An introduction to classical machine learning

Amira Abbas

IBM Quantum, University of KwaZulu-Natal

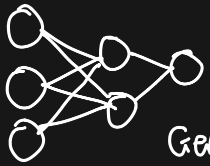
$x \to \boxed{f(x;\theta)} \to$ CAT DOG

data ⇒ model → cost

classification

TRAIN!

optimize

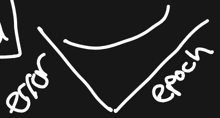


?

Generalization

$\theta = \begin{bmatrix} w_1 & w_2 & \cdots \\ \vdots & & \\ & & w_d \end{bmatrix}$

test set



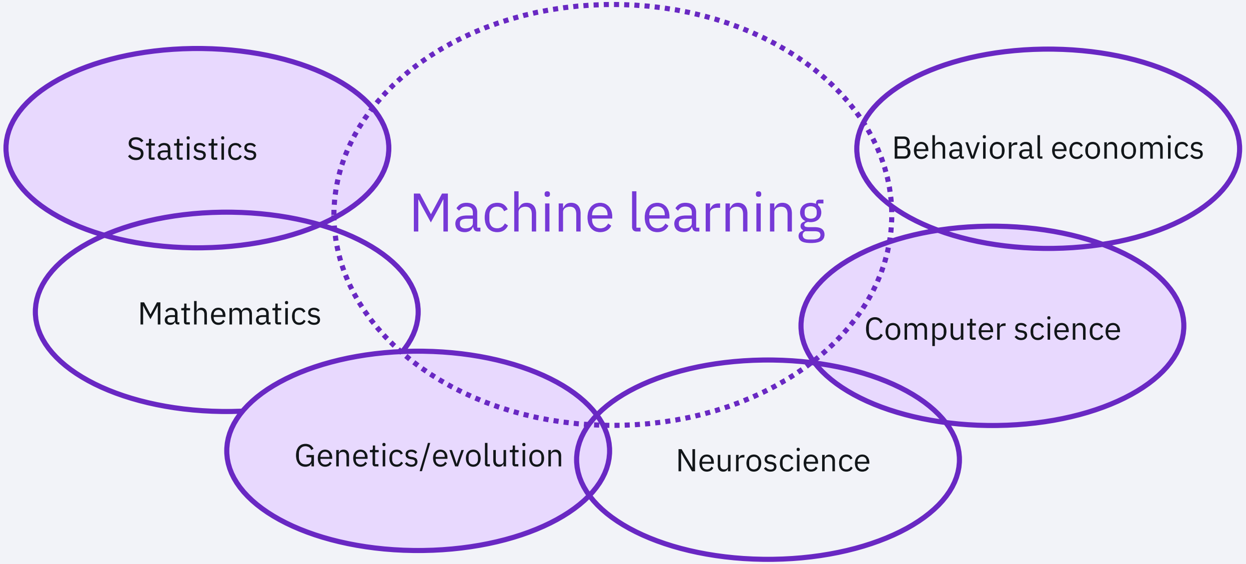
Regression

$X = \begin{bmatrix} x_1 \\ \vdots \\ x_d \end{bmatrix}$

$y \in [-1, 1]$

Qiskit

Machine learning



Machine learning 101

Qiskit

"Learning patterns from data in order to draw inferences"

Machine learning 101

Function approximation and optimization

$g(x)$

$f(\bar{x}; \theta)$

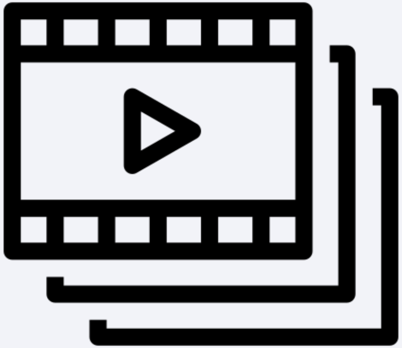
model

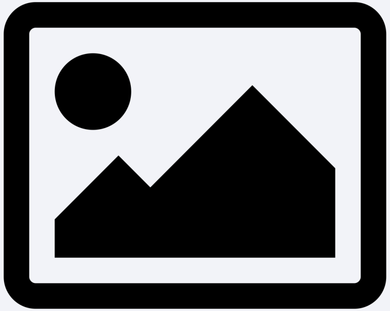
Machine learning ingredients

DATA

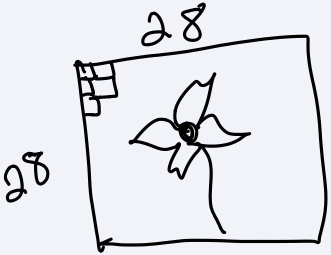
DATA

- Pictures
- Videos
- Spreadsheets
- Numerical/Categorical
- . . .





DATA



pixel $[-1, 1]$

dark

light

$$\text{data} = \underbrace{(\vec{x}_1, \ \dots \ \vec{x}_n)} \quad n \text{ samples}$$

$$X_{full} = \begin{array}{c|c} \left[\begin{array}{cccc} & & & \\ & & & \\ & & & \\ & & & \end{array}\right] & \begin{array}{c} 1 \\ 2 \\ \vdots \\ 784 \to d \end{array} \\ \begin{array}{cccc} \vec{x}_1 & \vec{x}_2 & \dots & \vec{x}_n \end{array} & \end{array}$$

$$\vec{x}_1 = \begin{array}{c} \\ \\ \\ \\ 784 \end{array} \left.\left[\begin{array}{c} 0.8 \\ 0.2 \\ \vdots \\ \vdots \\ -0.5 \end{array}\right]\right\} \text{features of the data} \to \text{dimension}$$

DATA

MODEL

$\vec{x} \longrightarrow \boxed{f(\vec{x};\theta)} \longrightarrow \hat{y} \mid f(\vec{x};\theta)$

sets

prediction

label

output / distribution

DATA

MODEL

COST

Score $\leftarrow$ $C(\underbrace{f(\vec{x};\theta)}_{\hat{y}}, \underbrace{\text{correct}}_{y})$

$$MSE := (\hat{y} - y)^2$$

CAT $\rightarrow$ 1

NO CAT $\rightarrow$ 0

$(1-0)^2 = 1$

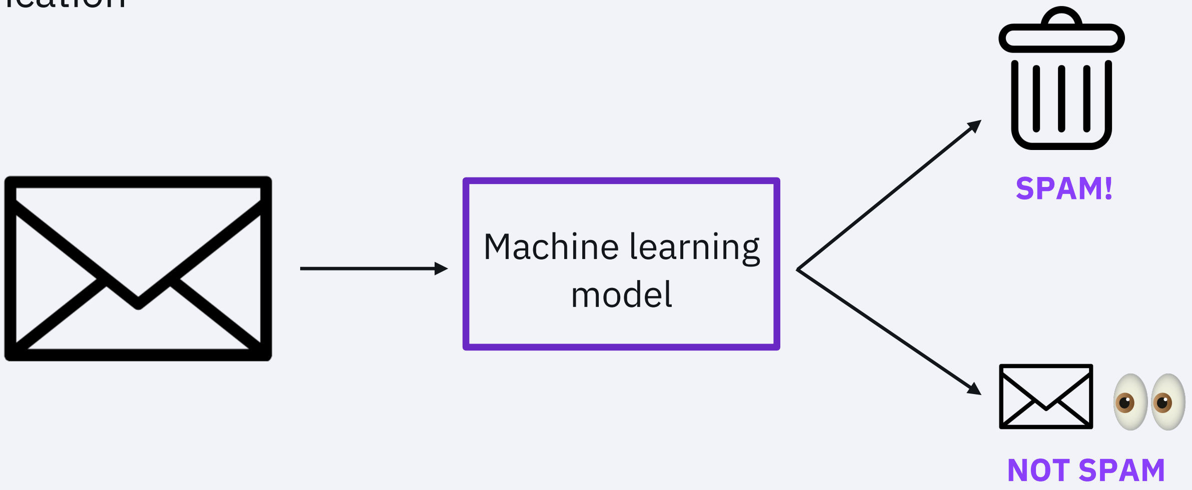
$(0-0)^2 = 0$

Machine learning examples

Machine learning examples

Qiskit

Classification

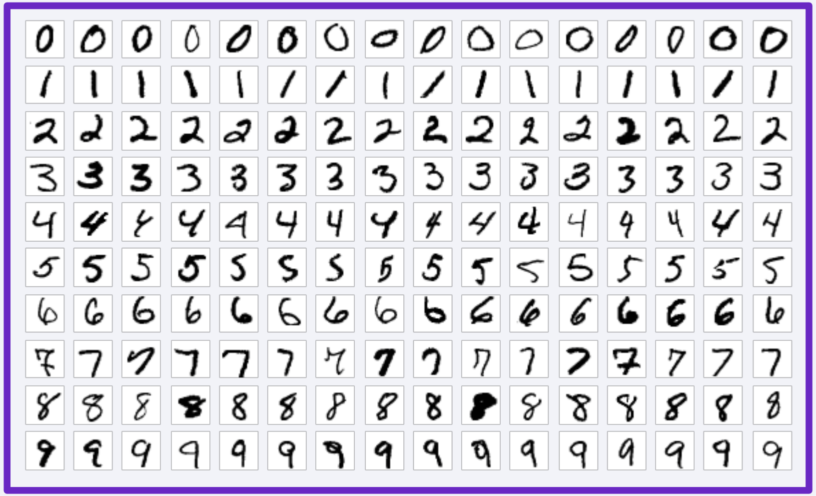


Task: use emails seen so far to produce rules to predict whether *future* emails are spam or not

Machine learning examples

Classification

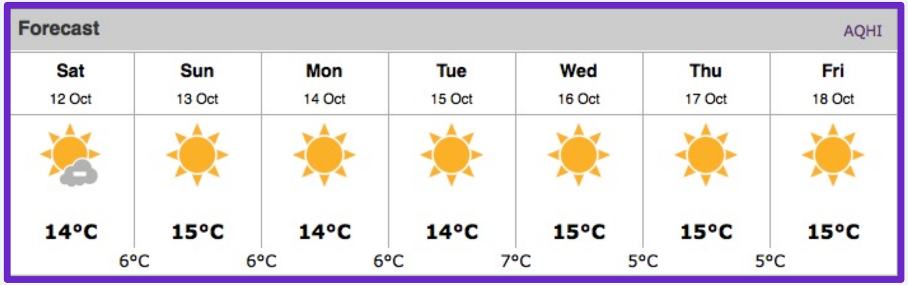
MNIST



Learn patterns from data and classify new data

Machine learning examples

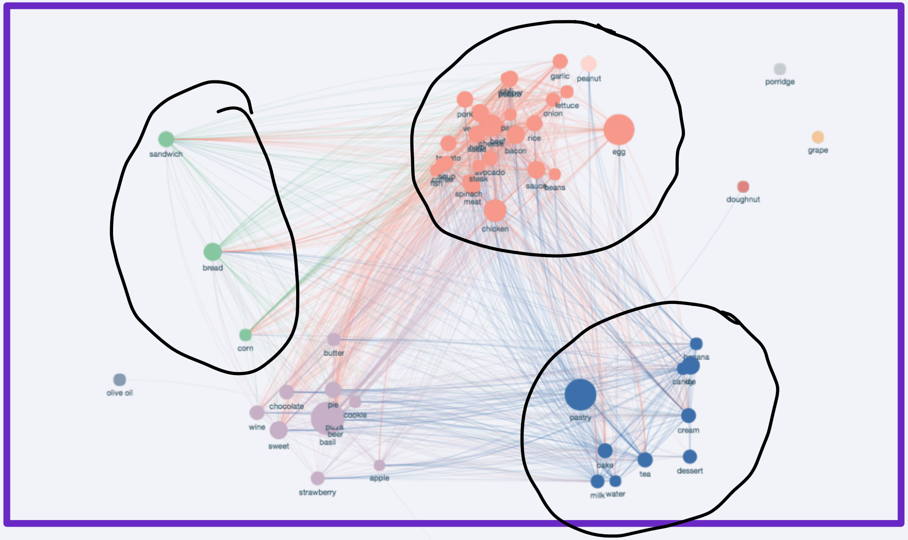
Regression



Predicting values based on relevant information

Machine learning examples

Clustering



Identify consumers with similar behavior

Machine learning examples

Qiskit

Generate data

Learn patterns or structure from data and reproduce them

Machine learning 101

Function approximation and optimization

Machine learning 101

Function approximation and optimization

Supervised setting

Machine learning 101

Function approximation and optimization

Supervised setting

$$(\vec{x}_1, y_1), \ldots (\vec{x}_n, y_n)$$

CAT $\rightarrow$ 1

DOG $\rightarrow$ -1

$$\vec{x}_1 = \begin{bmatrix} \\ \\ \end{bmatrix} \qquad y_1 = 1$$

Machine learning 101

Function approximation and optimization

Unsupervised setting

$$g(\vec{x})$$

$$f(\vec{x};\theta)$$

Machine learning 101

Function approximation and optimization

Unsupervised setting

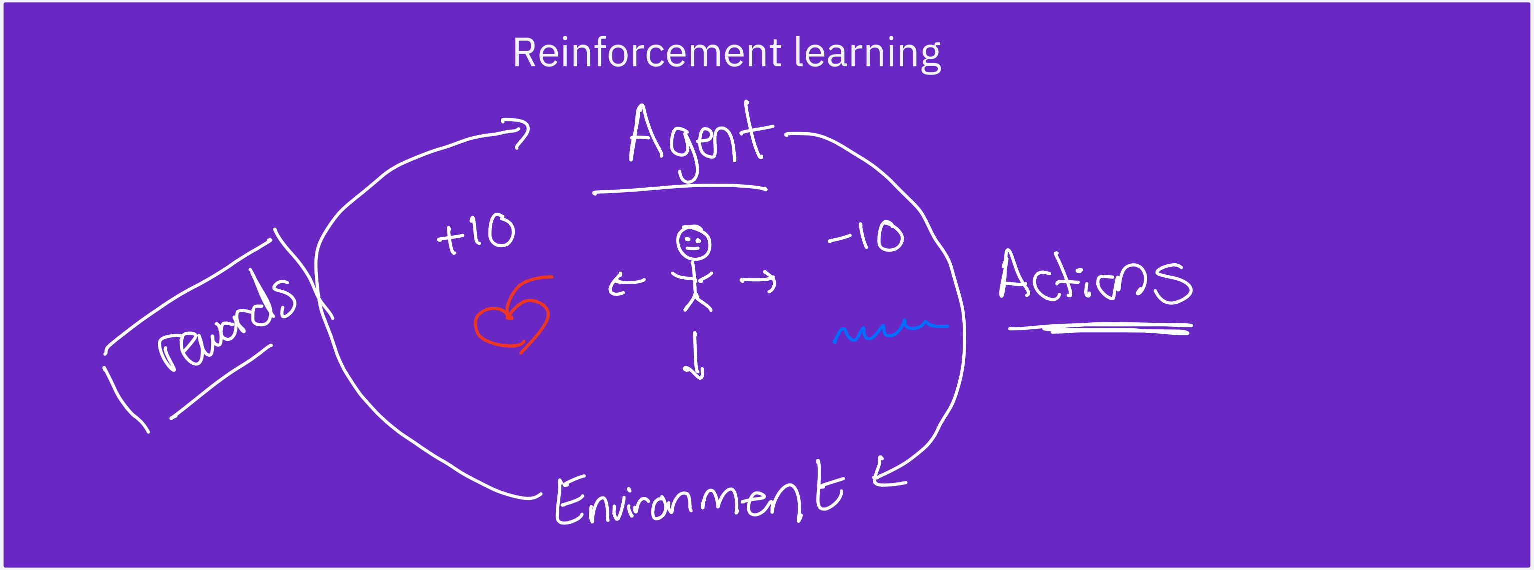
Machine learning 101

Function approximation and optimization

Unsupervised setting

Machine learning 101

Function approximation and optimization



Machine learning 101

Qiskit

Function approximation and optimization

Reinforcement learning

Machine learning ingredients

Supervised setting

Machine learning ingredients

Supervised setting

Classify the type of flower based on certain *features*

Machine learning ingredients

Qiskit

Supervised setting

Classify the type of flower based on certain *features*

1. DATA

y: roses $+1$, daisies -1

$$\vec{x} = \begin{bmatrix} \text{length petals} \\ \text{width bud} \end{bmatrix} \quad d = 2$$

$$\vec{x}_1 = \begin{bmatrix} 0.8 \\ 0.1 \end{bmatrix} \qquad y_1 = +1$$

Machine learning ingredients

Supervised setting

Classify the type of flower based on certain *features*

1. DATA

dataset → 120 samples

train

100

test

20

Machine learning ingredients

Qiskit

Supervised setting

Classify the type of flower based on certain *features*

2. MODEL

$f(\vec{x}; \theta) \to \hat{y}$

$d = \#\text{ features} = 2$

$\vec{x} = \begin{bmatrix} - \\ - \end{bmatrix}$

linear model

$\boxed{\hat{y} = \theta^T \cdot \vec{x}}$ $\leftarrow$ $(\vec{x}_1, y_1), (\vec{x}_2, y_2) \ldots$

Machine learning ingredients

Qiskit

Supervised setting

Classify the type of flower based on certain *features*

3. COST

For 100 samples (training)

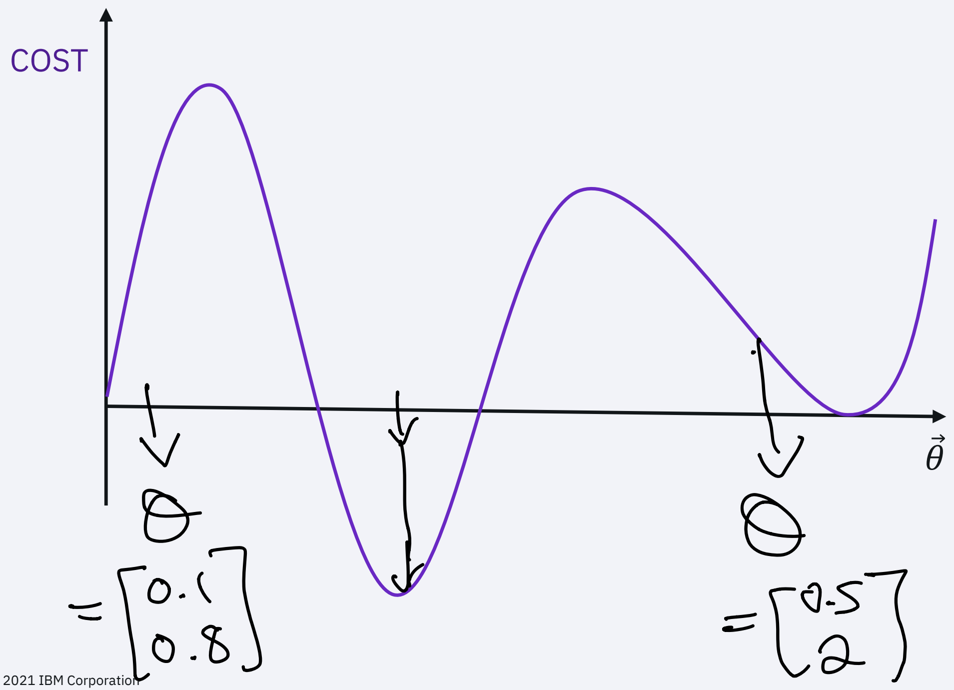
$$\text{cost} := \frac{1}{100} \sum_{i=1}^{100} (\hat{y}_i - y_i)^2 \longrightarrow \text{minimize}$$

$$\hat{y}_i = \theta^T \vec{x}_i$$

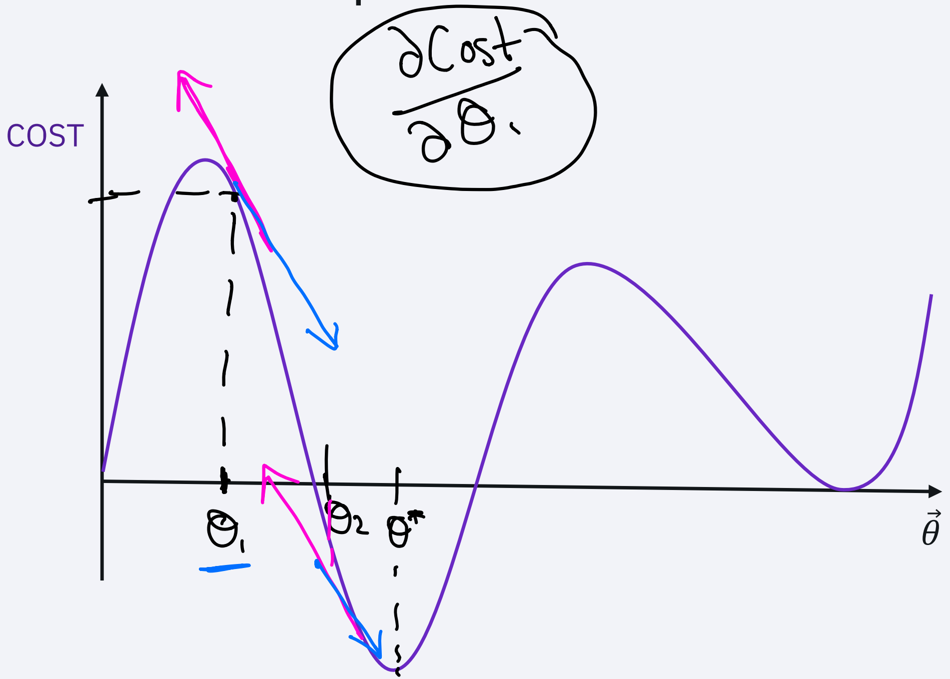
How do we optimize our model?

How do we optimize our model?

Qiskit



How do we optimize our model?



Gradient-descent

How do we optimize our model?

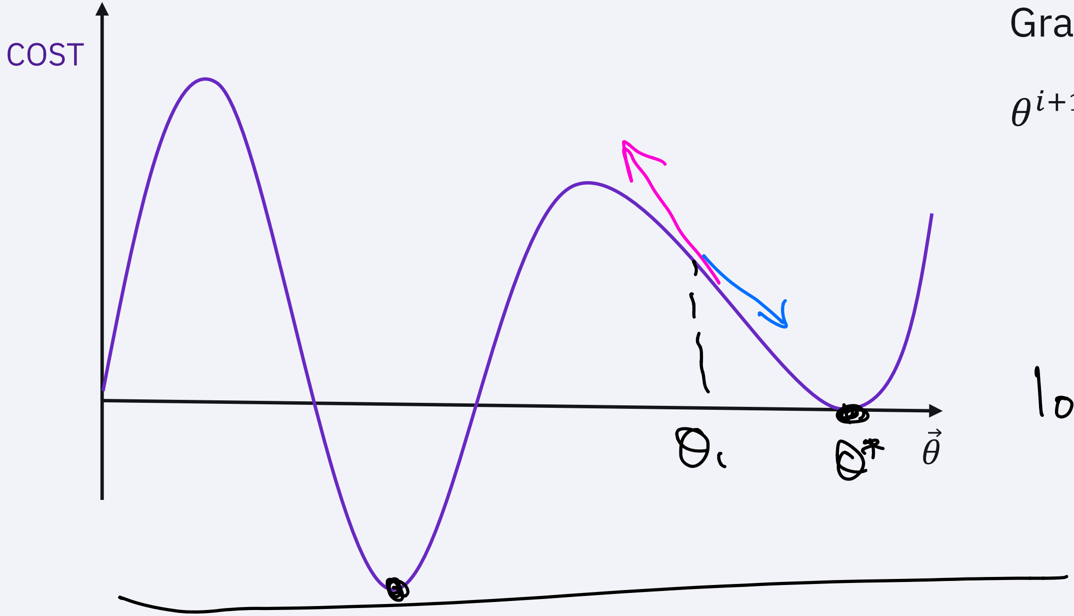
Qiskit

Gradient-descent

$$\theta^{i+1} = \theta^{i} - \eta \frac{\partial C}{\partial \theta^{i}}$$

learning rate

How do we optimize our model?



Gradient-descent

$$\theta^{i+1} = \theta^{i} - \eta \frac{\partial C}{\partial \theta^{i}}$$

local minimum

How do we optimize our model?

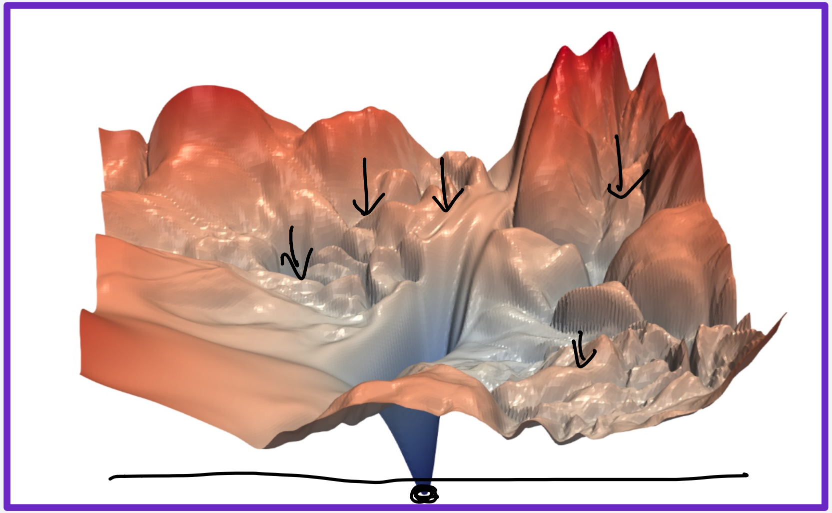


Image source: Li, H., Xu, Z., Taylor, G., Studer, C., & Goldstein, T. (2017). Visualizing the loss landscape of neural nets. arXiv preprint arXiv:1712.09913.

Cost function landscapes can be very complicated!

Stochastic gradient-descent methods work well to avoid local minima in practice

Training the model

Training the model

Training data: X_{train}

$(\vec{x}_1, \ldots, \vec{x}_{100}) \rightarrow$

Training the model

Training data: X_{train} $\longrightarrow$ Model: $f(X_{train}; \vec{\theta}) = \hat{y}$

$\hat{y}_1, \ldots, \hat{y}_{100}$

Training the model

Training data: X_{train} $\longrightarrow$ Model: $f(X_{train}; \vec{\theta}) = \hat{y}$ $\longrightarrow$ Cost: $C(f(X_{train}; \vec{\theta}), y)$

θ random

$\hat{y}_i$ to y

Training the model

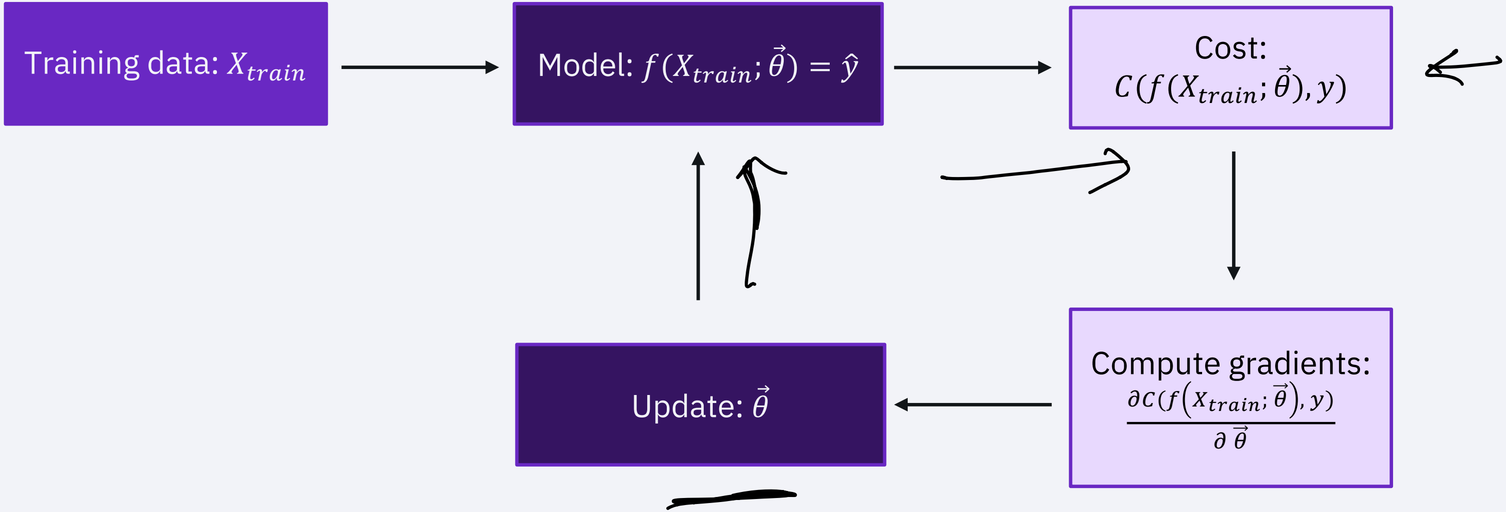
Qiskit

Training data: X_{train} → Model: $f(X_{train};\vec{\theta}) = \hat{y}$ → Cost: $C(f(X_{train};\vec{\theta}), y)$

↓

Compute gradients: $\frac{\partial C\left(f\left(X_{train};\vec{\theta}\right), y\right)}{\partial\,\vec{\theta}}$

Training the model



How do we assess our model?

How do we assess our model?

Test data: X_{test}

$\vec{x}_{101}, \ldots \vec{x}_{120} \longrightarrow$

How do we assess our model?

Qiskit

Test data: X_{test} $\longrightarrow$ Model: $f(X_{test}; \vec{\theta}^*) = \hat{y}$

θ^*

$\hat{y}_{101}, \ldots \hat{y}_{120}$

How do we assess our model?

Test data: X_{test} → Model: $f(X_{test};\vec{\theta}^*) = \hat{y}$ → Cost: $C(f(X_{test};\vec{\theta}), y)$

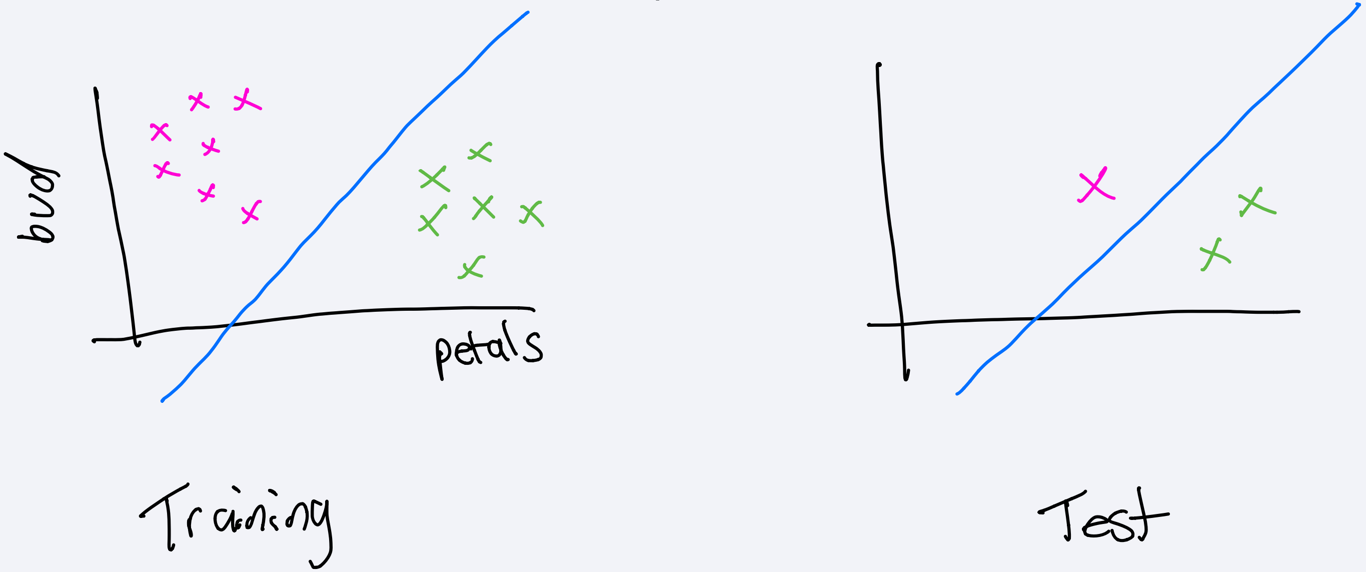
?

Choosing the right model

Generalization

Choosing the right model

Generalization: how well will our model perform on unseen data?



Choosing the right model

Generalization: how well will our model perform on unseen data?

Choosing the right model

Qiskit

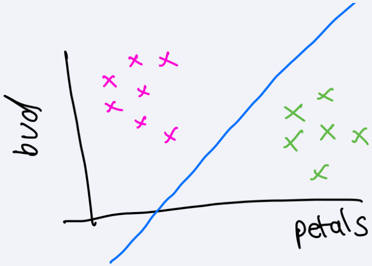
Generalization: how well will our model perform on unseen data?

?

Choosing the right model

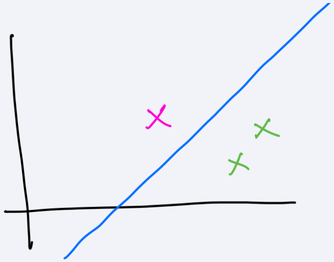
Generalization: how well will our model perform on unseen data?

Training data



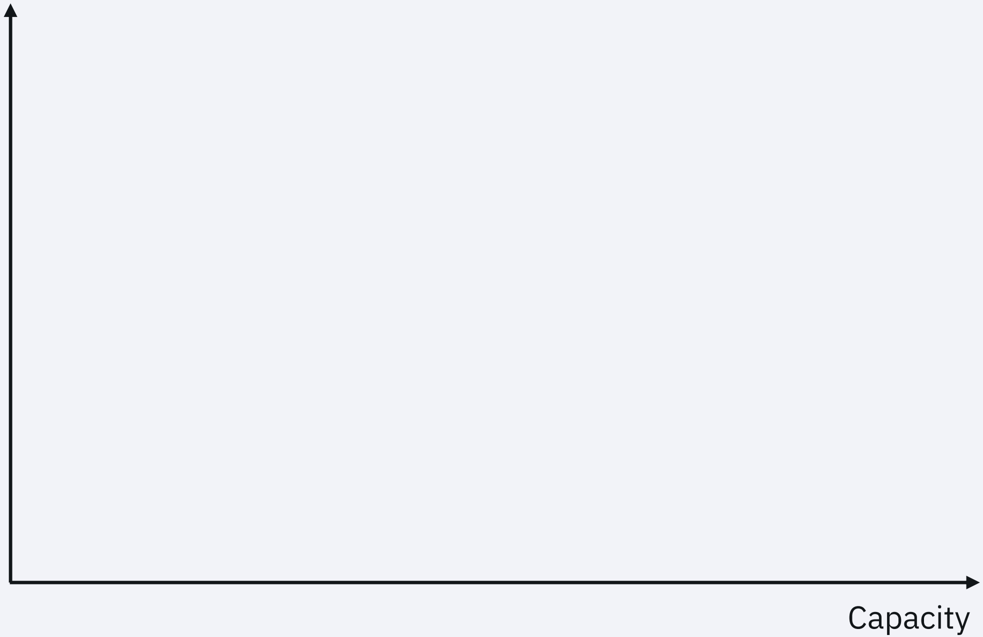
Training

Test data

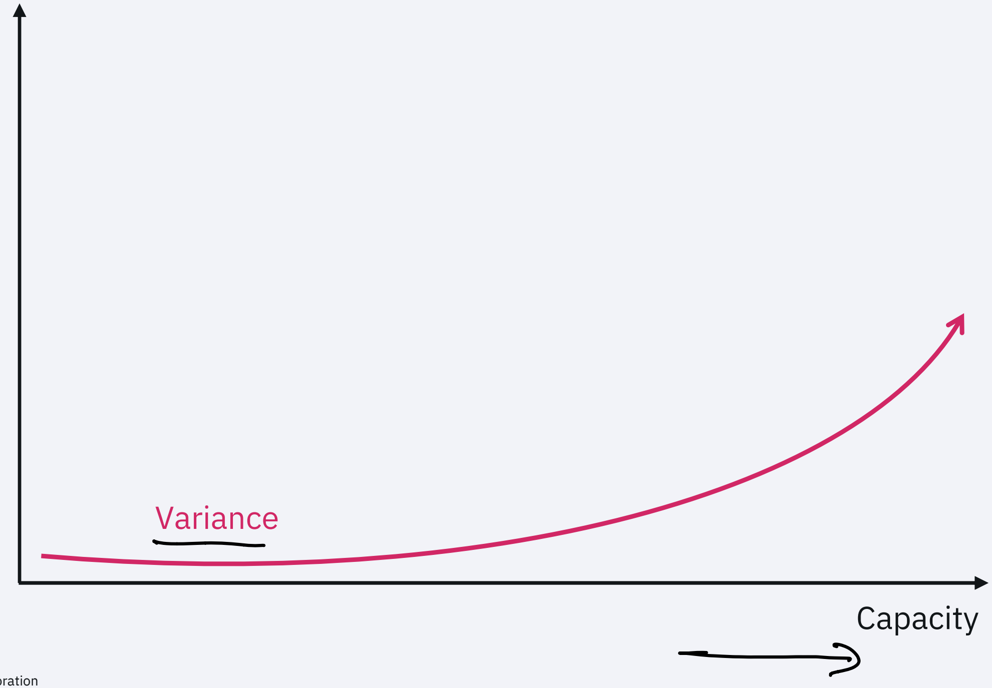


Test

Choosing the right model

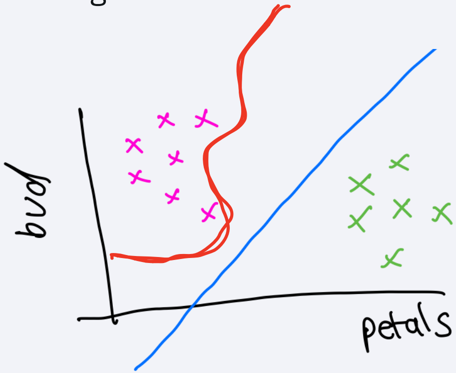


Choosing the right model



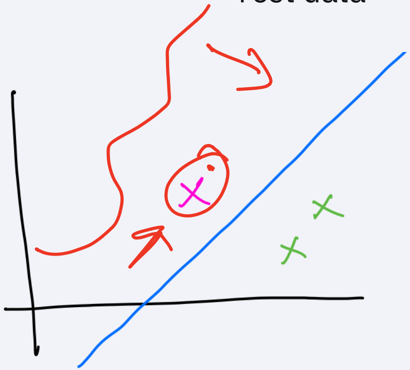
Models with high variance

Training data



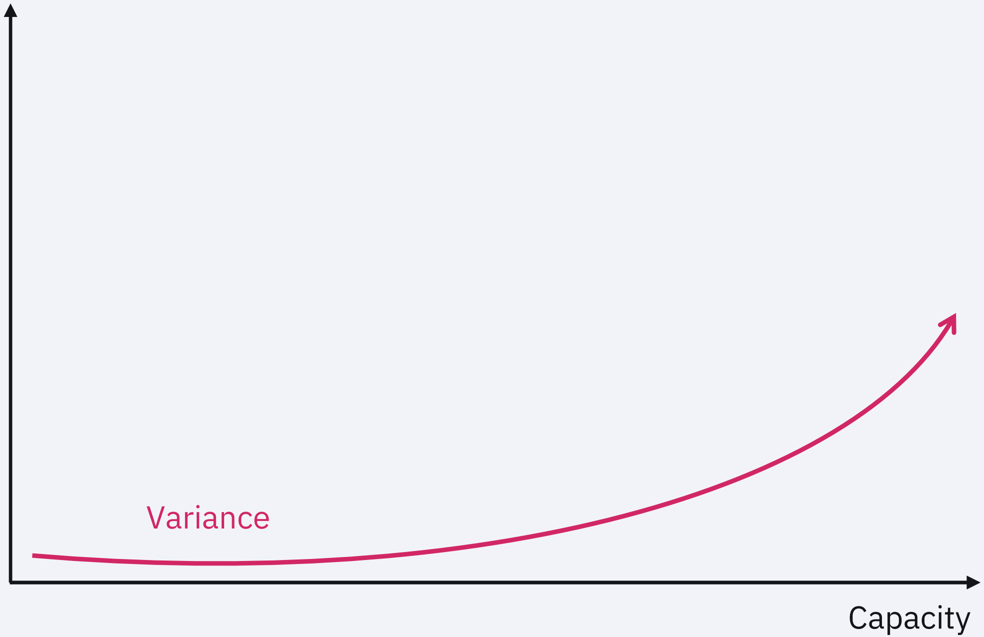
Training

Test data

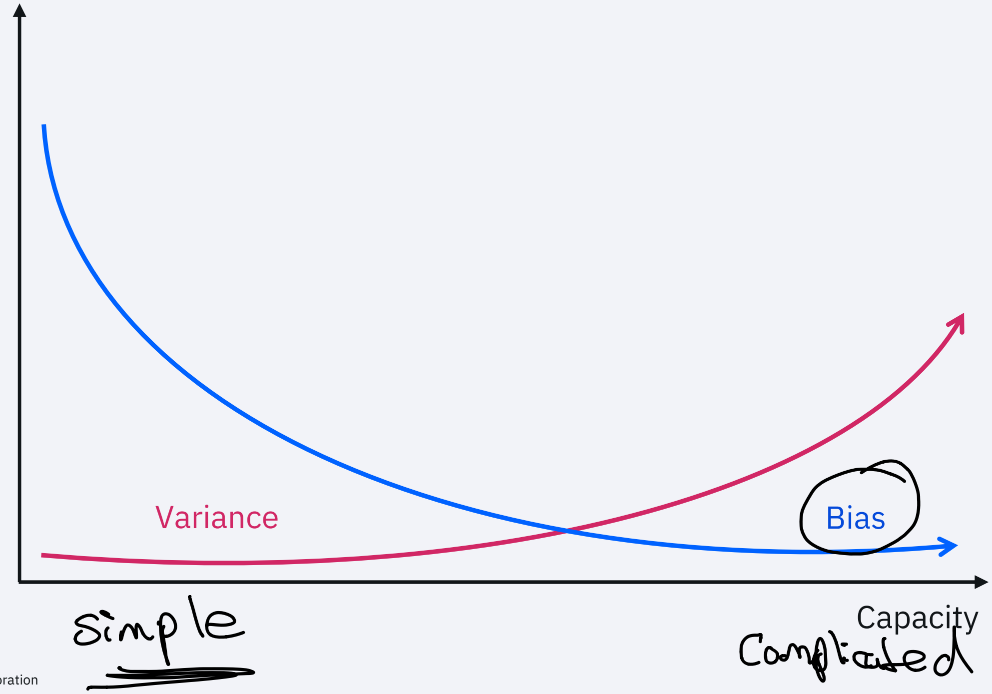


Test

Choosing the right model



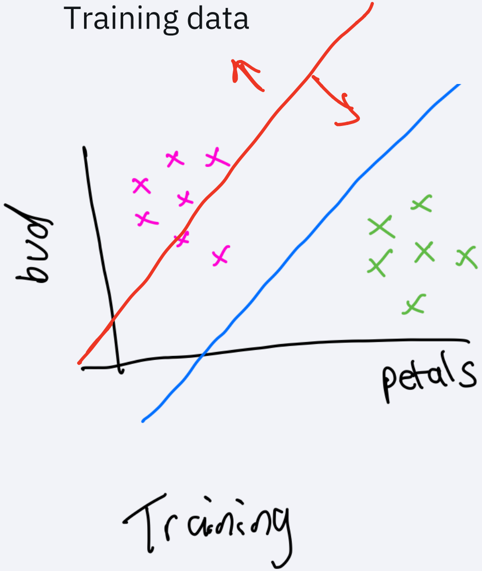
Choosing the right model



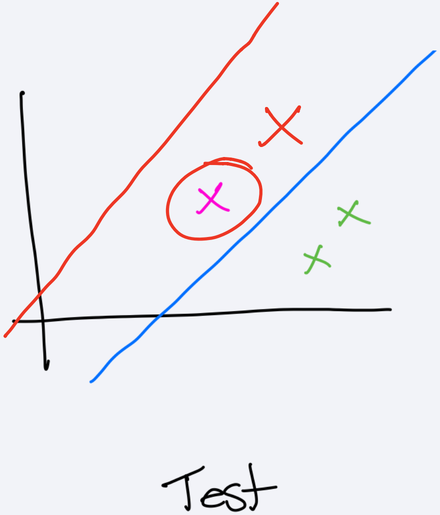
Models with high bias

Qiskit

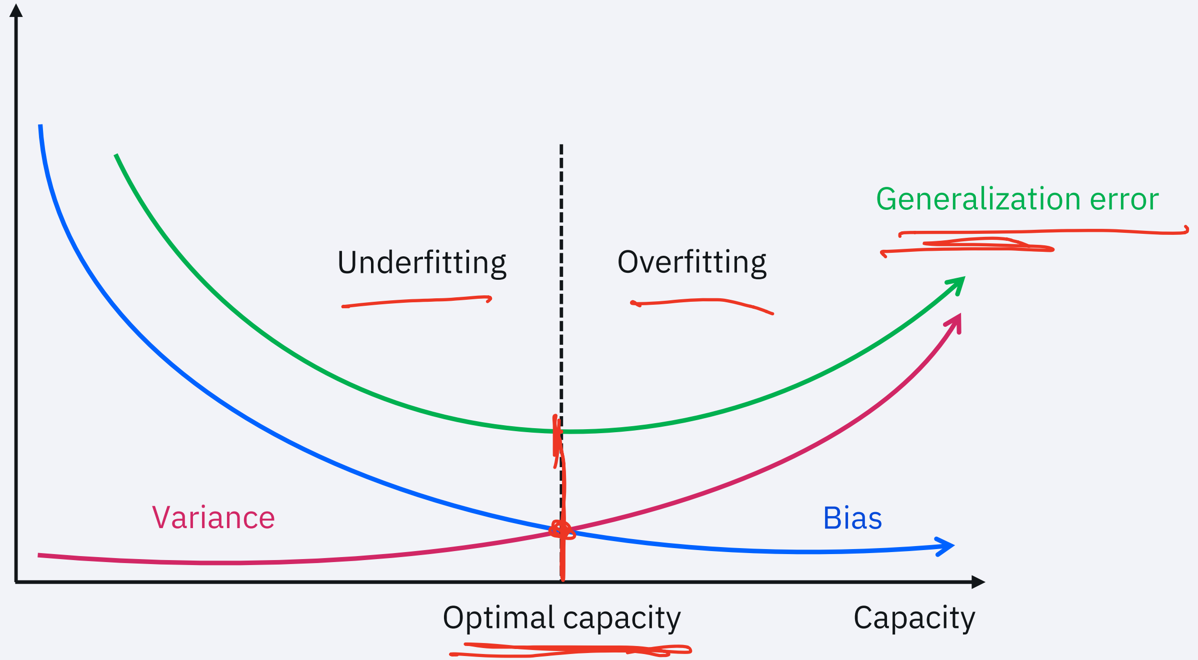
Training data



Test data



Choosing the right model



Recap

Recap

Machine learning ingredients

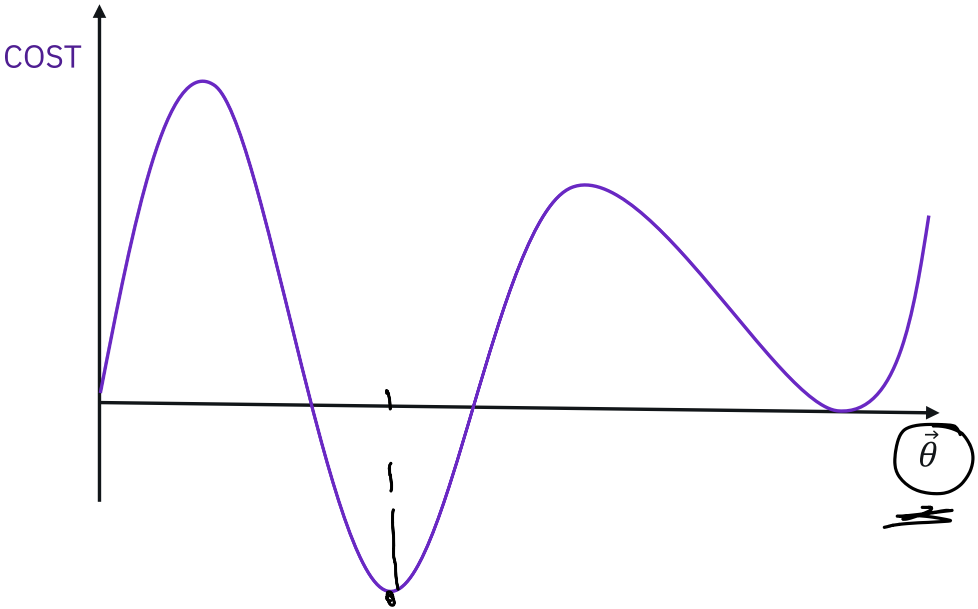
DATA MODEL COST

$$\vec{x} = \begin{bmatrix} \\ \\ \end{bmatrix} \longrightarrow \boxed{f(\vec{x};\theta)} \longrightarrow \hat{y} \rightarrow y_{true}$$

Recap

Training a model

- Minimize the cost function using training data
- $\boxed{\text{Gradient-based methods}}$

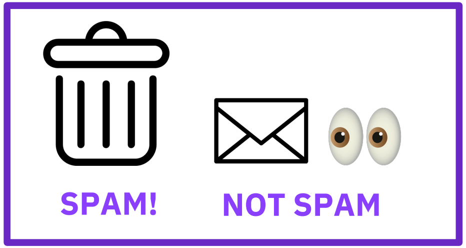


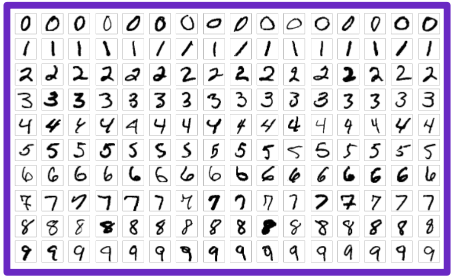
Recap

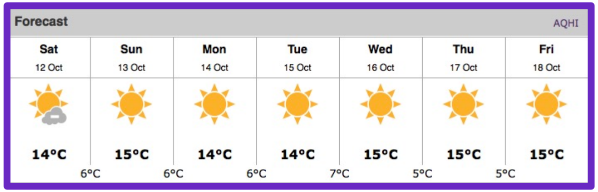
Qiskit

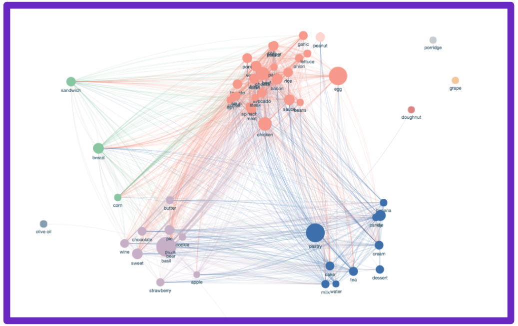
Machine learning examples:

- Classification
- Regression
- Clustering
- Generating data





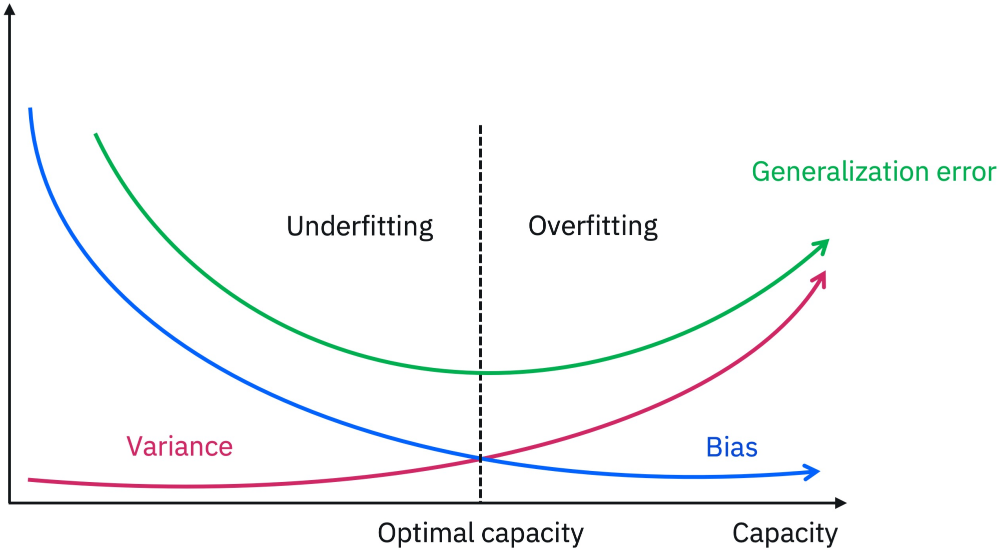




Recap

Assessing a model

- Want to choose a model that has a high enough ability to model the training data accurately
- The model should not overfit the training data
- Over/underfitting leads to poor generalization



DATA

MODEL

COST

Thank you!

Amira Abbas

@AmiraMorphism