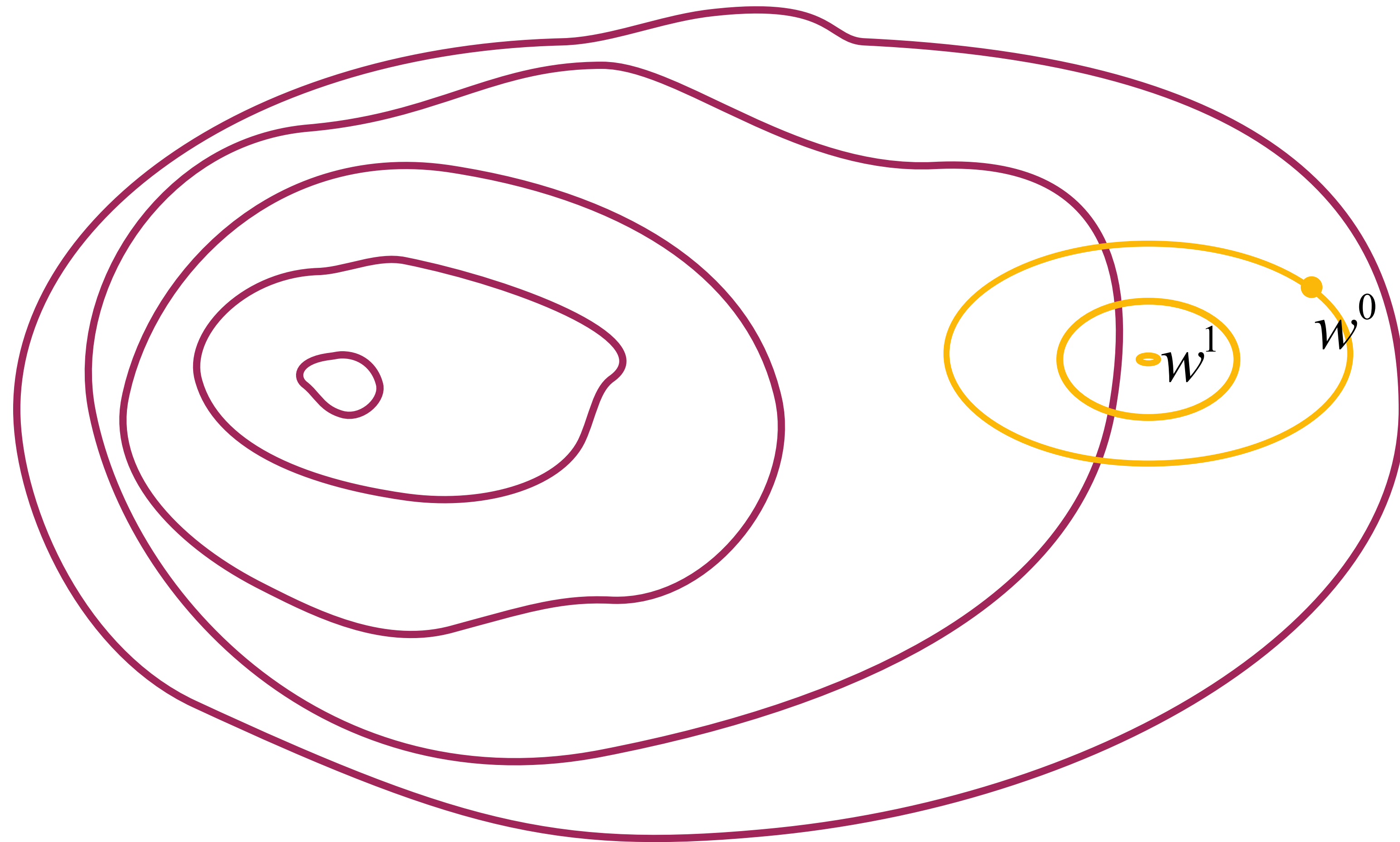


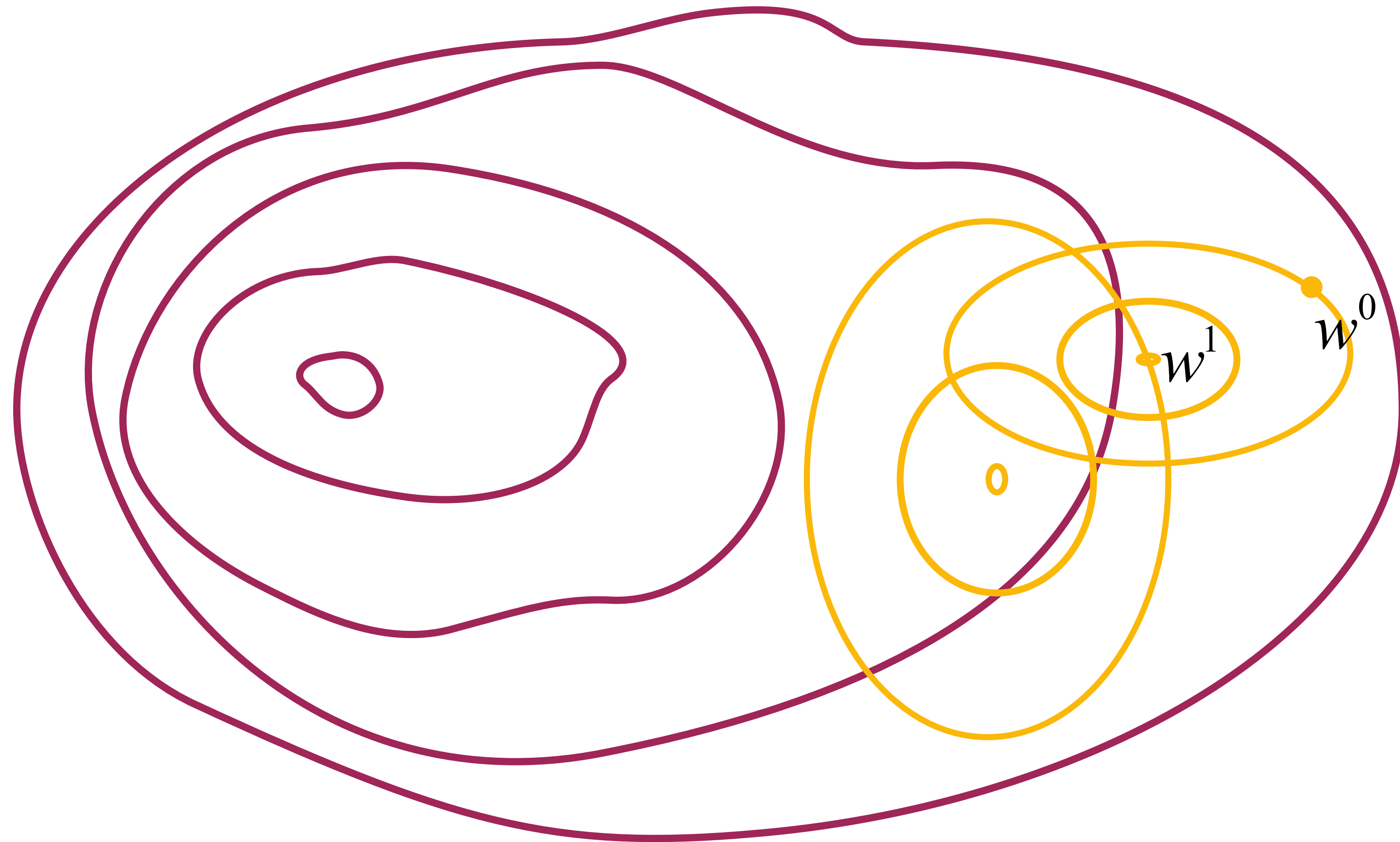


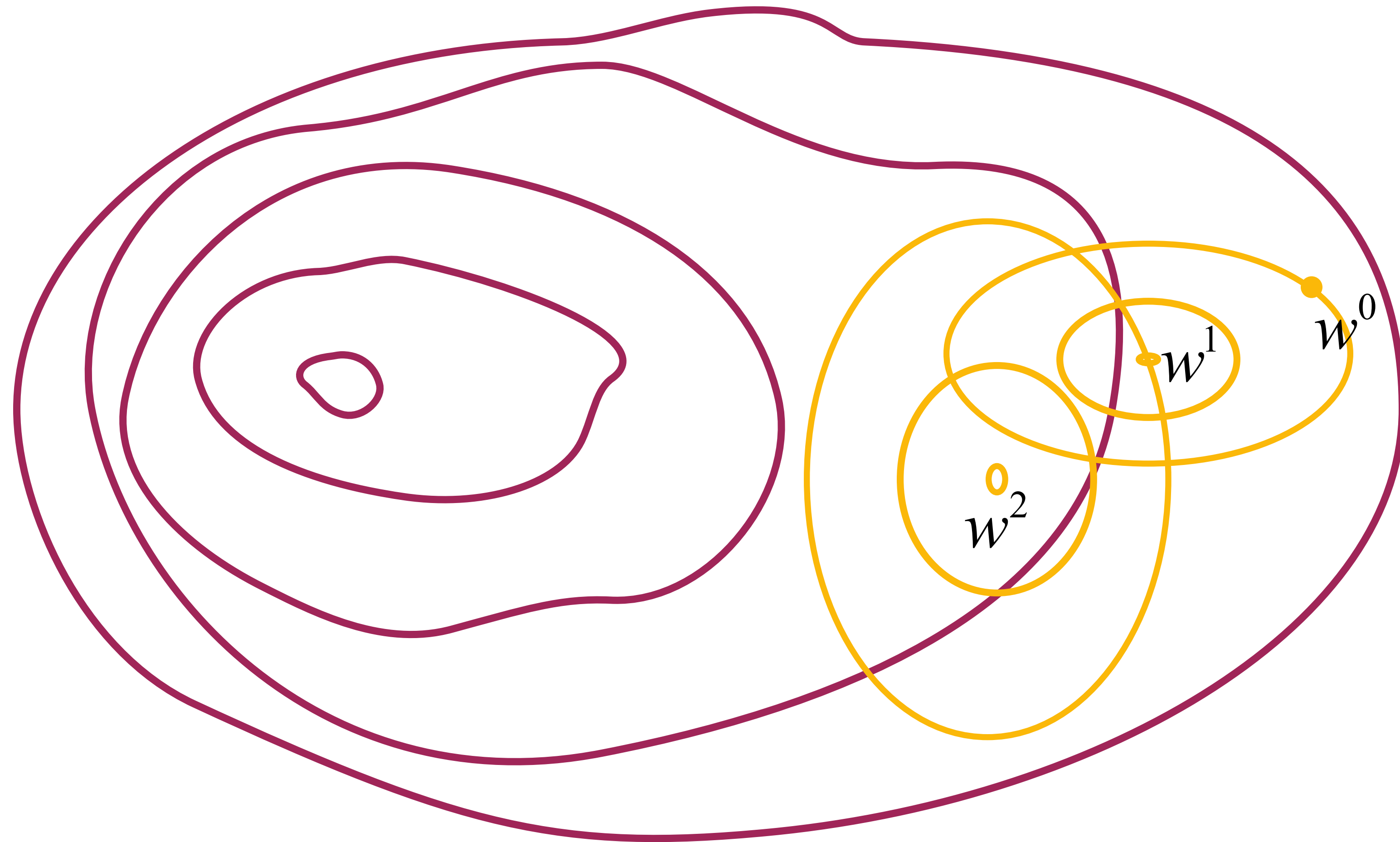
















$$L'(w) \quad \longrightarrow \quad \nabla L(w) = \begin{pmatrix} \frac{\partial L(w)}{\partial w_0} \\ \dots \\ \frac{\partial L(w)}{\partial w_n} \end{pmatrix}$$

$$L'(w) \quad \longrightarrow \quad \nabla L(w) = \begin{pmatrix} \frac{\partial L(w)}{\partial w_0} \\ \dots \\ \frac{\partial L(w)}{\partial w_n} \end{pmatrix}$$

$$L''(w) \quad \longrightarrow \quad H(L(w)) = \begin{pmatrix} \frac{\partial^2 L(w)}{\partial w_0^2} & \dots & \frac{\partial^2 L(w)}{\partial w_n \partial w_0} \\ \dots & \dots & \dots \\ \frac{\partial^2 L(w)}{\partial w_0 \partial w_n} & \dots & \frac{\partial^2 L(w)}{\partial w_n^2} \end{pmatrix}$$

$$L'(w) \quad \longrightarrow \quad \nabla L(w) = \begin{pmatrix} \frac{\partial L(w)}{\partial w_0} \\ \dots \\ \frac{\partial L(w)}{\partial w_n} \end{pmatrix}$$

$$L''(w) \quad \longrightarrow \quad H(L(w)) = \begin{pmatrix} \frac{\partial^2 L(w)}{\partial w_0^2} & \dots & \frac{\partial^2 L(w)}{\partial w_n \partial w_0} \\ \dots & \dots & \dots \\ \frac{\partial^2 L(w)}{\partial w_0 \partial w_n} & \dots & \frac{\partial^2 L(w)}{\partial w_n^2} \end{pmatrix}$$

$$w^{n+1} = w^n - \frac{L'(w)}{L''(w)} \quad \longrightarrow \quad w^{n+1} = w^n - H^{-1}(L(w)) \cdot \nabla L(w)$$

$$w^{n+1} = w^n - H^{-1}(L(w)) \cdot \nabla L(w)$$

## Problem

$$H^{-1}(L(w))$$

$n^3$  operations

10 features

1 000 operations

100 features

1 000 000 operations

1 000 features

1 000 000 000 operations

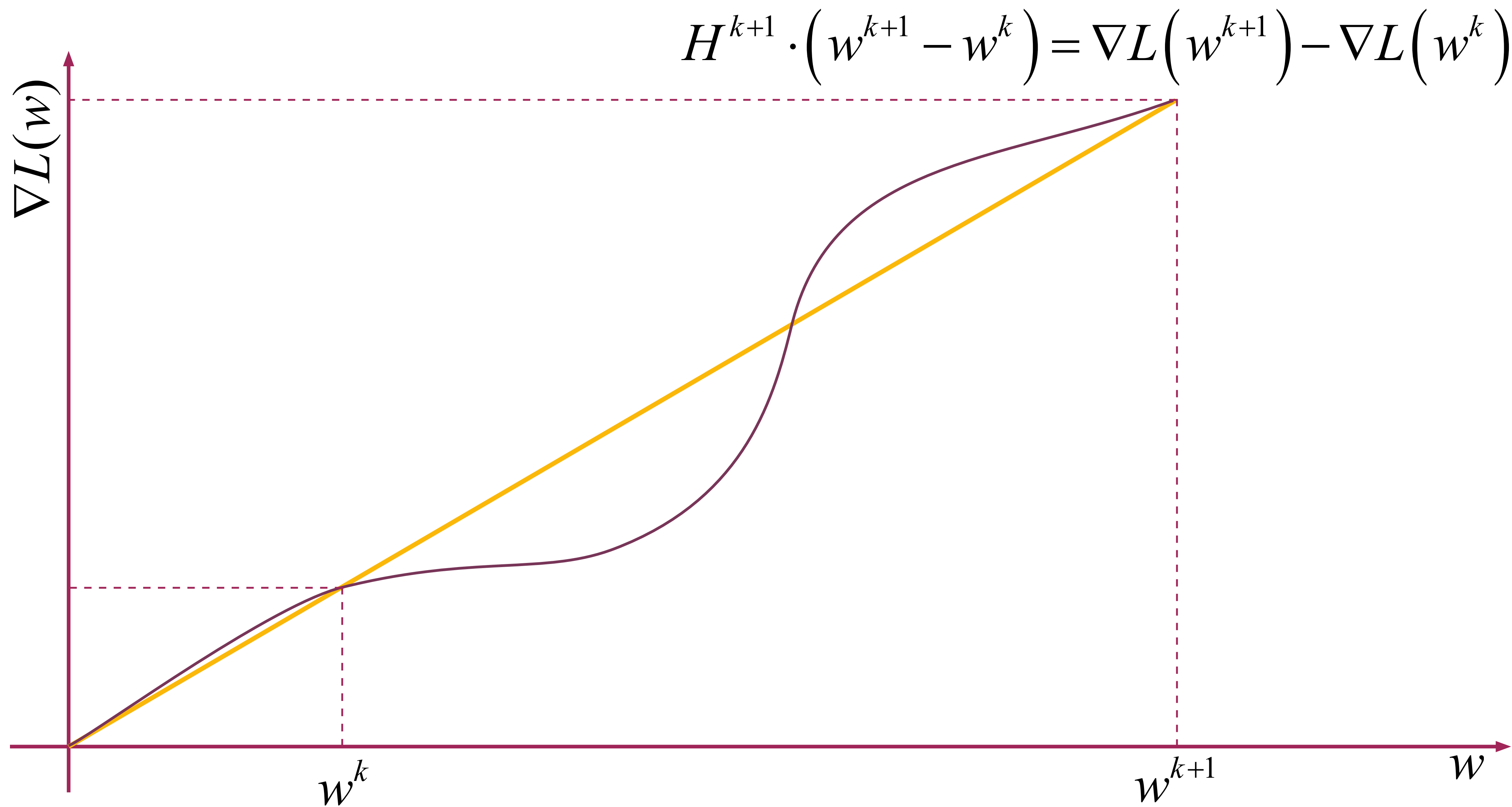
# BFGS

Broyden–Fletcher–Goldfarb–Shanno algorithm



# Approximate value

$$H^{-1}\left(L\left(w^{n+1}\right)\right)=H^{-1}\left(L\left(w^n\right)\right)+\delta \quad = \quad U \quad \begin{matrix} V^T \\ \hline \hline \end{matrix}$$



$$H^{-1}(L(w))$$

$n^2$  operations

10 features

100 operations

100 features

10 000 operations

1 000 features

1 000 000 operations



# L-BFGS

Limited memory

Broyden–Fletcher–Goldfarb–Shanno  
algorithm

$$H^{-1}\left(L\left(w^{k+1}\right)\right)=U^k \cdot V^{k^T}+H^{-1}\left(L\left(w^k\right)\right)$$



$$H^{-1}\left(L\left(w^{k+1}\right)\right)=U^k \cdot V^{k^T}+U^{k-1} \cdot V^{(k-1)^T}+H^{-1}\left(L\left(w^{k-1}\right)\right)$$



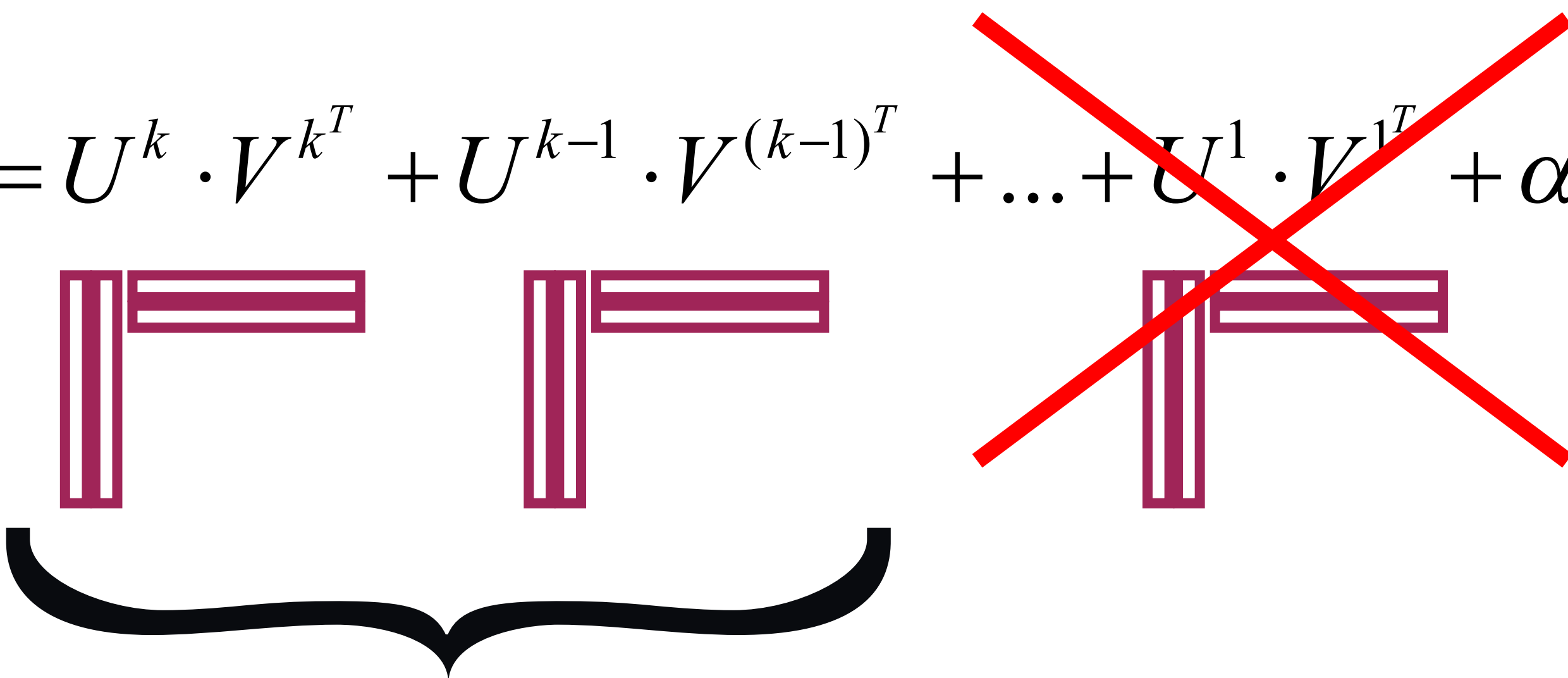
$$H^{-1}\left(L\left(w^{k+1}\right)\right)=\underbrace{U^k \cdot V^{k^T}}_{\text{}}+\underbrace{U^{k-1} \cdot V^{(k-1)^T}}_{\text{}}+\ldots+\underbrace{U^1 \cdot V^{1^T}}_{\text{}}+\underbrace{H^{-1}\left(L\left(w^0\right)\right)}_{\text{}}$$

$$\equiv \alpha \cdot I$$

$$\begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}$$

$$H^{-1}\left(L\left(w^{k+1}\right)\right)=U^k \cdot V^{k^T}+U^{k-1} \cdot V^{(k-1)^T}+\ldots+U^1 \cdot V^{1^T}+\alpha \cdot I$$



$$H^{-1}\left(L\left(w^{k+1}\right)\right)=\underbrace{U^k \cdot V^{k^T}+U^{k-1} \cdot V^{(k-1)^T}}_{\text{last } N=10}+\ldots+\cancel{U^1 \cdot V^{1^T}}+\alpha \cdot I$$


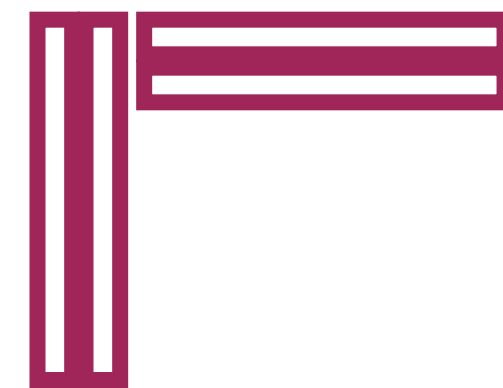
The diagram illustrates the summation of terms in the equation. The first two terms,  $U^k \cdot V^{k^T}$  and  $U^{k-1} \cdot V^{(k-1)^T}$ , are grouped by a bracket labeled "last N = 10". The third term,  $U^1 \cdot V^{1^T}$ , is crossed out with a large red X, indicating it is not part of the summation.

$$H^{-1}\left(L\left(w^{k+1}\right)\right)=U^k \cdot V^{k^T}+U^{k-1} \cdot V^{(k-1)^T}+\ldots+U^{(k-9)} \cdot V^{(k-9)^T}+\alpha \cdot I$$



$$w^{n+1} = w^n - H^{-1} \left( L(w) \right) \cdot \nabla L(w)$$

$$w^{n+1} = w^n - \left( U^k \cdot V^{k^T} + U^{k-1} \cdot V^{(k-1)^T} + \dots + U^{(k-9)} \cdot V^{(k-9)^T} + \alpha \cdot I \right) \cdot \nabla L(w)$$





$$w^{n+1} = w^n - \underbrace{U^k \cdot V^{k^T}}_{\text{matrix}} \cdot \underbrace{\nabla L(w)}_{\text{vector}} - \underbrace{U^{k-1} \cdot V^{(k-1)^T}}_{\text{matrix}} \cdot \underbrace{\nabla L(w)}_{\text{vector}} - \dots - U^{(k-9)} \cdot V^{(k-9)^T} \cdot \nabla L(w) - \alpha \cdot \nabla L(w)$$

n

# Linear regression

- Linear regression is a supervised learning algorithm that models the relationship between a continuous target variable and one or more feature variables.
- It assumes a linear relationship between the features and the target variable.
- The goal is to find the best-fitting line (or plane) that minimizes the error between the predicted and actual values.
- Linear regression is widely used in various applications, such as predicting house prices, stock prices, and sales volume.
- The model is trained using a dataset of input features and corresponding target values.
- The training process involves minimizing a cost function, typically the mean squared error (MSE).
- Once trained, the model can be used to predict the target variable for new input features.
- Linear regression is a simple and interpretable model, making it a popular choice for many applications.
- However, it may not perform well for non-linear relationships or data with high variance.
- In such cases, more complex models like decision trees or neural networks might be more appropriate.

# Linear regression



number of  
features  $< N$

# Linear regression

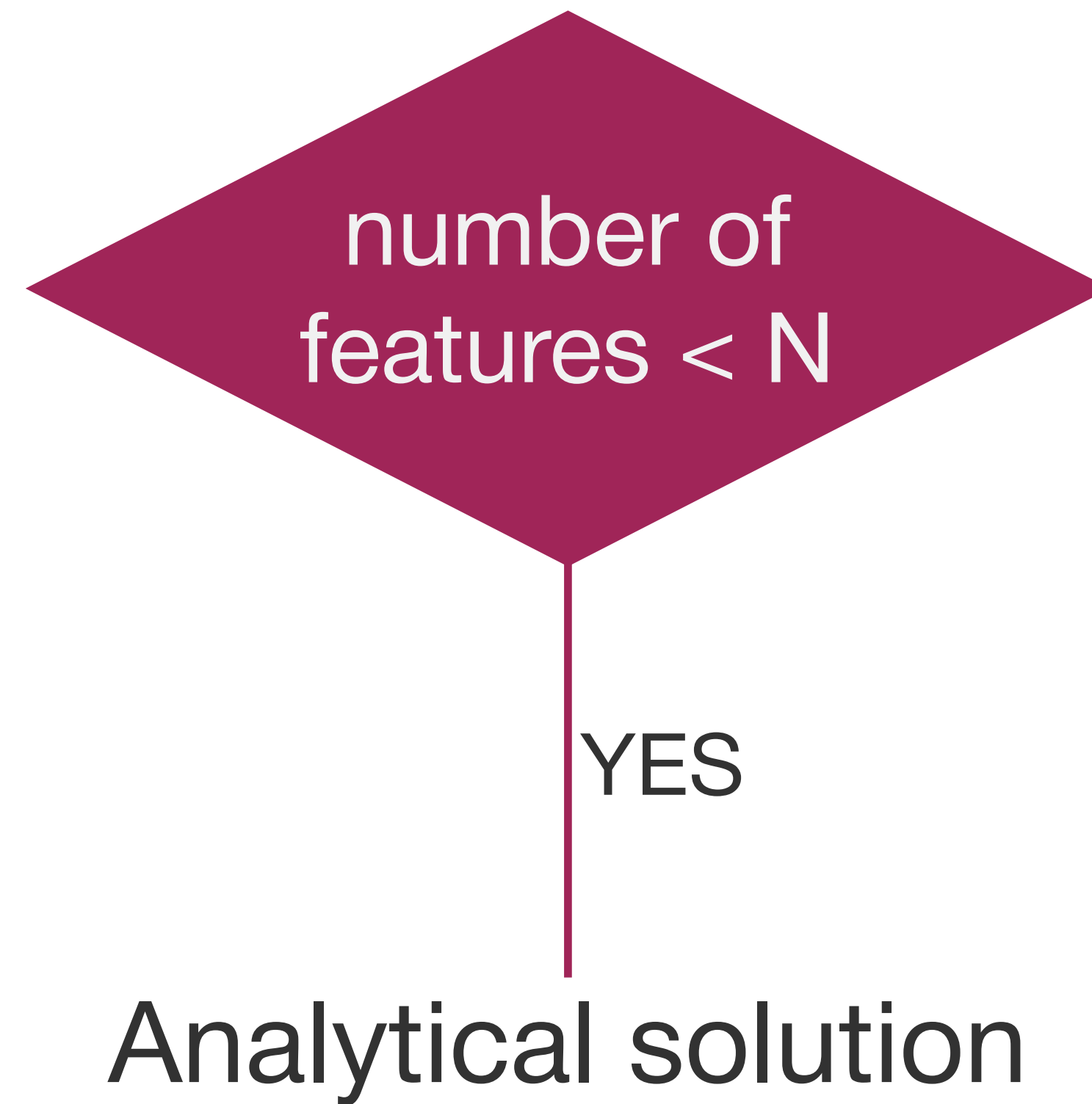
$N = 4096$



number of  
features  $< N$

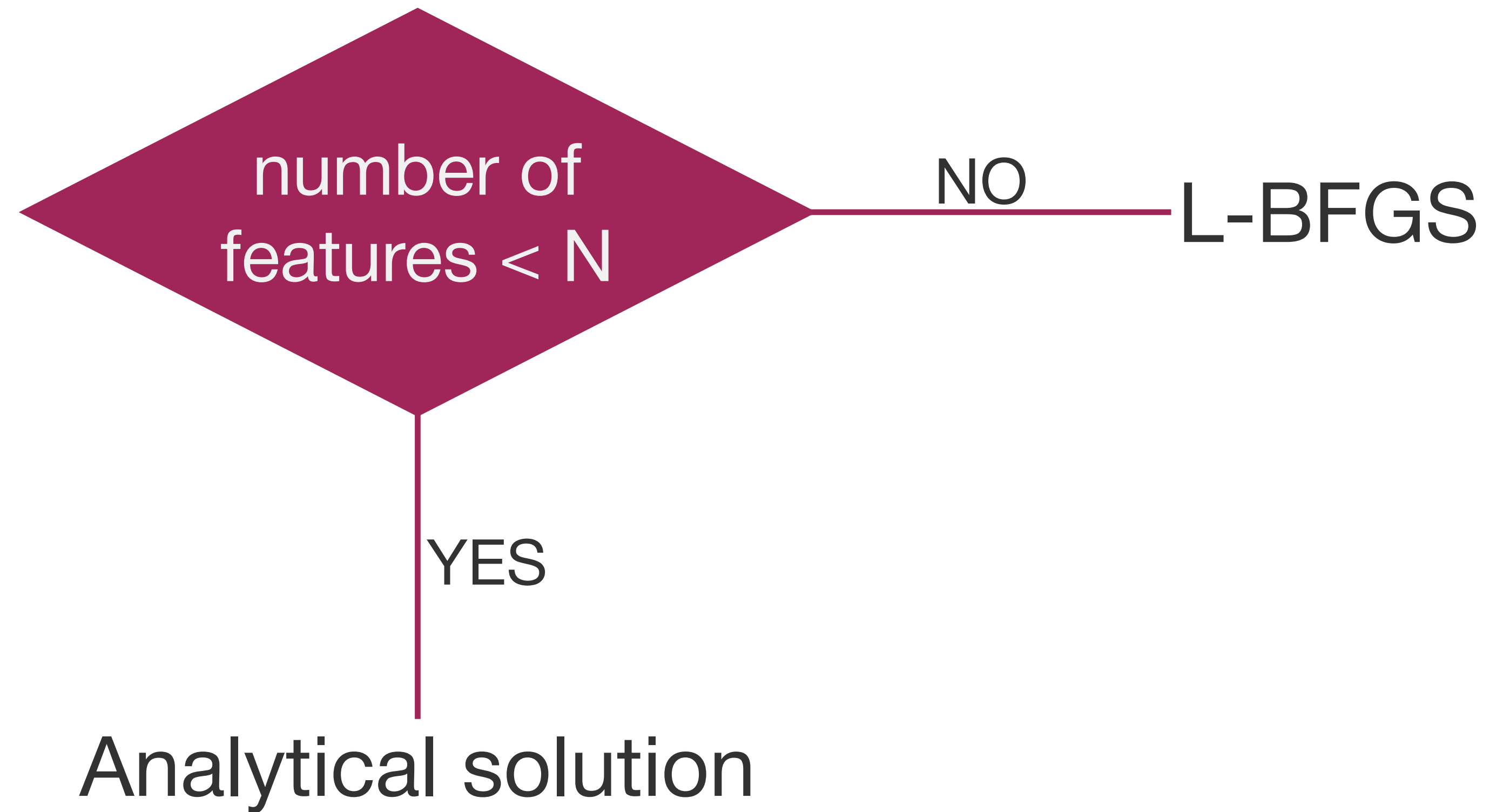
# Linear regression

$N = 4096$



# Linear regression

$N = 4096$



# Today you have learned about:

- The newton method
- The BFGS method
- The L-BFGS method
- The training of linear regression