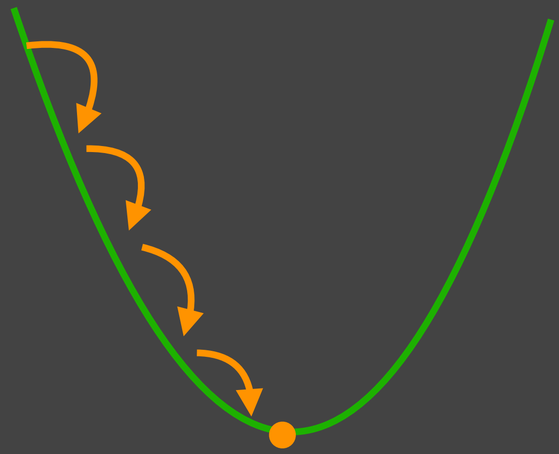
Machine Learning and Optimization Theory

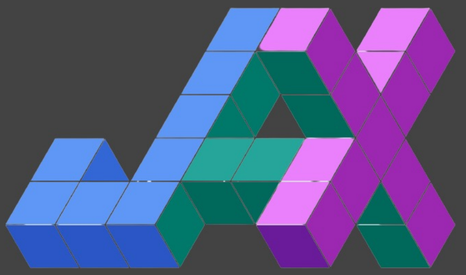
Lecturer: Göktuğ Güvercin

Organizer: BASIRA-CENTER

Program: 100TUNAI

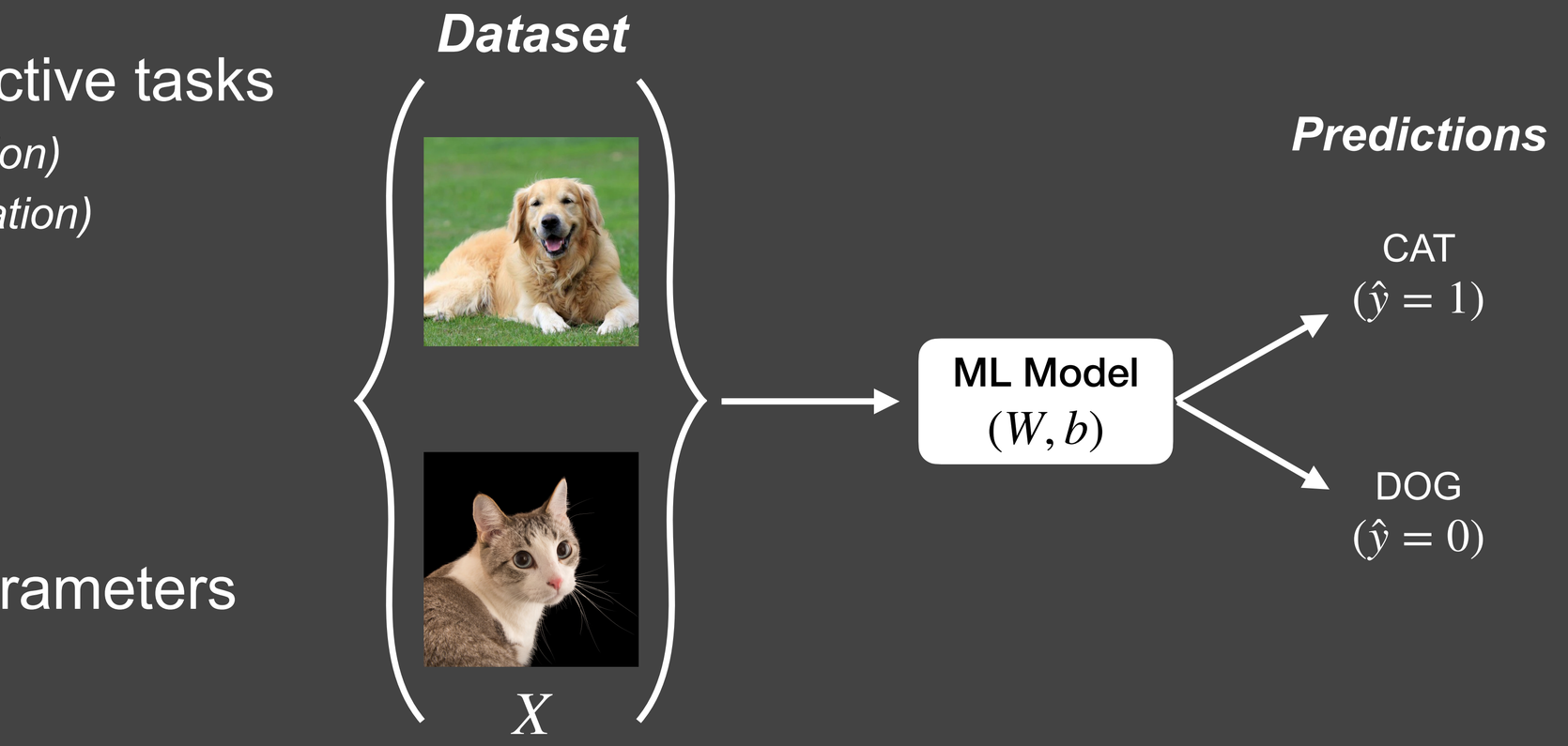
Table of Contents





Optimization Theory - Taxonomy of Machine Learning

- ML models to learn predictive tasks
  - House price prediction *(regression)*
  - Cat-Dog classification *(classification)*
- Requirement of dataset
  - Data samples X
  - Ground-truth labels y
- Optimization of model parameters
  - Weights W
  - Biases b



Optimization Theory - Taxonomy of Machine Learning

- Loss function for ML models
  - Performance measure for learning models
  - Comparison of model predictions with ground-truth labels
  - Quantification similarity between predictions and ground-truth labels

Optimization Theory - Taxonomy of Machine Learning

- Loss function for ML models
  - Performance measure for learning models
  - Comparison of model predictions with ground-truth labels
  - Quantification similarity between predictions and ground-truth labels

	Model Predictions	Ground-Truth Labels	Quality of Predictions	Loss Function
Model 1	18.2	18.4	Good	Low
Model 2	13.8	18.4	Bad	High

Optimization Theory - Taxonomy of Machine Learning

- Loss function for ML models
  - Performance measure for learning models
  - Comparison of model predictions with ground-truth labels
  - Quantification similarity between predictions and ground-truth labels

- Our purpose
  - Increasing quality of model predictions
  - Decreasing error rate (cost) of model predictions

	Model Predictions	Ground-Truth Labels	Quality of Predictions	Loss Function
Model 1	18.2	18.4	Good	Low
Model 2	13.8	18.4	Bad	High

Optimization Theory - Taxonomy of Machine Learning

- Loss function for ML models
  - Performance measure for learning models
  - Comparison of model predictions with ground-truth labels
  - Quantification similarity between predictions and ground-truth labels

- Our purpose
  - Increasing quality of model predictions
  - Decreasing error rate (cost) of model predictions

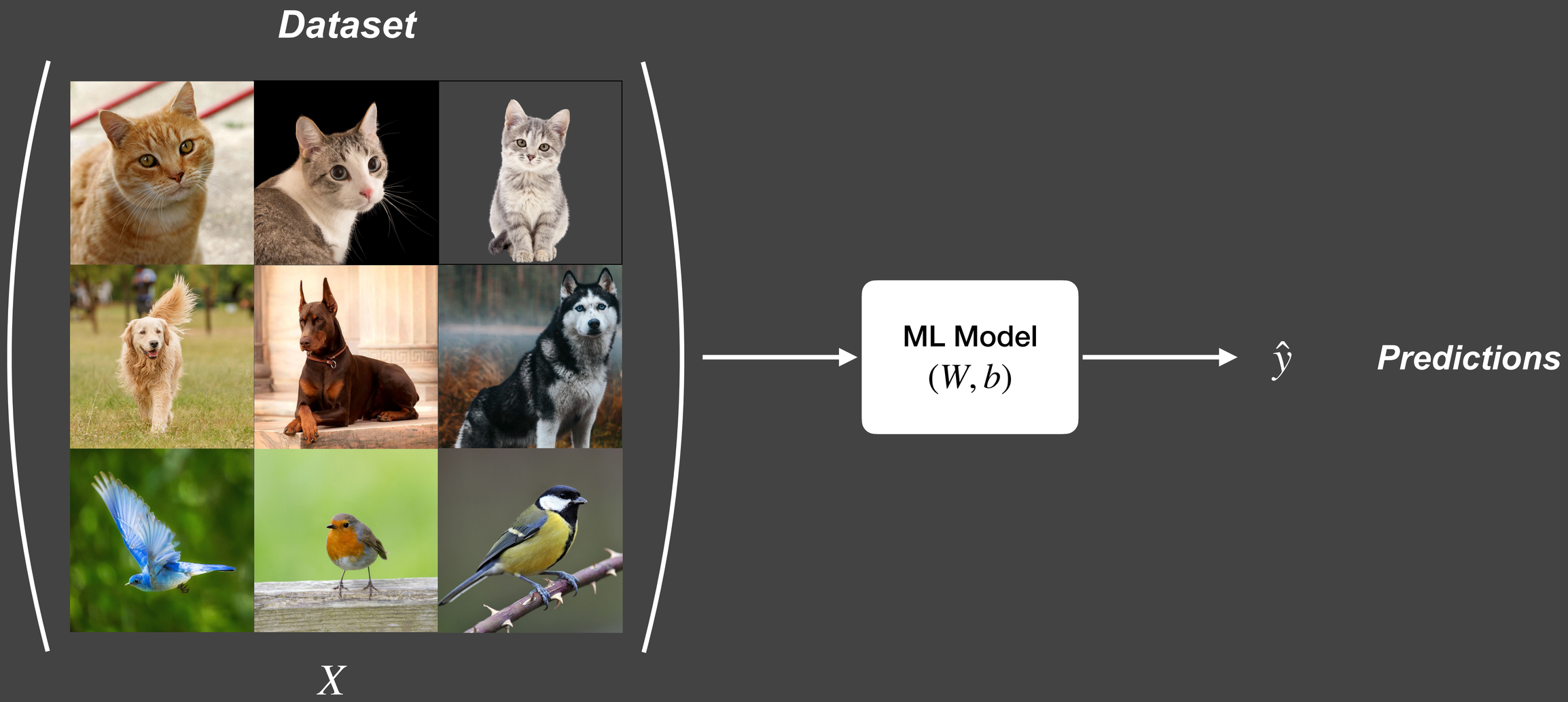
	Model Predictions	Ground-Truth Labels	Quality of Predictions	Loss Function
Model 1	18.2	18.4	Good	Low
Model 2	13.8	18.4	Bad	High

Optimization Theory - Taxonomy of Machine Learning

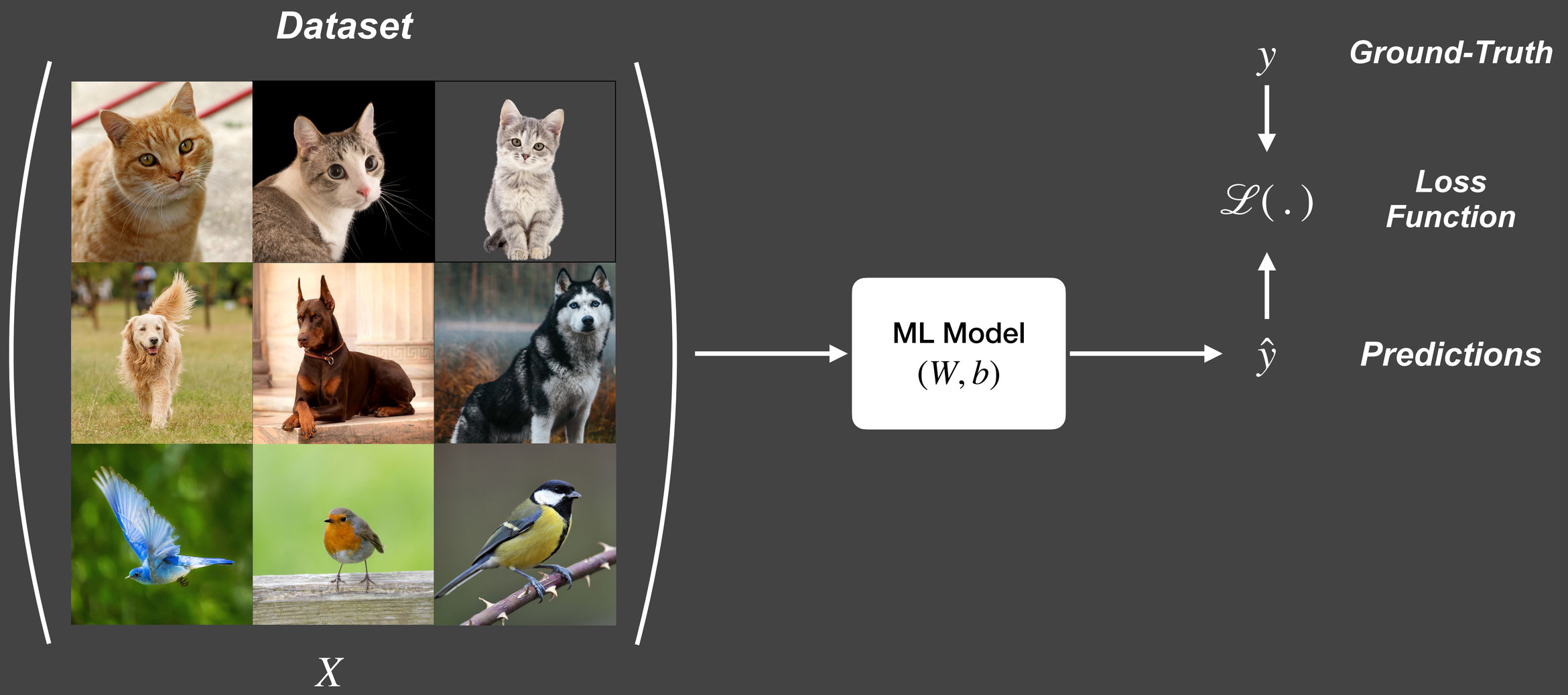
Dataset

X

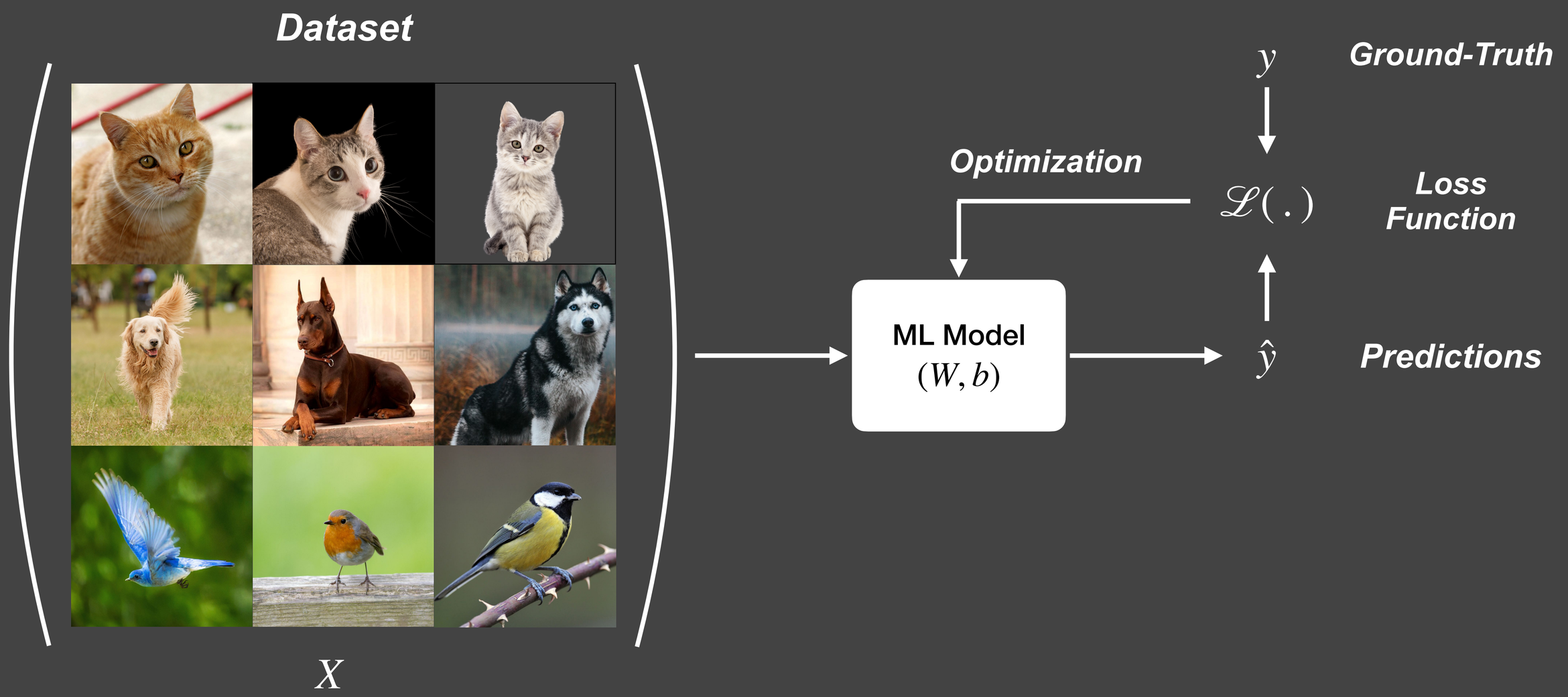
Optimization Theory - Taxonomy of Machine Learning



Optimization Theory - Taxonomy of Machine Learning



Optimization Theory - Taxonomy of Machine Learning



Optimization Theory - Taxonomy of Machine Learning

Dataset

- Resource for updating and optimizing model parameters
- Larger dataset, better model

Model and Loss Functions

- Model: Mathematical functions to learn relationship between features and labels
- Loss Function: Mathematical functions to compare predictions and labels

Optimization Algorithm

- Updating and optimizing model parameters

Optimization Theory - Taxonomy of Machine Learning

Dataset

- Resource for updating and optimizing model parameters
- Larger dataset, better model

Model and Loss Functions

- Model: Mathematical functions to learn relationship between features and labels
- Loss Function: Mathematical functions to compare predictions and labels

Optimization Algorithm

- Updating and optimizing model parameters

Optimization Theory - Data Notations

- Data sample (x^i, y^i)
  - Feature vector of d components x^i
  - Scalar ground-truth label y^i

Optimization Theory - Data Notations

- Data sample (x^i, y^i)
  - Feature vector of d components x^i
  - Scalar ground-truth label y^i

Optimization Theory - Data Notations

- Data sample (x^i, y^i)
  - Feature vector of d components x^i
  - Scalar ground-truth label y^i

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \rightarrow y$$

Optimization Theory - Data Notations

- Data sample (x^i, y^i)
  - Feature vector of d components x^i
  - Scalar ground-truth label y^i

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \rightarrow y$$

x_j^i → **Sample index**

x_j^i → **Feature index**

Optimization Theory - Data Notations

- Data sample (x^i, y^i)
  - Feature vector of d components x^i
  - Scalar ground-truth label y^i

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \rightarrow y$$

- Dataset $D = (X, Y)$
  - Matrix of N data samples X
  - Vector of N labels Y

Optimization Theory - Data Notations

- Data sample (x^i, y^i)
  - Feature vector of d components x^i
  - Scalar ground-truth label y^i

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \rightarrow y$$

- Dataset $D = (X, Y)$
  - Matrix of N data samples X
  - Vector of N labels Y

$$D = \{(x^1, y^1), (x^2, y^2), \cdots, (x^N, y^N)\}$$

$$\downarrow$$

$$D = (X, Y) = \left(\begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_d^1 \\ x_1^2 & x_2^2 & \cdots & x_d^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^N & x_2^N & \cdots & x_d^N \end{bmatrix}, \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^N \end{bmatrix} \right)$$

Optimization Theory - Data Notations

- Data sample (x^i, y^i)
  - Feature vector of d components x^i
  - Scalar ground-truth label y^i

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \rightarrow y$$

- Dataset $D = (X, Y)$
  - Matrix of N data samples X
  - Vector of N labels Y

$$D = \{(x^1, y^1), (x^2, y^2), \cdots, (x^N, y^N)\}$$

$$\downarrow$$

$$D = (X, Y) = \left(\begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_d^1 \\ x_1^2 & x_2^2 & \cdots & x_d^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^N & x_2^N & \cdots & x_d^N \end{bmatrix}, \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^N \end{bmatrix} \right)$$

Optimization Theory - Data Notations

- Data sample (x^i, y^i)
  - Feature vector of d components x^i
  - Scalar ground-truth label y^i

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \rightarrow y$$

- Dataset $D = (X, Y)$
  - Matrix of N data samples X
  - Vector of N labels Y

$$D = \{(x^1, y^1), (x^2, y^2), \cdots, (x^N, y^N)\}$$

$$\downarrow$$

$$D = (X, Y) = \left(\begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_d^1 \\ x_1^2 & x_2^2 & \cdots & x_d^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^N & x_2^N & \cdots & x_d^N \end{bmatrix}, \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^N \end{bmatrix} \right)$$

Optimization Theory - Taxonomy of Machine Learning

Dataset

- Resource for updating and optimizing model parameters
- Larger dataset, better model

Model and Loss Functions

- Model: Mathematical functions to learn relationship between features and labels
- Loss Function: Mathematical functions to compare predictions and labels

Optimization Algorithm

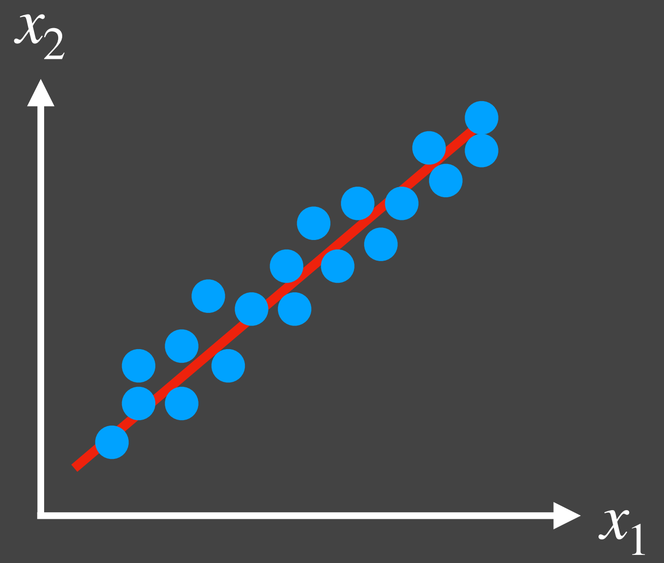
- Updating and optimizing model parameters

Optimization Theory - Model Notations

Model Function

$$f_{w,b}(x) = x^T \cdot w + b$$

$$x \in R^d, \quad w \in R^d, \quad b \in R$$

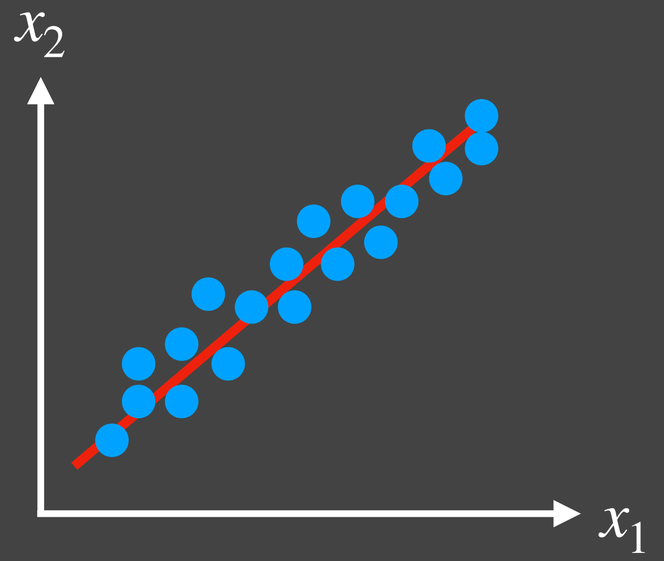


Optimization Theory - Model Notations

Model Function

$$f_{w,b}(x) = x^T \cdot w + b$$

$$x \in R^d, \quad w \in R^d, \quad b \in R$$

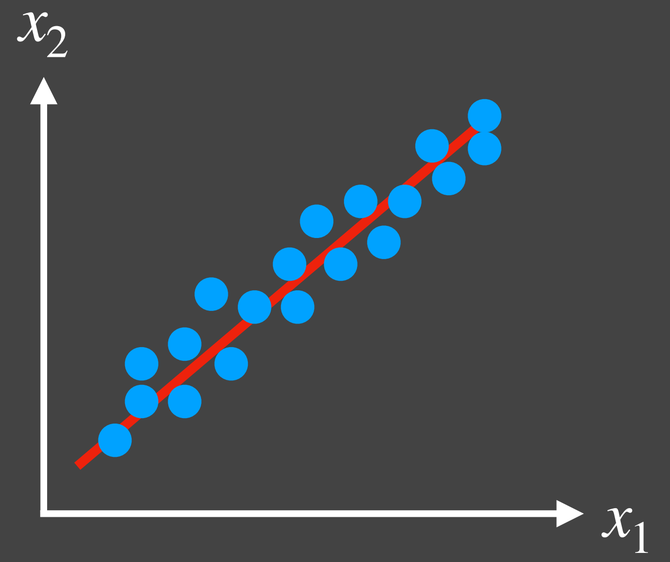


Optimization Theory - Model Notations

Model Function

$$f_{w,b}(x) = x^T \cdot w + b$$

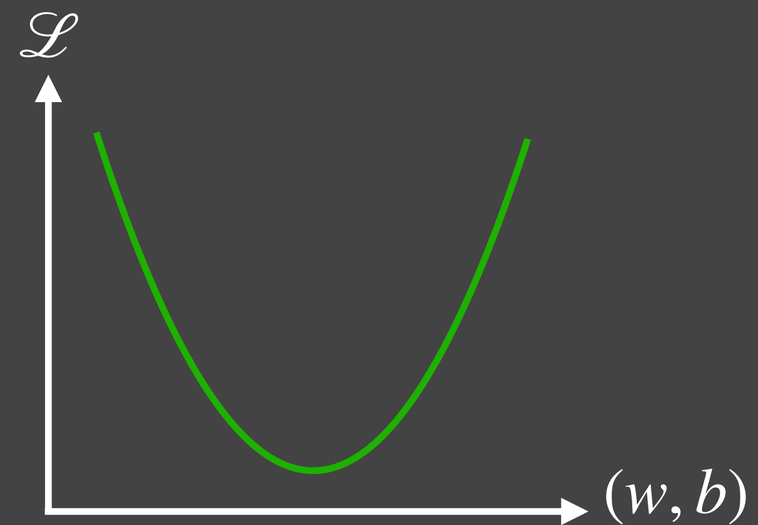
$$x \in R^d, \quad w \in R^d, \quad b \in R$$



Loss Function

$$\mathscr{L}(w,b) = \frac{1}{N}\sum_{i=1}^{N}(f_{w,b}(x^i) - y^i)^2$$

$$x^i \in R^d, \quad y^i \in R$$

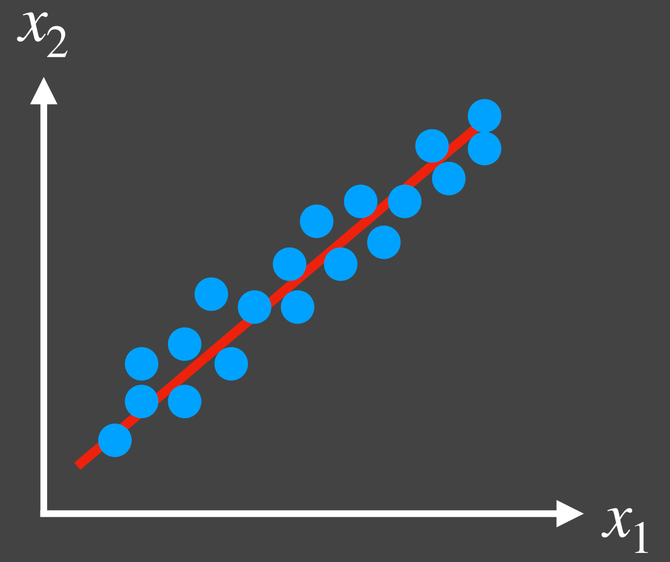


Optimization Theory - Model Notations

Model Function

$$f_{w,b}(x) = x^T \cdot w + b$$

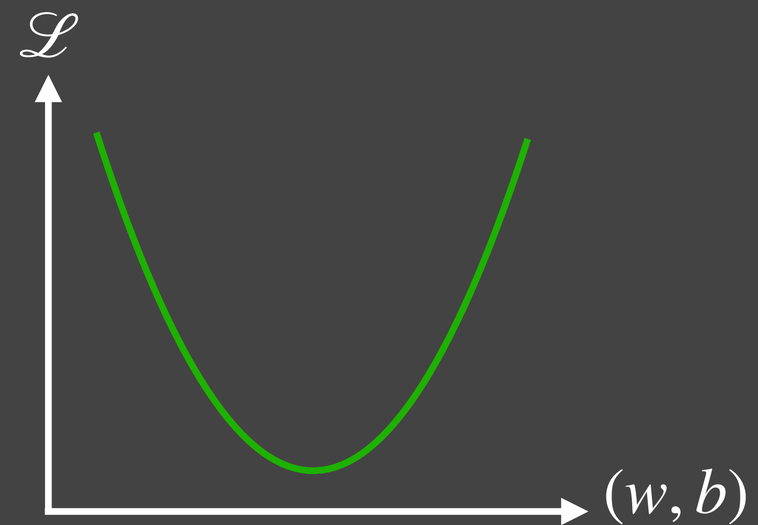
$$x \in R^d, \quad w \in R^d, \quad b \in R$$



Loss Function

$$\mathscr{L}(w,b) = \frac{1}{N}\sum_{i=1}^{N}(f_{w,b}(x^i) - y^i)^2$$

$$x^i \in R^d, \quad y^i \in R$$

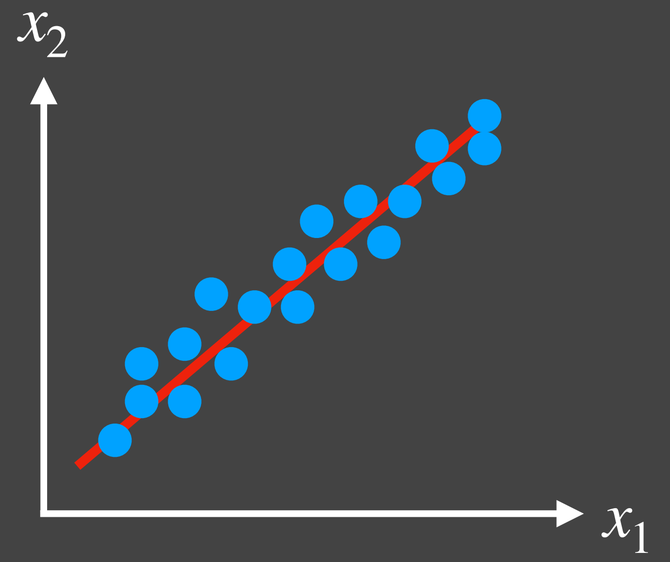


Optimization Theory - Model Notations

Model Function

$$f_{w,b}(x) = x^T \cdot w + b$$

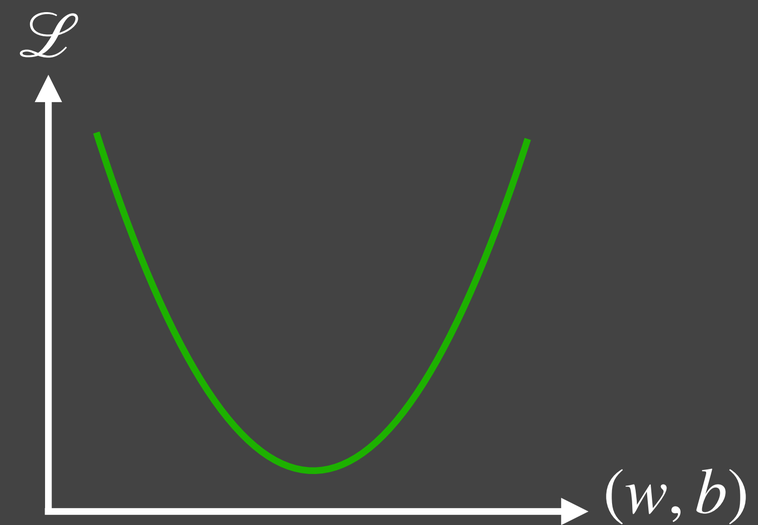
$$x \in R^d, \quad w \in R^d, \quad b \in R$$



Loss Function

$$\mathscr{L}(w,b) = \frac{1}{N}\sum_{i=1}^{N}(f_{w,b}(x^i) - y^i)^2$$

$$x^i \in R^d, \quad y^i \in R$$

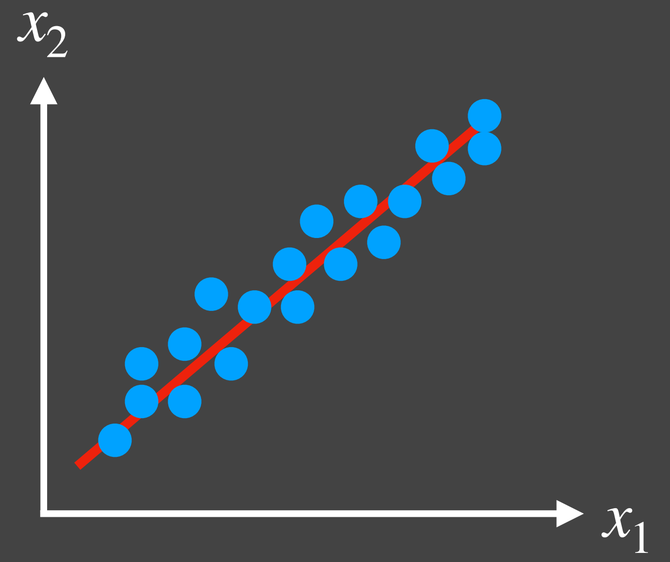


Optimization Theory - Model Notations

Model Function

$$f_{w,b}(x) = x^T \cdot w + b$$

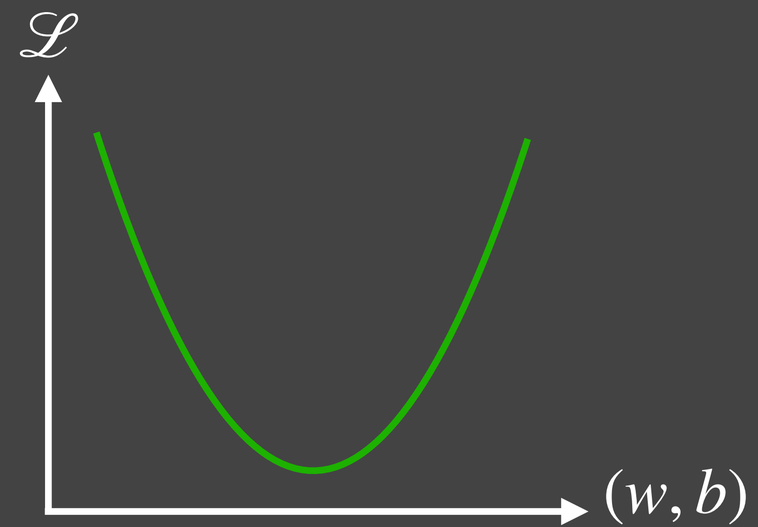
$$x \in R^d, \quad w \in R^d, \quad b \in R$$



Loss Function

$$\mathscr{L}(w,b) = \frac{1}{N}\sum_{i=1}^{N}(f_{w,b}(x^i) - y^i)^2$$

$$x^i \in R^d, \quad y^i \in R$$

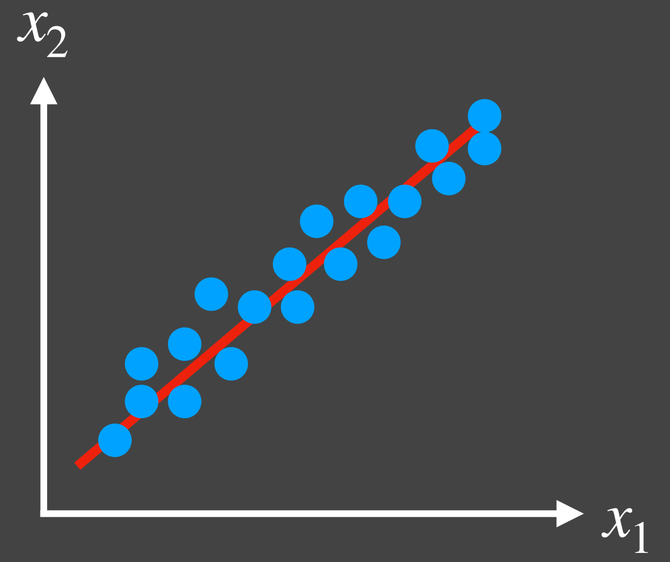


Optimization Theory - Model Notations

Model Function

$$f_{w,b}(x) = x^T \cdot w + b$$

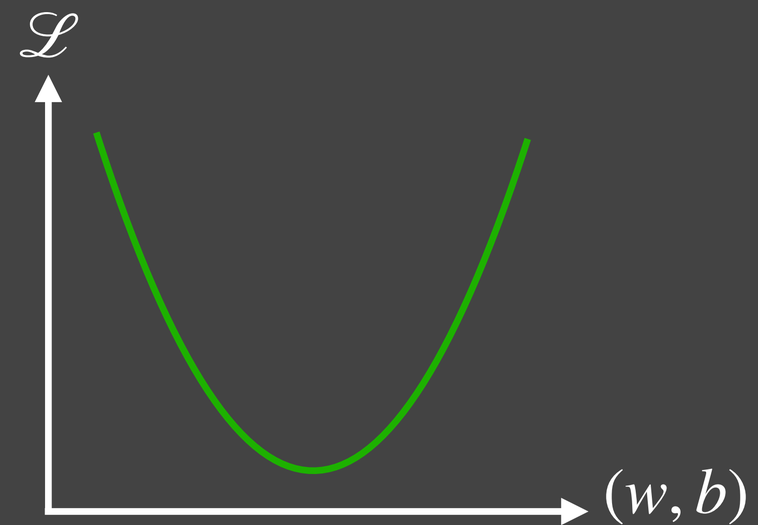
$$x \in R^d, \quad w \in R^d, \quad b \in R$$



Loss Function

$$\mathscr{L}(w,b) = \frac{1}{N}\sum_{i=1}^{N}(f_{w,b}(x^i) - y^i)^2$$

$$x^i \in R^d, \quad y^i \in R$$

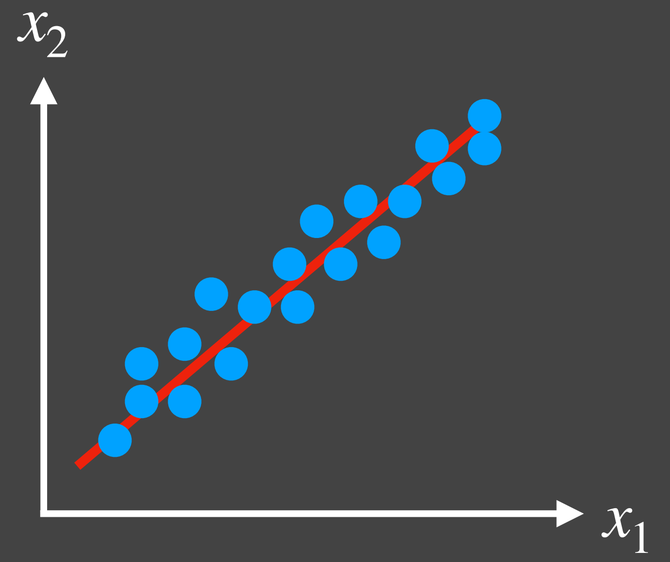


Optimization Theory - Model Notations

Model Function

$$f_{w,b}(x) = x^T \cdot w + b$$

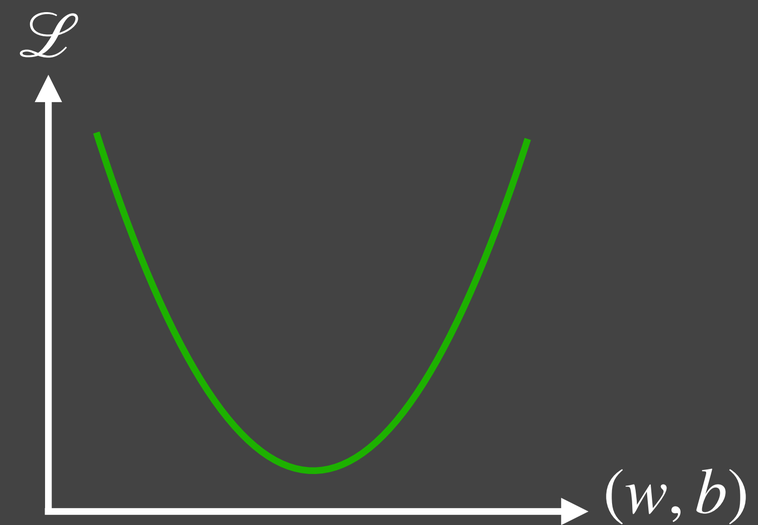
$$x \in R^d, \quad w \in R^d, \quad b \in R$$



MSE Loss

$$\mathscr{L}(w,b) = \frac{1}{N}\sum_{i=1}^{N}(f_{w,b}(x^i) - y^i)^2$$

$$x^i \in R^d, \quad y^i \in R$$



Optimization Theory - Taxonomy of Machine Learning

Dataset

- Resource for updating and optimizing model parameters
- Larger dataset, better model

Model and Loss Functions

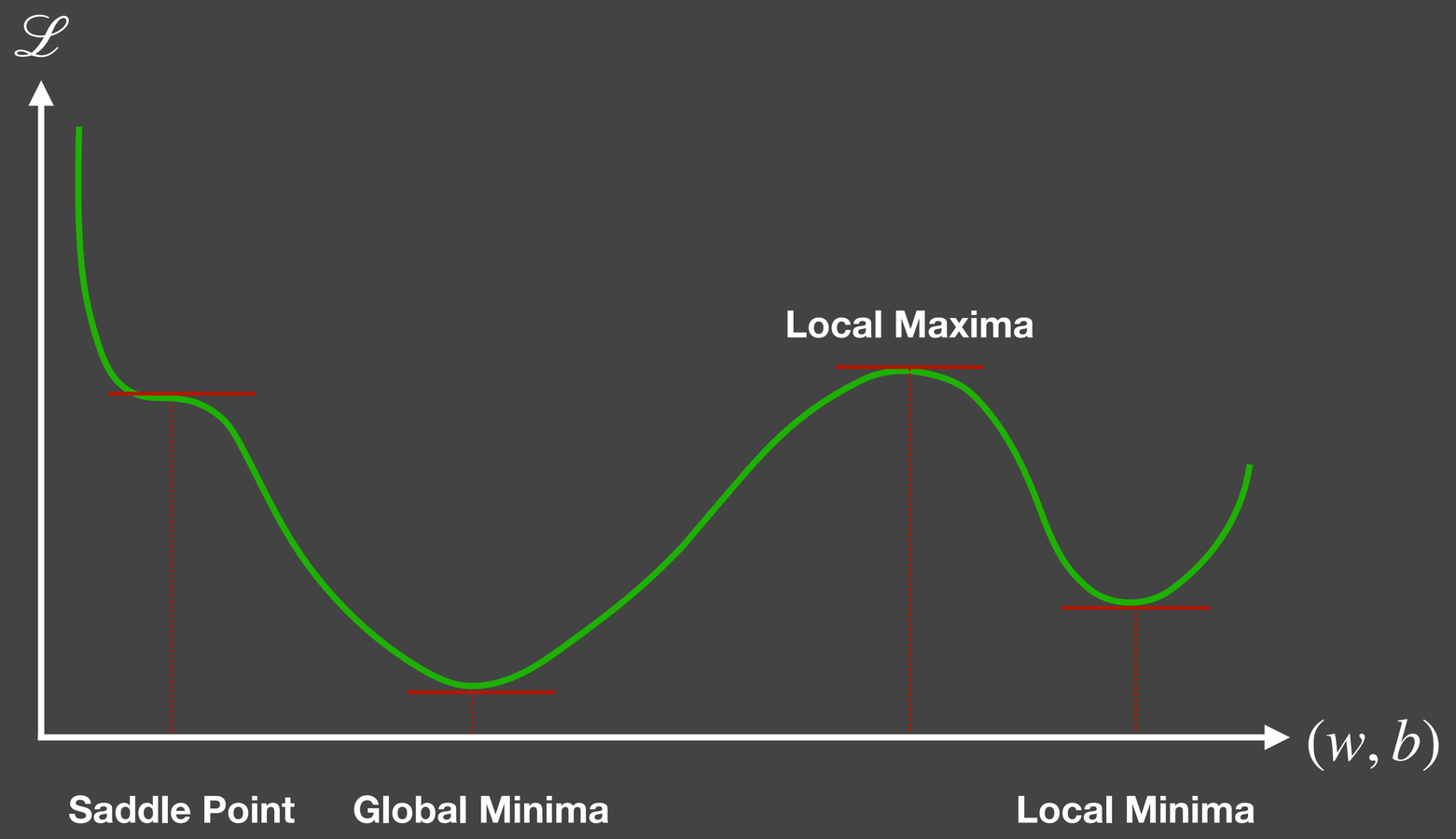
- Model: Mathematical functions to learn relationship between features and labels
- Loss Function: Mathematical functions to compare predictions and labels

Optimization Algorithm

- Updating and optimizing model parameters

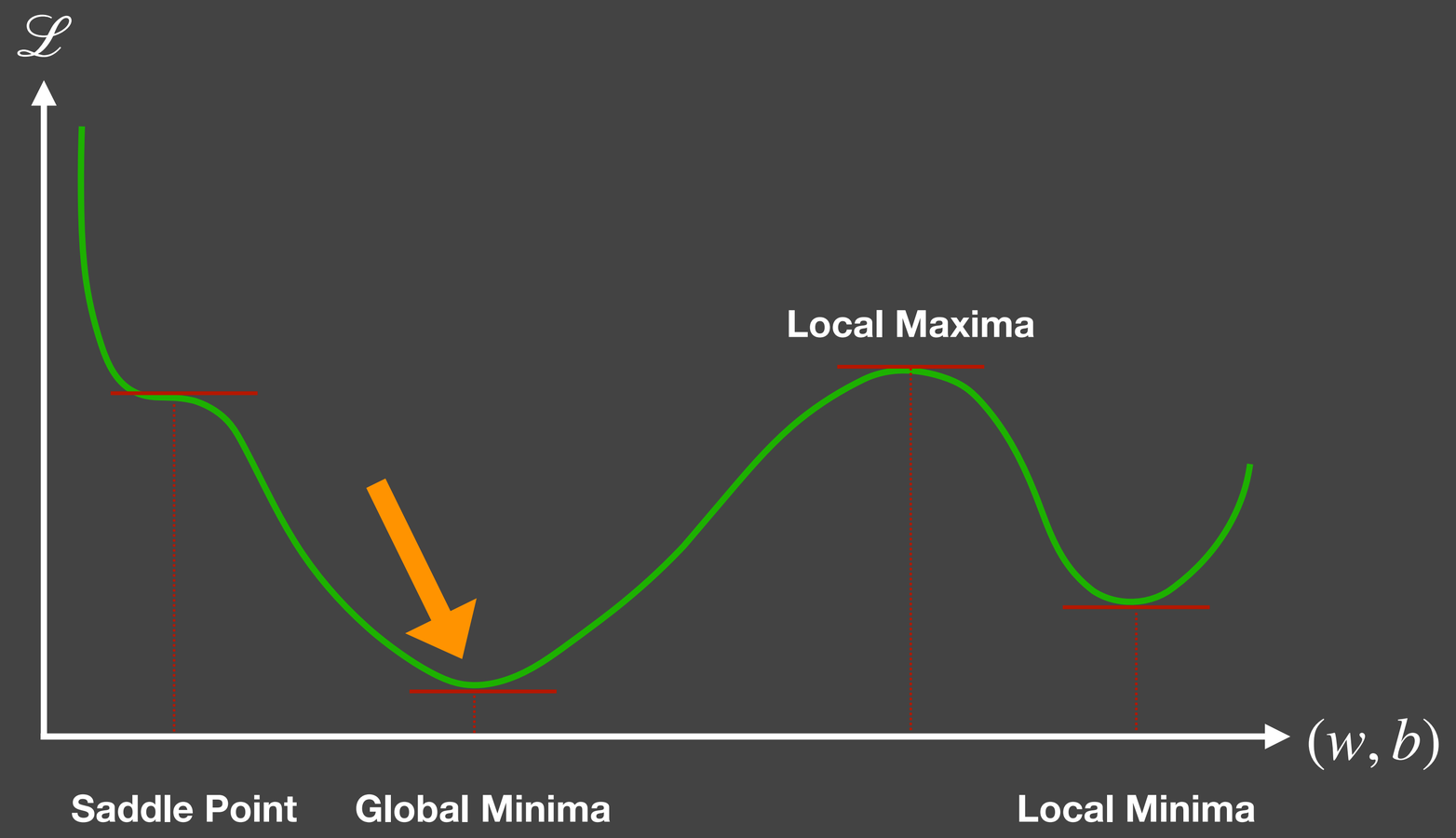
Optimization Theory - Stationary Points

- Stationary Points
  - Minima, maxima, saddle points
  - Derivative is zero



Optimization Theory - Stationary Points

- Stationary Points
  - Minima, maxima, saddle points
  - Derivative is zero



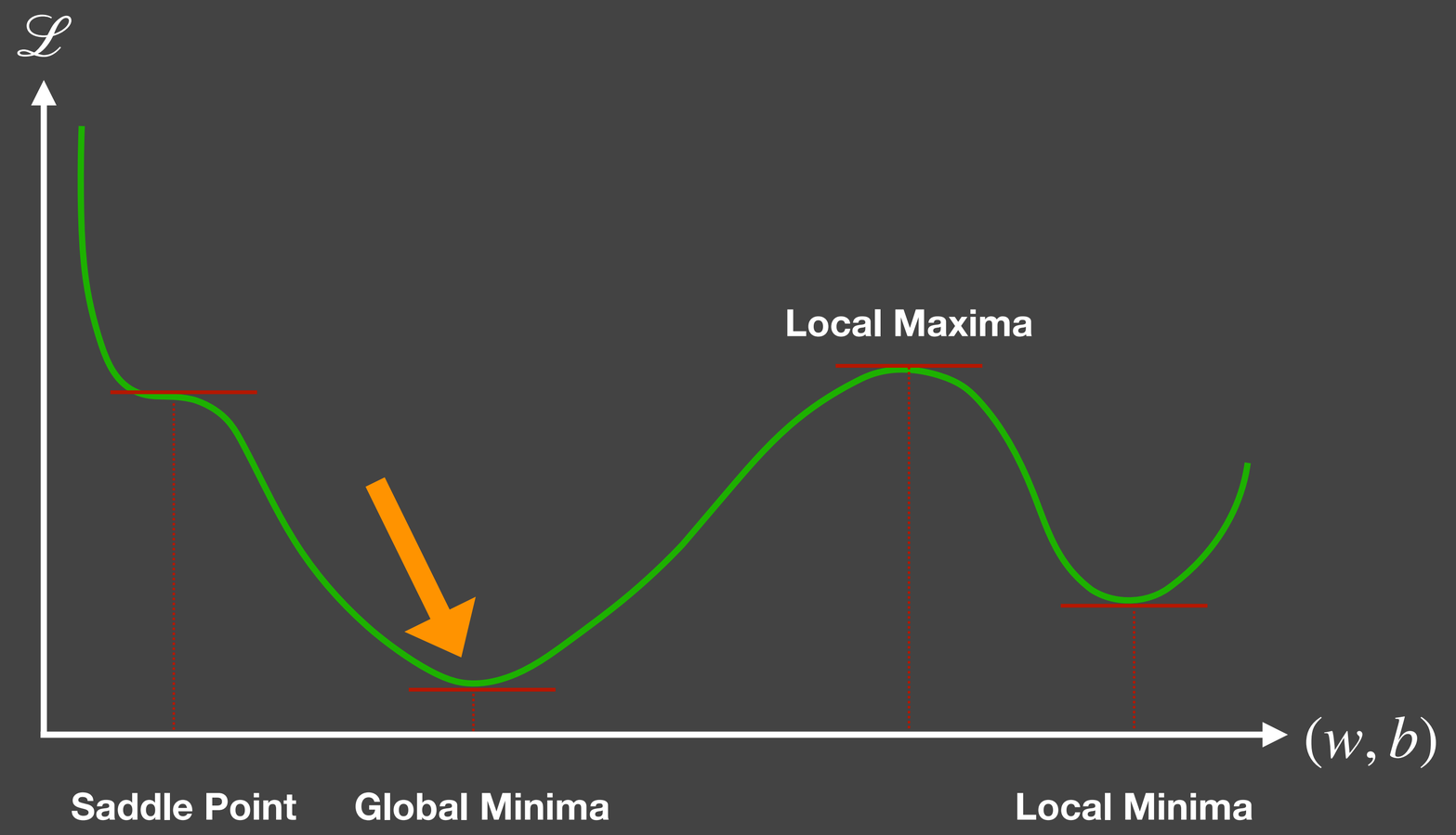
Optimization Theory - Stationary Points

- Stationary Points
  - Minima, maxima, saddle points
  - Derivative is zero

- Normal Equation

$$\frac{\partial \mathscr{L}}{\partial w} = 0$$

$$\frac{\partial \mathscr{L}}{\partial b} = 0$$



Optimization Theory - Differentiation and Gradients

- Univariate functions
  - $f : R \rightarrow R$
  - Single input, single output
  - Differentiation with single variable

$$f(x) = x^2 \tag{1}$$

$$\frac{df(x)}{dx} = 2x \tag{2}$$

$$f(x) \approx f(a) + \frac{df(a)}{dx}(x - a) \tag{3}$$

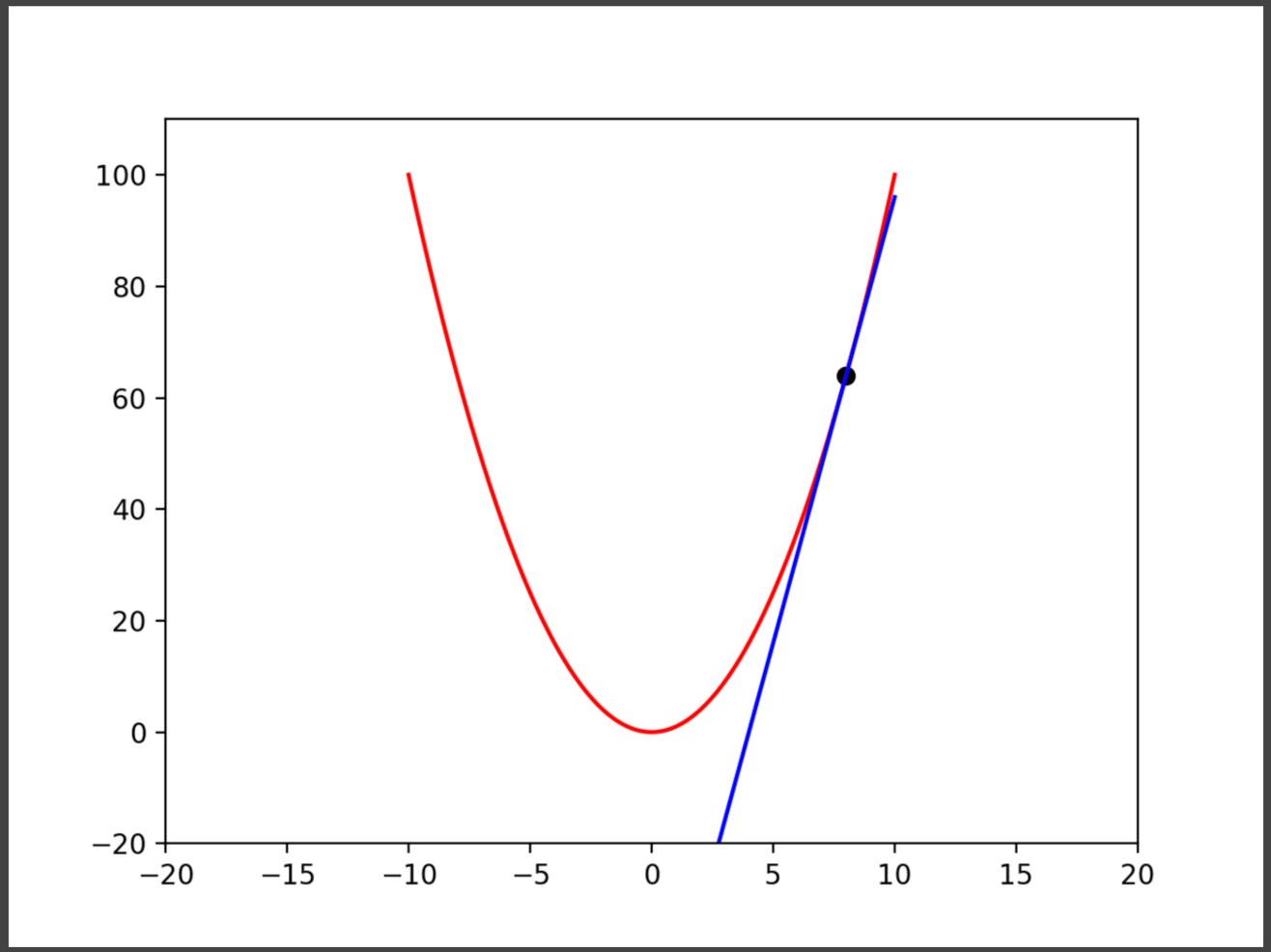
Optimization Theory - Differentiation and Gradients

- Univariate functions
  - $f : R \rightarrow R$
  - Single input, single output
  - Differentiation with single variable

$$f(x) = x^2 \tag{1}$$

$$\frac{df(x)}{dx} = 2x \tag{2}$$

$$f(x) \approx f(a) + \frac{df(a)}{dx}(x - a) \tag{3}$$



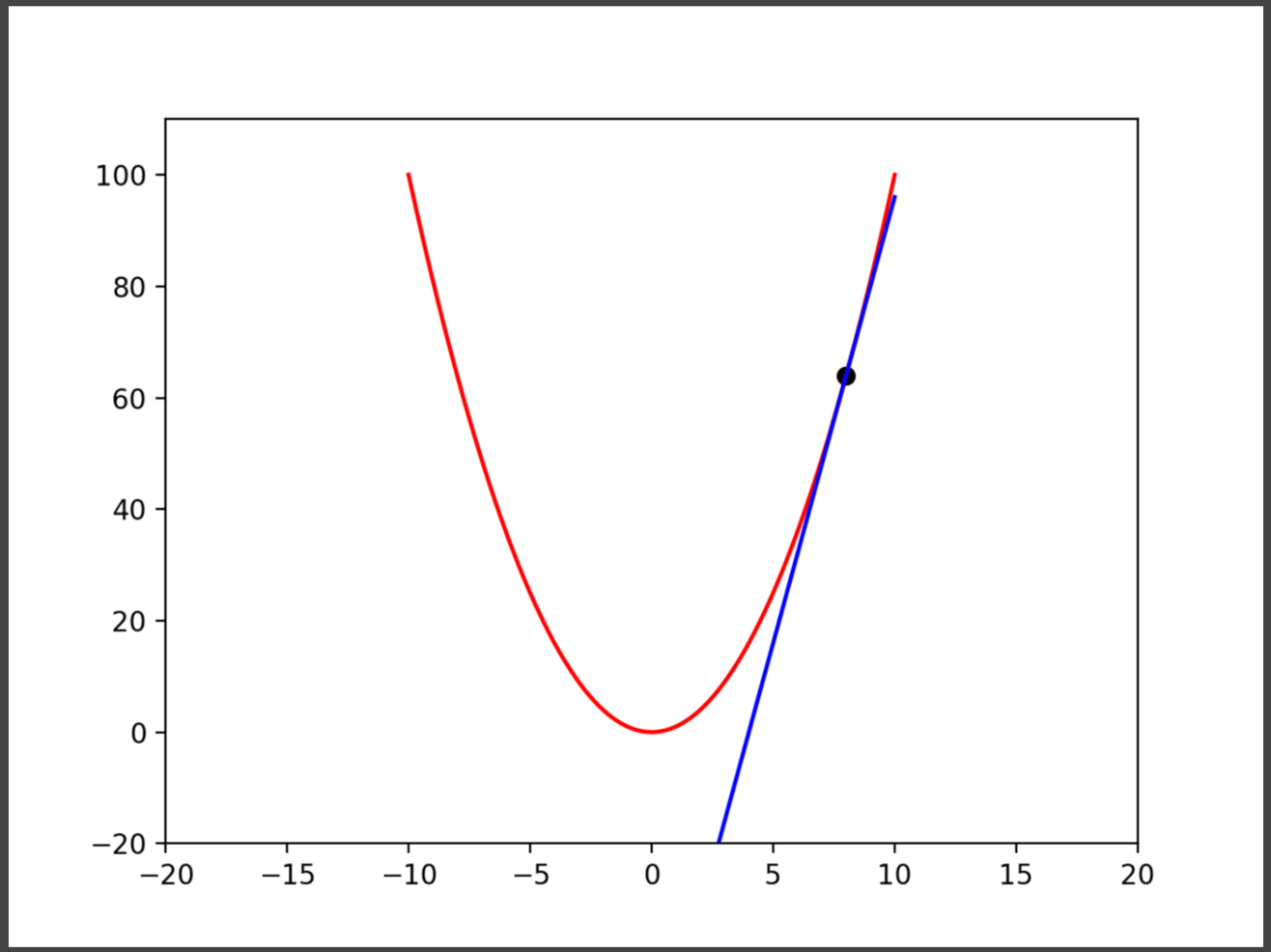
Optimization Theory - Differentiation and Gradients

- Univariate functions
  - $f : R \rightarrow R$
  - Single input, single output
  - Differentiation with single variable

$$f(x) = x^2 \tag{1}$$

$$\frac{df(x)}{dx} = 2x \tag{2}$$

$$f(x) \approx f(a) + \frac{df(a)}{dx}(x - a) \tag{3}$$



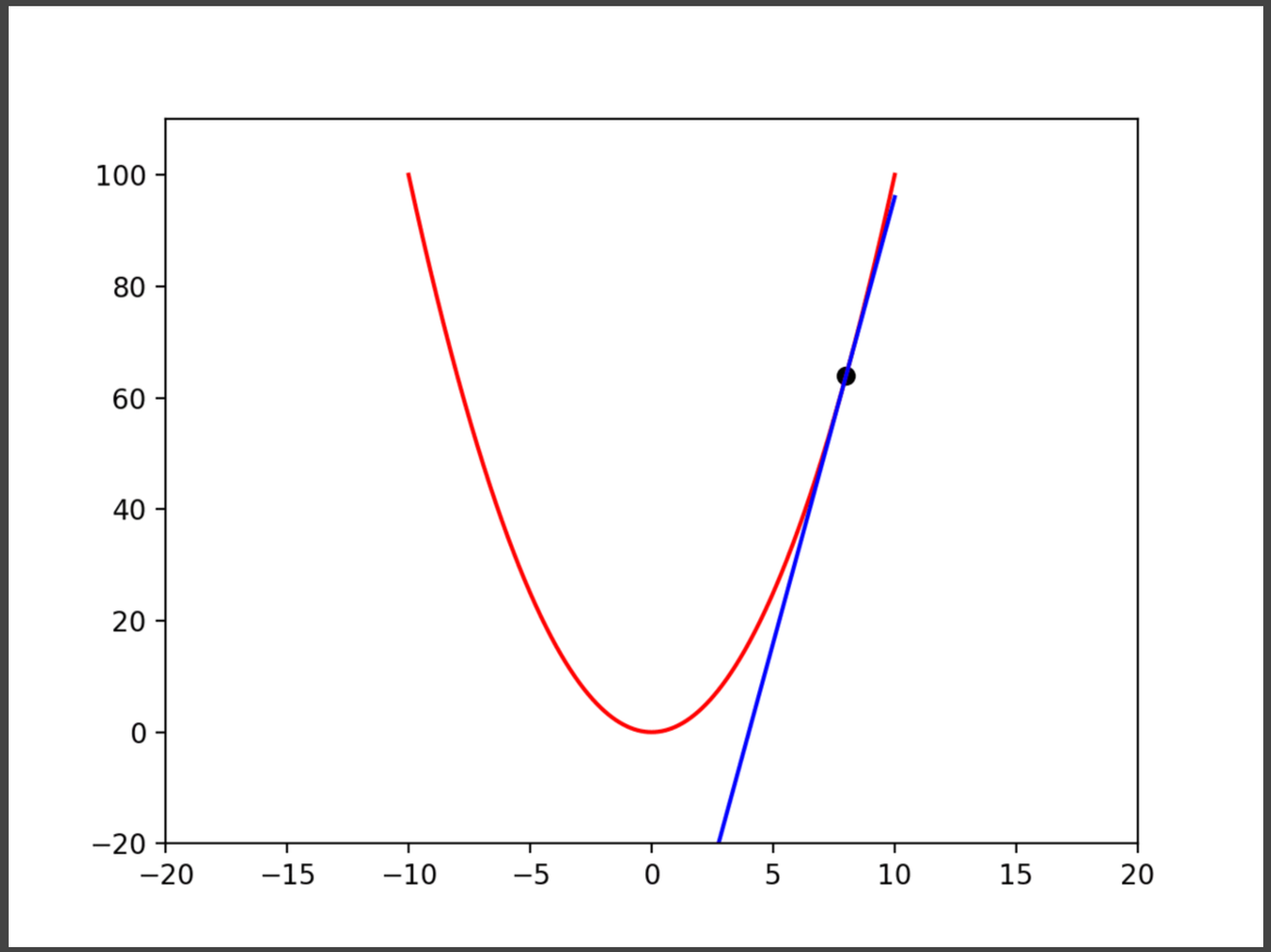
Optimization Theory - Differentiation and Gradients

- Univariate functions
  - $f : R \rightarrow R$
  - Single input, single output
  - Differentiation with single variable

$$f(x) = x^2 \tag{1}$$

$$\frac{df(x)}{dx} = 2x \tag{2}$$

$$f(x) \approx f(a) + \frac{df(a)}{dx}(x - a) \tag{3}$$



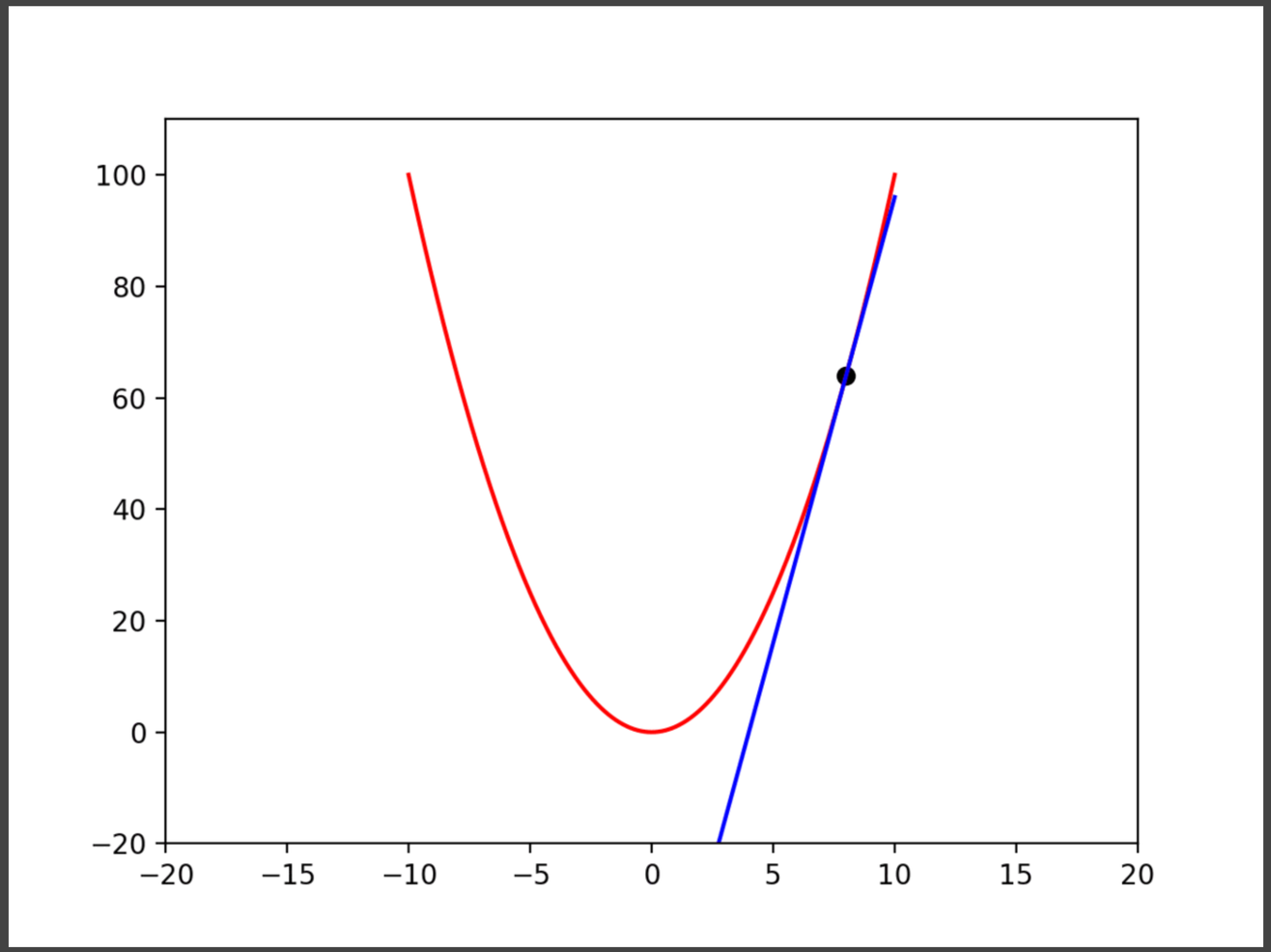
Optimization Theory - Differentiation and Gradients

- Univariate functions
  - $f : R \rightarrow R$
  - Single input, single output
  - Differentiation with single variable

$$f(x) = x^2 \tag{1}$$

$$\frac{df(x)}{dx} = 2x \tag{2}$$

$$f(x) \approx f(a) + \frac{df(a)}{dx}(x - a) \tag{3}$$



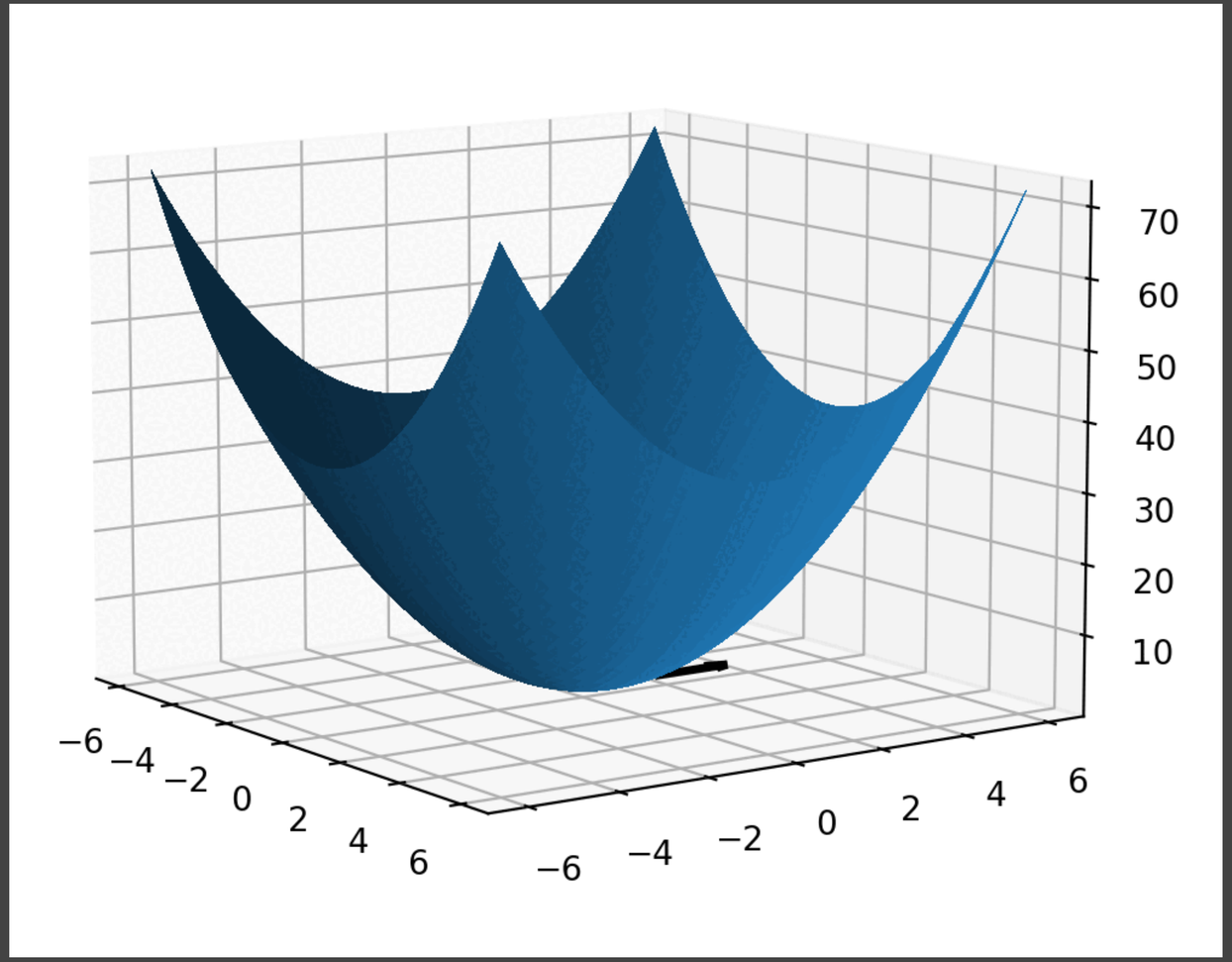
Optimization Theory - Differentiation and Gradients

- Multivariate functions
  - $g : R^2 \rightarrow R$
  - Multiple input, single output
  - Partial differentiation with each of multiple variables

$$g(x) = x^T \cdot x$$

$$\nabla g(x) = \left[\frac{\partial g(x)}{\partial x_1}, \frac{\partial g(x)}{\partial x_2}\right]^T$$

$$\nabla g(x) = \left[2x_1, 2x_2\right]^T$$



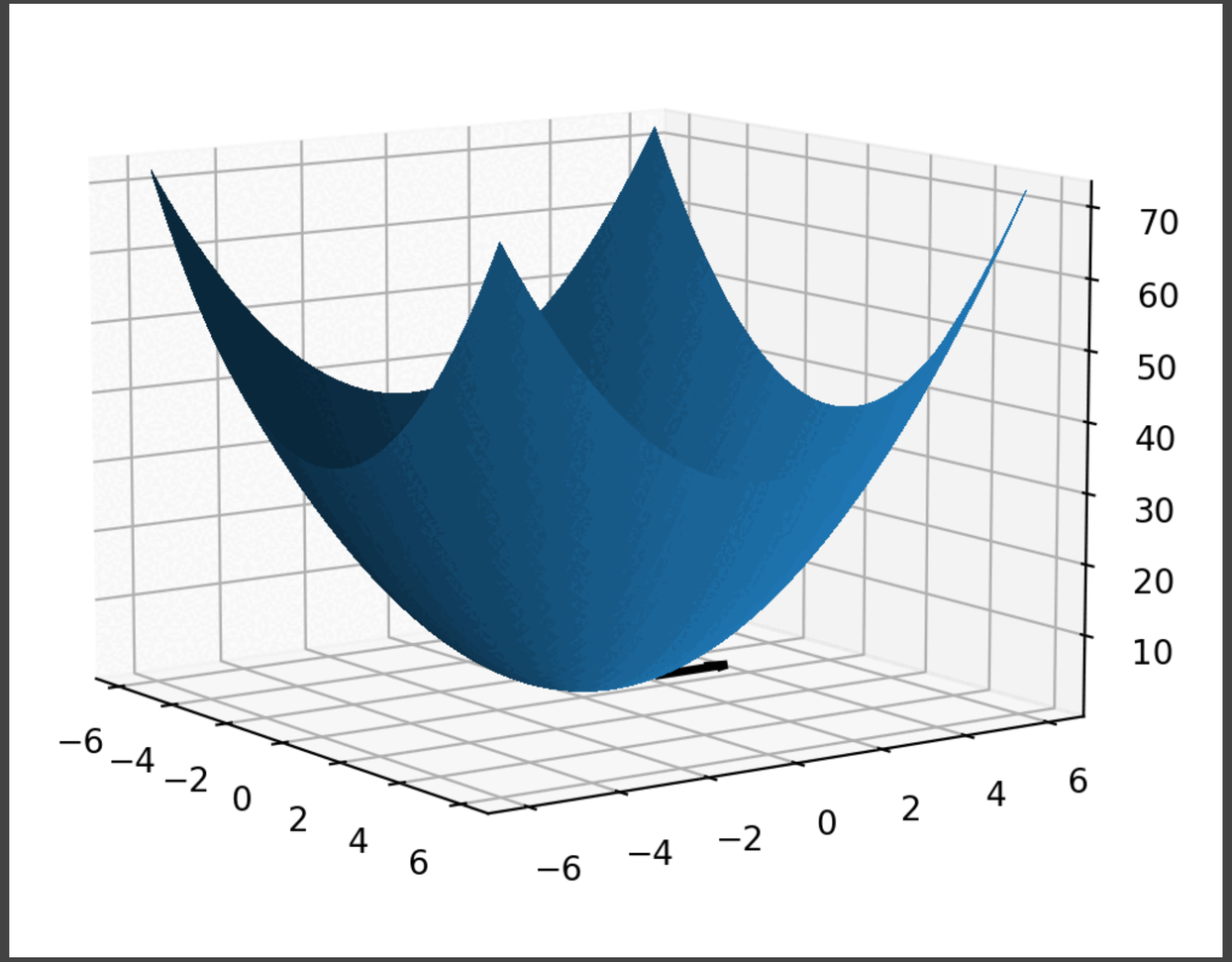
Optimization Theory - Differentiation and Gradients

- Multivariate functions
  - $g : R^2 \rightarrow R$
  - Multiple input, single output
  - Partial differentiation with each of multiple variables

$$g(x) = x^T \cdot x$$

$$\nabla g(x) = \left[\frac{\partial g(x)}{\partial x_1}, \frac{\partial g(x)}{\partial x_2} \right]^T$$

$$\nabla g(x) = \left[2x_1, 2x_2 \right]^T$$



Optimization Theory - Gradient Descent

- Reformulation
  - Bias and weight in one parameter vector
  - Updating model and cost functions to remove bias extension in model and cost functions

$$\hat{w} = \left[b,\ w\right]^T \quad \hat{x} = [1,\ x]^T$$

$$f_{\hat{w}}(x) = \hat{x}^T \cdot \hat{w}$$

$$\mathscr{L}(\hat{w}) = \frac{1}{N}\sum_{i=1}^{N}(f_{\hat{w}}(x^i) - y^i)^2$$

Optimization Theory - Gradient Descent

- Reformulation
  - Bias and weight in one parameter vector
  - Updating model and cost functions to remove bias extension in model and cost functions

$$\hat{w} = \left[b,\ w\right]^T \quad \hat{x} = [1,\ x]^T$$

$$f_{\hat{w}}(x) = \hat{x}^T \cdot \hat{w}$$

$$\mathscr{L}(\hat{w}) = \frac{1}{N}\sum_{i=1}^{N}(f_{\hat{w}}(x^i) - y^i)^2$$

Optimization Theory - Gradient Descent

- Reformulation
  - Bias and weight in one parameter vector
  - Updating model and cost functions to remove bias extension in model and cost functions

$$\hat{w} = \left[b,\ w\right]^T \quad \hat{x} = [1,\ x]^T$$

$$f_{\hat{w}}(x) = \hat{x}^T \cdot \hat{w}$$

$$\mathscr{L}(\hat{w}) = \frac{1}{N}\sum_{i=1}^{N}(f_{\hat{w}}(x^i) - y^i)^2$$

- Gradient Descent
  - Iterative optimization algorithm
  - Based on gradient vector

$$Init\ \ \hat{w}^0 = \left[b^0,\ w^0\right]$$

$$while(stop\ \ condition)$$

$$\hat{w}^{k+1} = \hat{w}^k - \alpha \nabla \mathscr{L}(\hat{w}^k)$$

$$k = k + 1$$

Optimization Theory - Gradient Descent

- Reformulation
  - Bias and weight in one parameter vector
  - Updating model and cost functions to remove bias extension in model and cost functions

$$\hat{w} = \left[b,\ w\right]^T \quad \hat{x} = [1,\ x]^T$$

$$f_{\hat{w}}(x) = \hat{x}^T \cdot \hat{w}$$

$$\mathscr{L}(\hat{w}) = \frac{1}{N}\sum_{i=1}^{N}(f_{\hat{w}}(x^i) - y^i)^2$$

- Gradient Descent
  - Iterative optimization algorithm
  - Based on gradient vector

$$Init \ \ \hat{w}^0 = \left[b^0,\ w^0\right]$$

$$while(stop \ \ condition)$$

$$\hat{w}^{k+1} = \hat{w}^k - \alpha \nabla \mathscr{L}(\hat{w}^k)$$

$$k = k + 1$$

Optimization Theory - Gradient Descent

- Reformulation
  - Bias and weight in one parameter vector
  - Updating model and cost functions to remove bias extension in model and cost functions

$$\hat{w} = \left[b,\ w\right]^T \quad \hat{x} = [1,\ x]^T$$

$$f_{\hat{w}}(x) = \hat{x}^T \cdot \hat{w}$$

$$\mathscr{L}(\hat{w}) = \frac{1}{N}\sum_{i=1}^{N}(f_{\hat{w}}(x^i) - y^i)^2$$

- Gradient Descent
  - Iterative optimization algorithm
  - Based on gradient vector

$$Init\ \ \hat{w}^0 = \left[b^0,\ w^0\right]$$

$$while(stop\ \ condition)$$

$$\hat{w}^{k+1} = \hat{w}^k - \alpha \nabla \mathscr{L}(\hat{w}^k)$$

$$k = k + 1$$

Optimization Theory - Gradient Descent

- Reformulation
  - Bias and weight in one parameter vector
  - Updating model and cost functions to remove bias extension in model and cost functions

$$\hat{w} = \left[b,\ w\right]^T \quad \hat{x} = [1,\ x]^T$$

$$f_{\hat{w}}(x) = \hat{x}^T \cdot \hat{w}$$

$$\mathscr{L}(\hat{w}) = \frac{1}{N}\sum_{i=1}^{N}(f_{\hat{w}}(x^i) - y^i)^2$$

- Gradient Descent
  - Iterative optimization algorithm
  - Based on gradient vector

$$Init\ \ \hat{w}^0 = \left[b^0,\ w^0\right]$$

$$while(stop\ \ condition)$$

$$\hat{w}^{k+1} = \hat{w}^k - \alpha \nabla \mathscr{L}(\hat{w}^k)$$

$$k = k + 1$$

Optimization Theory - Gradient Descent

- Reformulation
  - Bias and weight in one parameter vector
  - Updating model and cost functions to remove bias extension in model and cost functions

$$\hat{w} = \left[b,\ w\right]^T \quad \hat{x} = [1,\ x]^T$$

$$f_{\hat{w}}(x) = \hat{x}^T \cdot \hat{w}$$

$$\mathscr{L}(\hat{w}) = \frac{1}{N}\sum_{i=1}^{N}(f_{\hat{w}}(x^i) - y^i)^2$$

- Gradient Descent
  - Iterative optimization algorithm
  - Based on gradient vector

$$Init\ \ \hat{w}^0 = \left[b^0,\ w^0\right]$$

$$while(stop\ \ condition)$$

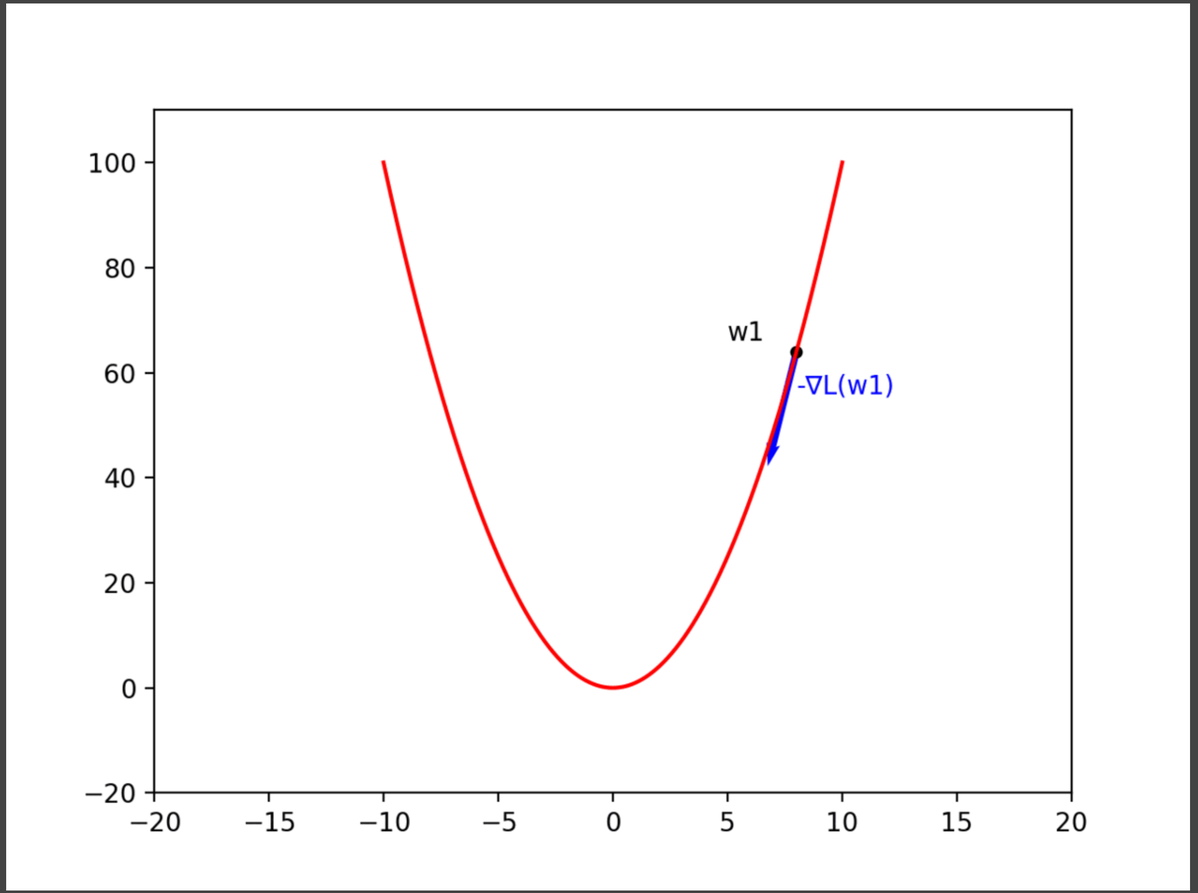
$$\hat{w}^{k+1} = \hat{w}^k - \alpha \nabla \mathscr{L}(\hat{w}^k)$$

$$k = k + 1$$

Optimization Theory - Gradient Descent

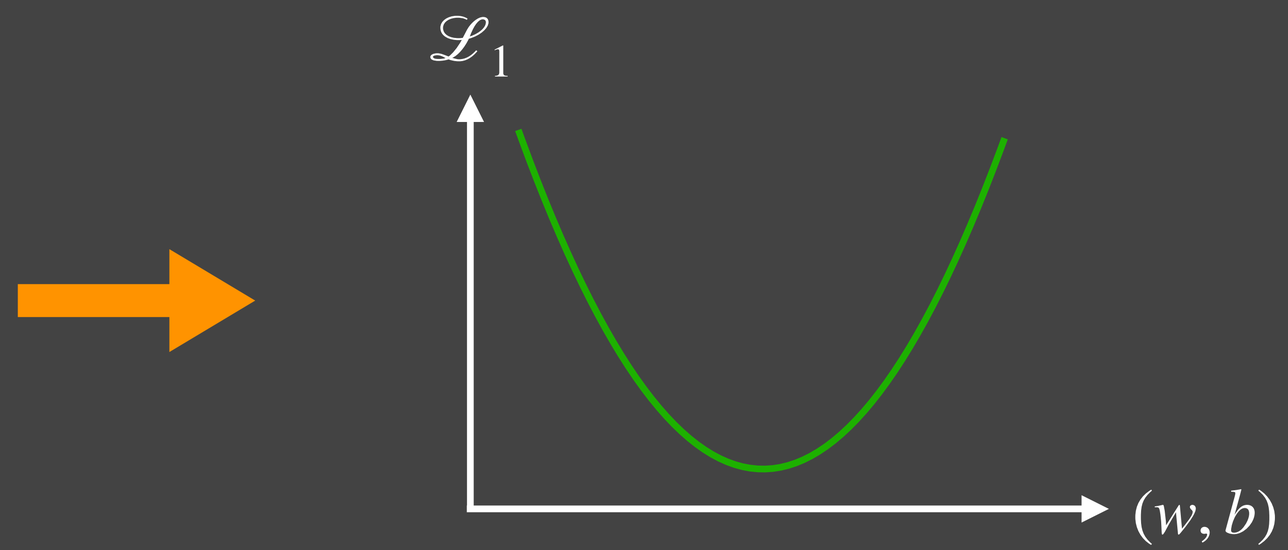
- Gradient descent optimization
  - Parabola in red, our loss function
  - W_1 as initial set of parameters and starting point
  - Moving along negative direction of gradient vector
  - Closer to global minima, smaller gradient vector
  - Step length to control how fast we converge

$$\hat{w}^{k+1} = \hat{w}^k - \alpha \nabla \mathscr{L}(\hat{w}^k)$$



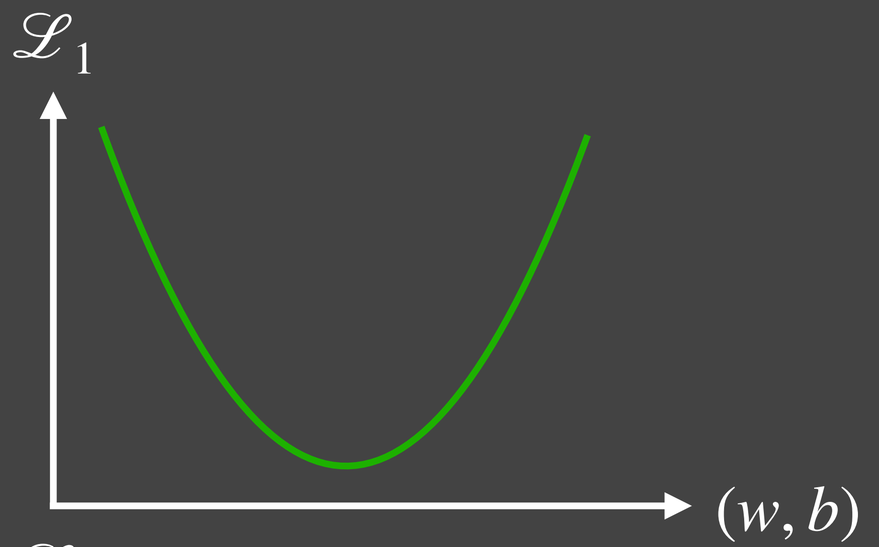
Optimization Theory - Convexity of Loss Functions

- Unique Minima
  - Global Minima
  - Optimal parameter set
  - Convex function

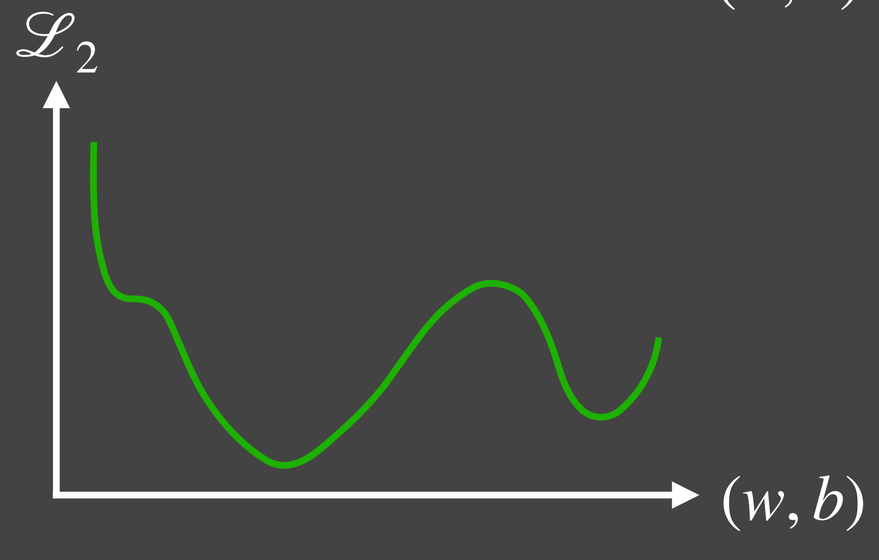


Optimization Theory - Convexity of Loss Functions

- Unique Minima
  - Global Minima
  - Optimal parameter set
  - Convex function

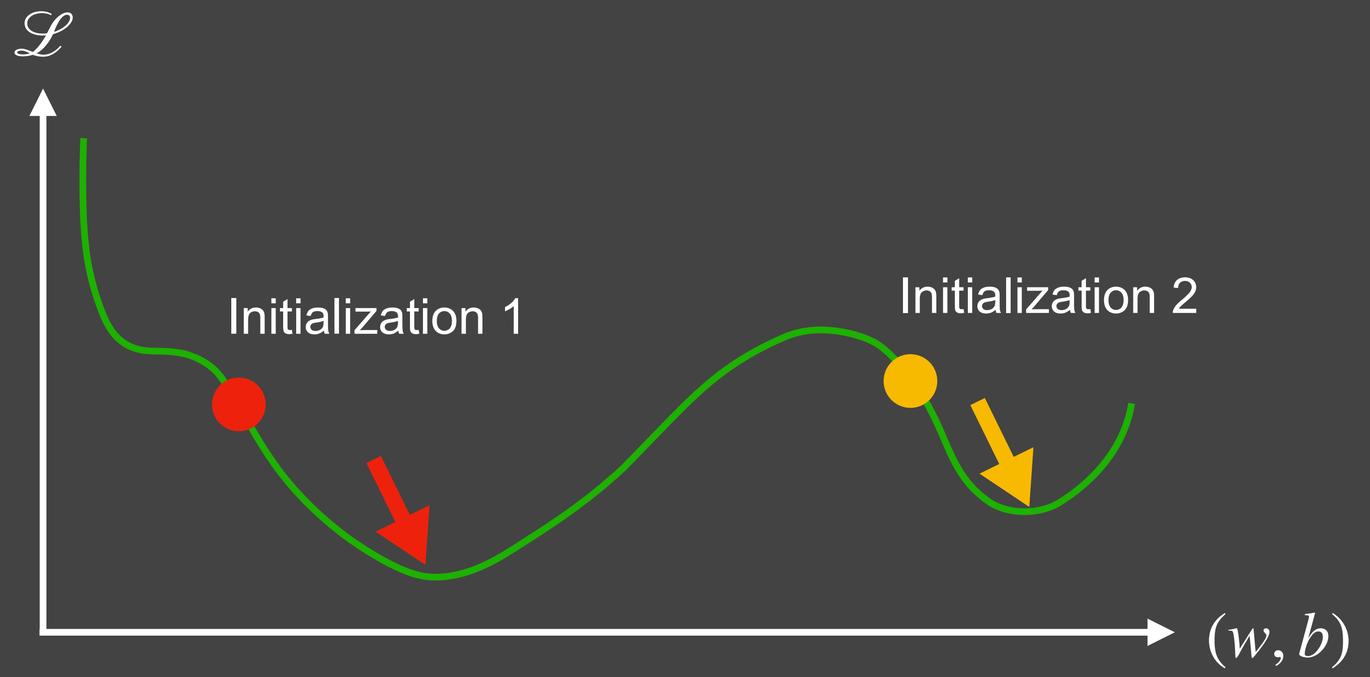


- Multiple Minimas
  - Global minima + Local minimas
  - 1 Optimal + Many sub-optimal solutions
  - Non-convex function



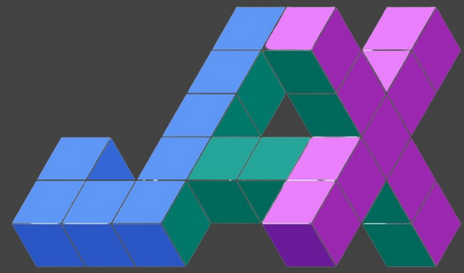
Optimization Theory - Convexity of Loss Functions

- Gradient descent on non-convex functions
  - Solution changes depending on initialization
  - Executing algorithm multiple times
- Convex optimization problems
  - Linear regression
  - Linear classification



JAX - Fundamentals

- Numerical computing library
  - Differentiable NumPy + Autograd
  - No usage of Tensor
  - XLA Compilation



- High level functionalities
  - Auto Vectorization
  - JIT Compilation

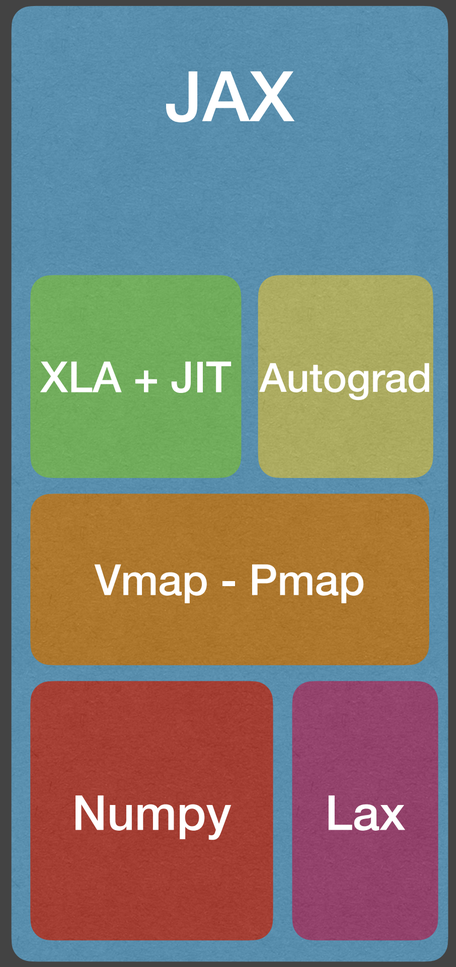
JAX - Numpy

JAX - Numpy

- Embedded in JAX
- Immutable
- Accelerator Support
- Vmap and JIT Support

Standard Numpy

- Standalone Library
- Mutable
- No GPU or TPU
- No Vmap or JIT



JAX - Composable Function Transformations

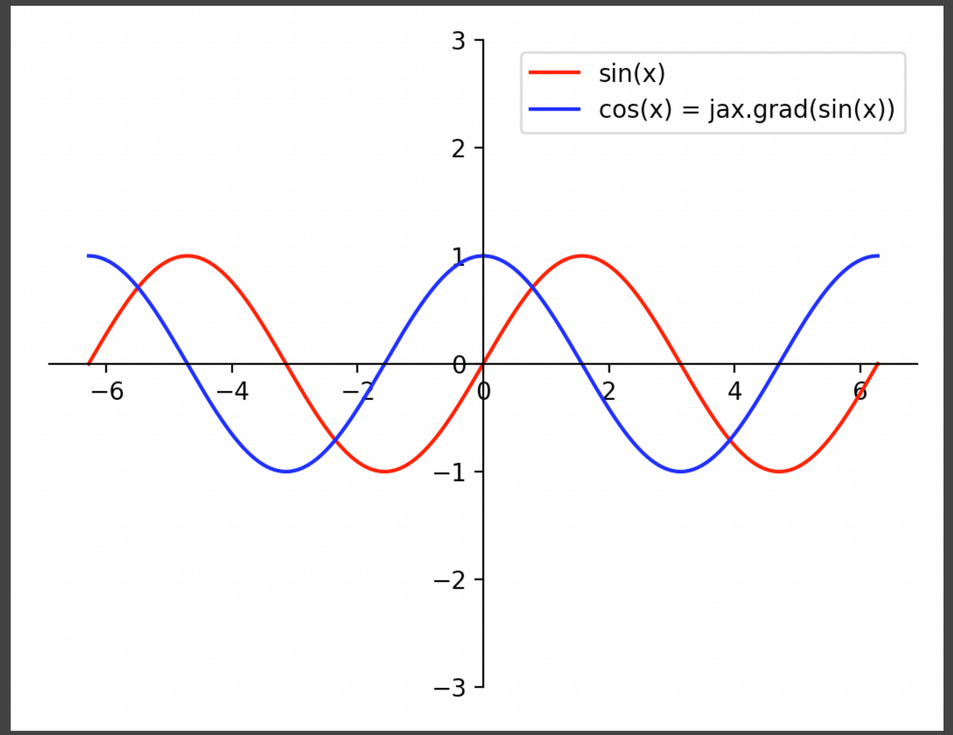
- Auto-Gradient
  - Differentiating a python function
  - Returning derivative as another function
  - $jax.grad(.)$

```
x = 0
y = jax.numpy.sin(x)
cos = jax.grad(jax.numpy.sin)
y = cos(x)
```

$$f(x) = sin(x)$$

$$\frac{df(x)}{dx} = cos(x)$$

$$jax.grad \approx \frac{d}{dx}$$



JAX - Composable Function Transformations

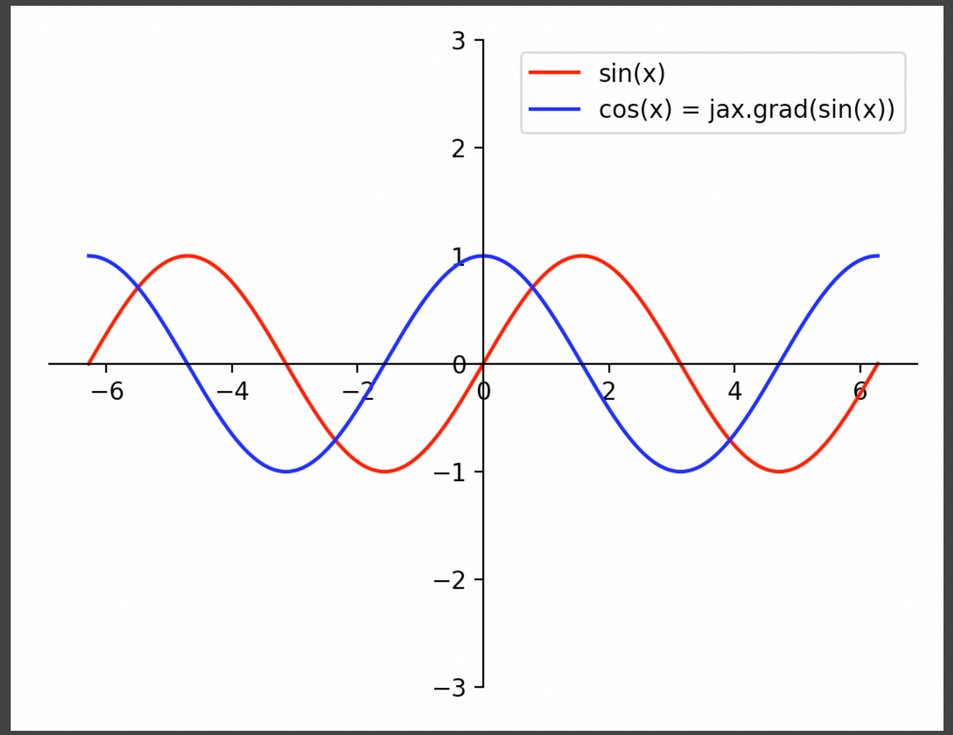
- Auto-Gradient
  - Differentiating a python function
  - Returning derivative as another function
  - $jax.grad(.)$

```
x = 0
y = jax.numpy.sin(x)
cos = jax.grad(jax.numpy.sin)
y = cos(x)
```

$$f(x) = sin(x)$$

$$\frac{df(x)}{dx} = cos(x)$$

$$jax.grad \approx \frac{d}{dx}$$



JAX - Composable Function Transformations

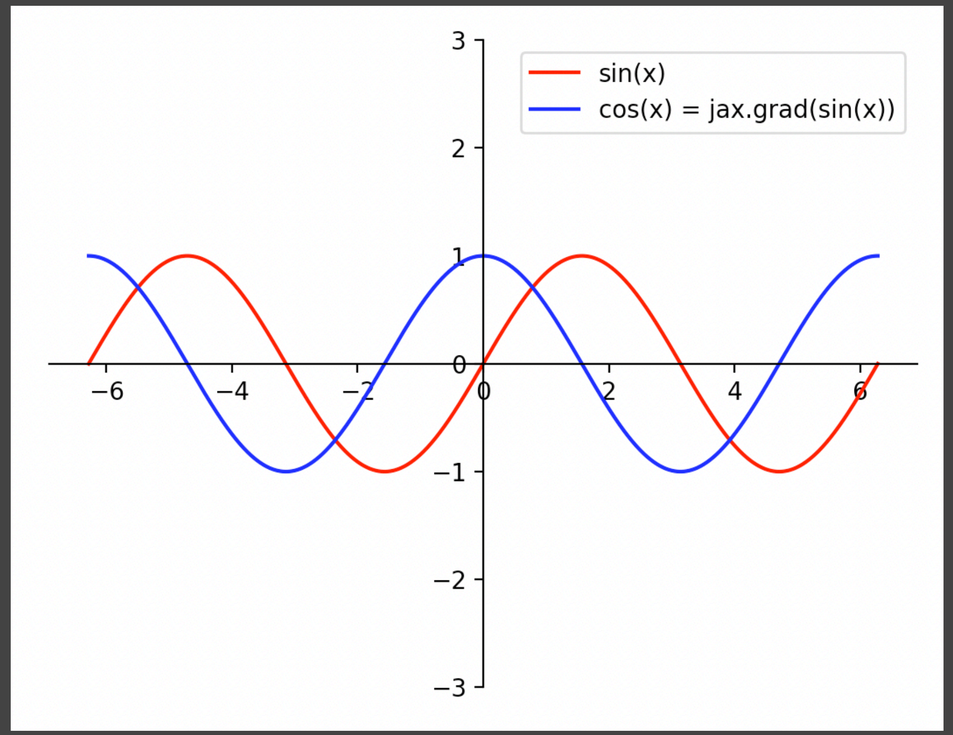
- Auto-Gradient
  - Differentiating a python function
  - Returning derivative as another function
  - $jax.grad(.)$

```
x = 0
y = jax.numpy.sin(x)
cos = jax.grad(jax.numpy.sin)
y = cos(x)
```

$$f(x) = sin(x)$$

$$\frac{df(x)}{dx} = cos(x)$$

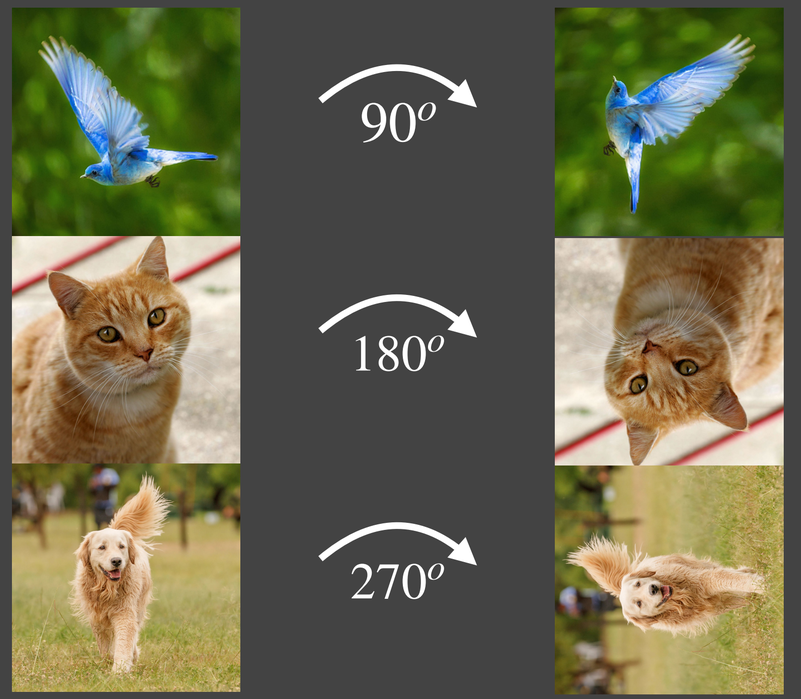
$$jax.grad \approx \frac{d}{dx}$$



JAX - Composable Function Transformations

- Auto-Vectorization
  - Rotating multiple images at the same time
  - Different rotation angle for each image
  - $jax.vmap(.)$

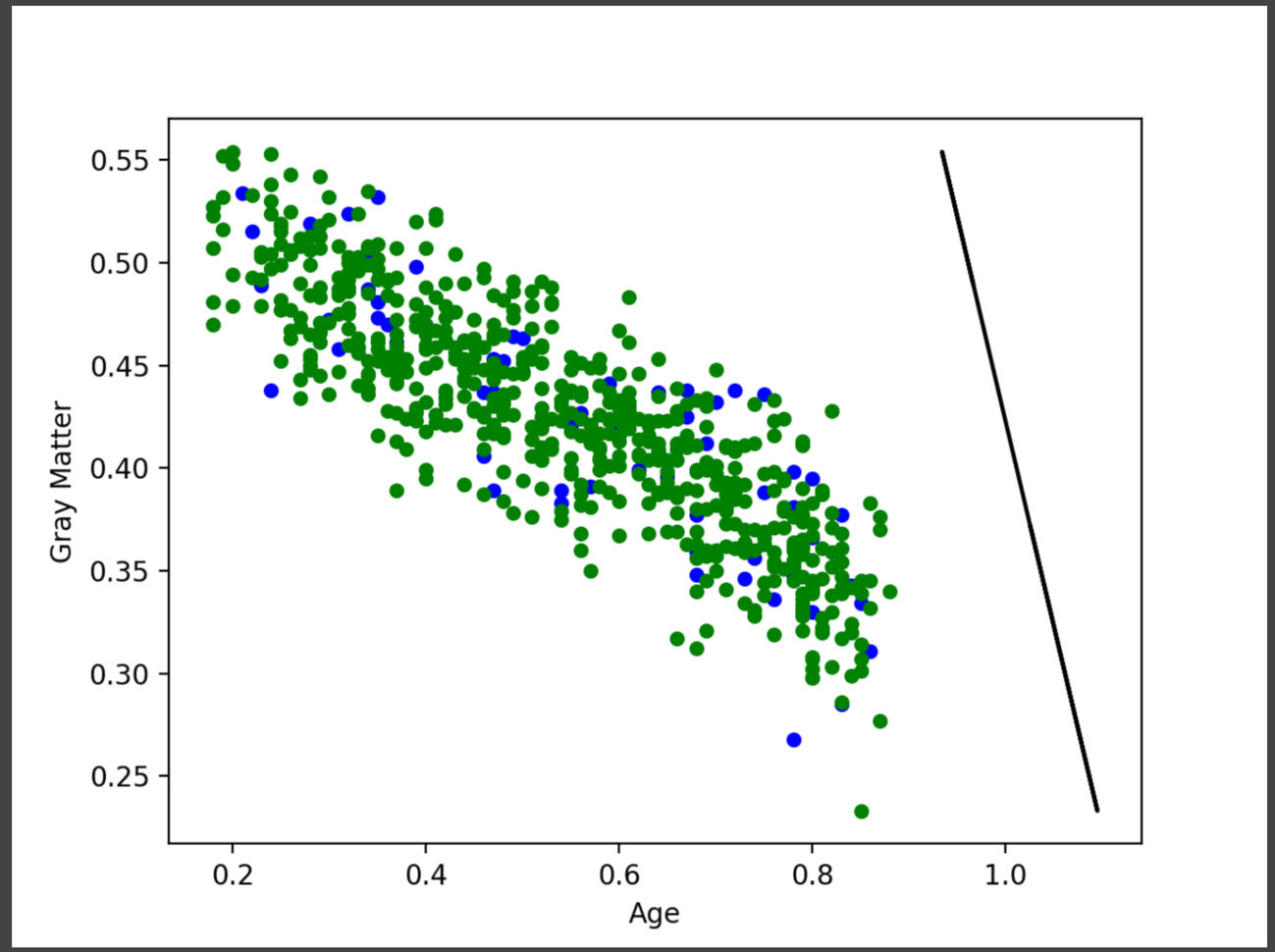
```
vec_rotate = jax.vmap(pix.rotate, in_axes=(0, 0), out_axes=0)
rotated_imgs = vec_rotate(imgs, radians)
```




```
rotated_img = pix.rotate(img, radian)
```

Coding Session - Brain Age Prediction

- Data Representation
  - Only 1 feature → Gray Matter
  - Samples → Training Data
  - Samples → Test Data
  - Older people with less gray matter
- Training Process
  - 15000 number of iterations
  - Min MSE Loss: 0.0091
  - Max R2 Score: 0.7344
- Testing Process
  - Min MSE Loss: 0.0105
  - Max R2 Score: 0.6980



References and Resources

- Additional JAX Resources and Tutorials
  - https://www.kaggle.com/code/goktugguvercin/introduction-to-jax
- Coding Session - Resource Colab File and Dataset
  - https://drive.google.com/drive/folders/1xbc07s6rLYEcb4Iy0MP8diB3U39uvfoz?usp=sharing
  - https://github.com/GoktugGuvercin/Machine-Learning-and-Optimization-Theory