Building Linear Regression Models Using TensorFlow

Vitthal Srinivasan
CO-FOUNDER, LOONYCORN
www.loonycorn.com

Implementing Regression in TensorFlow

Baseline

Non-TensorFlow implementation

Regular python code

Computation Graph

Neural network of 1 neuron

Affine transformation suffices

Cost Function

Mean Square Error (MSE)

Quantifying goodness-of-fit

Optimizer

Gradient Descent optimizers

Improving goodness-of-fit

Training

Invoke optimizer in epochs

Batch size for each epoch

Converged Model

Values of W and b

Compare to baseline

Simple Regression

Cause

Changes in S&P 500 equity index

Effect

Changes in price of Google stock

Implementing Regression in TensorFlow

Baseline

Non-TensorFlow implementation

Regular python code

Regression in Python

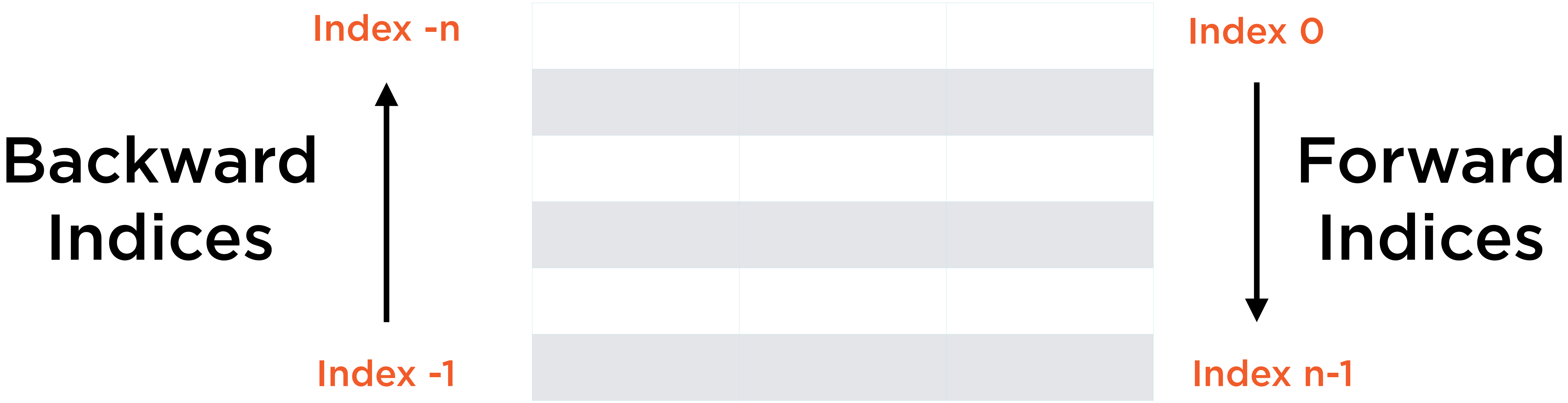
Pandas for dataframes

NumPy for arrays

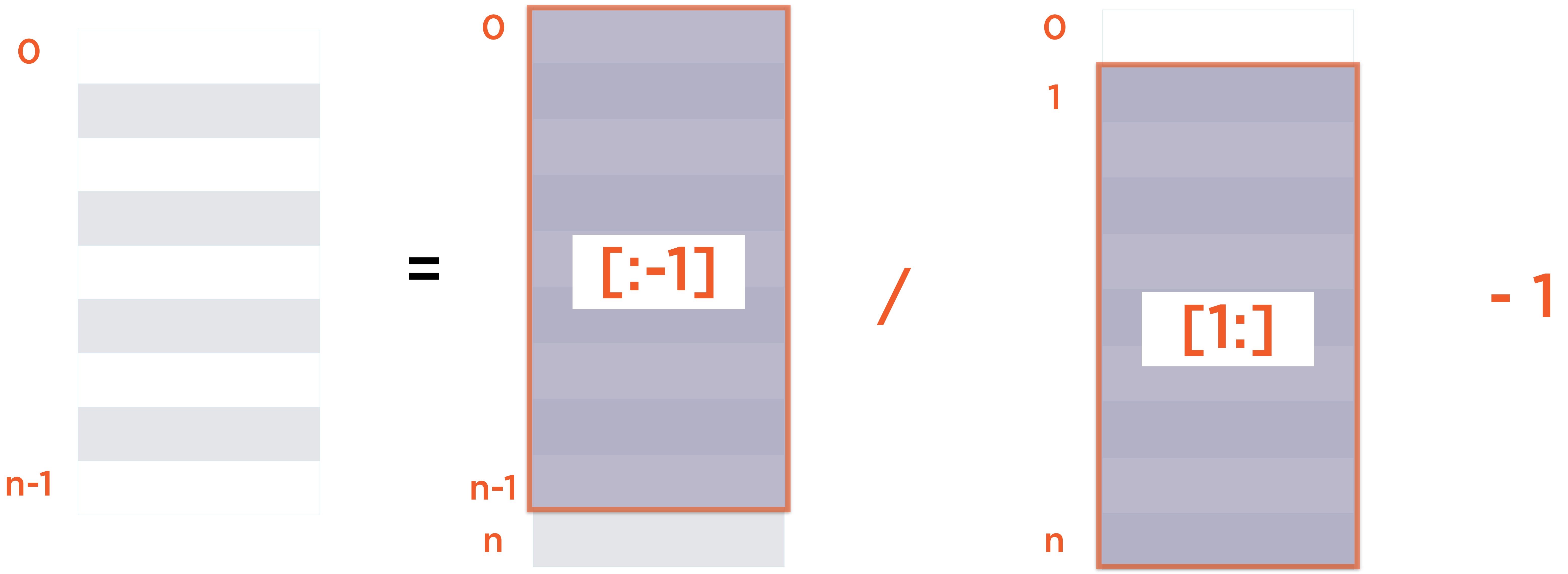
Statsmodels for regression

Matplotlib for plots

Negative Indices In Python

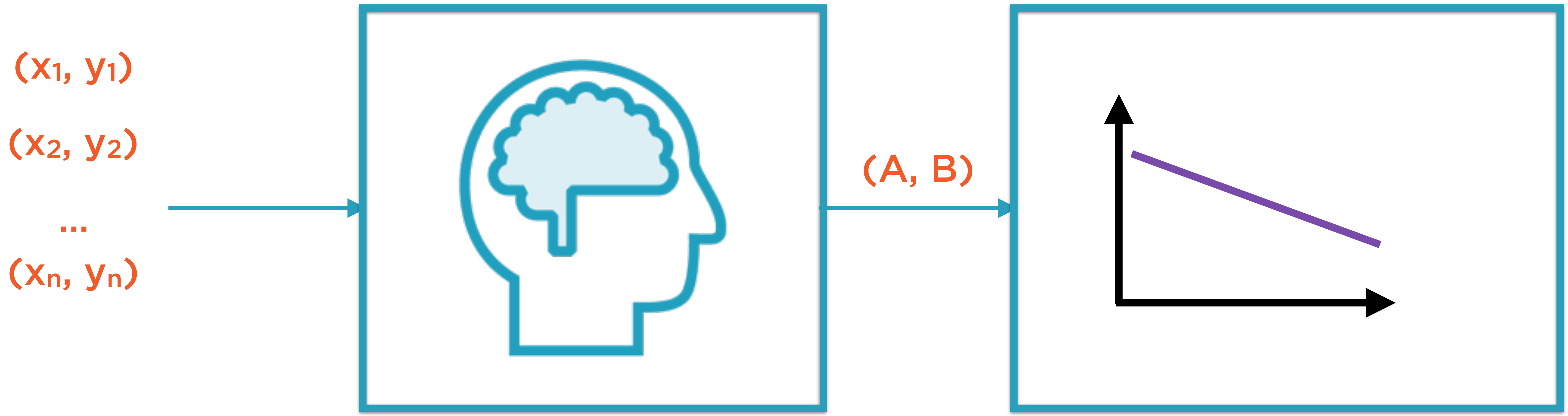


Prices to Returns



Returns = Prices[:-1] / Prices[1:] - 1

ML-based Regression Model

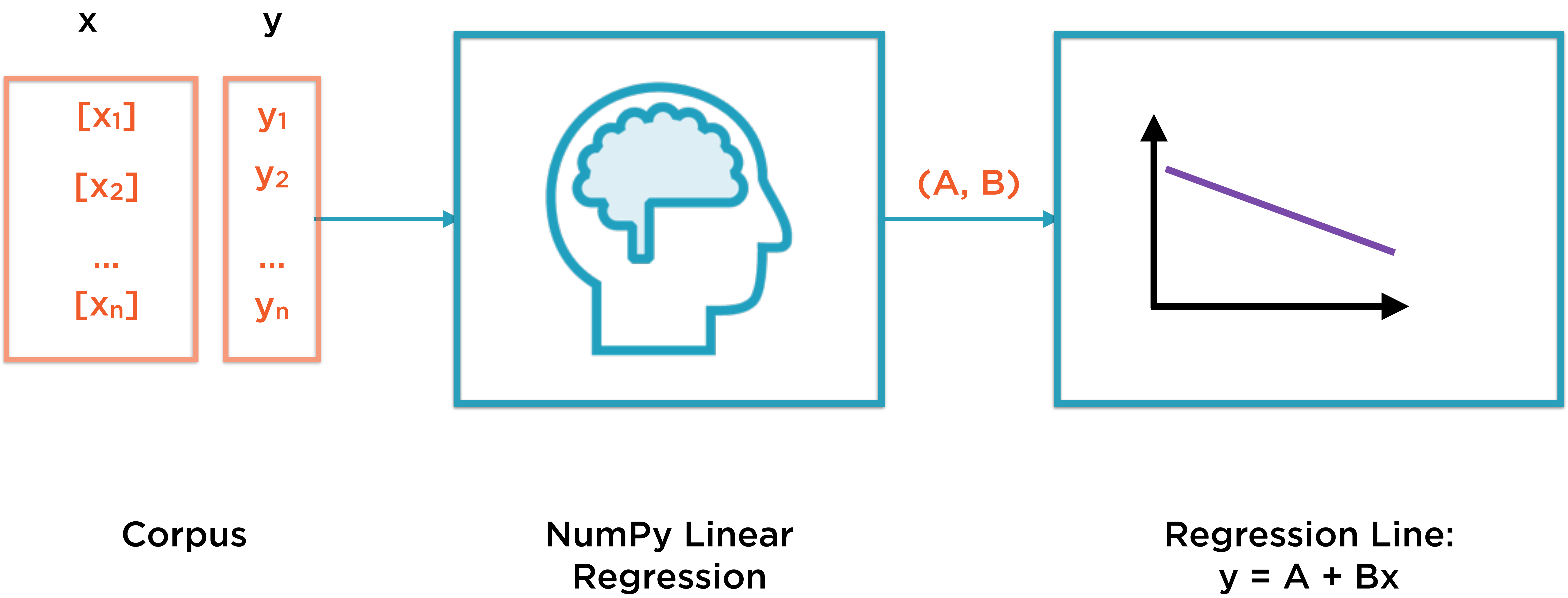


Corpus

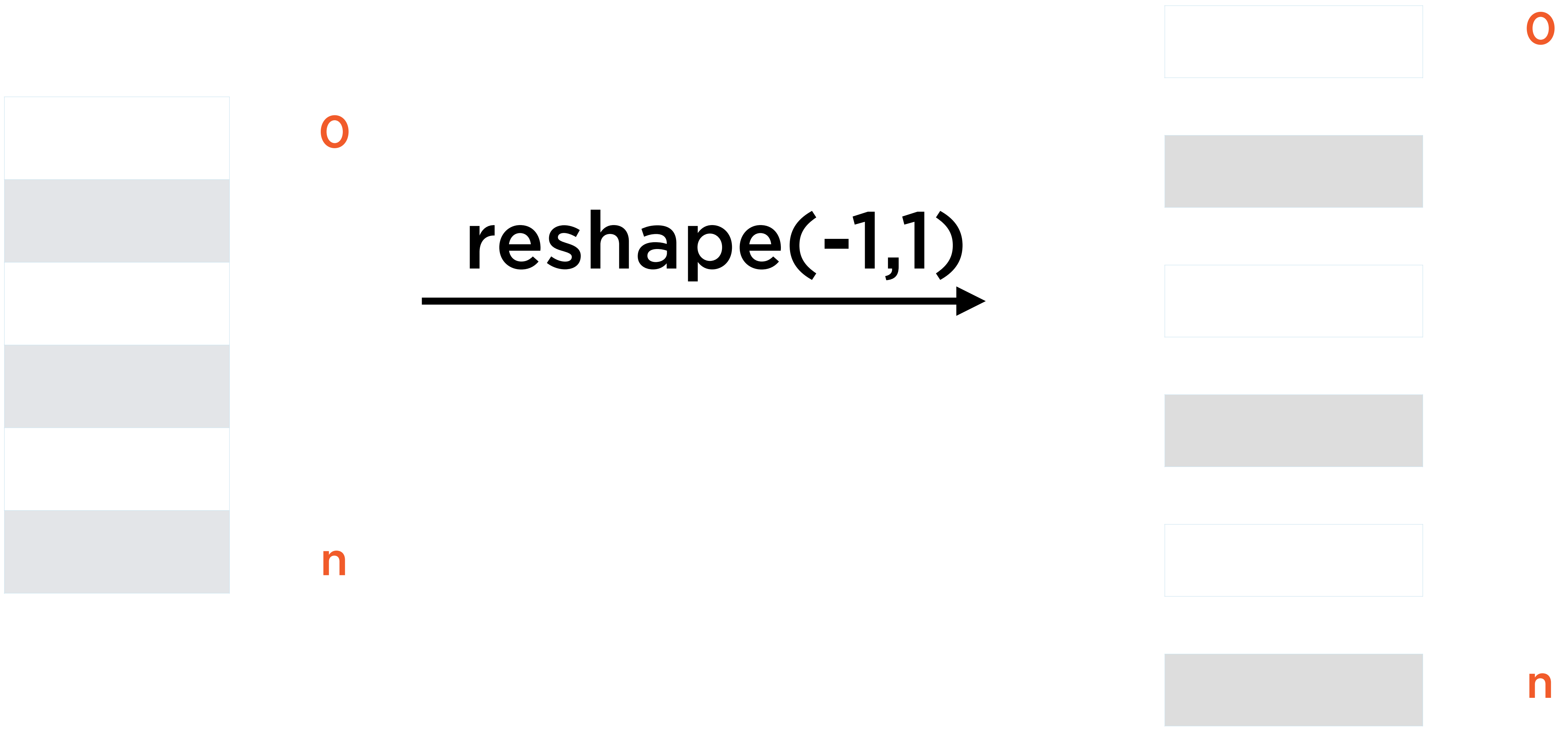
Regression Algorithm

Regression Line: $y = A + Bx$

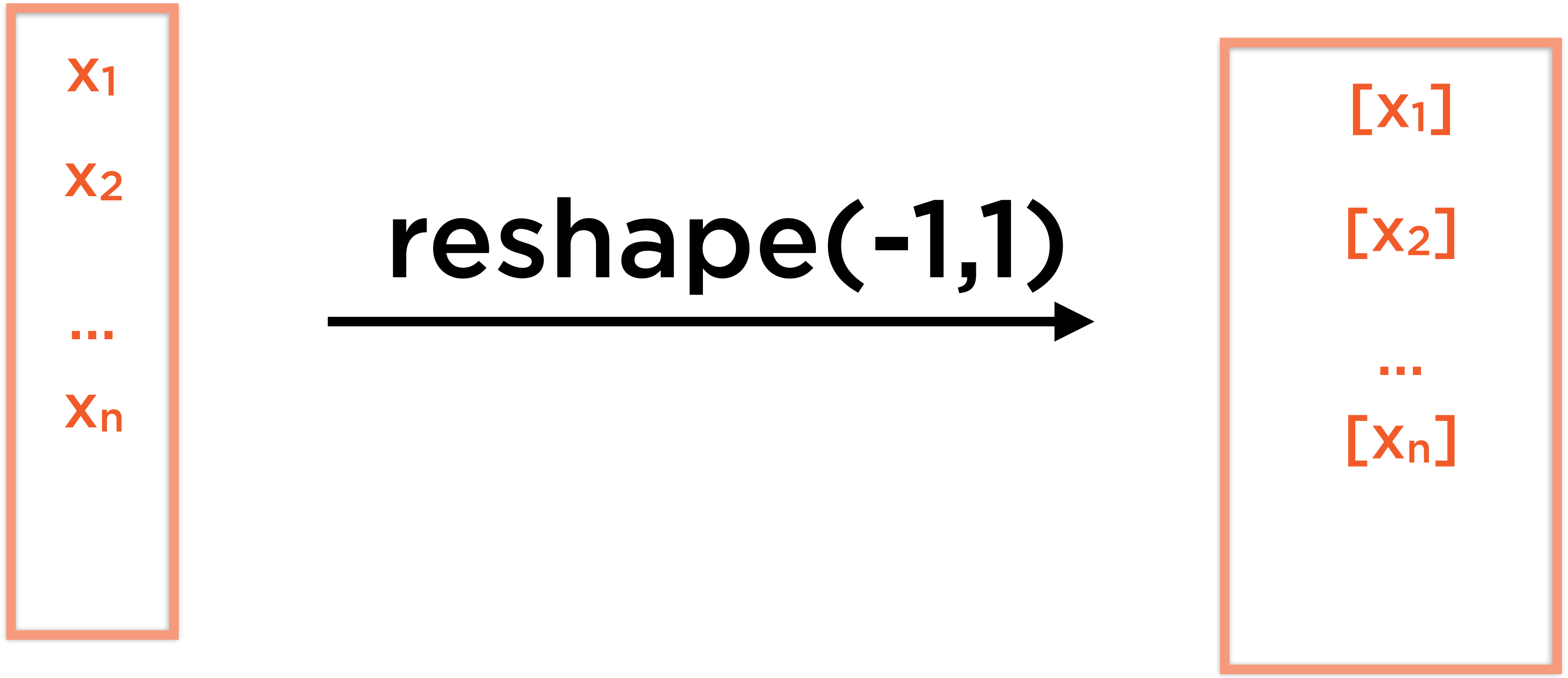
ML-based Regression Model



Reshaping in NumPy



Reshaping in NumPy



Implementing Regression in TensorFlow

Baseline

Non-TensorFlow implementation

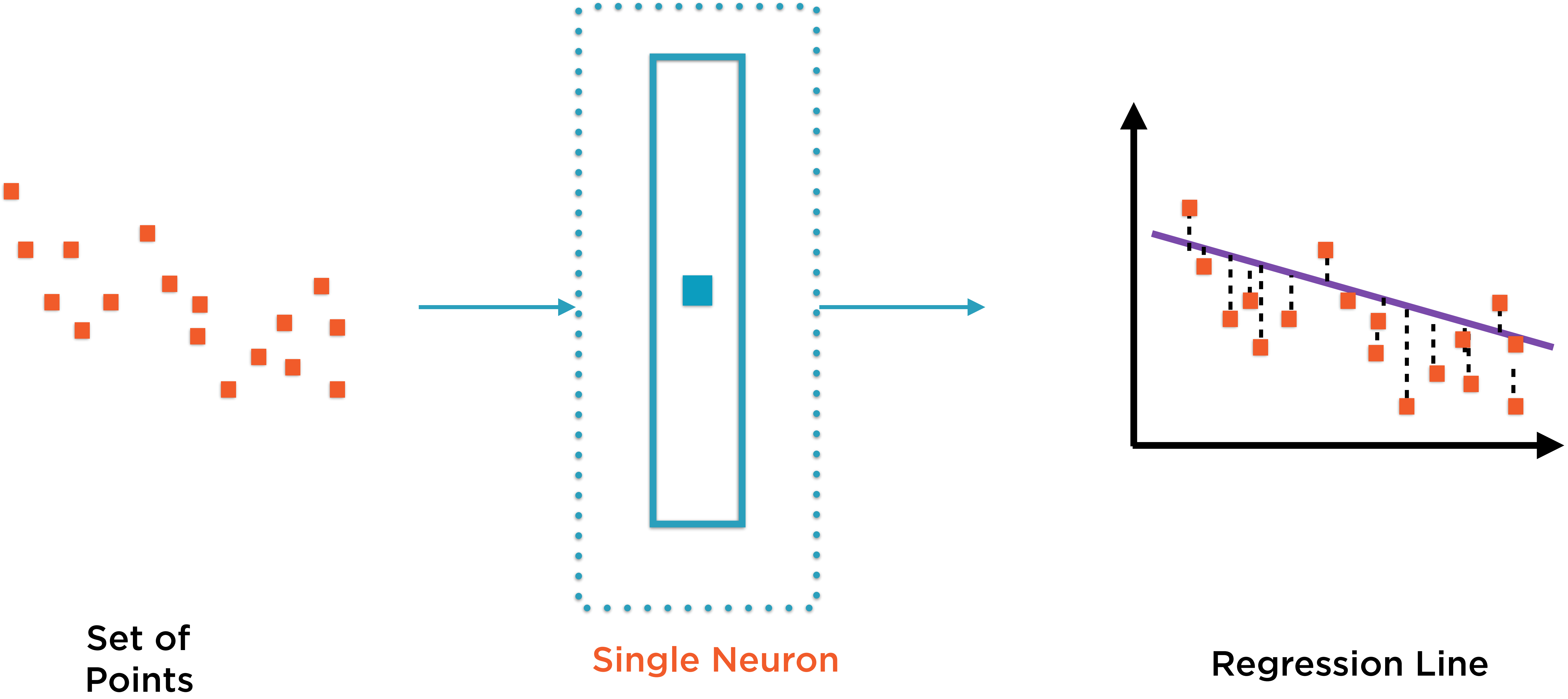
Regular python code

Computation Graph

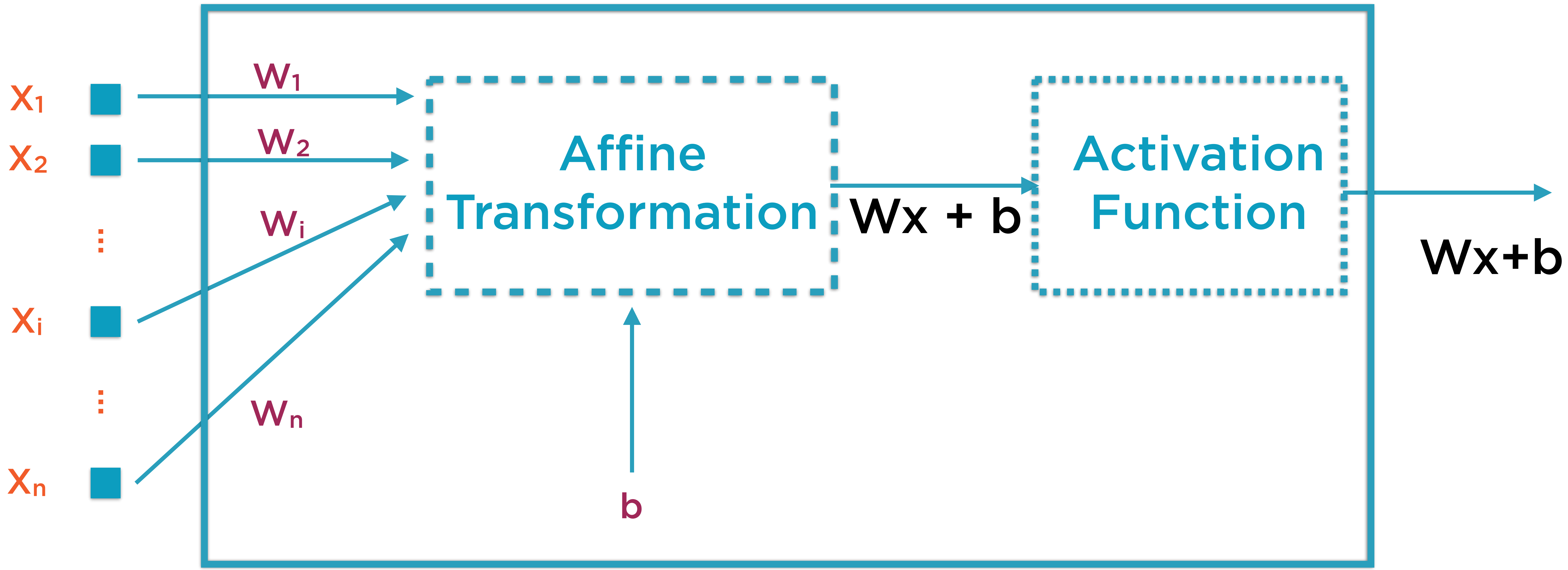
Neural network of 1 neuron

Affine transformation suffices

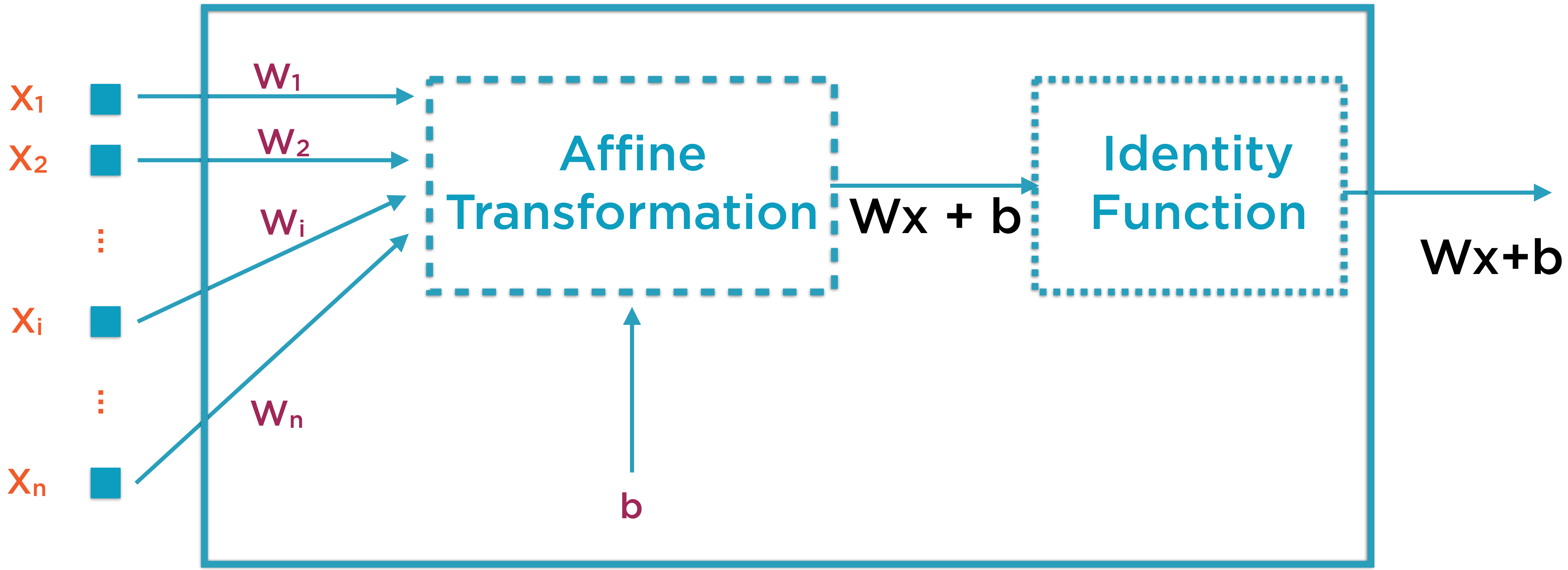
Regression: The Simplest Neural Network



Regression: The Simplest Neural Network



Regression: The Simplest Neural Network



Implementing Regression in TensorFlow

Baseline

Non-TensorFlow implementation

Regular python code

Cost Function

Mean Square Error (MSE)

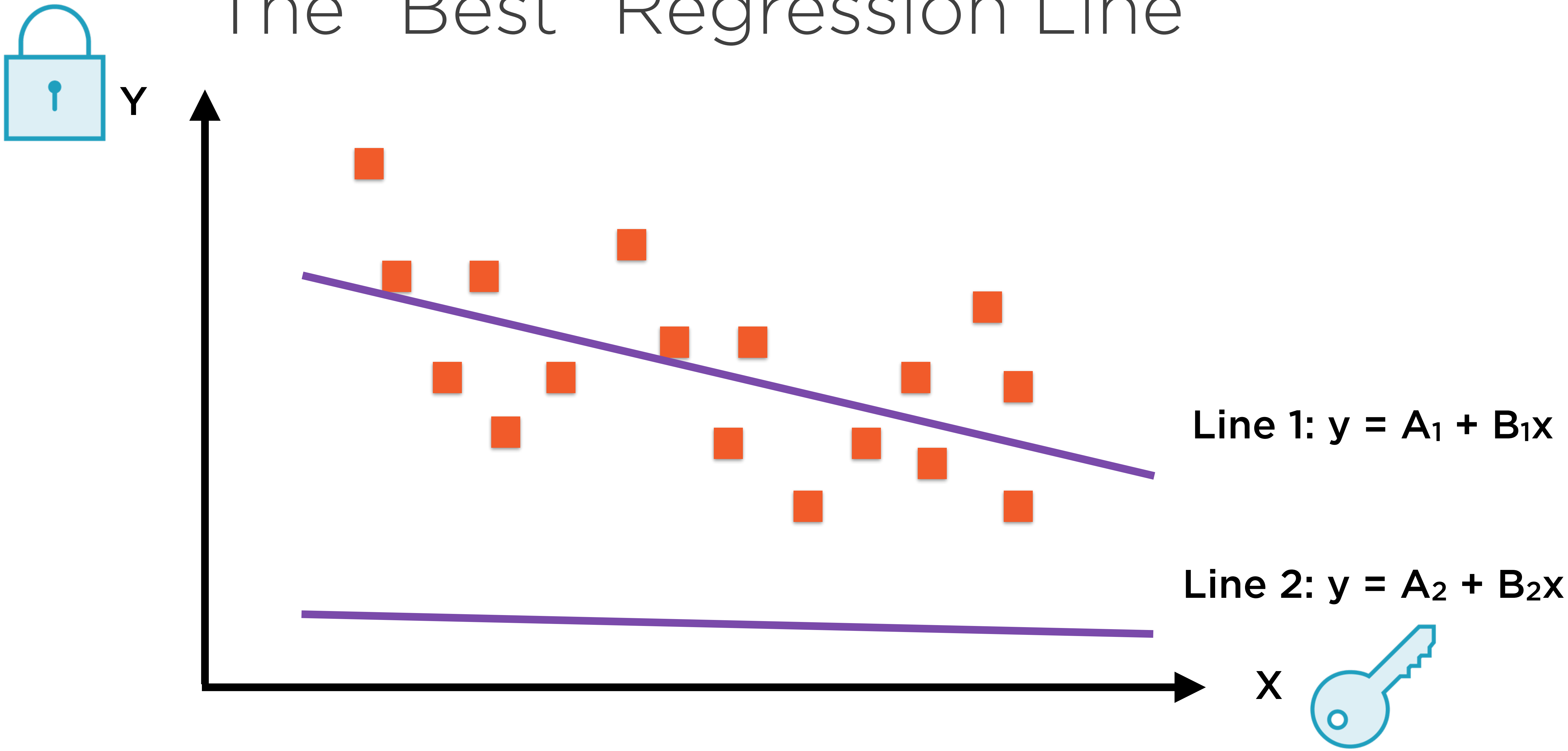
Quantifying goodness-of-fit

Computation Graph

Neural network of 1 neuron

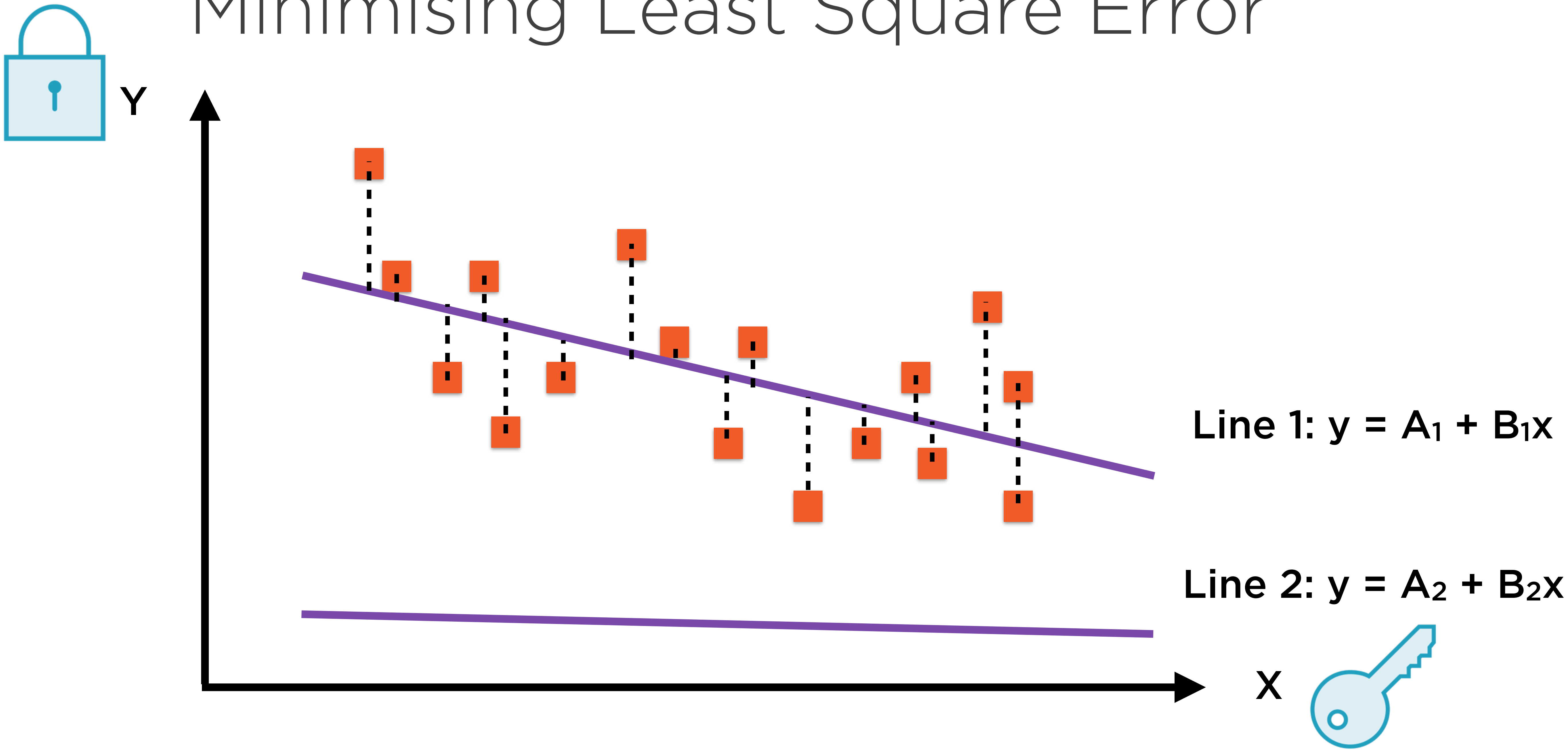
Affine transformation suffices

The "Best" Regression Line



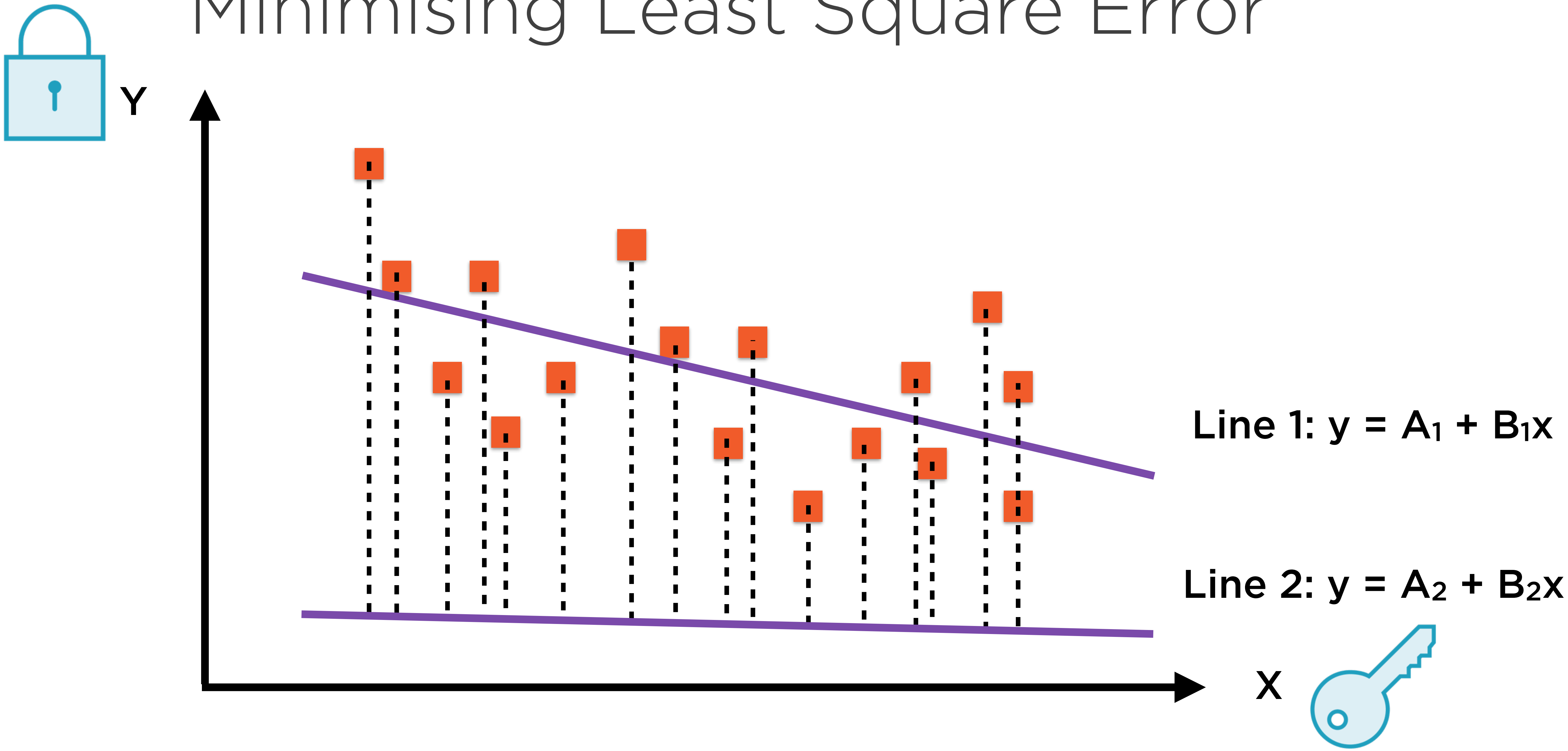
Let's compare two lines, Line 1 and Line 2

Minimising Least Square Error



Drop vertical lines from each point to the lines A and B

Minimising Least Square Error



Drop vertical lines from each point to the lines A and B

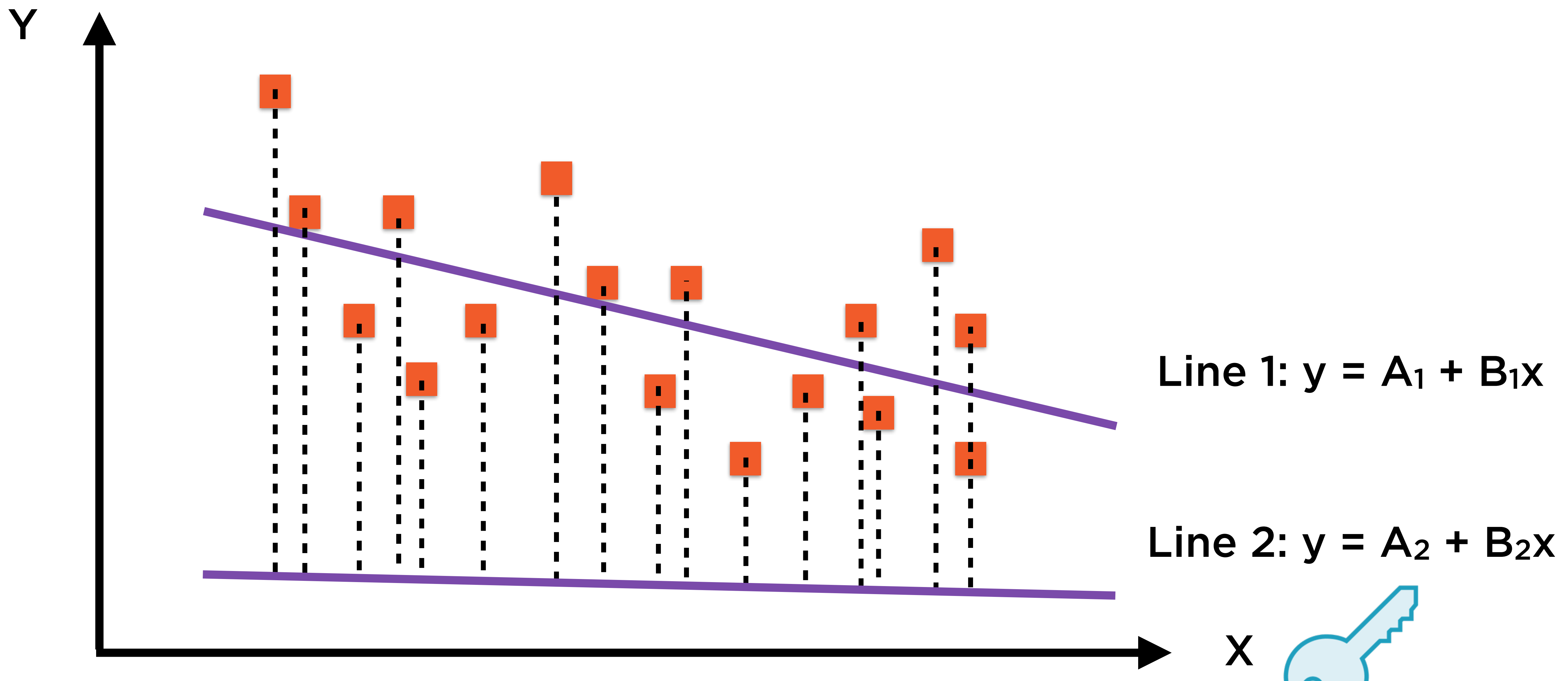
Simple Regression

Regression Equation:

$$y = A + Bx$$

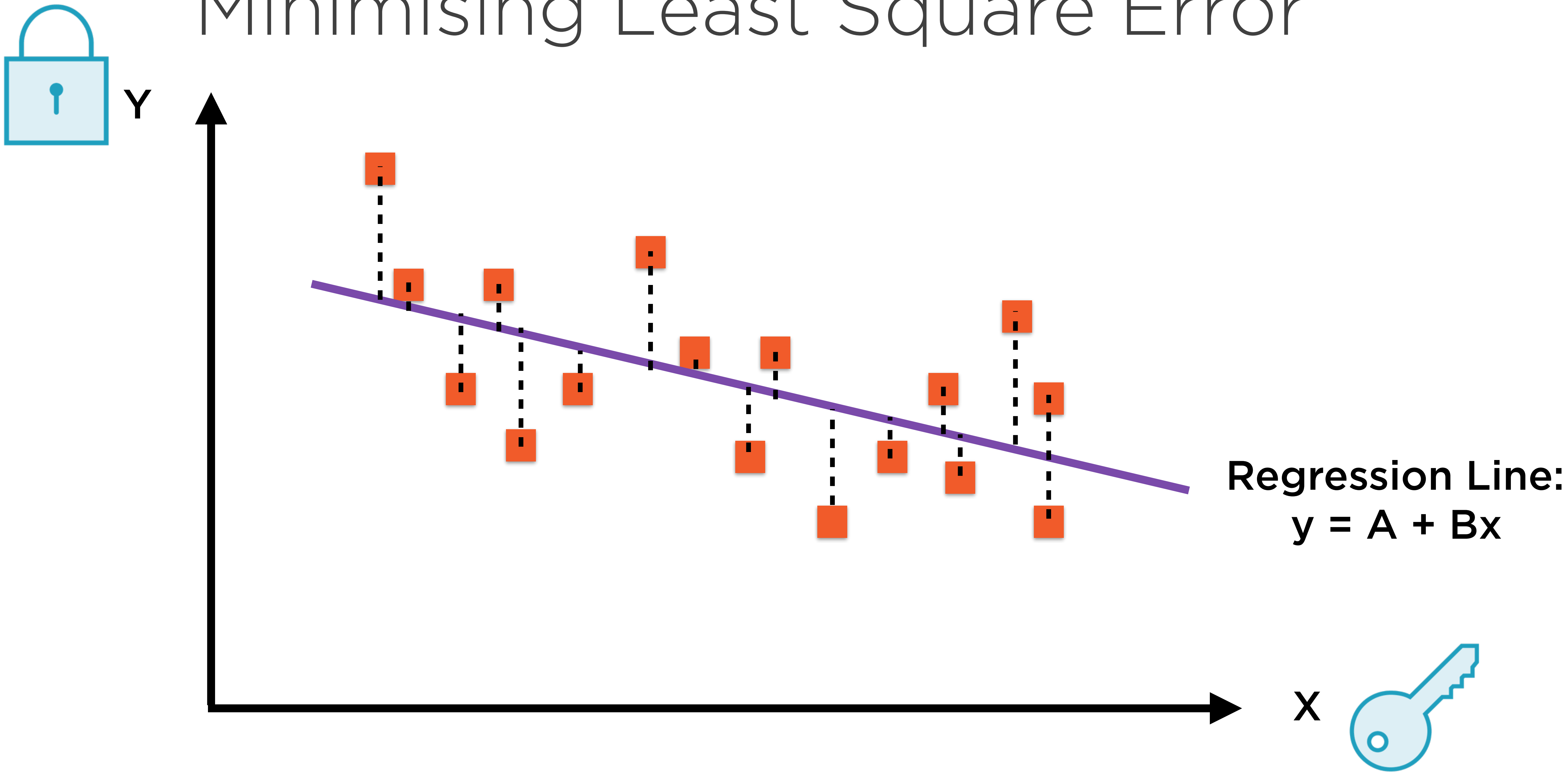
$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \dots \\ y_n \end{bmatrix} = A \begin{bmatrix} 1 \\ 1 \\ 1 \\ \dots \\ 1 \end{bmatrix} + B \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \dots \\ x_n \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \dots \\ e_n \end{bmatrix}$$

Minimising Least Square Error



The "best fit" line is the one where the sum of the squares of the lengths of these dotted lines is minimum

Minimising Least Square Error



The “best fit” line is called the regression line

Implementing Regression in TensorFlow

Baseline

Non-TensorFlow implementation

Regular python code

Cost Function

Mean Square Error (MSE)

Quantifying goodness-of-fit

Computation Graph

Neural network of 1 neuron

Affine transformation suffices

Optimizer

Gradient Descent optimizers

Improving goodness-of-fit

Why Choosing Is Complicated

What do we really want to achieve?

What is slowing us down?

What do we really control?

Choosing involves answering complicated questions

Why Optimization Helps

What do we really want to achieve?

What is slowing us down?

What do we really control?

Optimization forces us to mathematically pin down answers to these questions

Framing the Optimization Problem

Objective Function

What we would like to achieve

Constraints

What slows us down

Decision Variables

What we really control

Collectively, these answers constitute the optimization problem

Linear Regression as an Optimization Problem

Objective Function

Minimize variance of the residuals (MSE)

Constraints

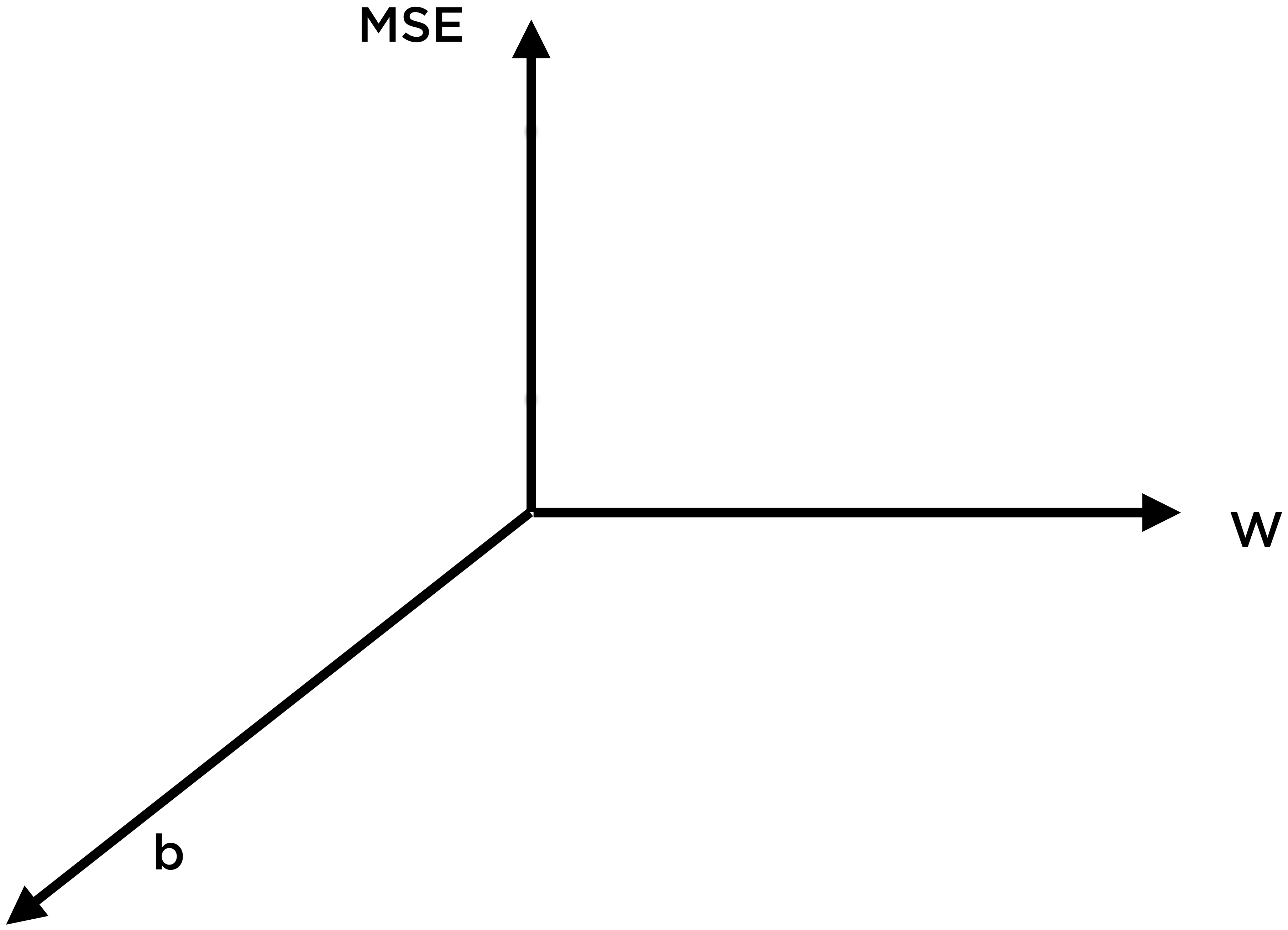
Express relationship as a straight line

$y = Wx + b$

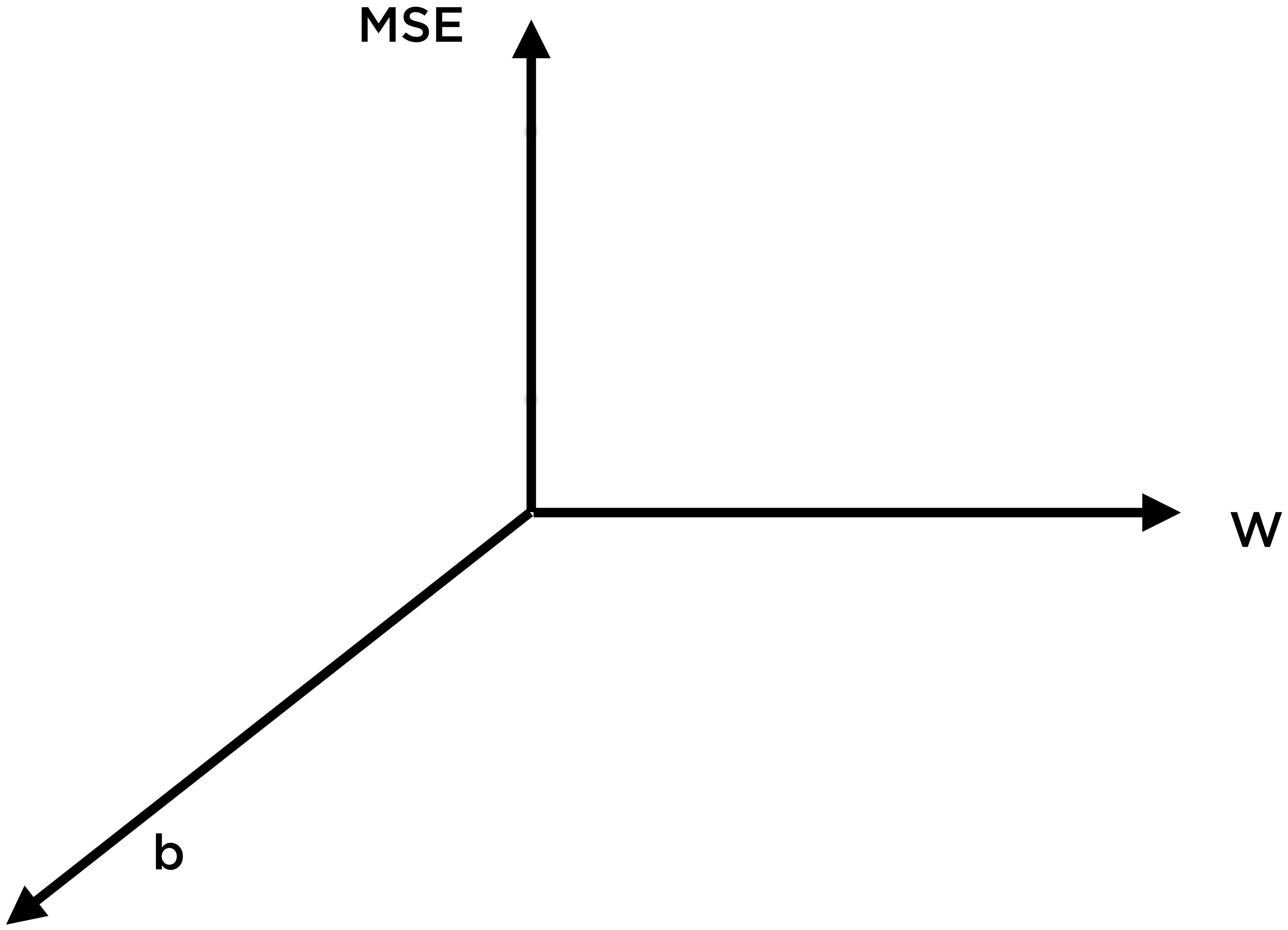
Decision Variables

Values of W and b

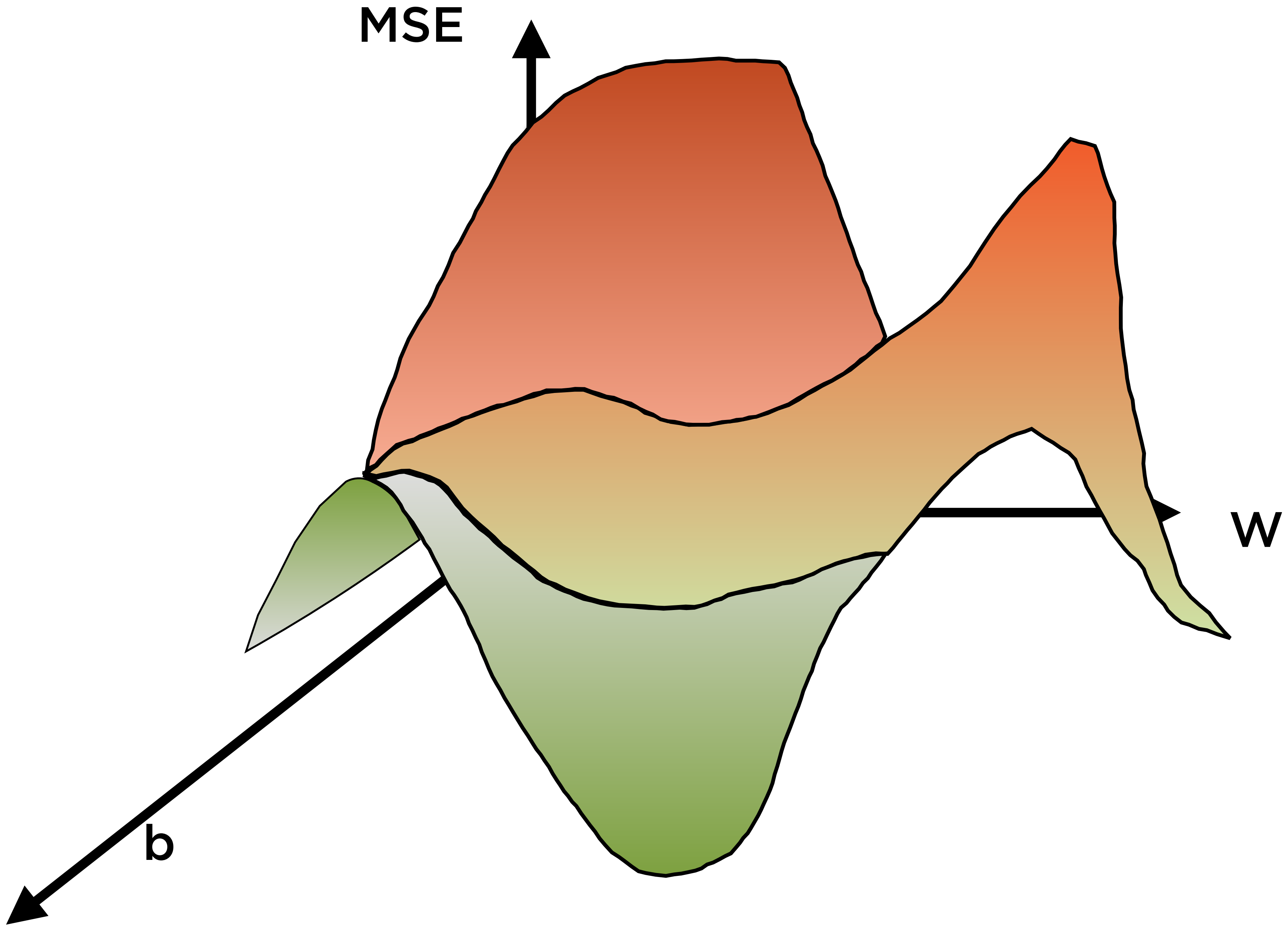
Minimizing MSE



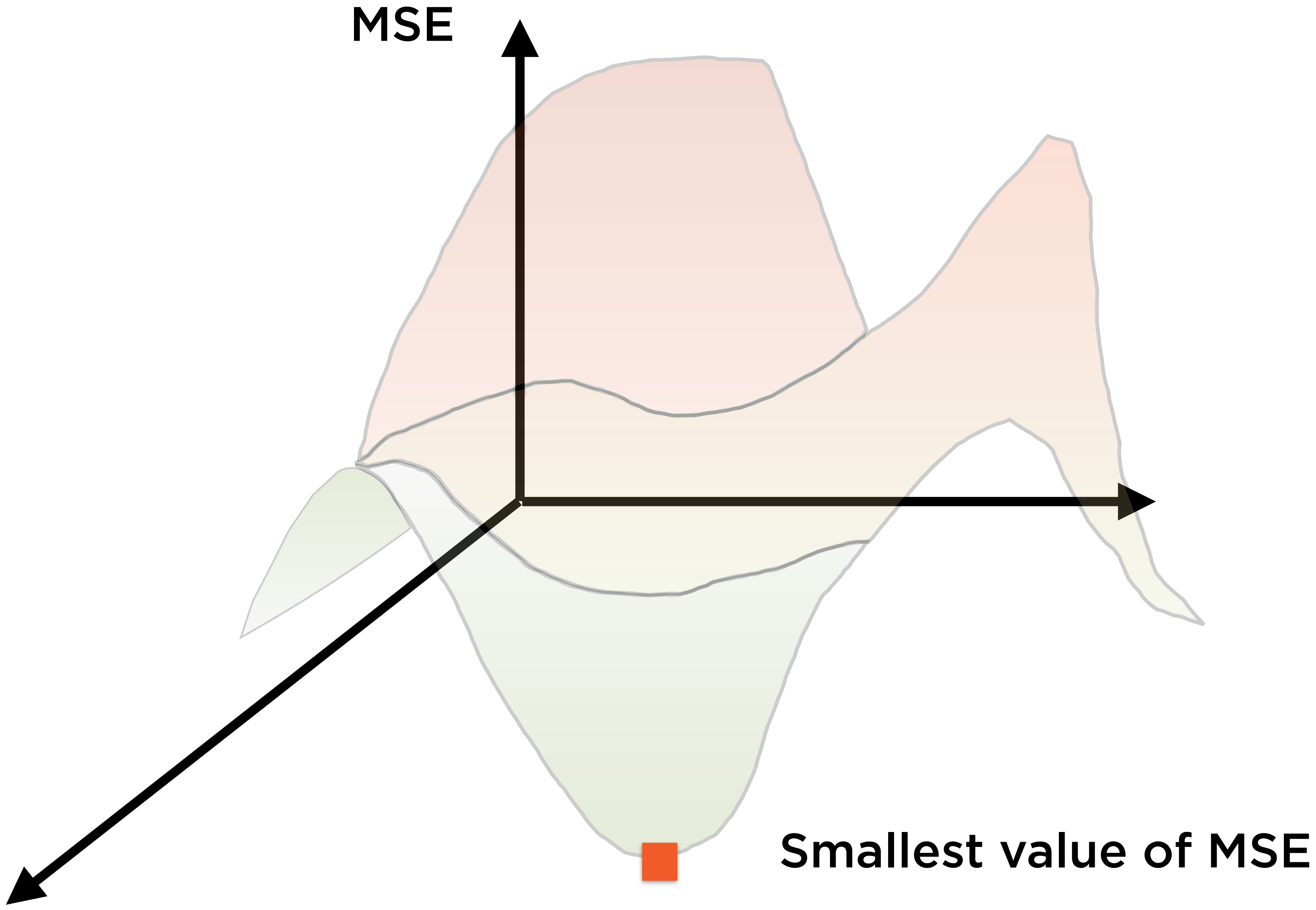
Minimizing MSE



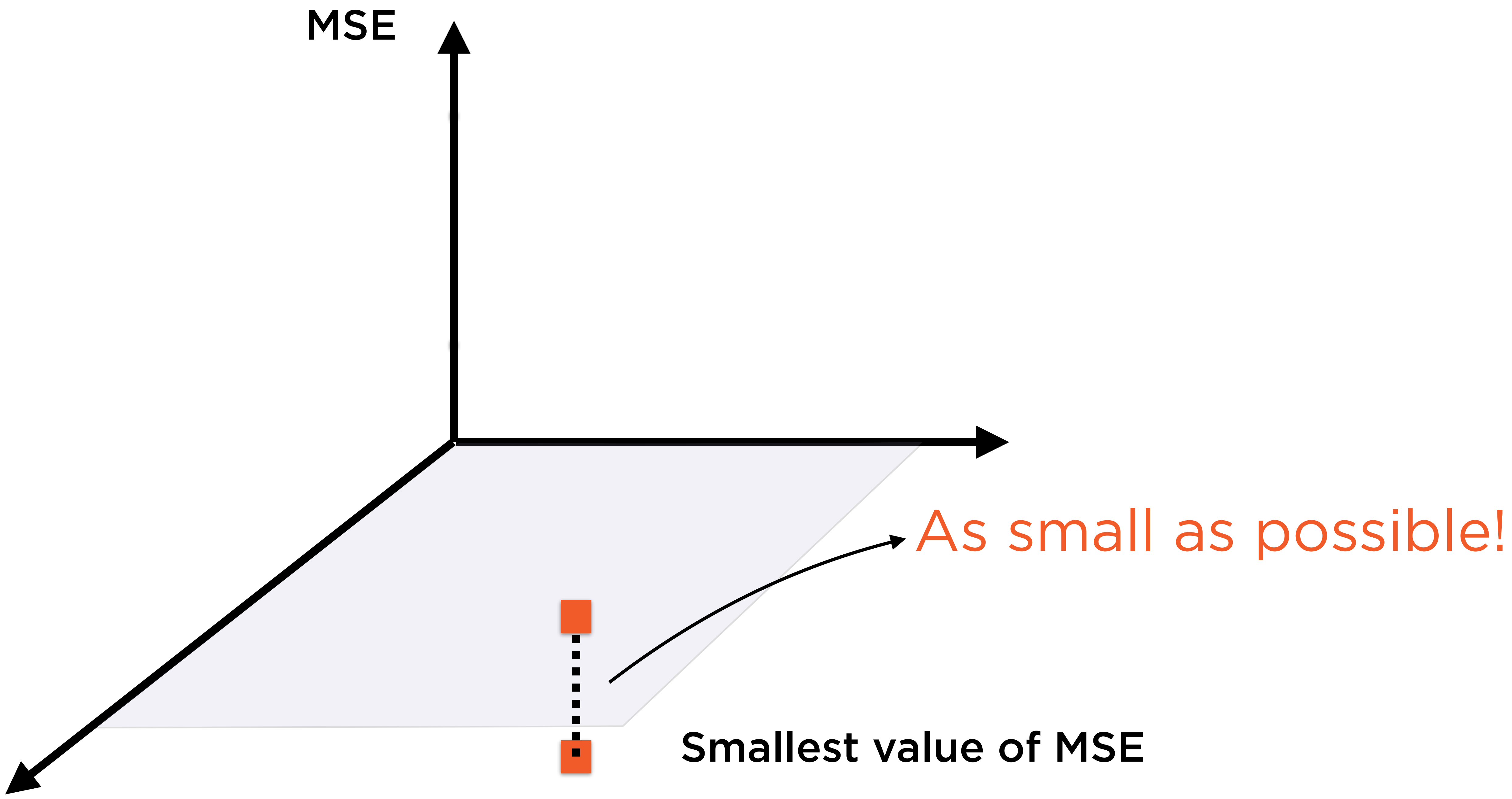
Minimizing MSE



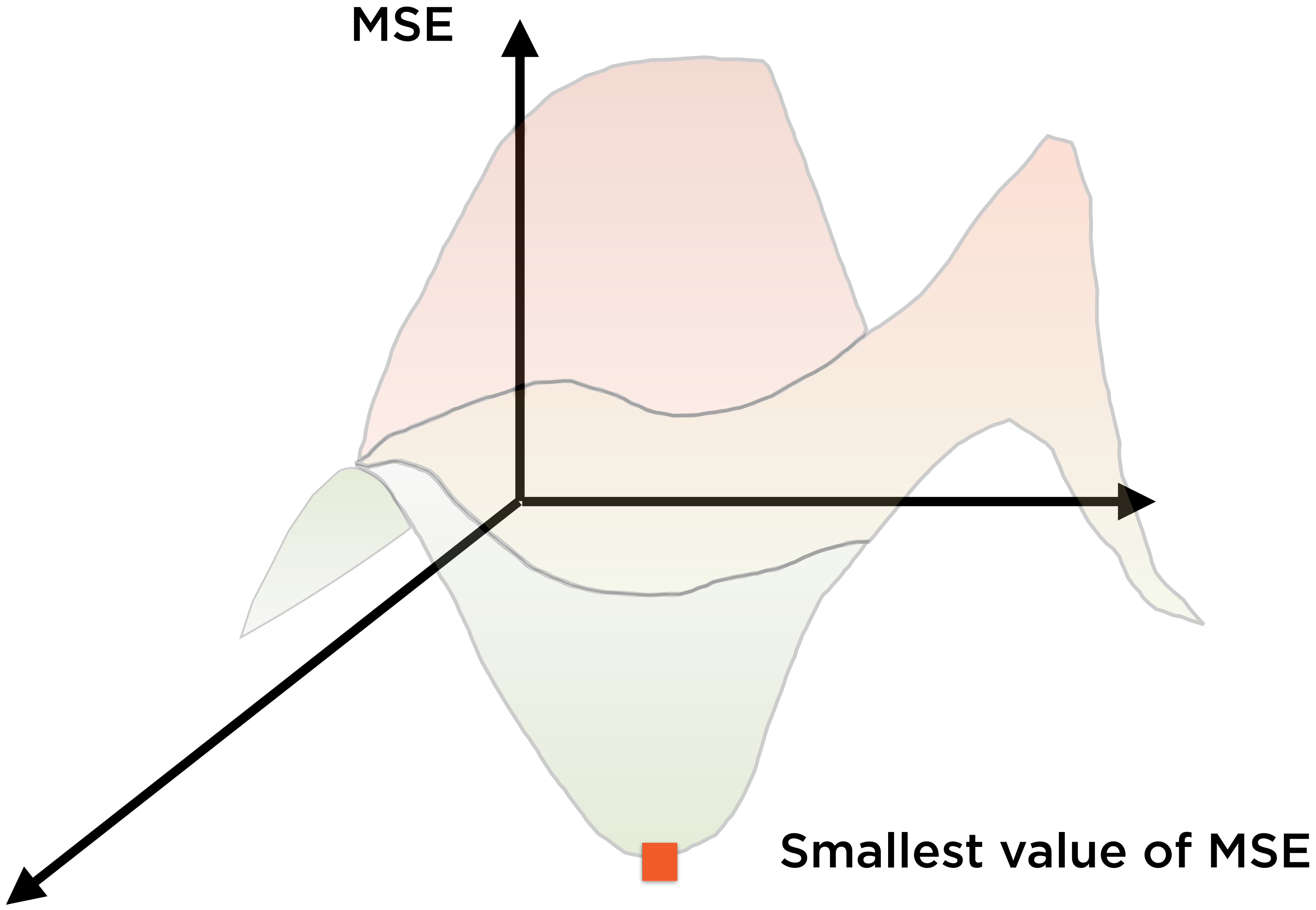
Minimizing MSE



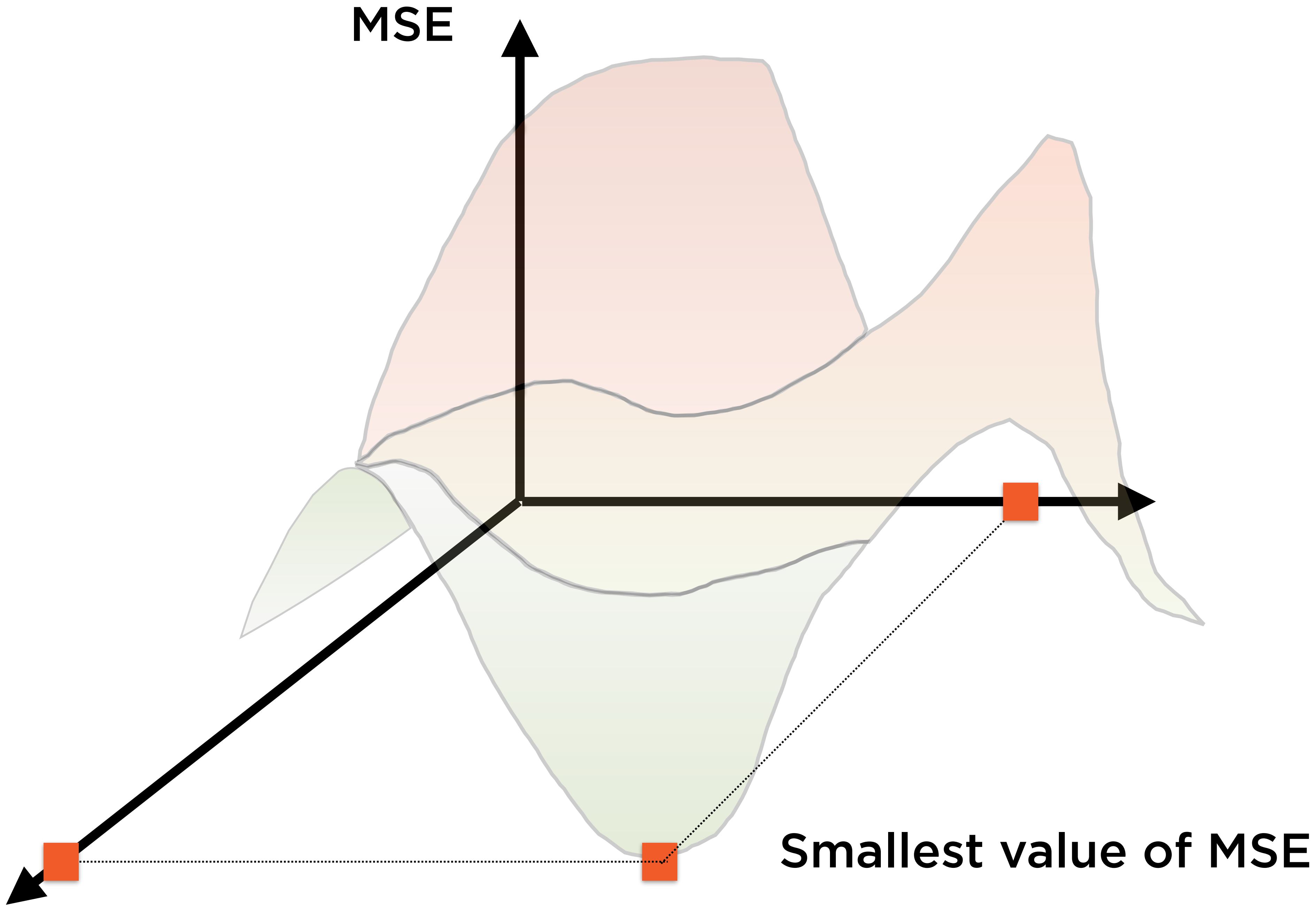
Minimizing MSE



Minimizing MSE

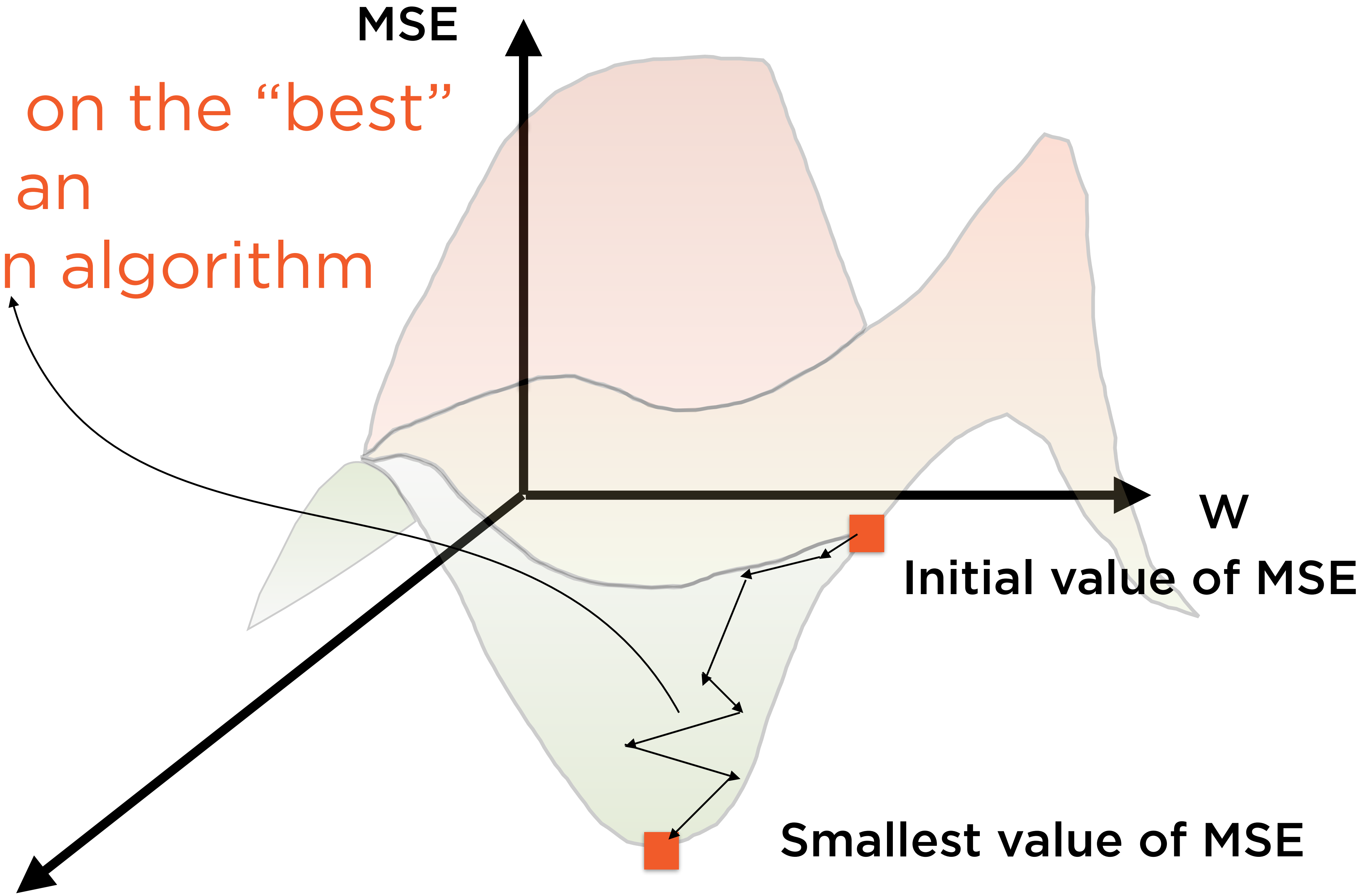


Minimizing MSE

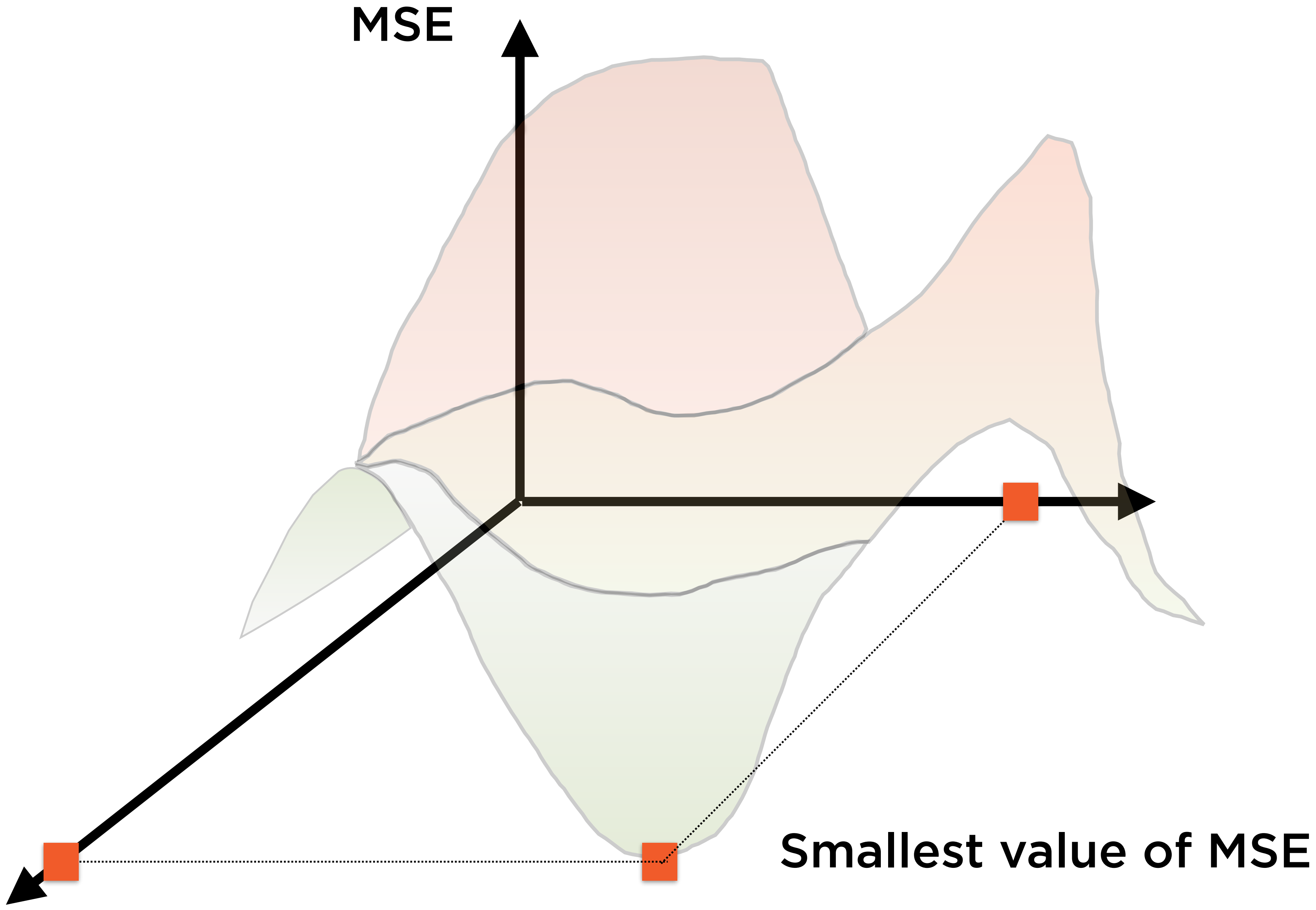


“Gradient Descent”

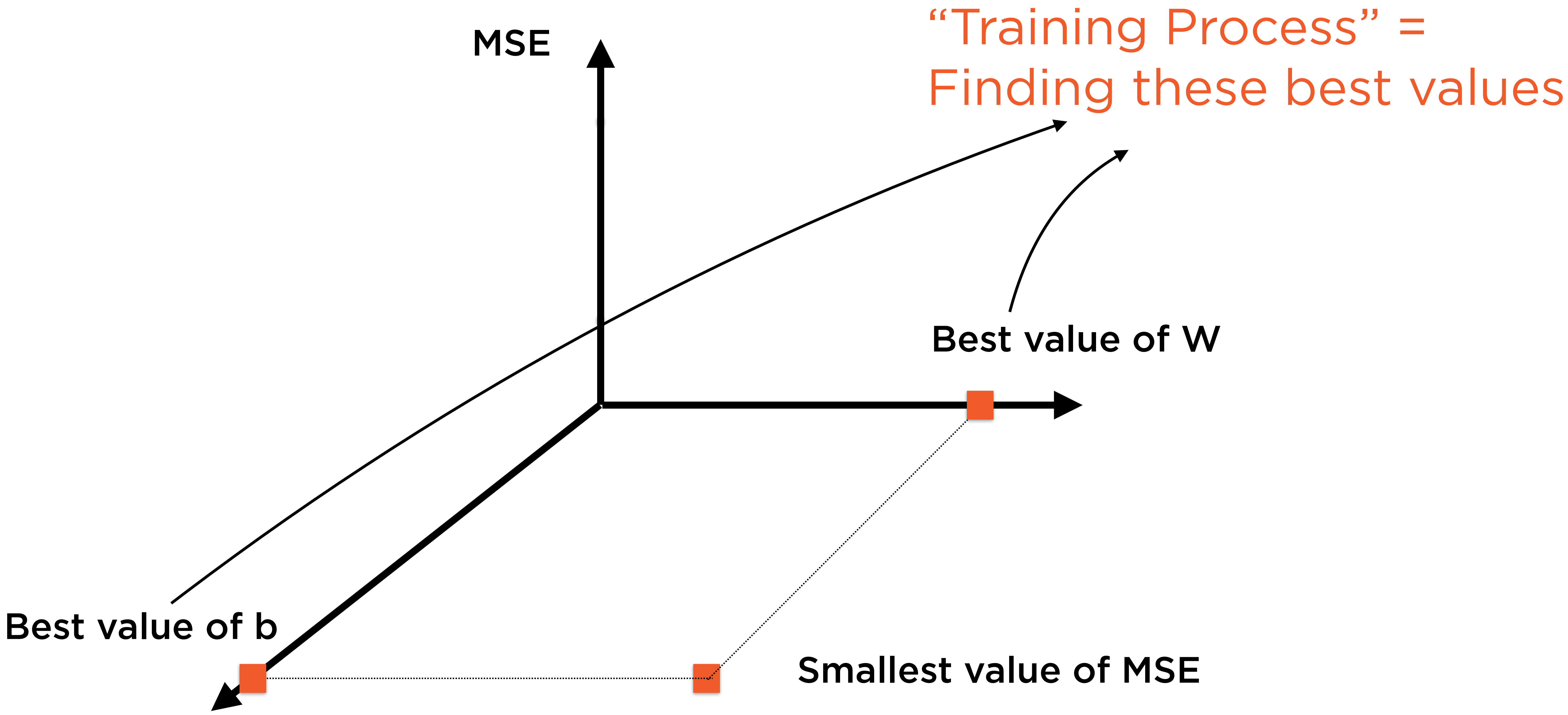
Converging on the “best” value using an optimization algorithm



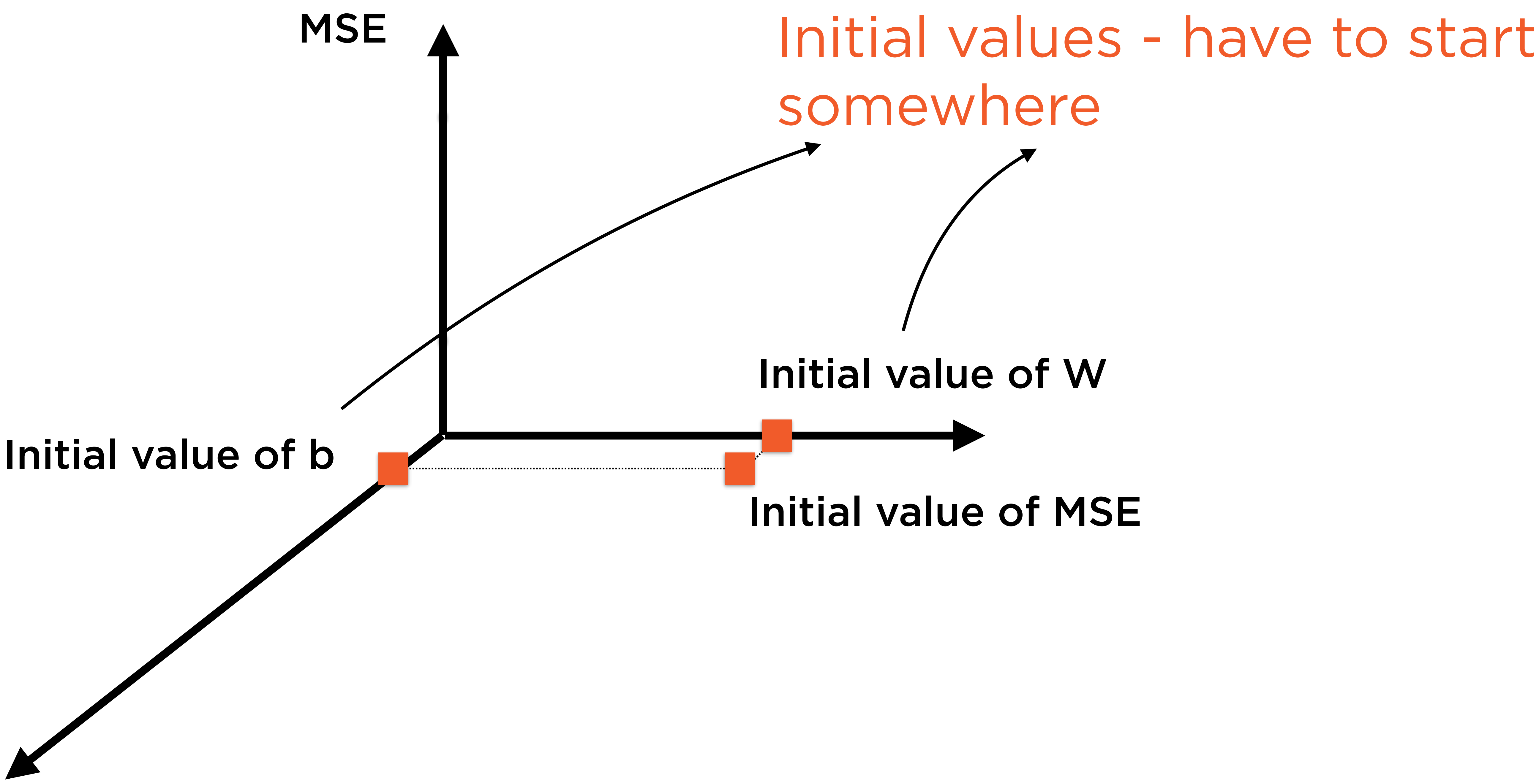
Minimizing MSE



“Training” the Algorithm

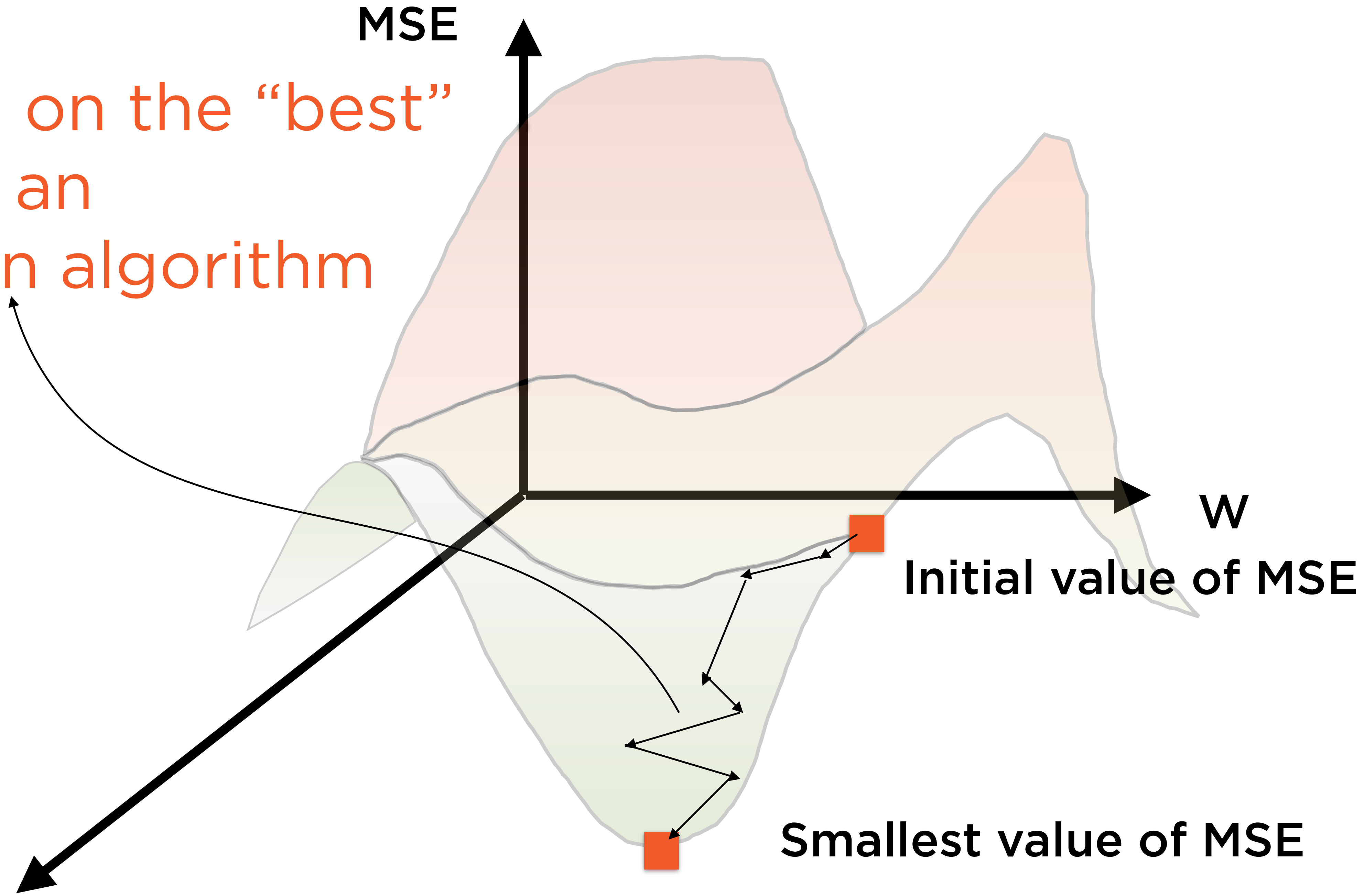


Start Somewhere



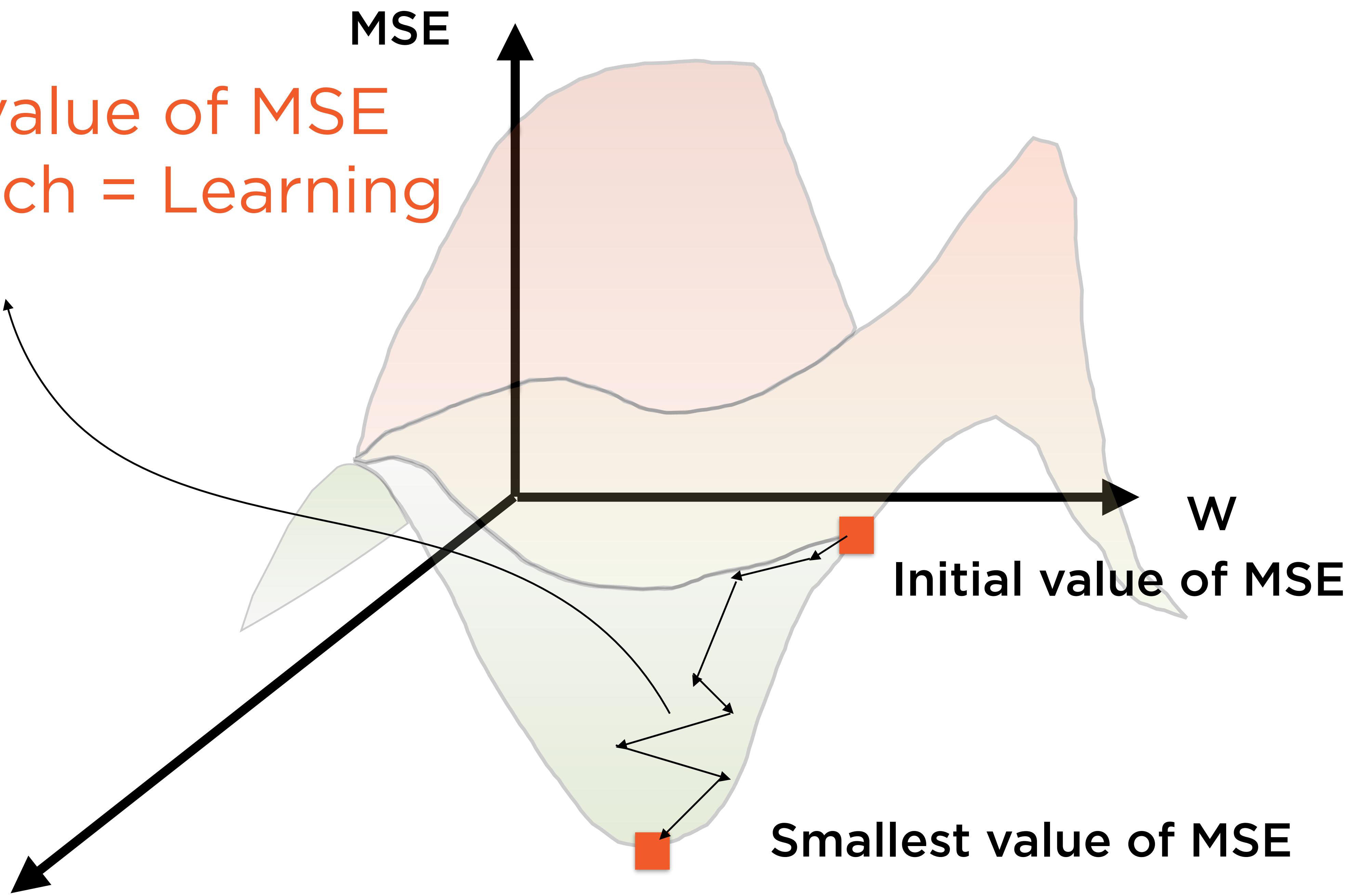
“Gradient Descent”

Converging on the “best” value using an optimization algorithm



“Learning Rate”

Change in value of MSE in each epoch = Learning Rate



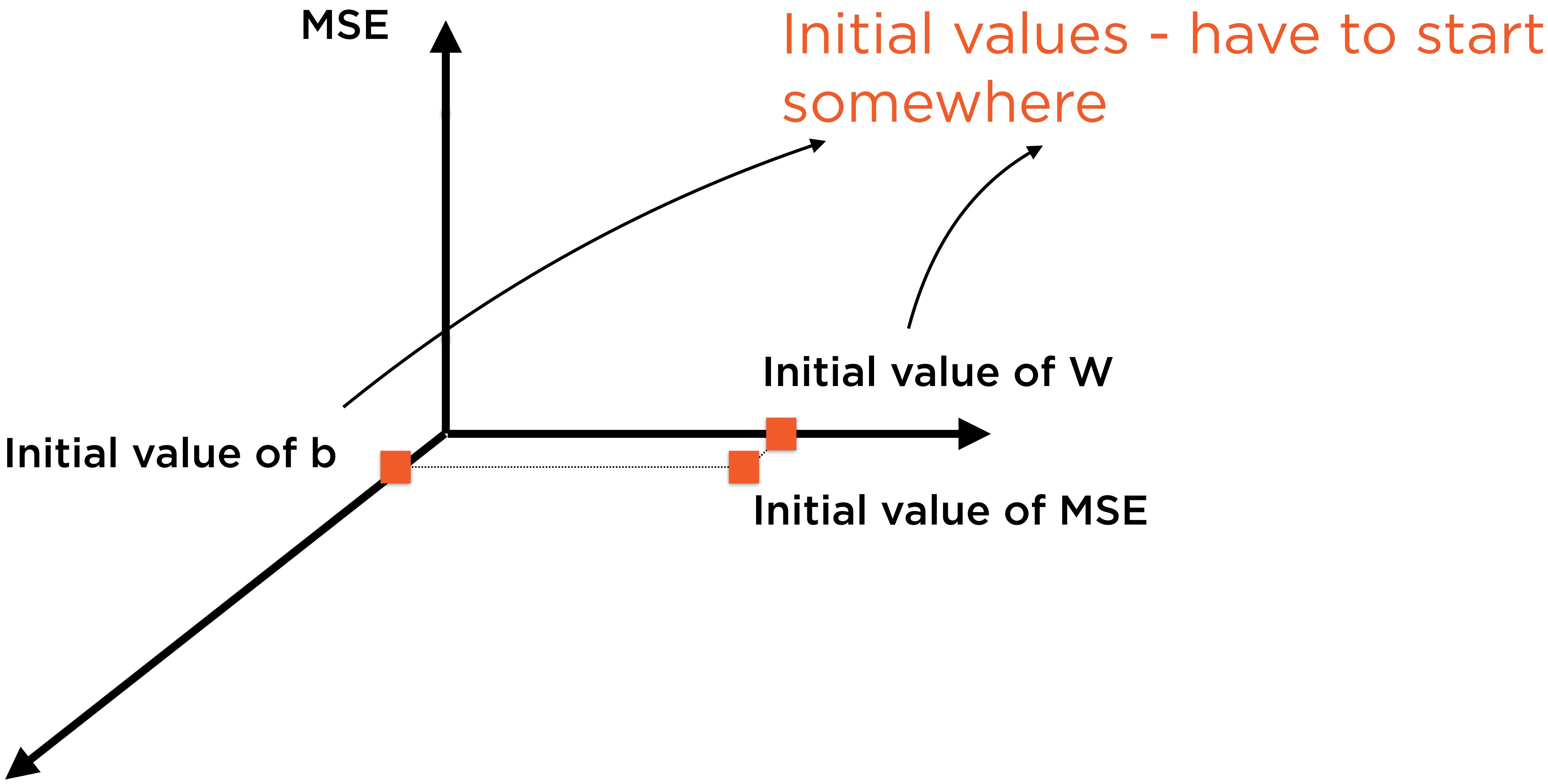
Gradient Descent Optimizers

tf.train.GradientDescentOptimizer

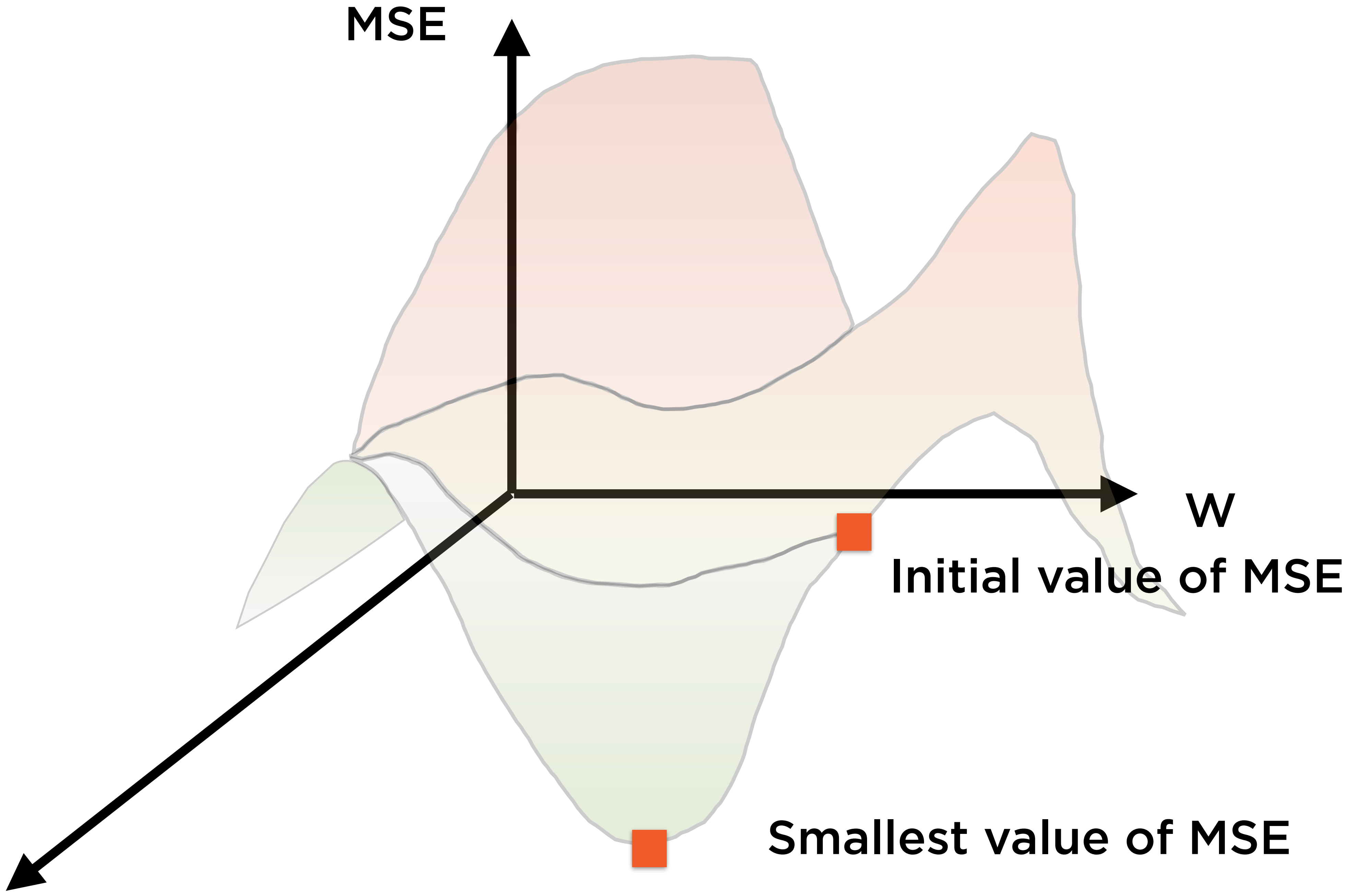
tf.train.AdamOptimizer

tf.train.FtrlOptimizer

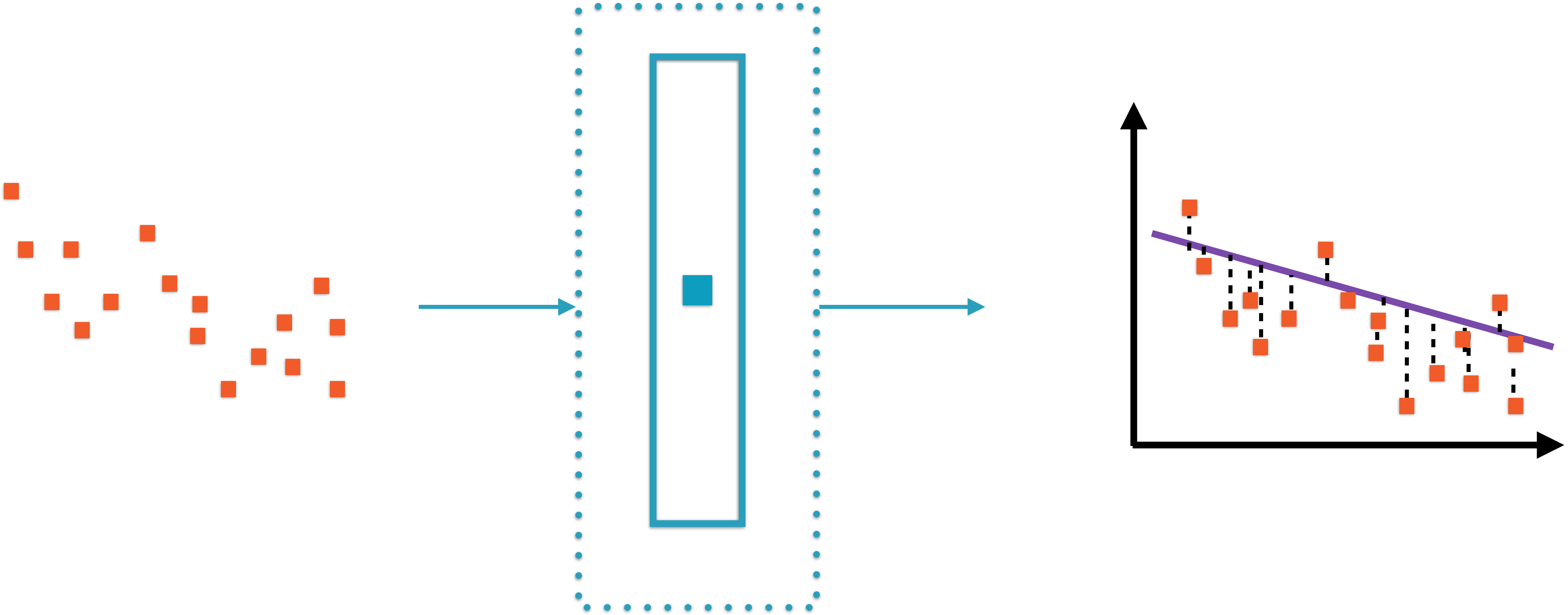
Start Somewhere



Minimizing MSE



Regression: The Simplest Neural Network

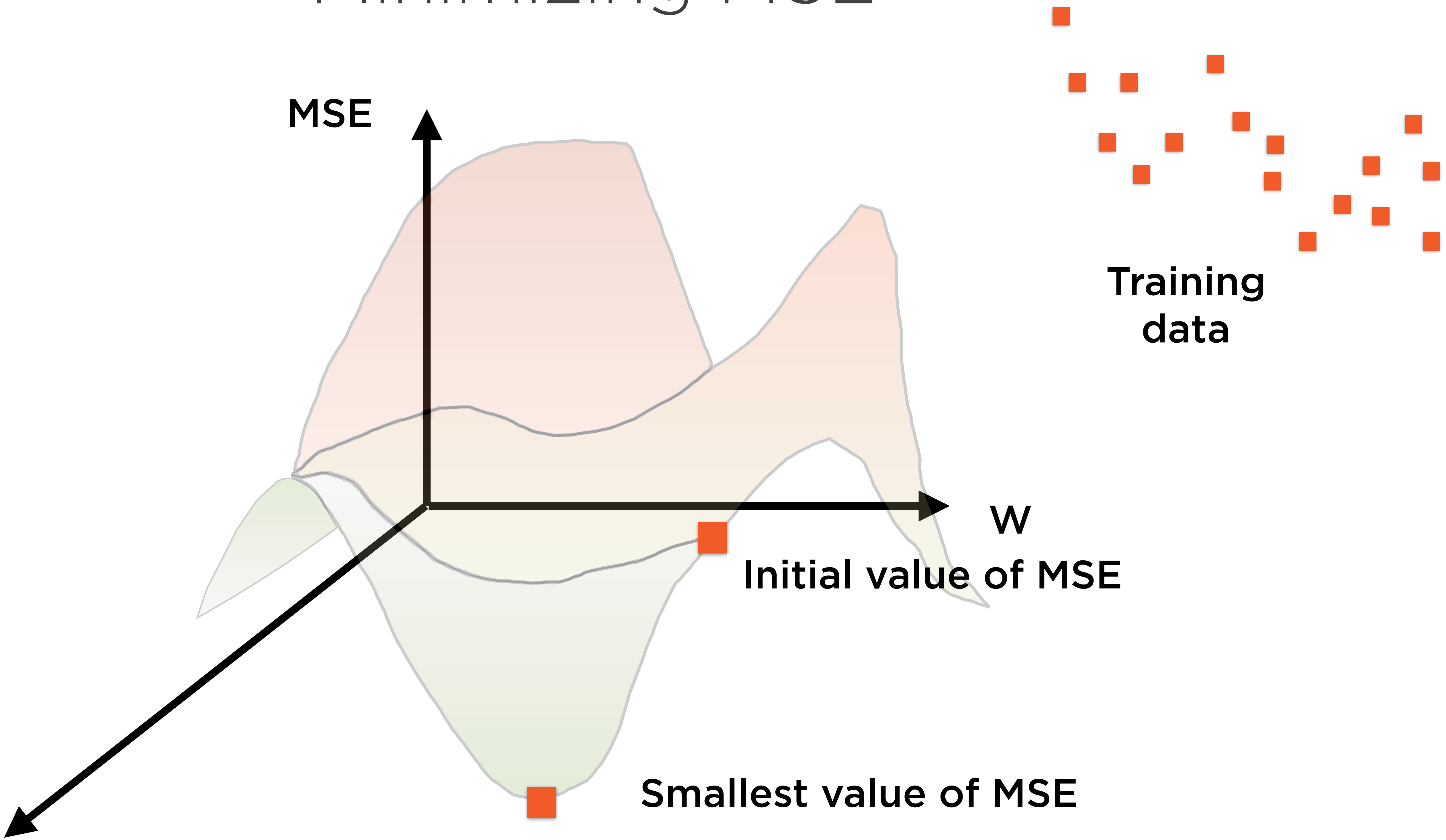


Training data

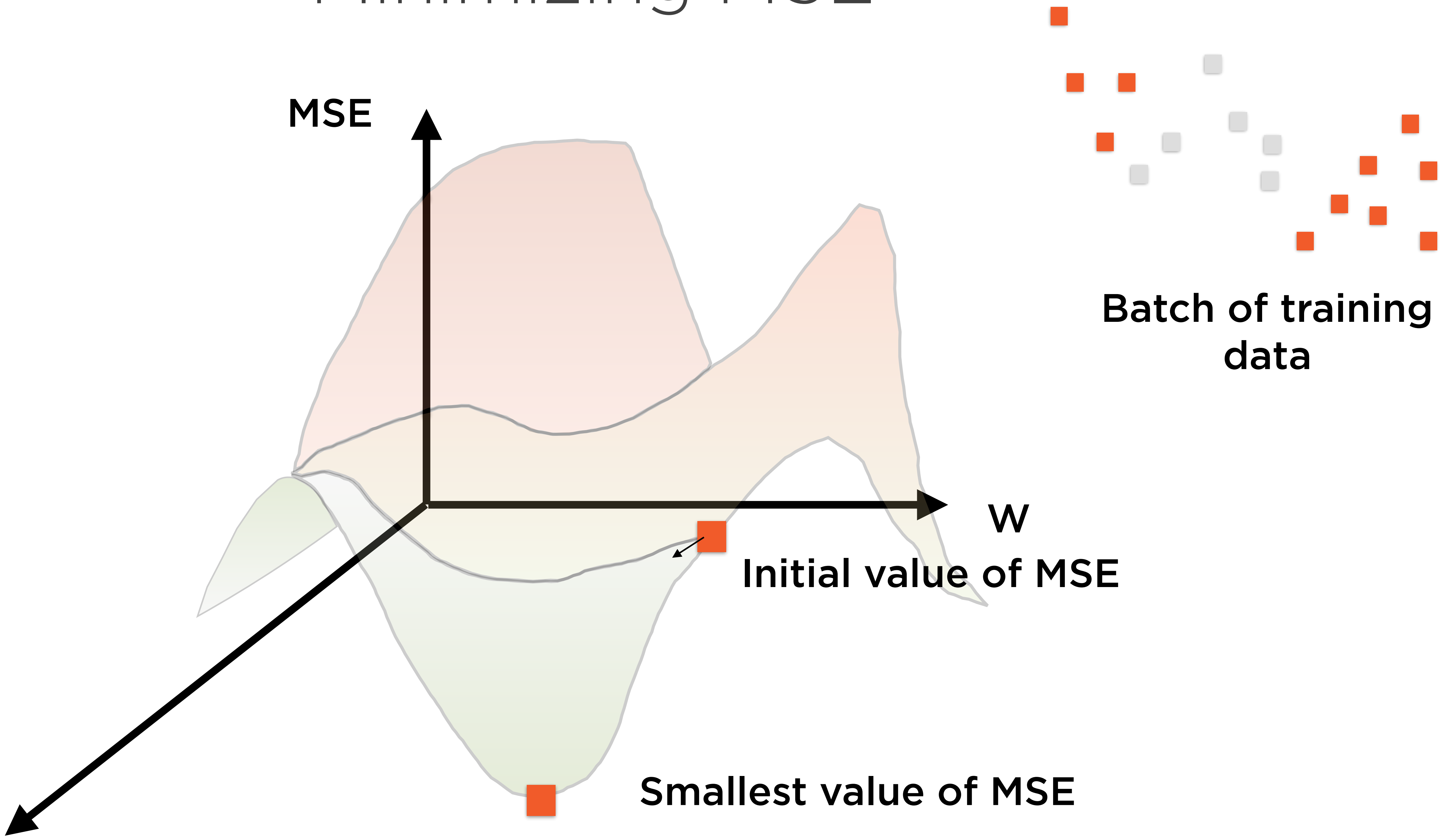
Single Neuron

Regression Line

Minimizing MSE

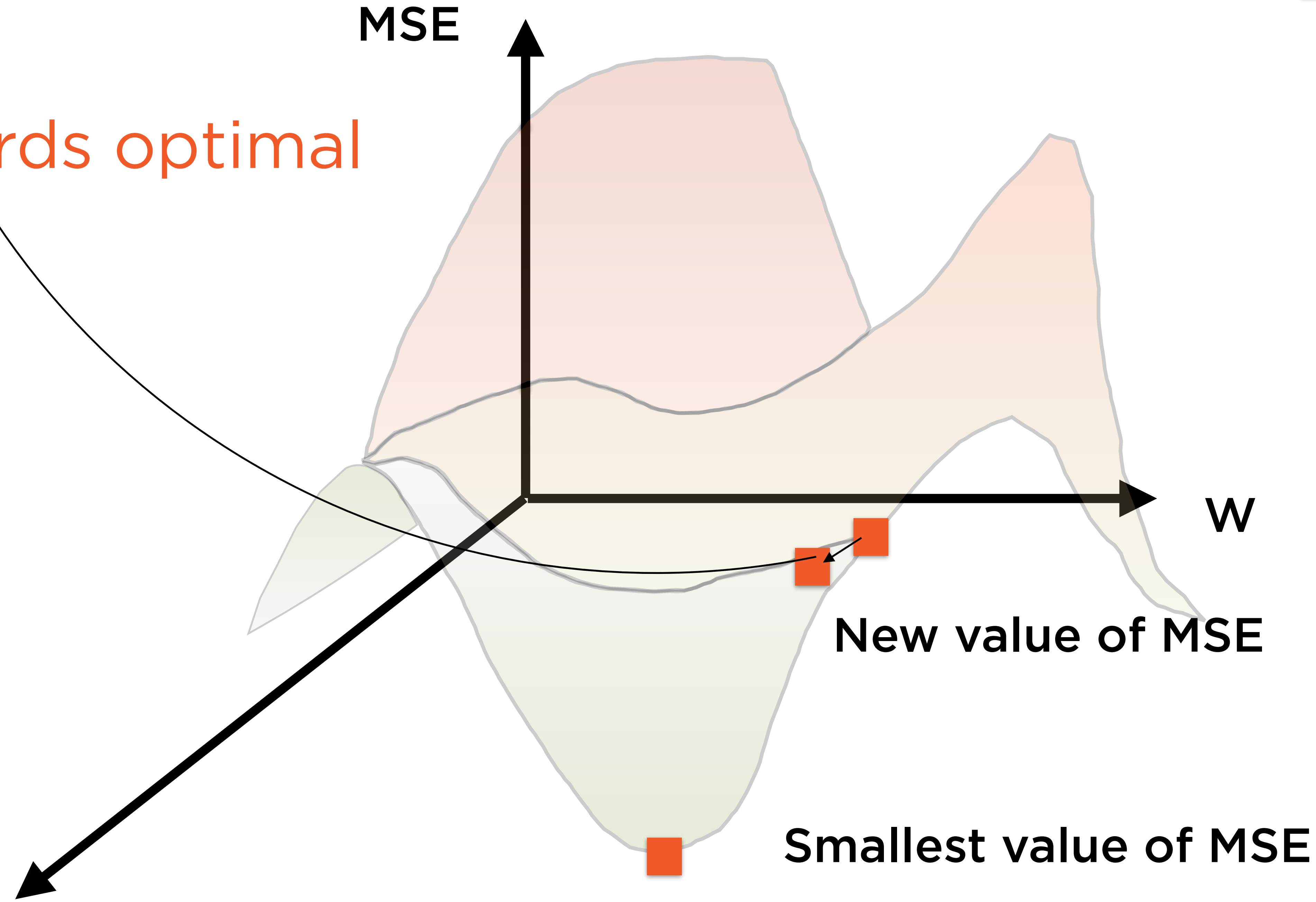


Minimizing MSE



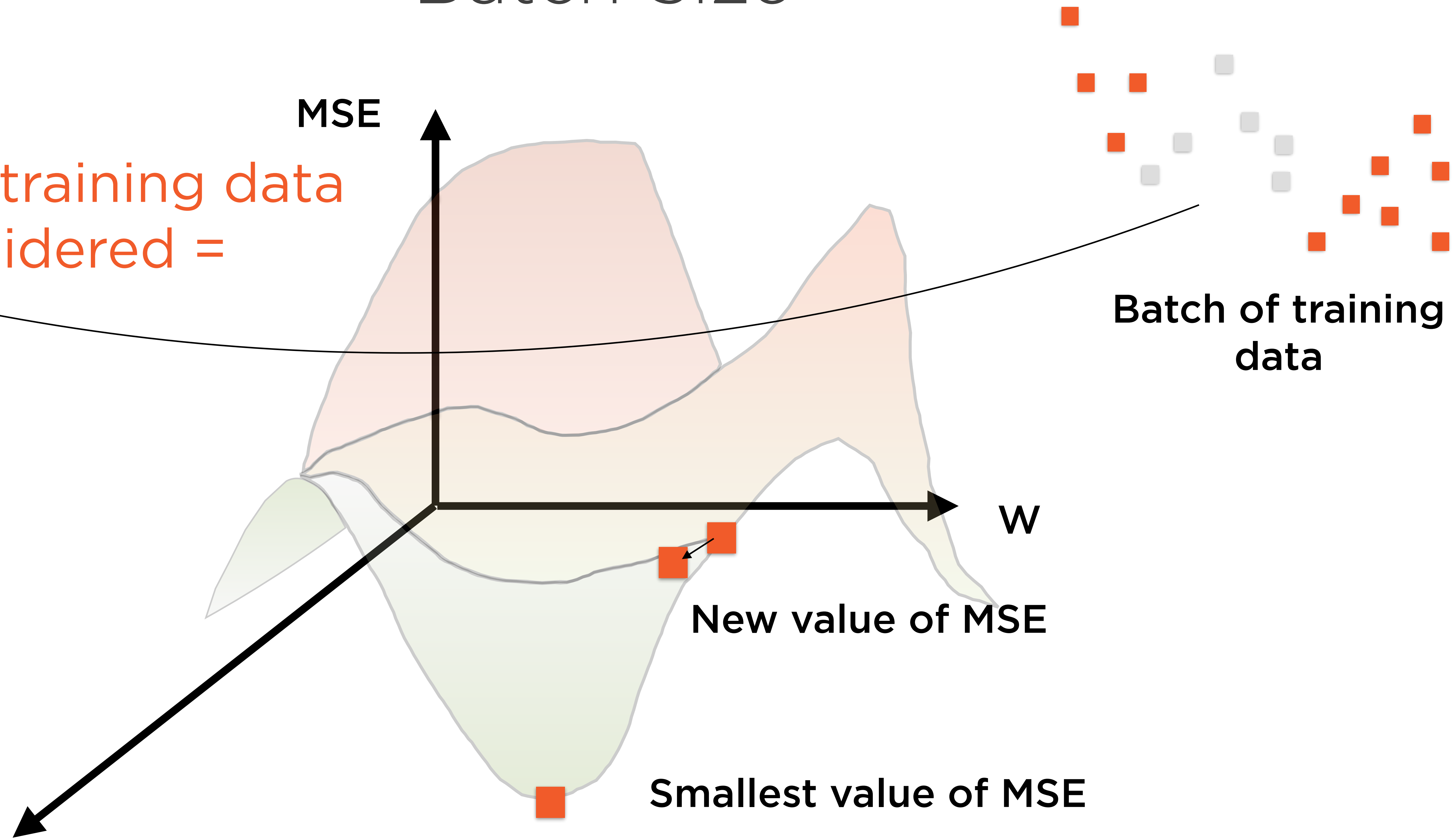
“Epoch”

1 step towards optimal
= 1 epoch



“Batch Size”

Number of training data points considered = batch size



"Batch Size"

Stochastic Gradient Descent

1 point at a time

Mini-batch Gradient Descent

Some subset in each batch

Batch Gradient Descent

All training data in each batch

Implementing Regression in TensorFlow

Baseline

Non-TensorFlow implementation

Regular python code

Computation Graph

Neural network of 1 neuron

Affine transformation suffices

Cost Function

Mean Square Error (MSE)

Quantifying goodness-of-fit

Optimizer

Gradient Descent optimizers

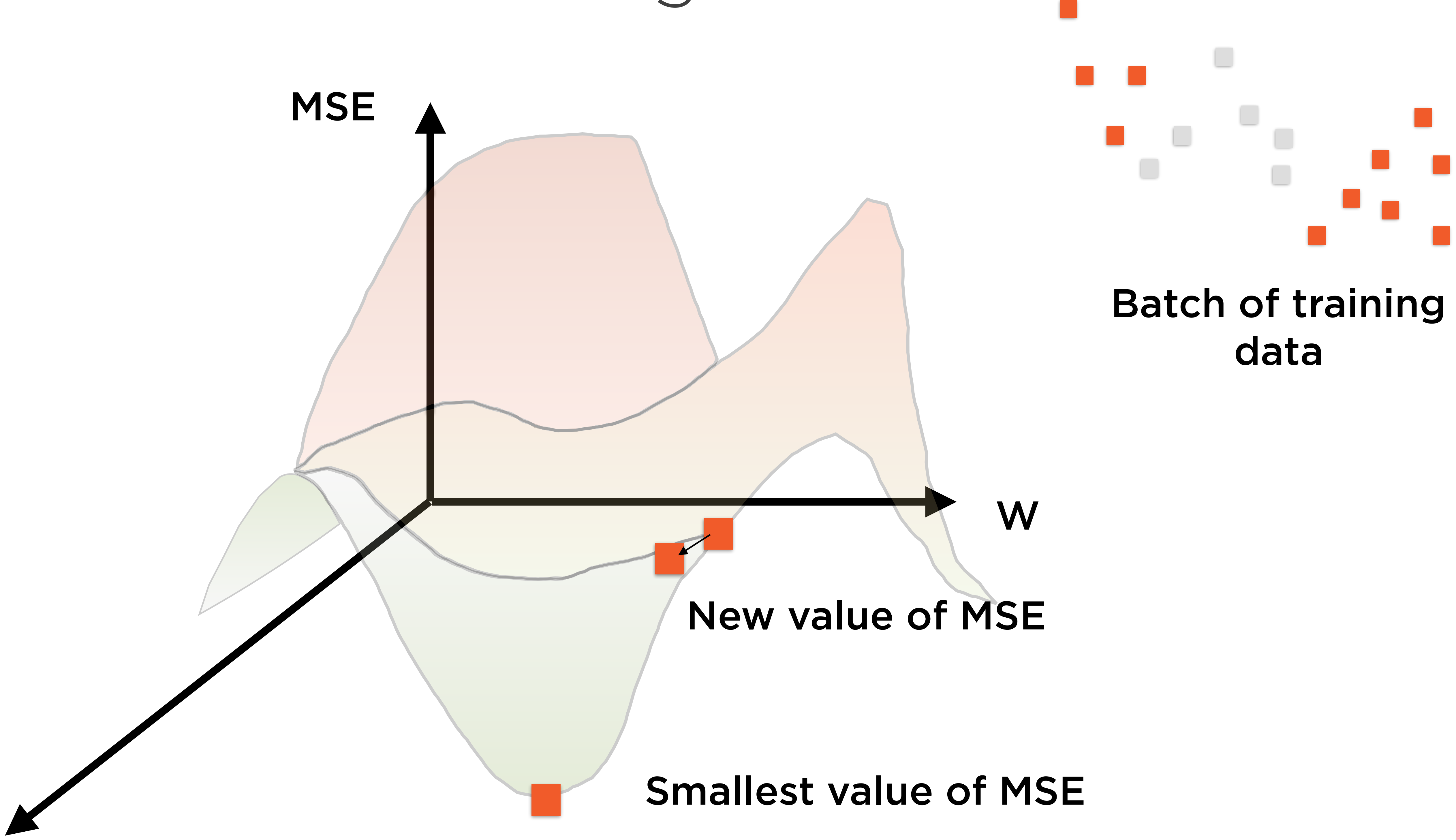
Improving goodness-of-fit

Training

Invoke optimizer in epochs

Batch size for each epoch

Minimizing MSE



Decisions in Training

Initial values

Type of optimizer

Number of epochs

Batch size

Simple Regression

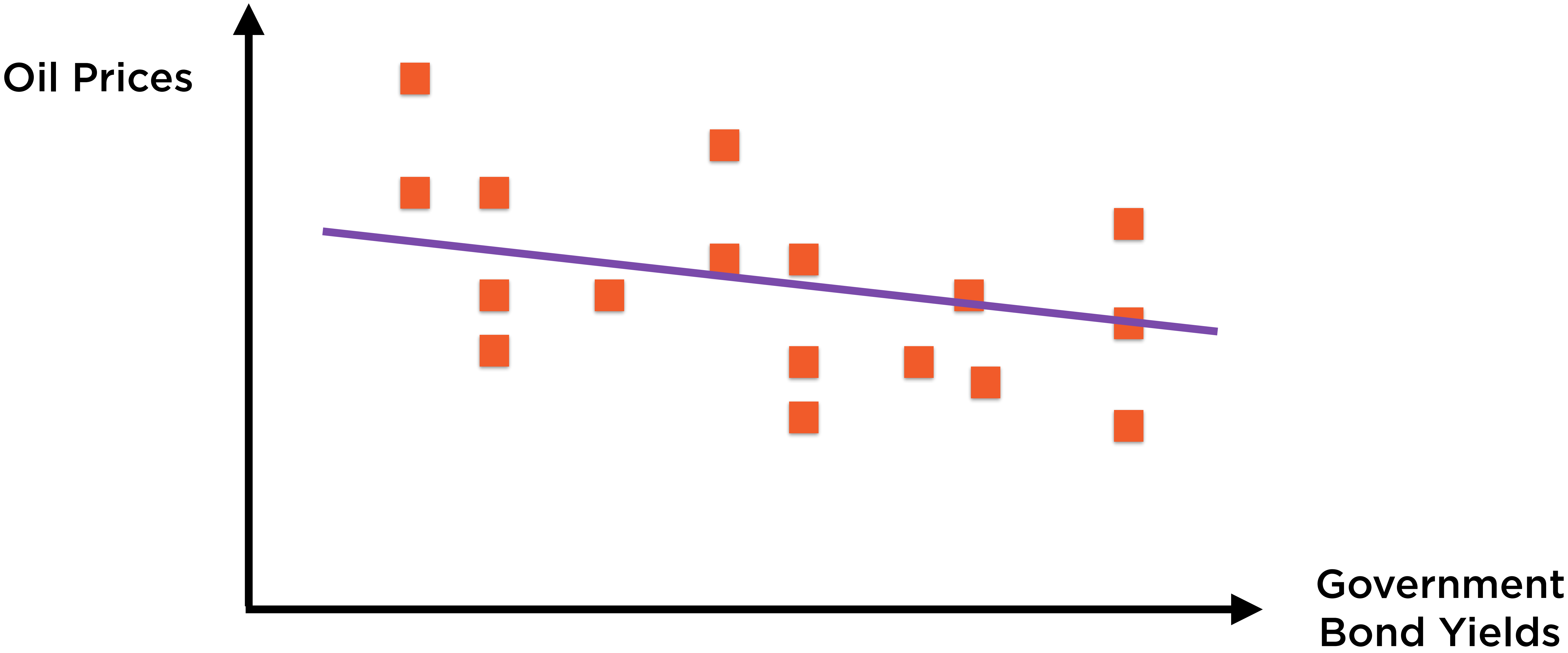
Cause

Independent variable

Effect

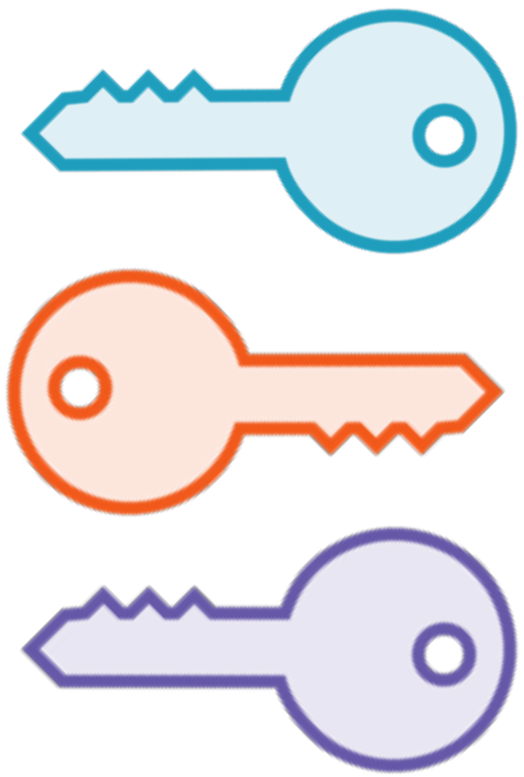
Dependent variable

Simple Regression



One cause, one effect

Multiple Regression



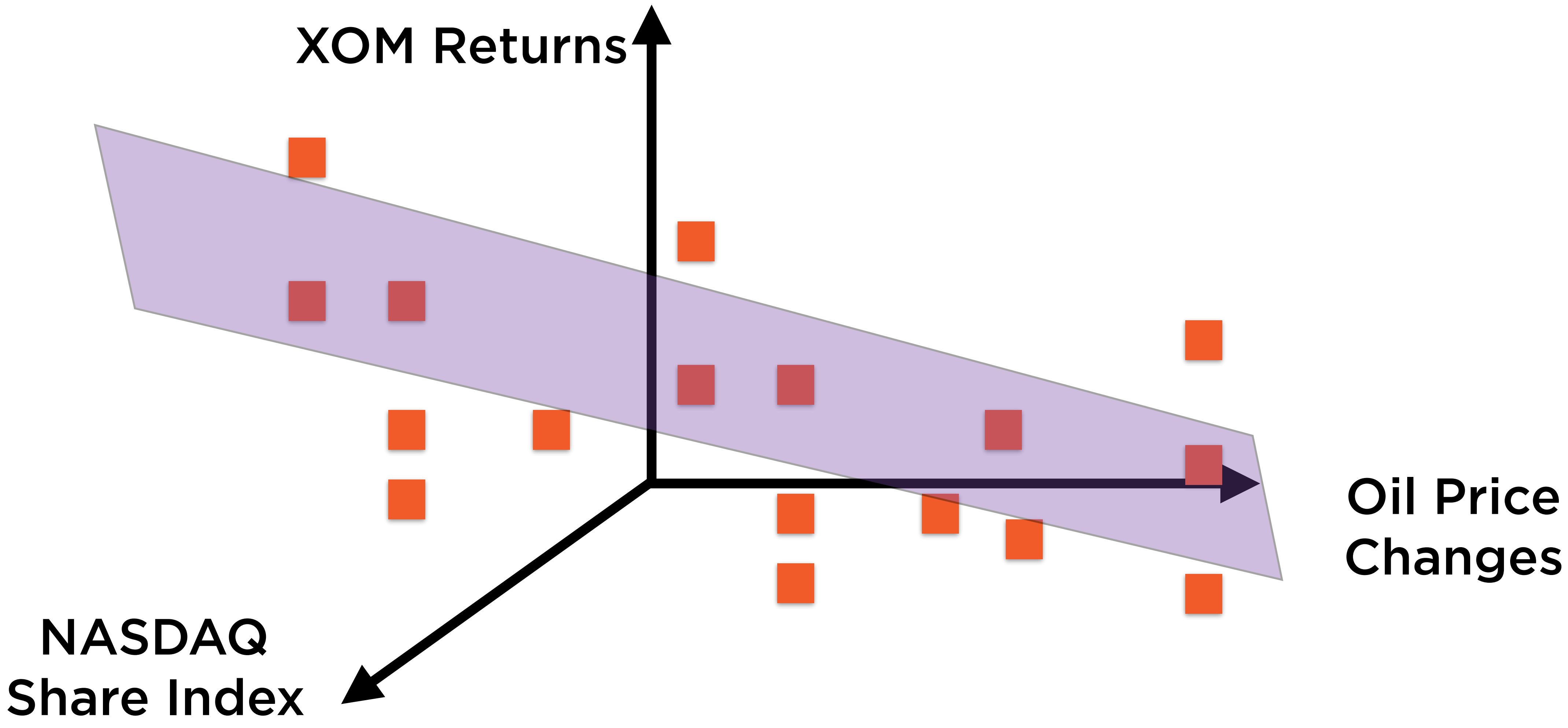
Causes

Independent variables

Effect

Dependent variable

Multiple Regression



Many causes, one effect

Implementing Regression in TensorFlow

Baseline

Non-TensorFlow implementation

Regular python code

Computation Graph

Neural network of 1 neuron

Affine transformation suffices

Cost Function

Mean Square Error (MSE)

Quantifying goodness-of-fit

Optimizer

Gradient Descent optimizers

Improving goodness-of-fit

Training

Invoke optimizer in epochs

Batch size for each epoch

Converged Model

Values of W and b

Compare to baseline

Summary

Implement linear regression without using TensorFlow

Define computation graph of just one neuron

Set up the cost function

Use various gradient descent optimizers

Understand gradient descent process

Train the model to get a converged regression model