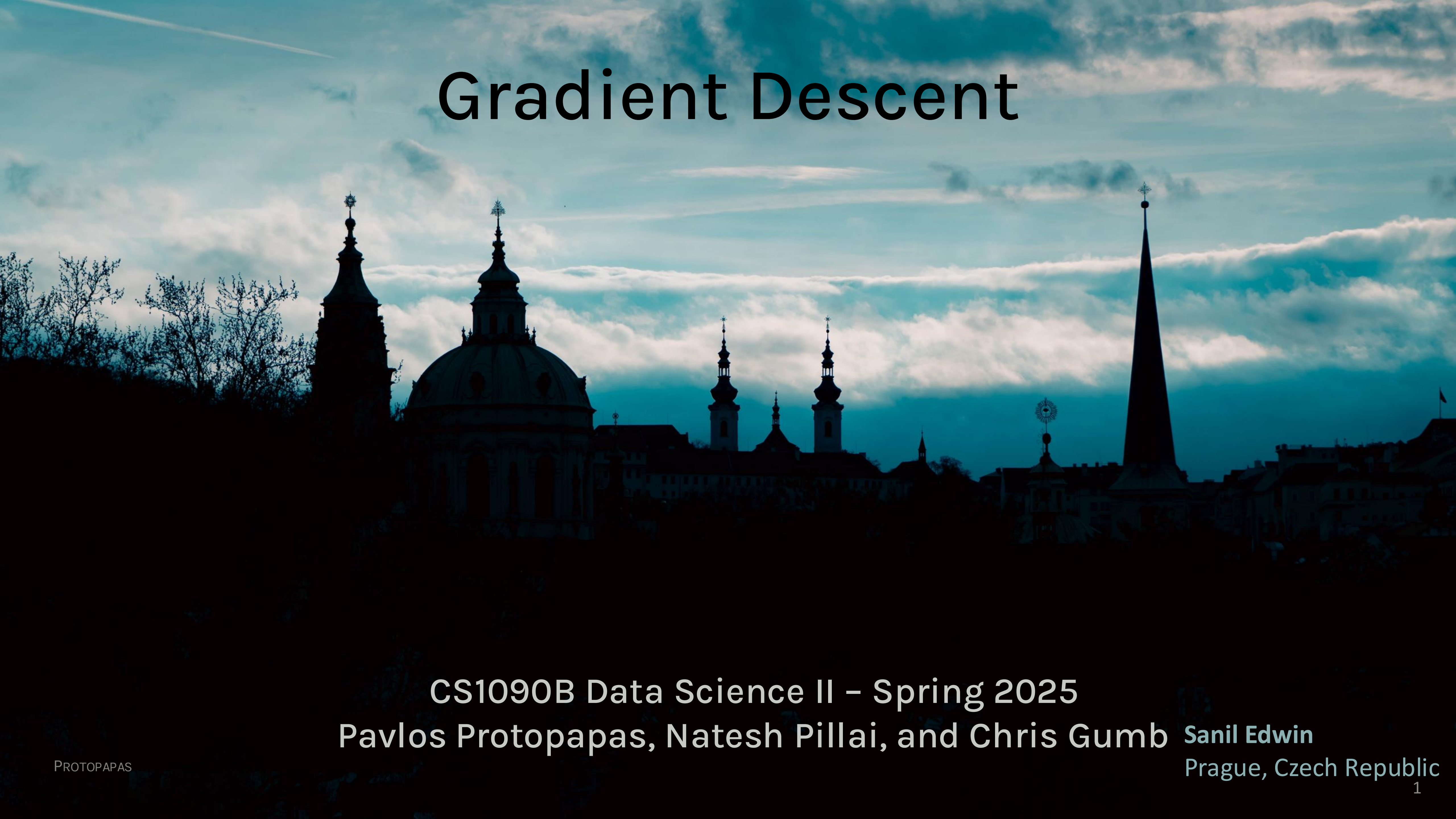
Gradient Descent



Sanil Edwin
Prague, Czech Republic

CS1090B Data Science II – Spring 2025

Pavlos Protopapas, Natesh Pillai, and Chris Gumb

Outline

- Gradient Descent
- Stochastic Gradient Descent

Outline

- **Gradient Descent**
- Stochastic Gradient Descent

Gradient Descent - Introduction

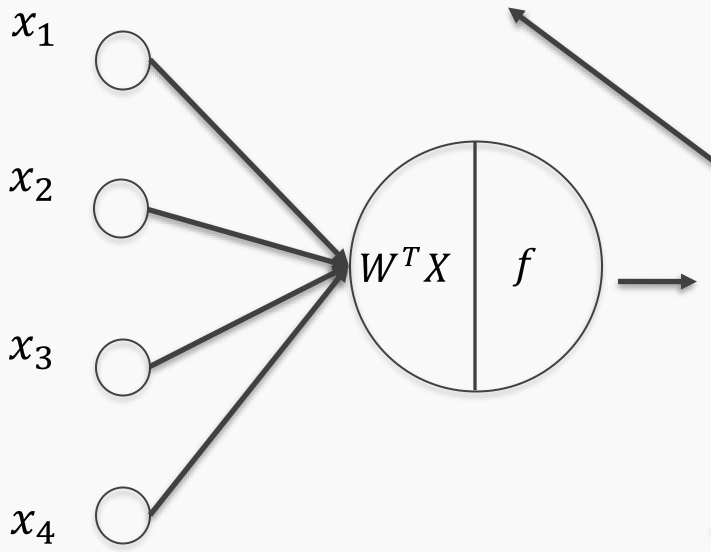
Start with single neuron

Binary Classification:
Activation is sigmoid

$$f(X) = \frac{1}{1+e^{-W^T X}}$$

Regression:
Activation is linear

$$f(X) = W^T X$$



$$Y = f(W_0 + W_1 x_1 + W_2 x_2 + W_3 x_3 + W_4 x_4)$$

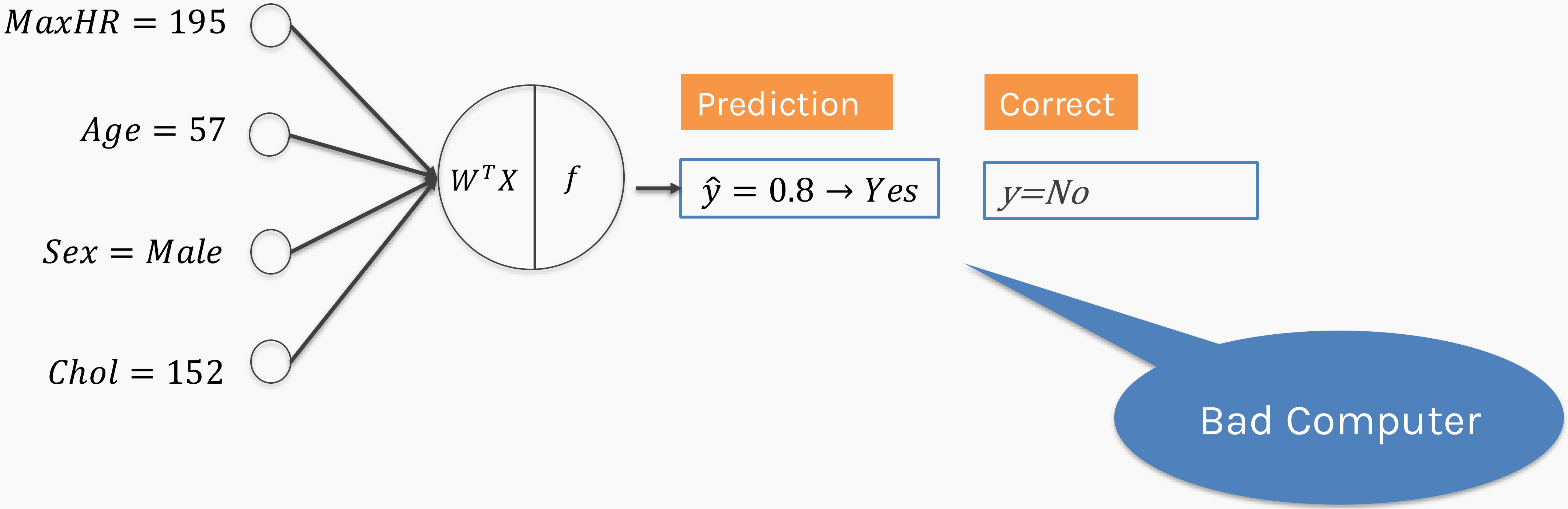
Intercept or **Bias**

Coefficients or **Weights**

Gradient Descent - Introduction

Start with all randomly selected weights.

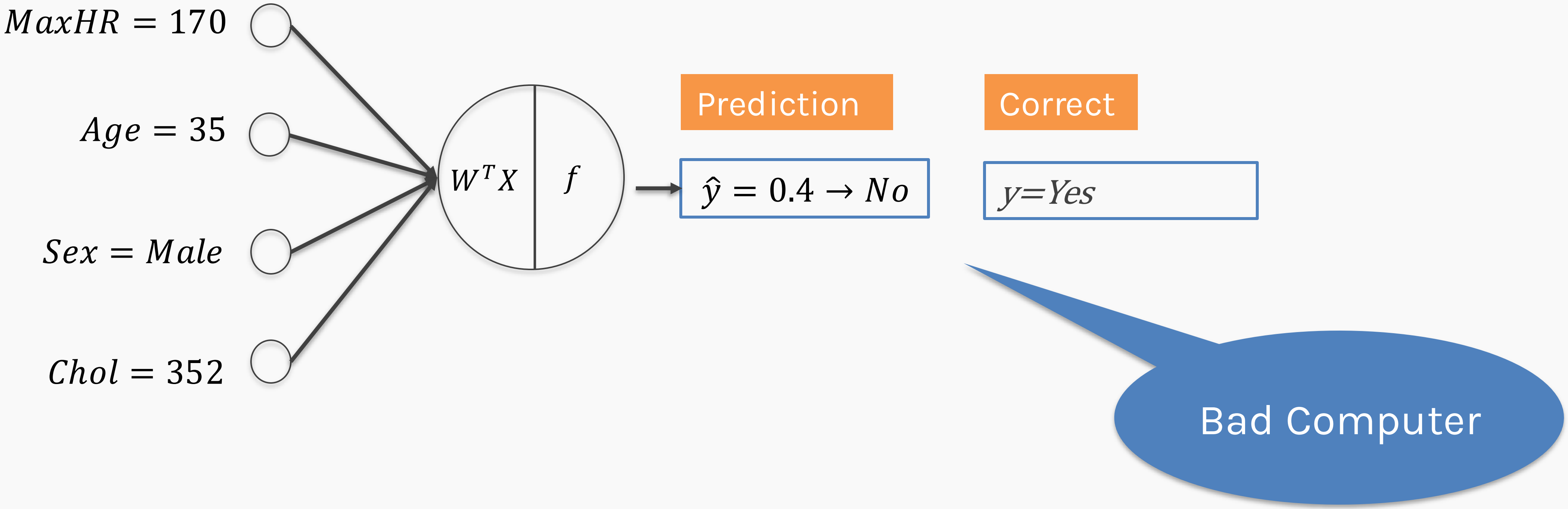
Most likely it will perform horribly. For example, in our heart data, the model will be giving us the wrong answer.



Gradient Descent - Introduction

Start with all randomly selected weights.

Most likely it will perform horribly. For example, in our heart data, the model will be giving us the wrong answer.



Gradient Descent - Introduction

- The **loss function** takes all these results and averages them and tells us how bad or good the computer or those weights are.

- Telling the computer whether it is **bad** or **good** does not help.

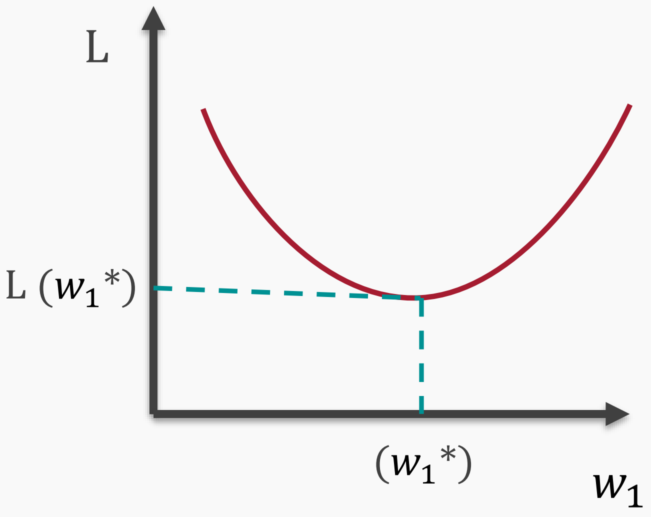
- You also want to tell it how to change those weights, so it gets better.

Minimizing the Loss function

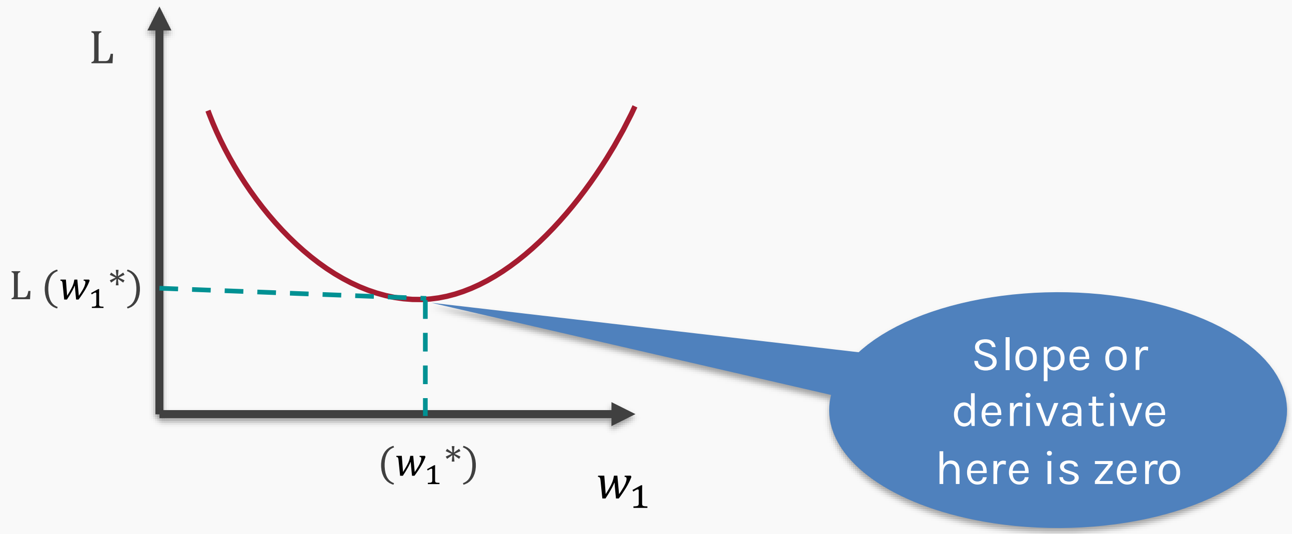
Loss function: $\mathcal{L}(w_0, w_1, w_2, w_3, w_4)$

For now, let's only consider a single weight, w_1, and ignore the dependance of the loss function on all other weights: $\mathcal{L}(w_1)$.

Ideally, we want to know the value of w_1 that gives the minimal $\mathcal{L}(w_1)$.



Minimizing the Loss function



How do we find the optimal point of a function $\mathcal{L}(w_1)$?

- We take the derivative wrt the weight, w_1, and equate it to 0: $\frac{d\mathcal{L}(w_1)}{dw_1} = 0$
- Find the value of w_1 that satisfies the above equation.

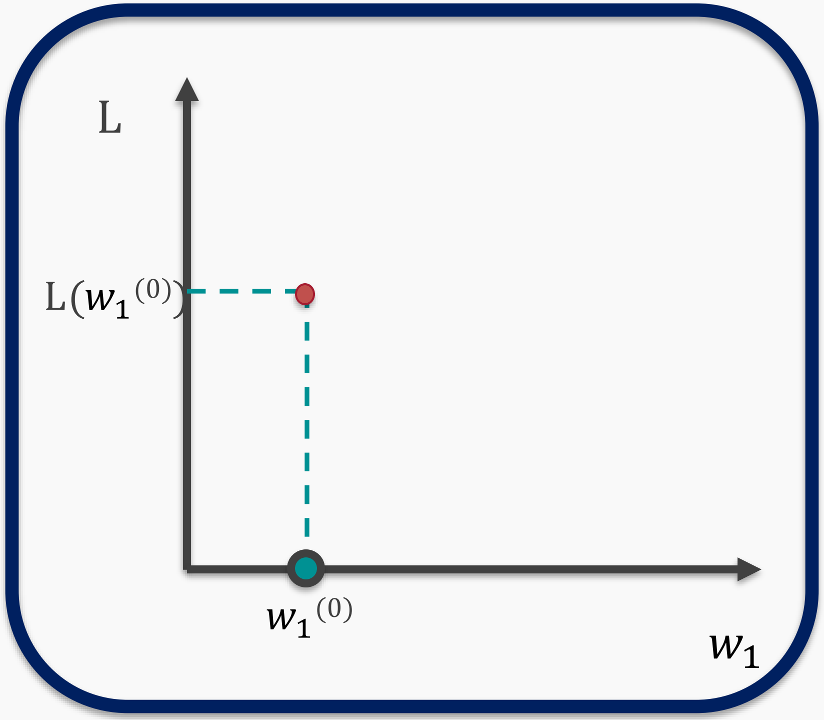
Sometimes there is no explicit solution for that.

Gradient Descent

A more flexible method would be:

Gradient Descent

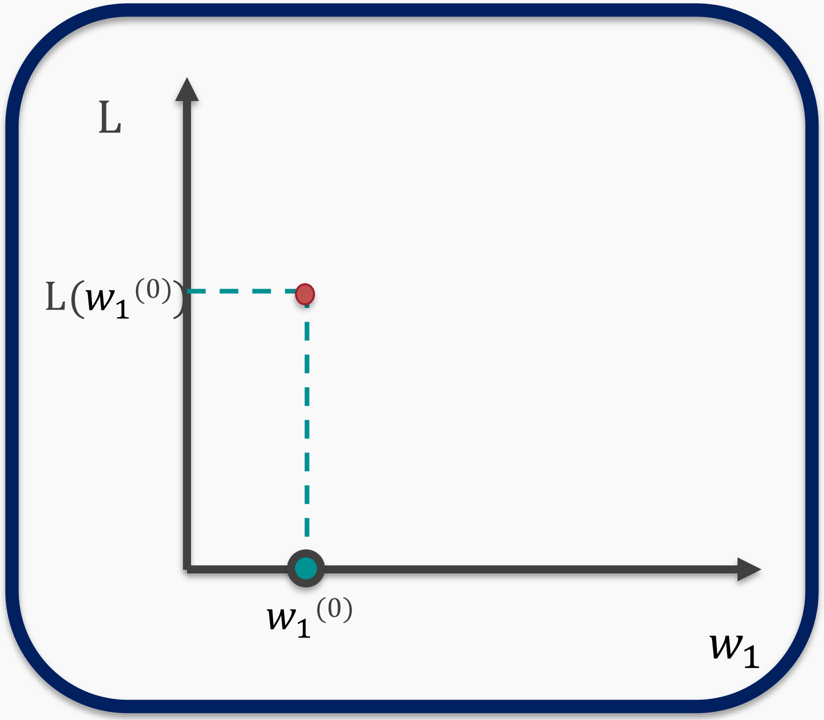
A more flexible method would be:



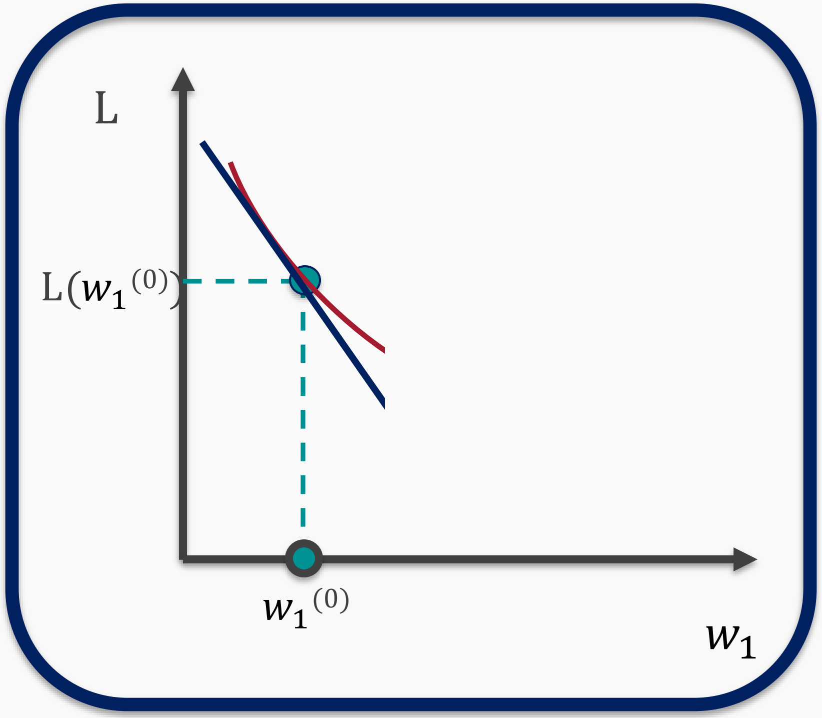
Start from a random point

Gradient Descent

A more flexible method would be:



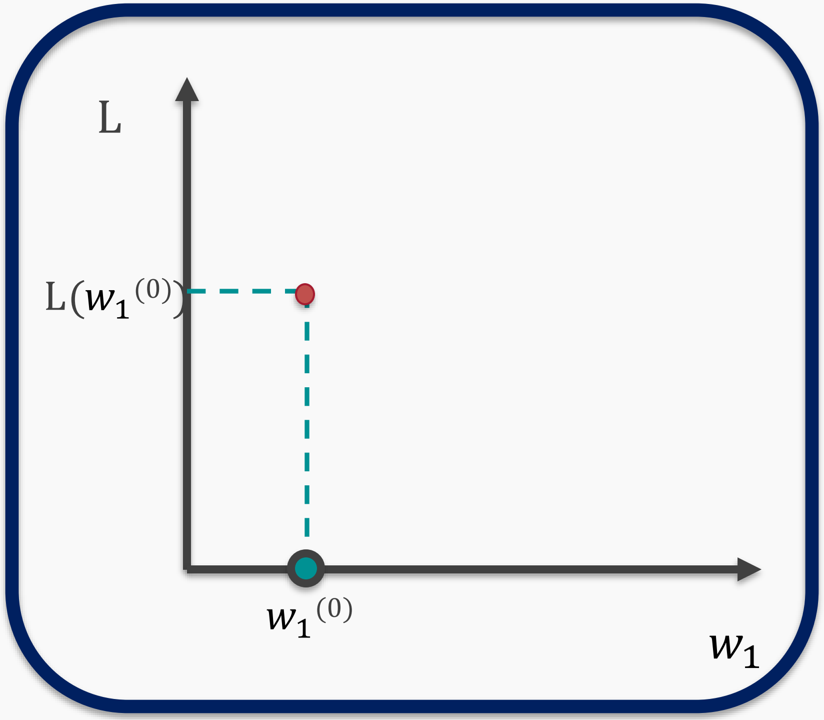
Start from a random point



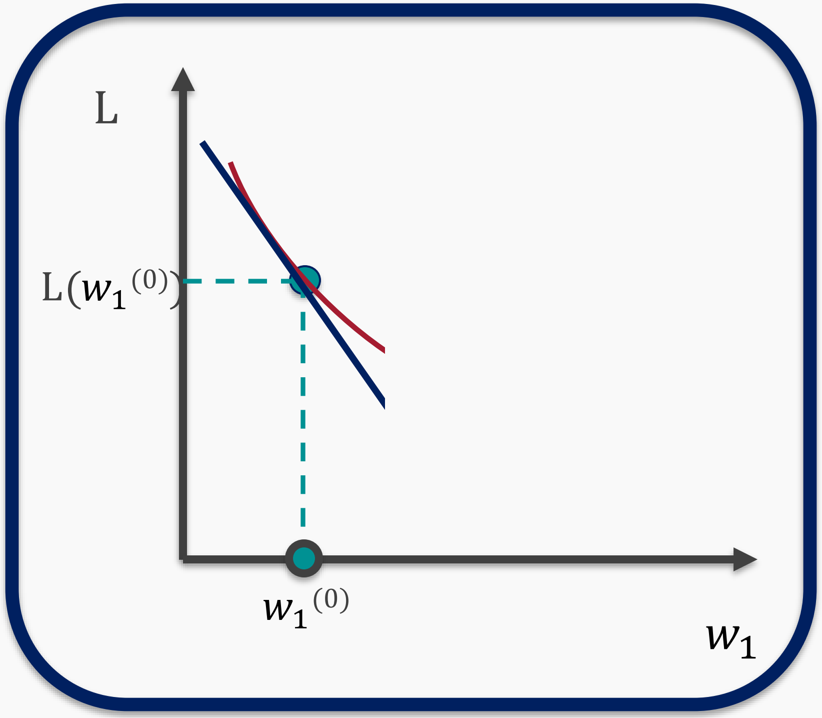
Compute the slope/derivative at this point

Gradient Descent

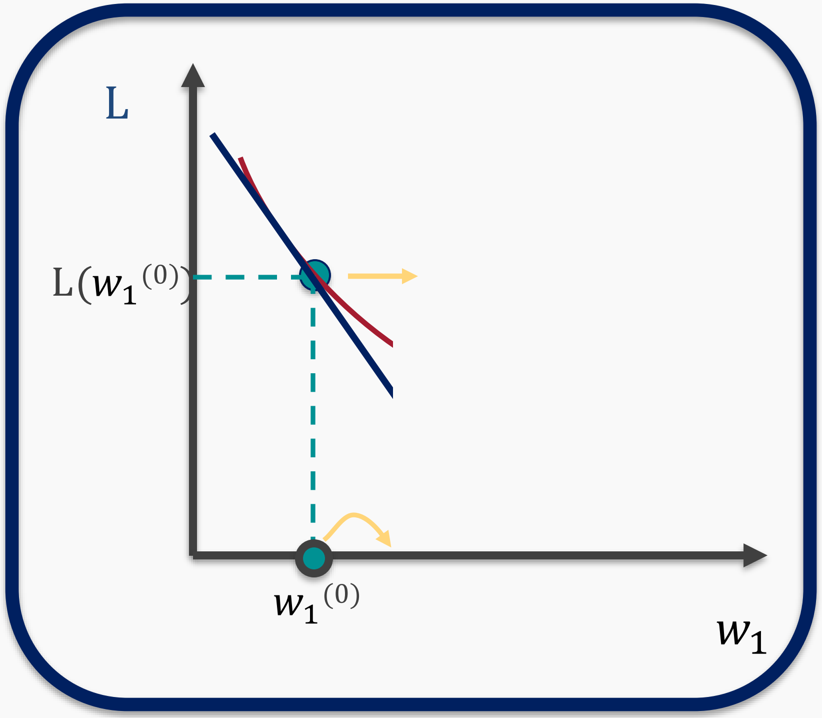
A more flexible method would be:



Start from a random point



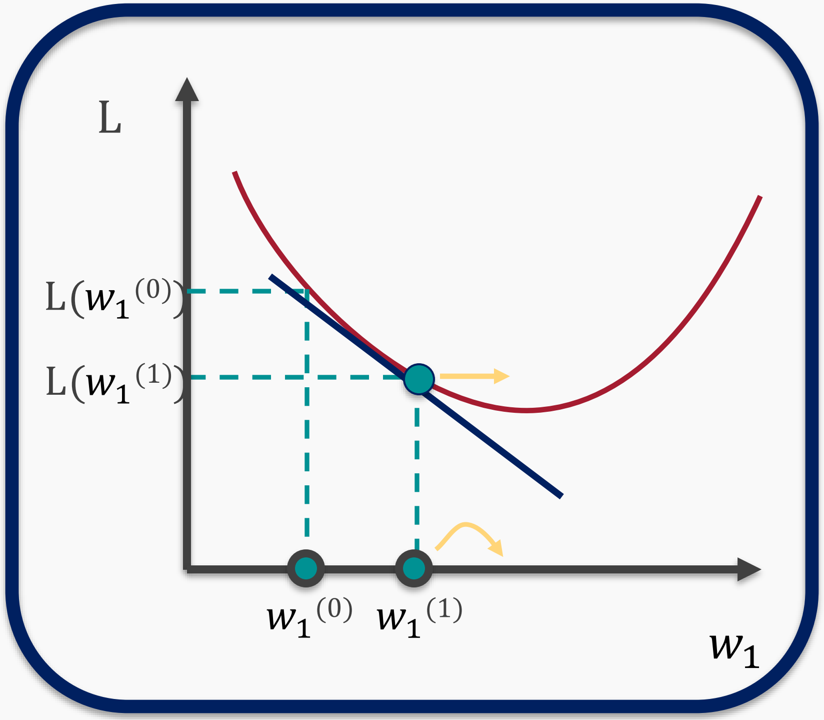
Compute the slope/derivative at this point



Step to the opposite direction of the derivative

Gradient Descent

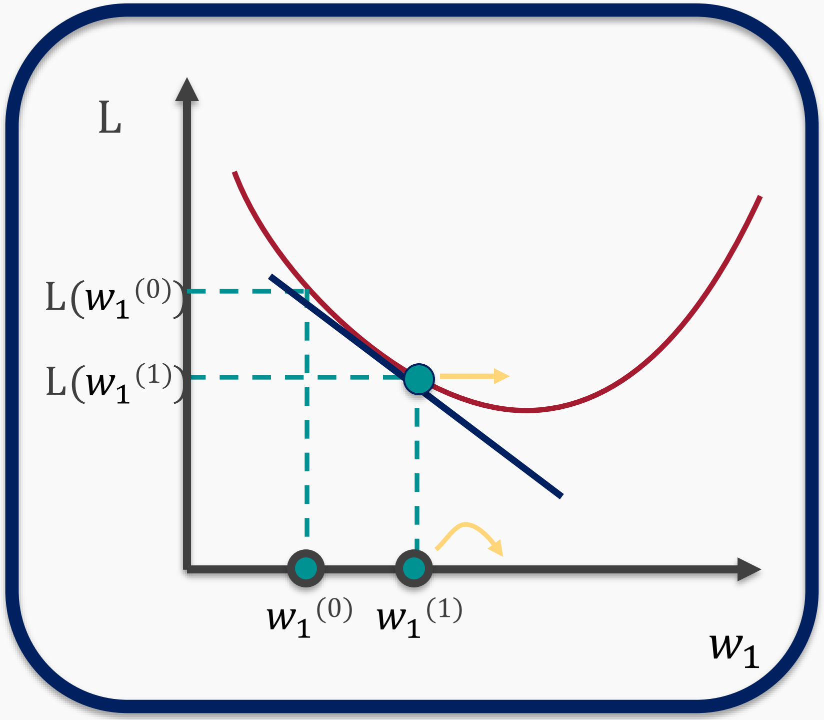
A more flexible method would be:



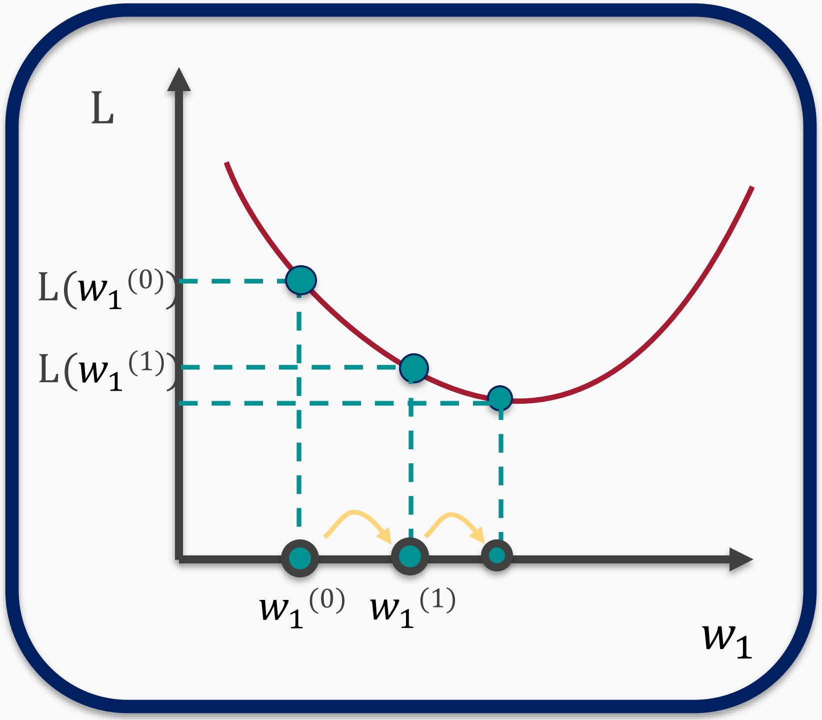
Compute the slope/derivative at $w_1{}^{(1)}$ and step again in the opposite direction of the derivative.

Gradient Descent

A more flexible method would be:



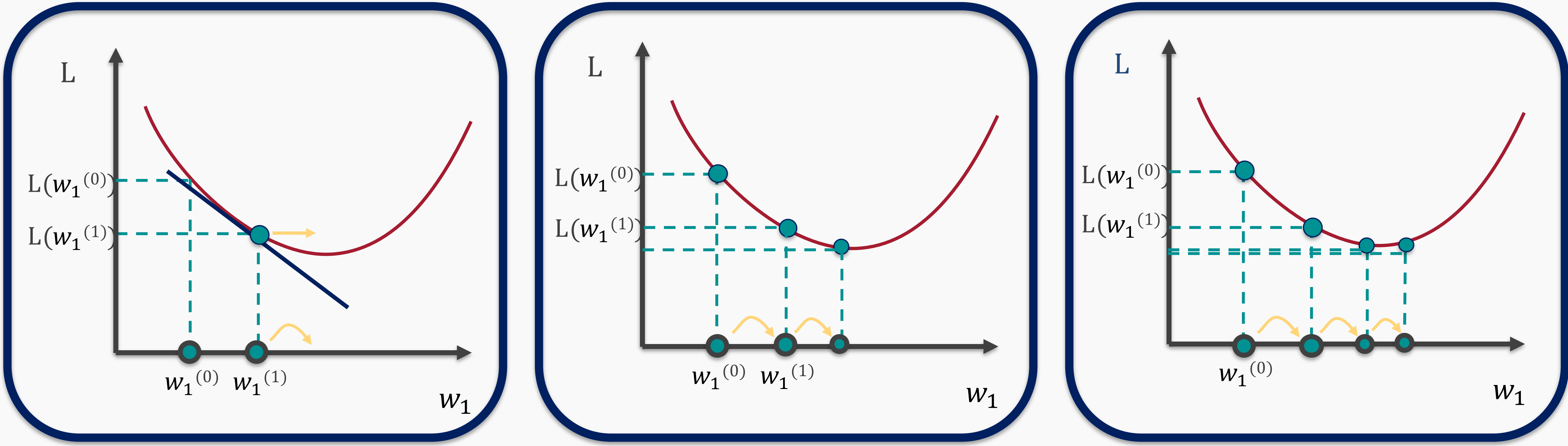
Compute the slope/derivative at $w_1{}^{(1)}$ and step again in the opposite direction of the derivative.



Continue,

Gradient Descent

A more flexible method would be:



Compute the slope/derivative at $w_1^{(1)}$ and step again in the opposite direction of the derivative.

Continue,

Stop when no more improvement or after a certain number of iterations.

Gradient Descent

Question: How do we generalize this to more than one weight?

Take the gradient:

$$\boldsymbol{\nabla_W L(W)} = \left[\frac{\partial \boldsymbol{L}}{\partial \boldsymbol{W_1}}, \frac{\partial \boldsymbol{L}}{\partial \boldsymbol{W_2}}, \ldots, \frac{\partial \boldsymbol{L}}{\partial \boldsymbol{W_p}}\right]^{\mathbf{T}}$$

This symbol is called the **nabla** symbol.

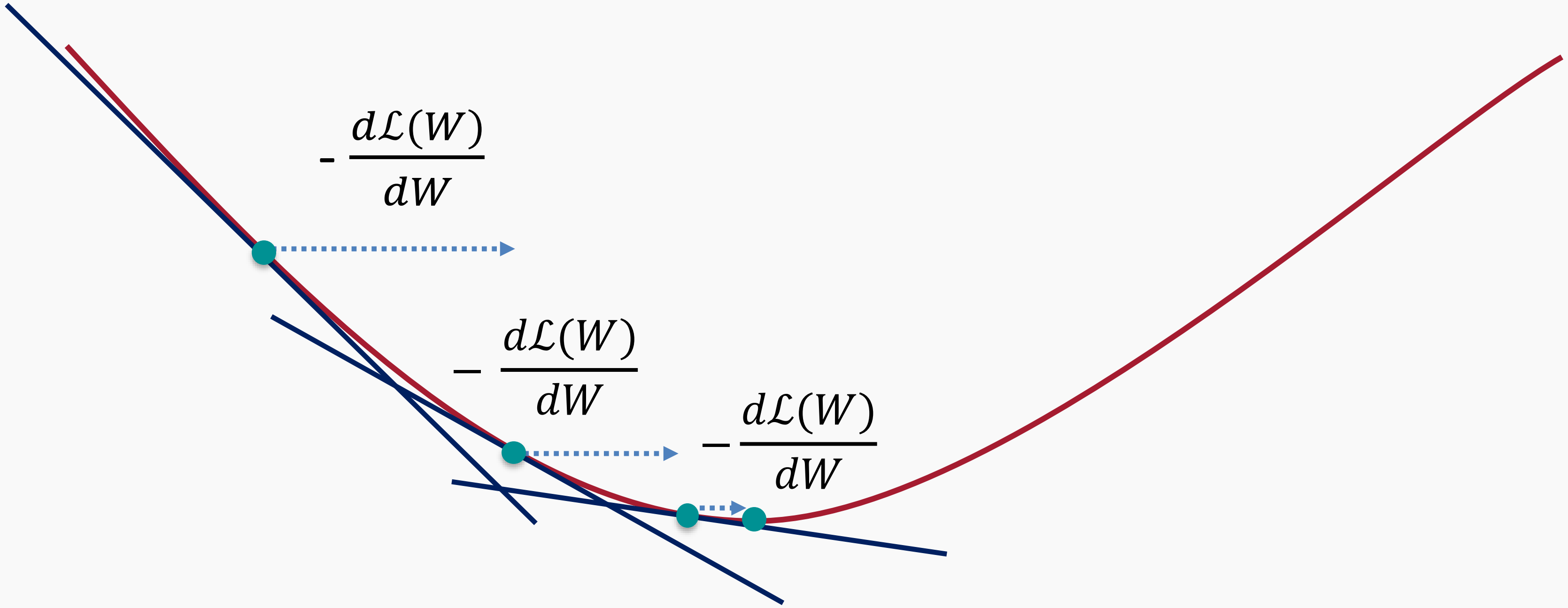
A gradient is a vector that points in the direction of greatest increase of L.

Gradient Descent

Question: With respect to step size, what do you think would be a good strategy for improving the model when using gradient descent?

Gradient Descent (cont.)

If the step is proportional to the slope, then you avoid overshooting the minimum. How?



Gradient Descent

We know that we want to go in the opposite direction of the derivative, and we know we want to be making a step proportional to the derivative.

Making a step means:

$$\boxed{w^{new} = w^{old} + step}$$

Opposite direction of the derivative and proportional to the derivative means:

$$\boxed{w^{new} = w^{old} - \eta \frac{d\mathcal{L}}{dw}}$$

Step size is proportional to derivative

Learning Rate

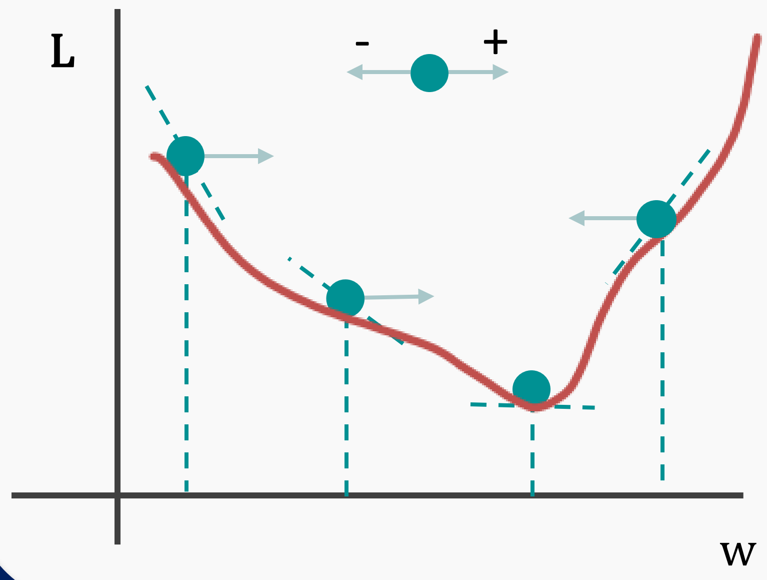
Change to more conventional notation:

$$\boxed{w^{(i+1)} = w^{(i)} - \eta \frac{d\mathcal{L}}{dw}}$$

Gradient Descent

- Algorithm for optimization of first order to finding a minimum of a function.
- It is an iterative method.
- L is decreasing much faster in the direction of the negative derivative.
- The learning rate is controlled by the magnitude of η.

$$w^{(i+1)} = w^{(i)} - \eta \frac{d\mathcal{L}}{dw}$$

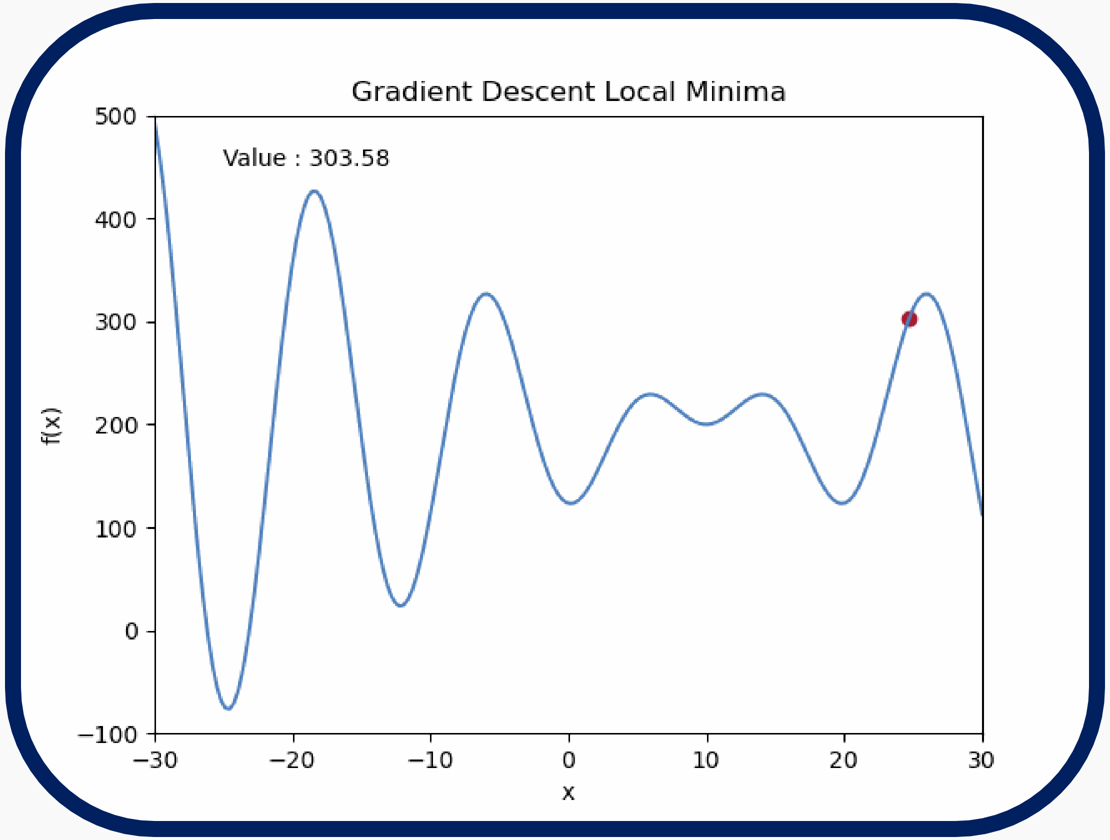


Gradient Descent – Local Minima

- During training, gradient descent seeks the lowest possible point (global minimum) to minimize error. However, it can mistakenly settle in a local minimum, thinking it has found the optimal solution.

- **Local minima** represent the shallow valleys within the vast landscape of the network's loss function

- The key challenge is to find the global minima.



Game Time

What should be considered when specifying the gradient descent algorithm? (Select all that apply)

A. Calculating the derivatives

B. How to set the learning rate

C. Avoid local minima

D. It should ensure that the gradients remain large

Gradient Descent Considerations

- We still need to calculate the derivatives.
- We need to set the learning rate.
- Local vs global minima.
- The full likelihood function includes summing up all individual ‘*errors*’. Sometimes this includes hundreds of thousands of examples.

Calculate the Derivatives

Can we do it? Can we calculate the derivative of any loss function?

Wolfram Alpha can do it for us!

However, we need a formalism to deal with these derivatives.

Chain Rule

Chain rule for computing gradients:

$$y = g(x) \qquad z = f(y) = f\big(g(x)\big)$$

$$\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx}$$

$$\boldsymbol{y} = g(\boldsymbol{x}) \qquad z = f(\boldsymbol{y}) = f\big(g(\boldsymbol{x})\big)$$

$$\frac{\partial z}{\partial x_i} = \sum_j \frac{\partial z}{\partial y_j}\frac{\partial y_j}{\partial x_i}$$

For longer chains:

$$\frac{\partial z}{\partial x_i} = \sum_{j_1} \dots \sum_{j_m} \frac{\partial z}{\partial y_{j_1}} \dots \frac{\partial y_{j_m}}{\partial x_i}$$

Example: Logistic Regression derivatives

For logistic regression, the negative log of the likelihood is:

$$\mathcal{L} = \sum_i \mathcal{L}_i = -\sum_i \log L_i = -\sum_i [y_i \log p_i + (1 - y_i) \log(1 - p_i)]$$

$$\mathcal{L}_i = -y_i \log \frac{1}{1 + e^{-W^T X_i}} - (1 - y_i) \log(1 - \frac{1}{1 + e^{-W^T X_i}})$$

To simplify the analysis let us split it into two parts,

$$\mathcal{L}_i = \mathcal{L}_i^A + \mathcal{L}_i^B$$

So, the derivative with respect to W is:

$$\frac{d\mathcal{L}}{dW} = \sum_i \frac{d\mathcal{L}_i}{dW} = \sum_i (\frac{d\mathcal{L}_i^A}{dW} + \frac{d\mathcal{L}_i^B}{dW})$$

$$\mathcal{L}_i^A = -y_i \log \frac{1}{1 + e^{\boxed{-W^T X_i}}}$$

Variables	Derivatives	Expanded Derivatives
$\xi_1 = -W^T X_\mathrm{i}$	$\frac{d\xi_1}{\mathrm{d}W} = -X_\mathrm{i}$	$\frac{d\xi_1}{\mathrm{d}W} = -X_\mathrm{i}$

$$\mathcal{L}_i^A = -y_i \log \frac{1}{1 + e^{\boxed{-W^T X_i}}}$$

Variables	Derivatives	Expanded Derivatives
$\xi_1 = -W^T X_\mathrm{i}$	$\frac{d\xi_1}{\mathrm{d}W} = -X_\mathrm{i}$	$\frac{d\xi_1}{\mathrm{d}W} = -X_\mathrm{i}$

$$\mathcal{L}_i^A = -y_i \log \frac{1}{1 + \boxed{e^{-W^T X_i}}}$$

Variables	Derivatives	Expanded Derivatives
$\xi_1 = -W^T X_\mathrm{i}$	$\frac{\mathrm{d}\xi_1}{\mathrm{d}W} = -X_\mathrm{i}$	$\frac{\mathrm{d}\xi_1}{\mathrm{d}W} = -X_\mathrm{i}$
$\xi_2 = e^{\xi_1} = e^{-W^T X_i}$	$\frac{\mathrm{d}\xi_2}{\mathrm{d}\xi_1} = e^{\xi_1}$	$\frac{\mathrm{d}\xi_2}{\mathrm{d}\xi_1} = e^{-W^T X_\mathrm{i}}$

$$\mathcal{L}_i^A = -y_i \log \frac{1}{\boxed{1 + e^{-W^T X_i}}}$$

Variables	Derivatives	Expanded Derivatives
$\xi_1 = -W^T X_\mathrm{i}$	$\frac{\mathrm{d}\xi_1}{\mathrm{d}W} = -X_\mathrm{i}$	$\frac{\mathrm{d}\xi_1}{\mathrm{d}W} = -X_\mathrm{i}$
$\xi_2 = e^{\xi_1} = e^{-W^T X_\mathrm{i}}$	$\frac{\mathrm{d}\xi_2}{\mathrm{d}\xi_1} = e^{\xi_1}$	$\frac{\mathrm{d}\xi_2}{\mathrm{d}\xi_1} = e^{-W^T X_\mathrm{i}}$
$\xi_3 = 1 + \xi_2 = 1 + e^{-W^T X_i}$	$\frac{\mathrm{d}\xi_3}{\mathrm{d}\xi_2} = 1$	$\frac{\mathrm{d}\xi_3}{\mathrm{d}\xi_2}$ =1

$$\mathcal{L}_i^A = -y_i \log \boxed{\frac{1}{1+e^{-W^T X_i}}}$$

Variables	Derivatives	Expanded Derivatives
$\xi_1 = -W^T X_\mathrm{i}$	$\frac{\mathrm{d}\xi_1}{\mathrm{d}W} = -X_\mathrm{i}$	$\frac{\mathrm{d}\xi_1}{\mathrm{d}W} = -X_\mathrm{i}$
$\xi_2 = e^{\xi_1} = e^{-W^T X_\mathrm{i}}$	$\frac{\mathrm{d}\xi_2}{\mathrm{d}\xi_1} = e^{\xi_1}$	$\frac{\mathrm{d}\xi_2}{\mathrm{d}\xi_1} = e^{-W^T X_\mathrm{i}}$
$\xi_3 = 1 + \xi_2 = 1 + e^{-W^T X_\mathrm{i}}$	$\frac{\mathrm{d}\xi_3}{\mathrm{d}\xi_2} = 1$	$\frac{\mathrm{d}\xi_3}{\mathrm{d}\xi_2} = 1$
$\xi_4 = \frac{1}{\xi_3} = \frac{1}{1+e^{-W^T X_\mathrm{i}}} = p$	$\frac{\mathrm{d}\xi_4}{\mathrm{d}\xi_3} = -\frac{1}{\xi_3^2}$	$\frac{\mathrm{d}\xi_4}{\mathrm{d}\xi_3} = -\frac{1}{\left(1+e^{-W^T X_i}\right)^2}$

$$\mathcal{L}_i^A = -y_i \boxed{\log \frac{1}{1+e^{-W^T X_i}}}$$

Variables	Derivatives	Expanded Derivatives
$\xi_1 = -W^T X_\mathrm{i}$	$\frac{\mathrm{d}\xi_1}{\mathrm{d}W} = -X_\mathrm{i}$	$\frac{\mathrm{d}\xi_1}{\mathrm{d}W} = -X_\mathrm{i}$
$\xi_2 = e^{\xi_1} = e^{-W^T X_\mathrm{i}}$	$\frac{\mathrm{d}\xi_2}{\mathrm{d}\xi_1} = e^{\xi_1}$	$\frac{\mathrm{d}\xi_2}{\mathrm{d}\xi_1} = e^{-W^T X_\mathrm{i}}$
$\xi_3 = 1 + \xi_2 = 1 + e^{-W^T X_\mathrm{i}}$	$\frac{\mathrm{d}\xi_3}{\mathrm{d}\xi_2} = 1$	$\frac{\mathrm{d}\xi_3}{\mathrm{d}\xi_2} = 1$
$\xi_4 = \frac{1}{\xi_3} = \frac{1}{1+e^{-W^T X_\mathrm{i}}} = p$	$\frac{\mathrm{d}\xi_4}{\mathrm{d}\xi_3} = -\frac{1}{\xi_3^2}$	$\frac{\mathrm{d}\xi_4}{\mathrm{d}\xi_3} = -\frac{1}{\left(1+e^{-W^T X_\mathrm{i}}\right)^2}$
$\xi_5 = \log \xi_4 = \log p = \log \frac{1}{1+e^{-W^T X_\mathrm{i}}}$	$\frac{\mathrm{d}\xi_5}{\mathrm{d}\xi_4} = \frac{1}{\xi_4}$	$\frac{\mathrm{d}\xi_5}{\mathrm{d}\xi_4} = 1 + e^{-W^T X_\mathrm{i}}$

$$\mathcal{L}_i^A = -y_i \log \frac{1}{1+e^{-W^T X_i}}$$

Variables	Derivatives	Expanded Derivatives
$\xi_1 = -W^T X_\mathrm{i}$	$\frac{\mathrm{d}\xi_1}{\mathrm{d}W} = -X_\mathrm{i}$	$\frac{\mathrm{d}\xi_1}{\mathrm{d}W} = -X_\mathrm{i}$
$\xi_2 = e^{\xi_1} = e^{-W^T X_\mathrm{i}}$	$\frac{\mathrm{d}\xi_2}{\mathrm{d}\xi_1} = e^{\xi_1}$	$\frac{\mathrm{d}\xi_2}{\mathrm{d}\xi_1} = e^{-W^T X_\mathrm{i}}$
$\xi_3 = 1 + \xi_2 = 1 + e^{-W^T X_\mathrm{i}}$	$\frac{\mathrm{d}\xi_3}{\mathrm{d}\xi_2} = 1$	$\frac{\mathrm{d}\xi_3}{\mathrm{d}\xi_2} = 1$
$\xi_4 = \frac{1}{\xi_3} = \frac{1}{1+e^{-W^T X_\mathrm{i}}} = p$	$\frac{\mathrm{d}\xi_4}{\mathrm{d}\xi_3} = -\frac{1}{\xi_3^2}$	$\frac{\mathrm{d}\xi_4}{\mathrm{d}\xi_3} = -\frac{1}{\left(1+e^{-W^T X_\mathrm{i}}\right)^2}$
$\xi_5 = \log \xi_4 = \log p = \log \frac{1}{1+e^{-W^T X_\mathrm{i}}}$	$\frac{\mathrm{d}\xi_5}{\mathrm{d}\xi_4} = \frac{1}{\xi_4}$	$\frac{\mathrm{d}\xi_5}{\mathrm{d}\xi_4} = 1 + e^{-W^T X_\mathrm{i}}$
$\mathcal{L}_i^A = -y_\mathrm{i}\xi_5$	$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\xi_5} = -y_\mathrm{i}$	$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\xi_5} = -y_\mathrm{i}$
$\frac{d\mathcal{L}_i^A}{\mathrm{d}W} = \frac{\mathrm{d}\mathcal{L}_i}{\mathrm{d}\xi_5}\frac{\mathrm{d}\xi_5}{\mathrm{d}\xi_4}\frac{\mathrm{d}\xi_4}{\mathrm{d}\xi_3}\frac{\mathrm{d}\xi_3}{\mathrm{d}\xi_2}\frac{\mathrm{d}\xi_2}{\mathrm{d}\xi_1}\frac{\mathrm{d}\xi_1}{\mathrm{d}W}$		$\frac{d\mathcal{L}_i^A}{\mathrm{d}W} = -y_\mathrm{i} X_\mathrm{i} e^{-W^T X_\mathrm{i}} \frac{1}{\left(1+e^{-W^T X_\mathrm{i}}\right)}$

$$\mathcal{L}_i^B = -(1-y_i)\log[1-\frac{1}{1+e^{-W^T X_i}}]$$

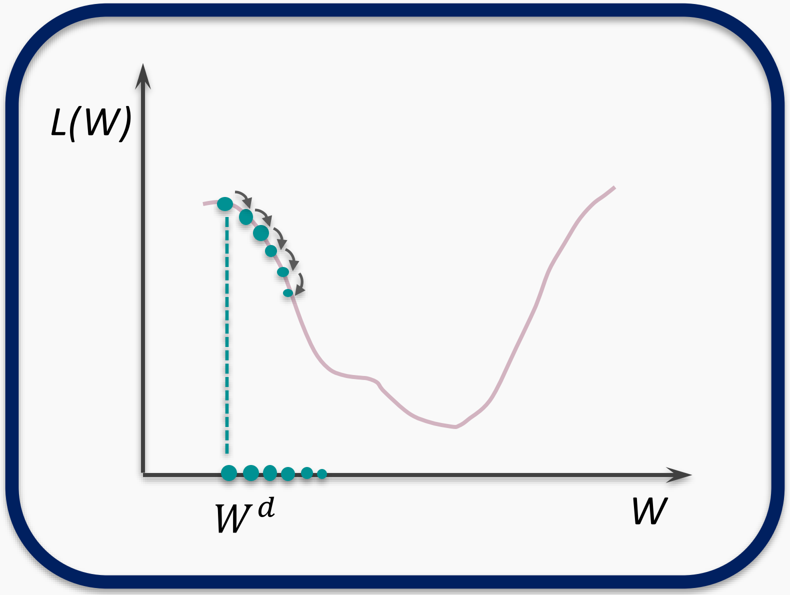
Variables	Derivatives	Expanded Derivatives
$\xi_1 = -W^T X_\mathrm{i}$	$\frac{\mathrm{d}\xi_1}{\mathrm{d}W} = -X_\mathrm{i}$	$\frac{\mathrm{d}\xi_1}{\mathrm{d}W} = -X_\mathrm{i}$
$\xi_2 = e^{\xi_1} = e^{-W^T X_\mathrm{i}}$	$\frac{\mathrm{d}\xi_2}{\mathrm{d}\xi_1} = e^{\xi_1}$	$\frac{\mathrm{d}\xi_2}{\mathrm{d}\xi_1} = e^{-W^T X_\mathrm{i}}$
$\xi_3 = 1 + \xi_2 = 1 + e_i^{-W^T X}$	$\frac{\mathrm{d}\xi_3}{\mathrm{d}\xi_2} = 1$	$\frac{\mathrm{d}\xi_3}{\mathrm{d}\xi_2}$ =1
$\xi_4 = \frac{1}{\xi_3} = \frac{1}{1+e^{-W^T X_\mathrm{i}}} = p$	$\frac{\mathrm{d}\xi_4}{\mathrm{d}\xi_3} = -\frac{1}{\xi_3^2}$	$\frac{\mathrm{d}\xi_4}{\mathrm{d}\xi_3} = -\frac{1}{\left(1+e^{-W^T X_\mathrm{i}}\right)^2}$
$\xi_5 = 1 - \xi_4 = 1 - \frac{1}{1+e_i^{-W^T X}}$	$\frac{\mathrm{d}\xi_5}{\mathrm{d}\xi_4} = -1$	$\frac{\mathrm{d}\xi_5}{\mathrm{d}\xi_4}$ =-1
$\xi_6 = \log\xi_5 = \log(1-p) \;= \log\frac{1}{1+e^{-W^T X_\mathrm{i}}}$	$\frac{\mathrm{d}\xi_6}{\mathrm{d}\xi_5} = \frac{1}{\xi_5}$	$\frac{\mathrm{d}\xi_6}{\mathrm{d}\xi_5} = \frac{1+e^{-W^T X_\mathrm{i}}}{e^{-W^T X_\mathrm{i}}}$
$\mathcal{L}_i^B = (1-y_\mathrm{i})\xi_6$	$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\xi_6} = 1 - y_\mathrm{i}$	$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\xi_6} = 1 - y_\mathrm{i}$
$\frac{d\mathcal{L}_i^B}{\mathrm{d}W} = \frac{\mathrm{d}\mathcal{L}_i^B}{\mathrm{d}\xi_6}\frac{\mathrm{d}\xi_6}{\mathrm{d}\xi_5}\frac{\mathrm{d}\xi_5}{\mathrm{d}\xi_4}\frac{\mathrm{d}\xi_4}{\mathrm{d}\xi_3}\frac{\mathrm{d}\xi_3}{\mathrm{d}\xi_2}\frac{\mathrm{d}\xi_2}{\mathrm{d}\xi_1}\frac{\mathrm{d}\xi_1}{\mathrm{d}W}$		$\frac{d\mathcal{L}_i^B}{\mathrm{d}W} = -(1-y_\mathrm{i})X_\mathrm{i}\frac{1}{(1+e^{-W\,X_\mathrm{i}})}$

Gradient Descent Considerations

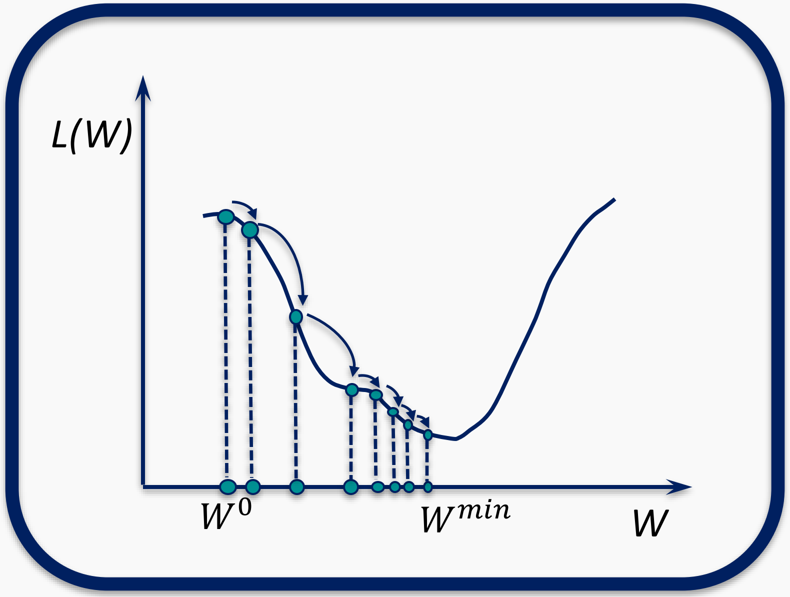
- We still need to calculate the derivatives.
- **We need to set the** learning rate.
- Local vs global minima.
- The full likelihood function includes summing up all individual '*errors*'. Sometimes this includes hundreds of thousands of examples.

Learning Rate

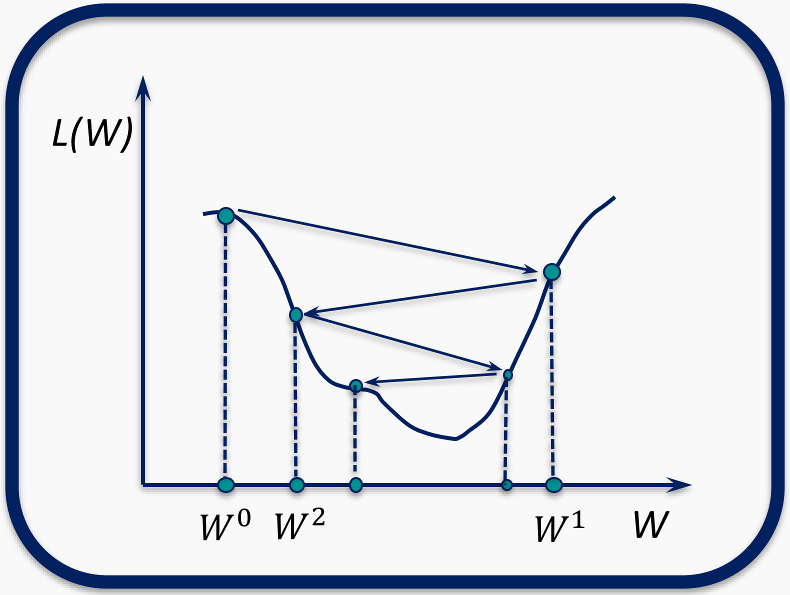
Our choice of the learning rate has a significant impact on the performance of gradient descent.



When η is too small, the algorithm makes very little progress.



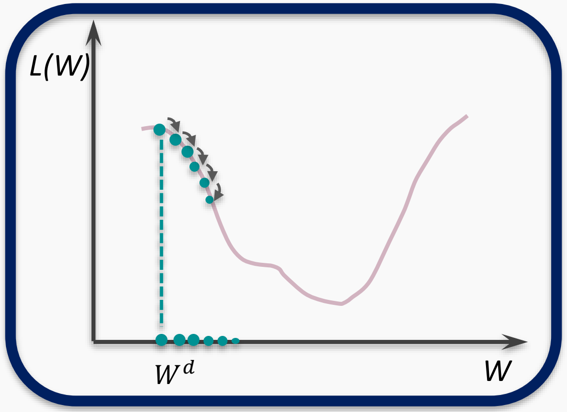
When η is appropriate, the algorithm will find the minimum. The algorithm **converges**!

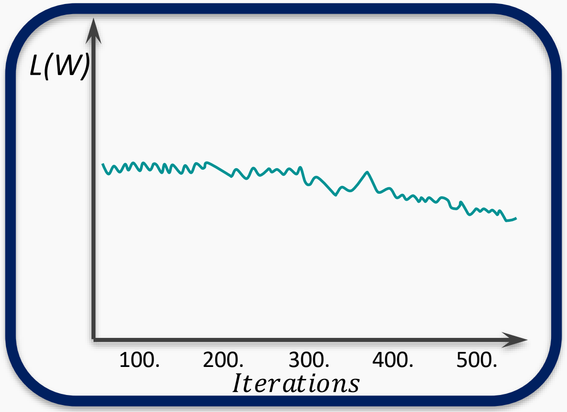


When η is too large, the algorithm may overshoot the minimum and has crazy oscillations.

How can we tell when gradient descent is converging? We visualize the loss function at each step of gradient descent. This is called the **trace plot**.

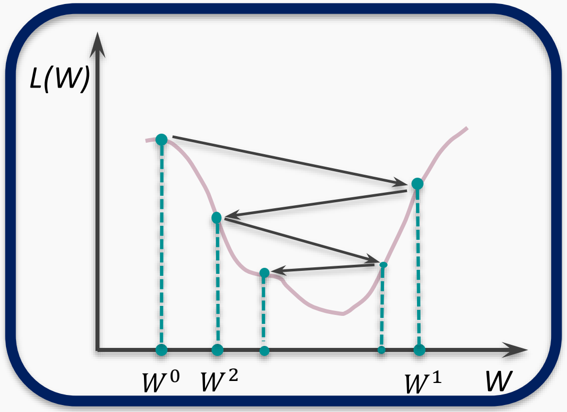
η is too small

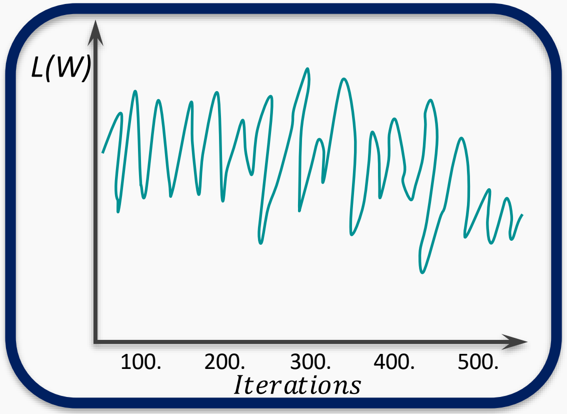




While the loss is decreasing throughout training, it does not look like descent hit the bottom.

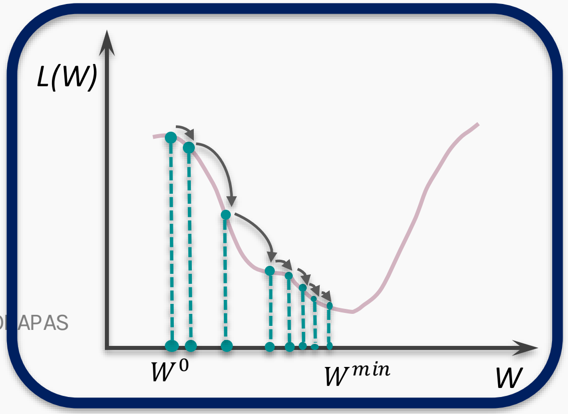
η is too large

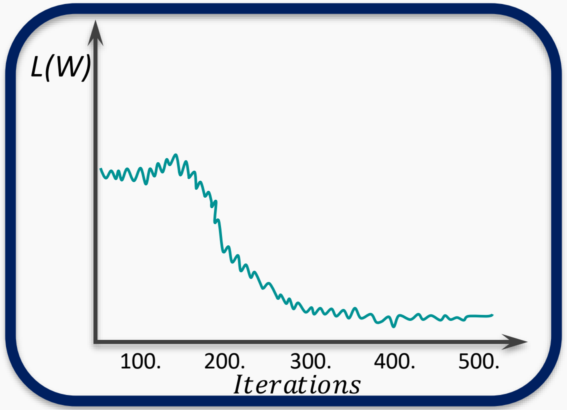




Loss is mostly oscillating between values rather than converging.

η is appropriate





The loss has decreased significantly during training. Towards the end, the loss stabilizes and it can't decrease further.

Learning Rate

There are many alternative methods which address how to set or adjust the learning rate, using the derivative or second derivatives and/or the momentum.

More on this later

Gradient Descent Considerations

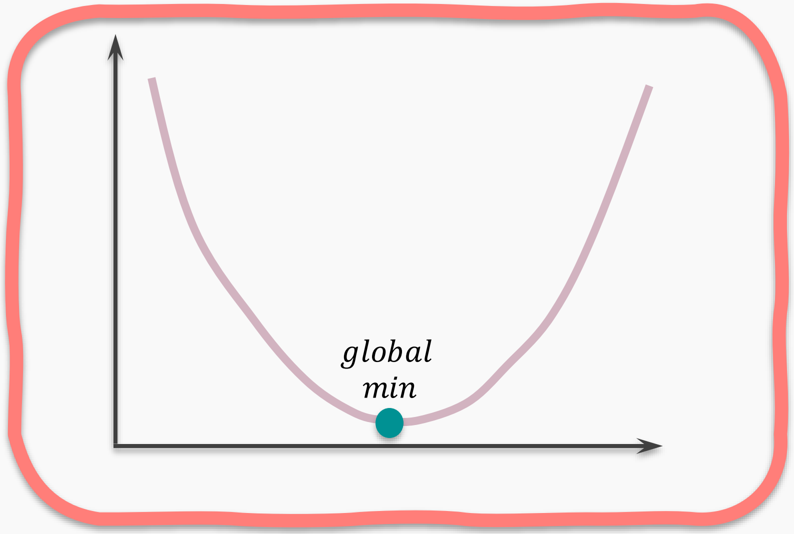
- We still need to calculate the derivatives.
- We need to set the learning rate.
- **Local vs global minima.**
- The full likelihood function includes summing up all individual '*errors*'. Sometimes this includes hundreds of thousands of examples.

Local vs Global Minima

If we choose η correctly, then gradient descent will converge to a stationary point. But will this point be a **global minimum**?

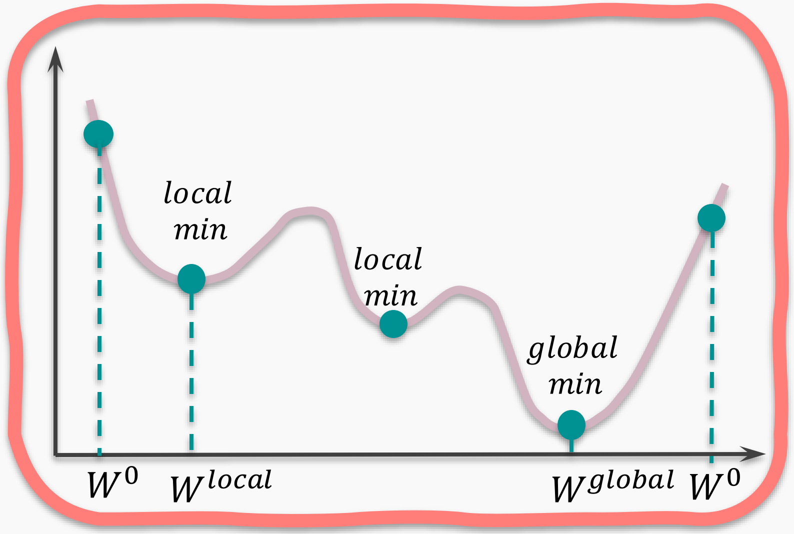
If the function is convex then the stationary point will be a global minimum.

For linear and polynomial regressions Loss Functions are Convex



Hessian (2^{nd} derivative) positive semi-definite everywhere. Every stationary point of the gradient is a global min

Neural Network Loss Functions are not Convex



Neural networks with different weights can correspond to the same function.
Most stationary points are local minima but not global optima.

Local vs Global Minima

There is still no guarantee that we get the global minimum.

Game Time

What would be a good strategy to increase our chances to converge to the global minimum?

A. Restart training with different weight initialization

B. Take bigger steps when doing gradient descent

C. Add noise to the loss function

D. We cannot increase our chances of getting to the global minimum

Gradient Descent Considerations

- We still need to calculate the derivatives. ☑
- We need to set the learning rate. ☑
- Local vs global minima. ☑
- The full likelihood function includes summing up all individual '*errors*'. Sometimes this includes hundreds of thousands of examples.