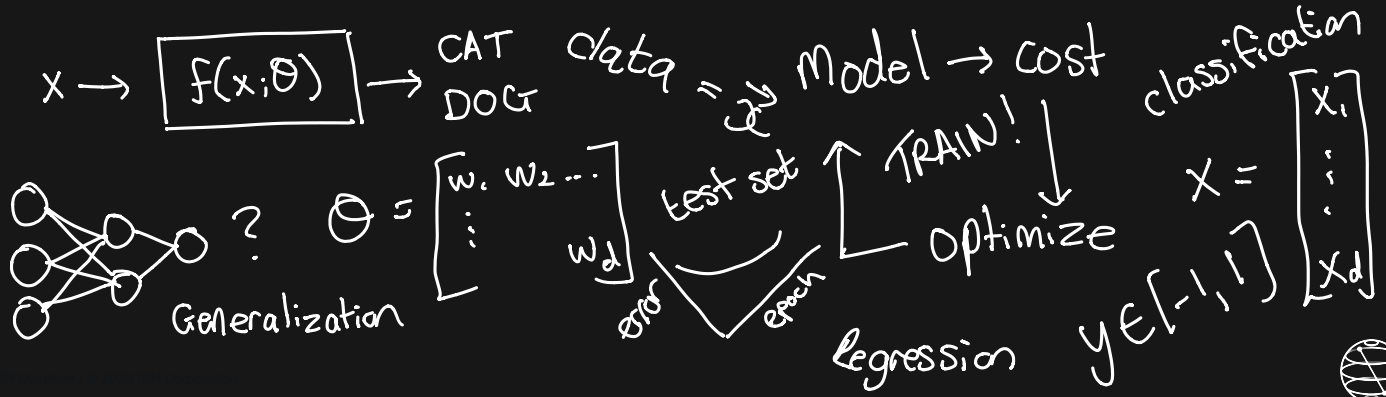
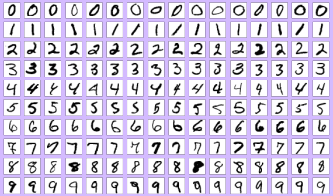
Classical machine learning models

Amira Abbas

IBM Quantum, University of KwaZulu-Natal

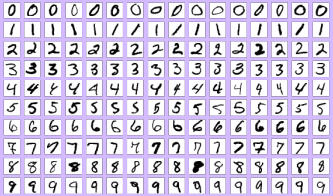


Recap



Recap

DATA

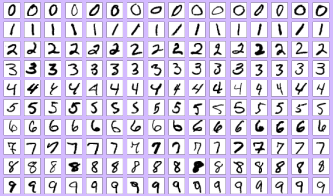


MODEL

$f(X_{train}; \vec{\theta})$

Recap

DATA



MODEL

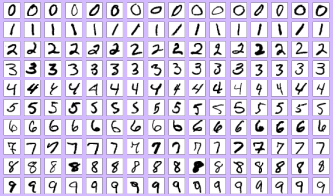
$$f(X_{train}; \vec{\theta})$$

COST

$$C(f(X_{train}; \vec{\theta}), y)$$

Recap

DATA



MODEL

$$f(X_{train}; \vec{\theta})$$

COST

$$C(f(X_{train}; \vec{\theta}), y)$$

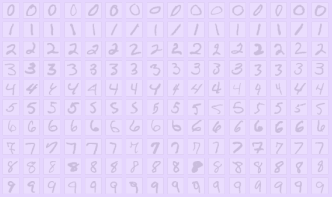
OPTIMIZATION

$$\frac{\partial C(f(X_{train}; \vec{\theta}),\ y)}{\partial \vec{\theta}}$$

Qiskit

Recap

DATA



MODEL

$f(X_{train}; \vec{\theta})$

COST

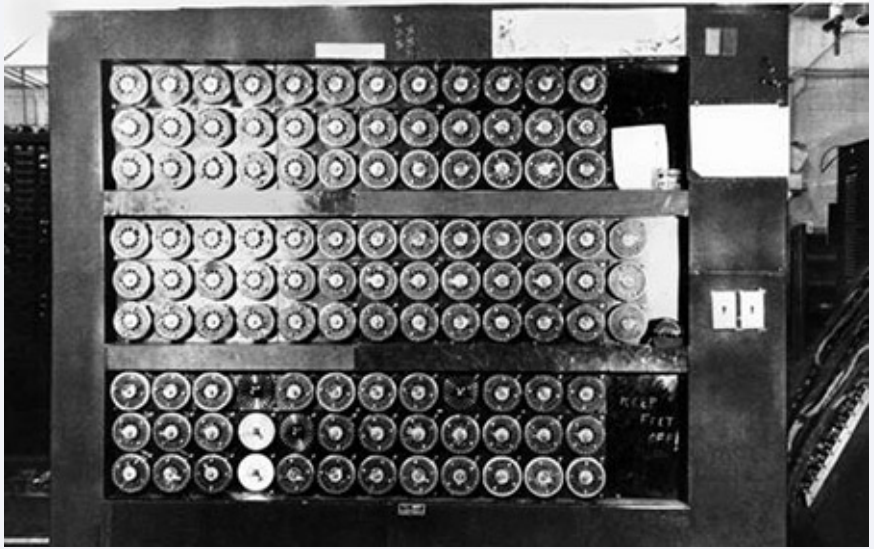
$C(f(X_{train}; \vec{\theta}), y)$

OPTIMIZATION

$$\frac{\partial C(f(X_{train}; \vec{\theta}), y)}{\partial \vec{\theta}}$$

History of machine learning

- 1950: Turing's Learning Machine

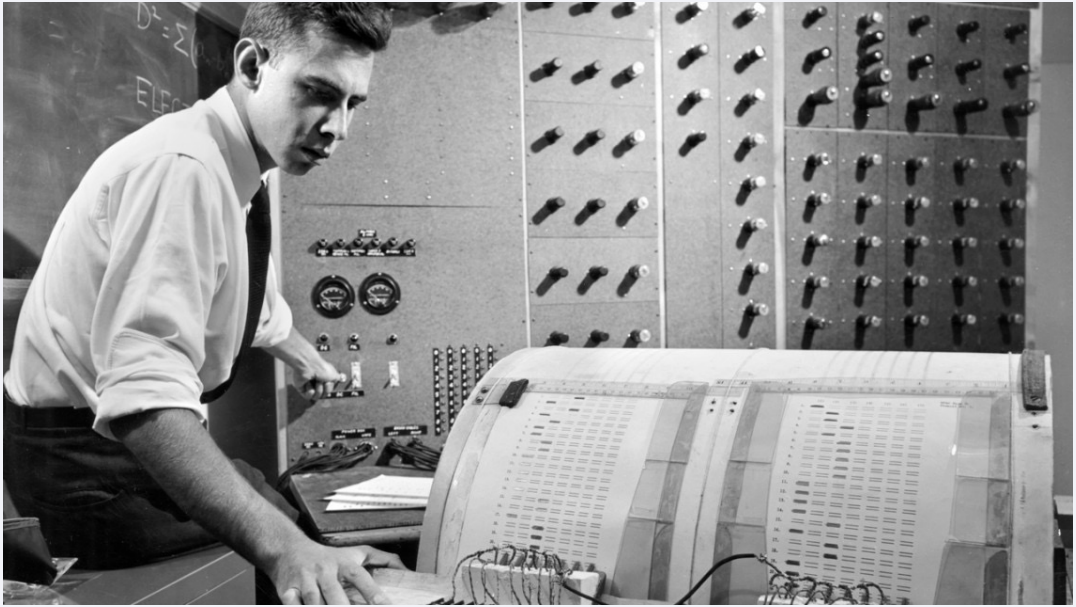


History of machine learning

- *Machine learning* was coined in 1952 by Arthur Samuel, an American IBMer

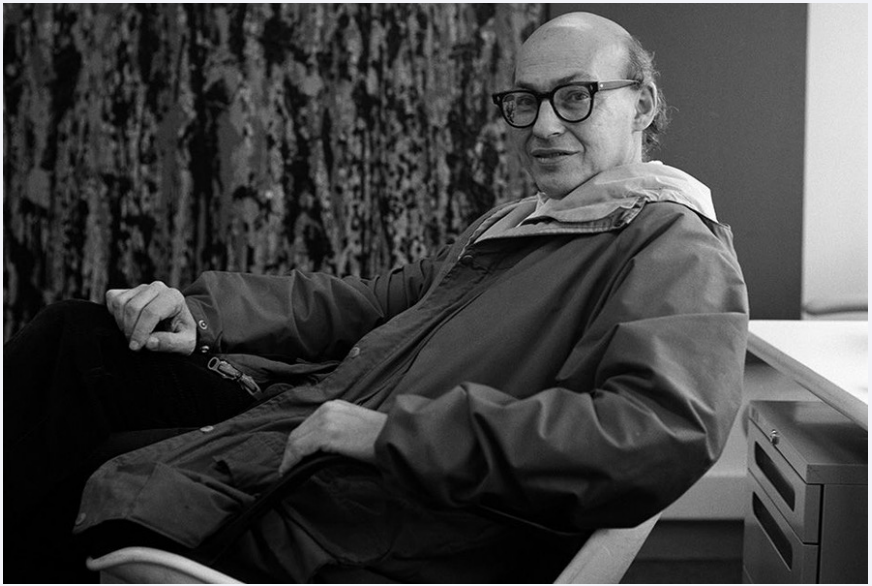
History of machine learning

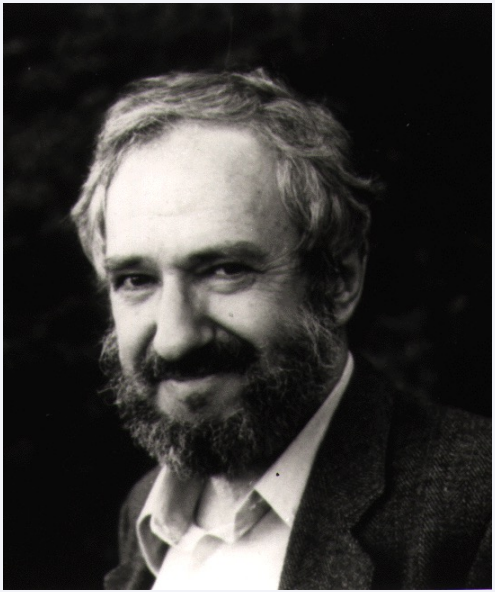
- 1957: Enter the perceptron model created by Frank Rosenblatt, Cornell



History of machine learning

- 1969: Limitations of neural networks, Marvin Minsky & Seymour Papert



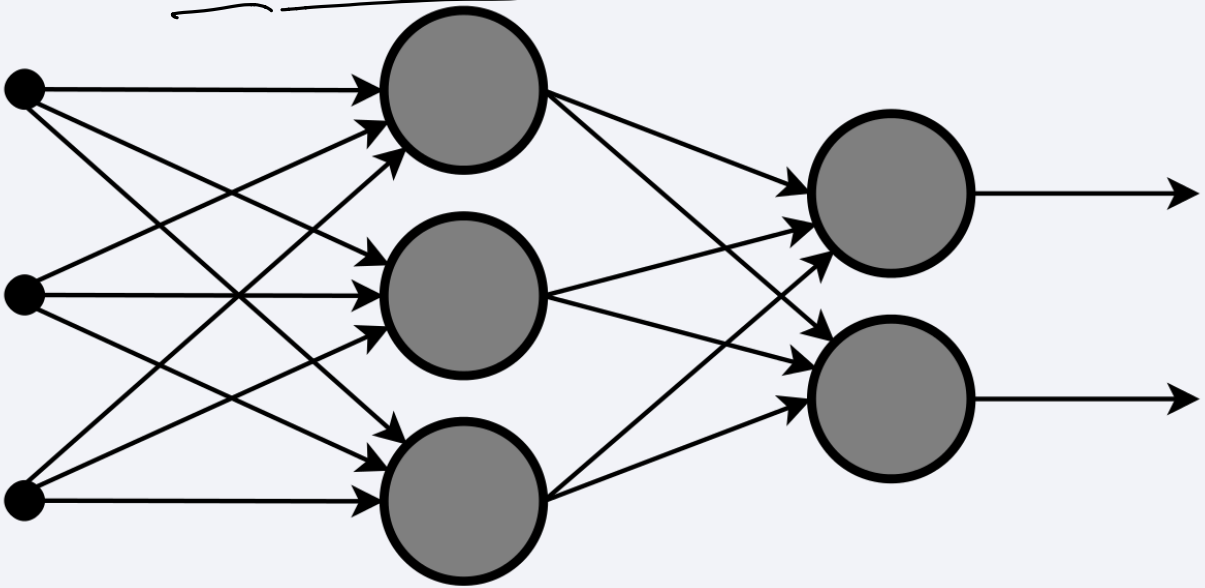


History of machine learning

- 1970’s: AI winter

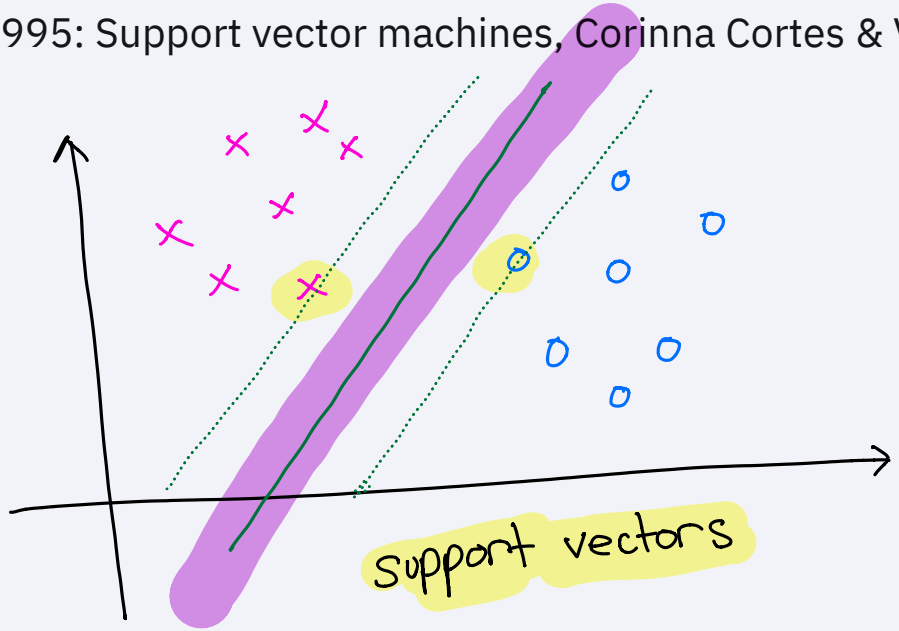
History of machine learning

- 1970 & 1986: Backpropagation, Seppo Linnainmaa & Geoffrey Hinton



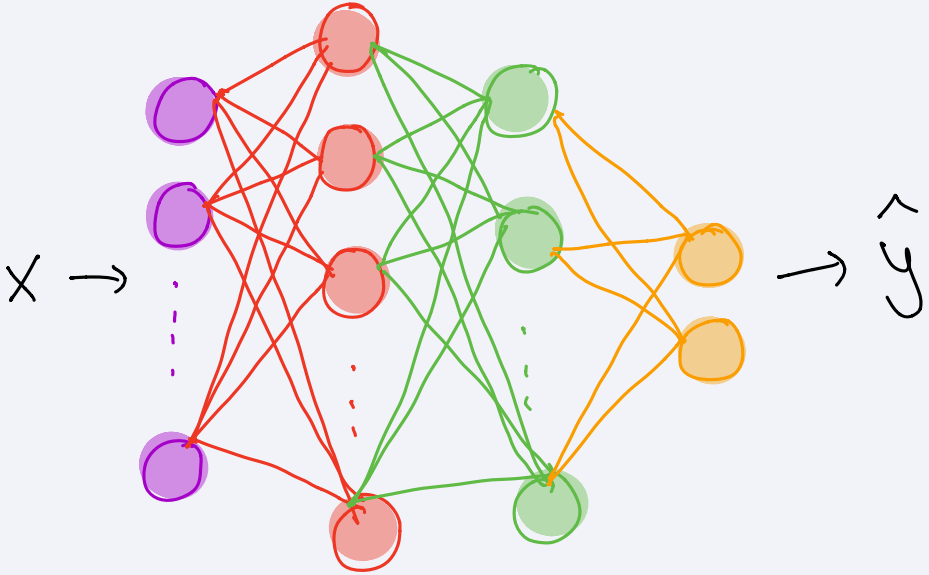
History of machine learning

- 1995: Support vector machines, Corinna Cortes & Vladimir Vapnik



History of machine learning

- 2010s: Deep learning became feasible in practice



Qiskit

Linear models

Linear models

Linear models

1-dimensional data, regression problem

$\vec{x}_1 = [-]$ dataset $= (\vec{x}_1, \ldots, \vec{x}_n)$

d- dimensiond data

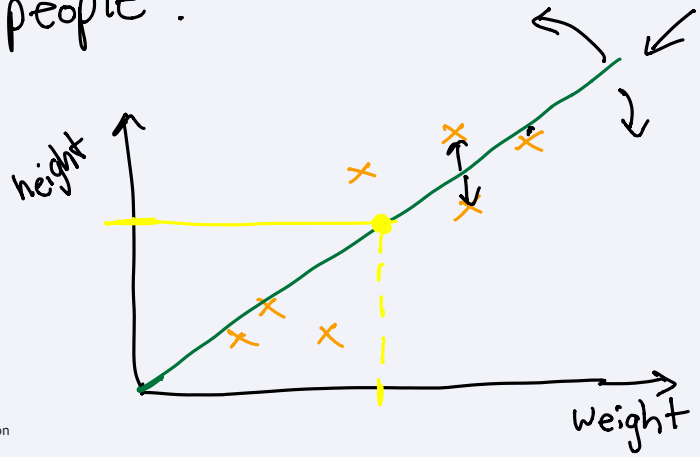
$\vec{x}_1 = \begin{bmatrix} \\ \\ \vdots \\ \\ d \end{bmatrix}$ $d \rightarrow$ features

Linear models

1-dimensional data, regression problem

Task: predict the height of a person, based on their weight

$n = 10$ samples, we know heights (y_i) and weights (x_i) of 10 people.



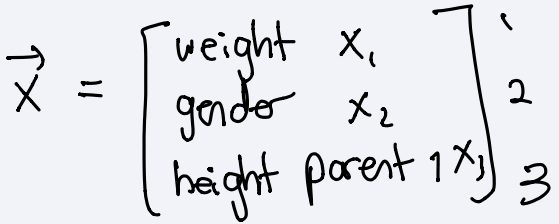
$$\hat{y}_i = \theta x_i$$

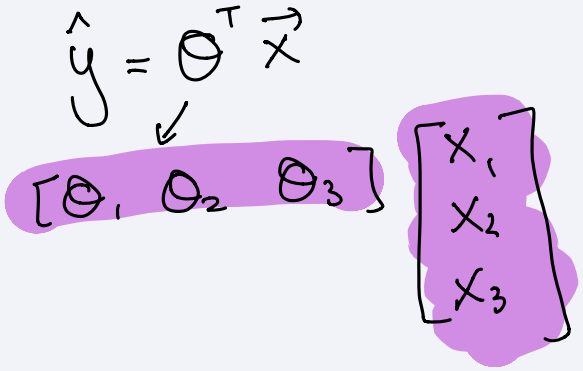
$$\text{MSE}: \frac{1}{10} \sum_{i=1}^{10} (\hat{y}_i - y_i)$$

Linear models

Higher dimensional data, regression problem

Task: predict the height of a person, based on their weight, gender, height of their parents, etc.



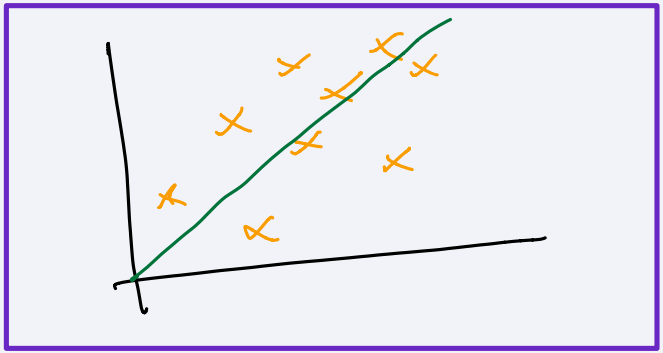


Linear models

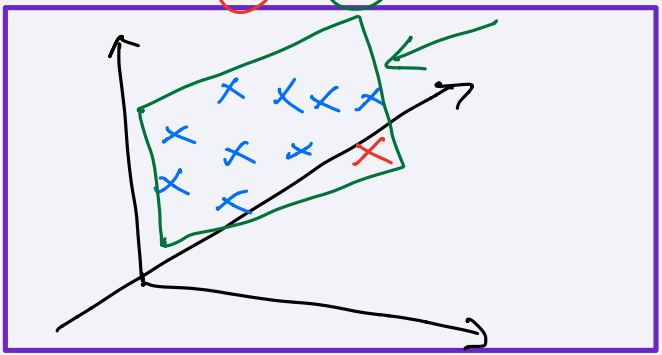
Higher dimensional data, regression problem

Task: predict the height of a person, based on their weight, gender, height of their parents, etc.

$$\hat{y} = \theta x$$



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



Qiskit

Neural networks

Neural networks

2-dimensional data, classification

$$\vec{x}_i = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad \hat{y}_i \rightarrow 1, -1$$

Neural networks

Qiskit

Formula representation

Linear model $\hat{y} = \theta^T \vec{x} + b$

Perceptron $\hat{y} = \sigma(\theta^T \vec{x})$ activation fn

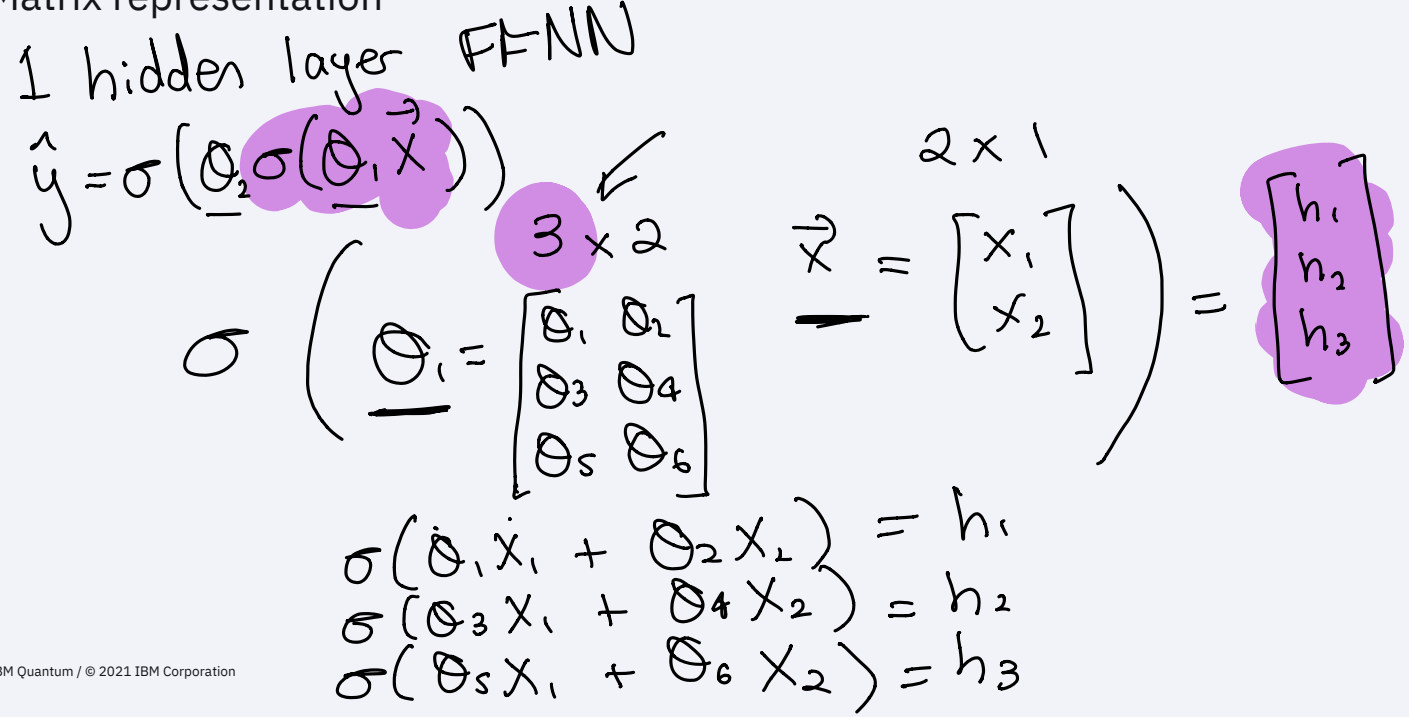
Feed forward NNs $\hat{y} = \sigma(\theta_2 \sigma(\theta_1^T \vec{x}))$ perceptron

$\hat{y} = \sigma(\theta_3 \sigma(\theta_2 \sigma(\theta_1 \vec{x})))$ ←

Neural networks

Qiskit

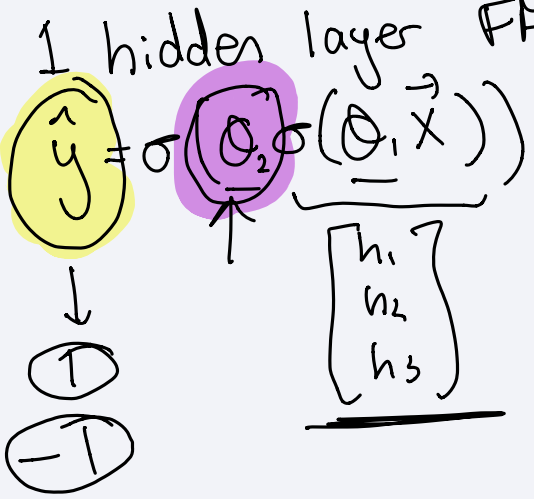
Matrix representation



Neural networks

Qiskit

Matrix representation



$$\hat{y} = \sigma(\theta_2 \sigma(\theta_1 \vec{x}))$$

$1 \times 3 \quad 3 \times 1$

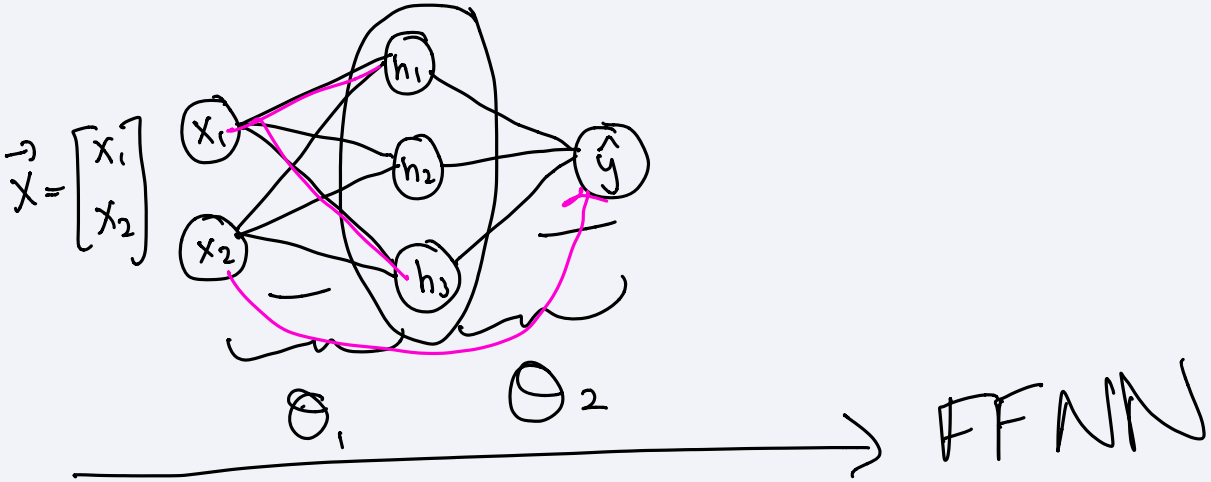
$$\theta_2 = [\theta_1' \ \theta_2' \ \theta_3'] \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix} \leftarrow \text{neurons}$$

$$\hat{y} = \sigma(\theta_1' h_1 + \theta_2' h_2 + \theta_3' h_3)$$

Neural networks

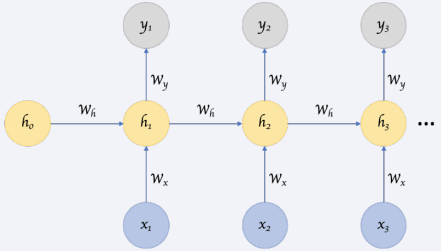
Qiskit

Graphical representation

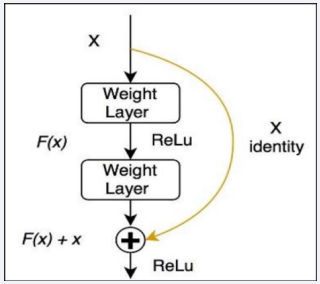


Neural networks

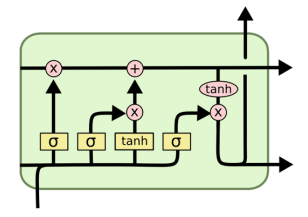
The success of neural networks



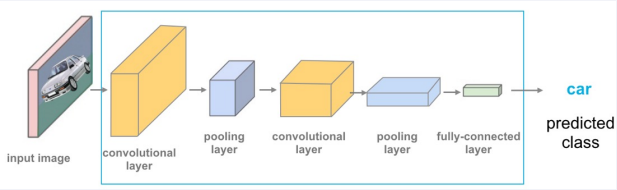
Mikolov et al. "Extensions of recurrent neural network language model." 2011 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2011.



Hao, et al. "Visualizing the loss landscape of neural nets." Advances in Neural Information Processing Systems. 2018.



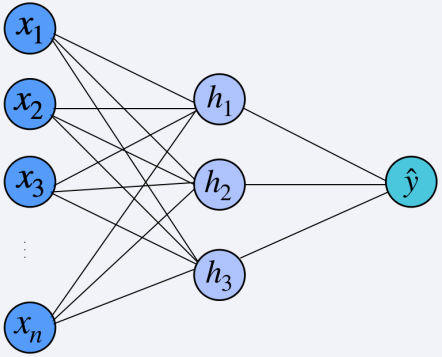
Sak et. al. "Long short-term memory recurrent neural network architectures for large scale acoustic modeling." (2014).



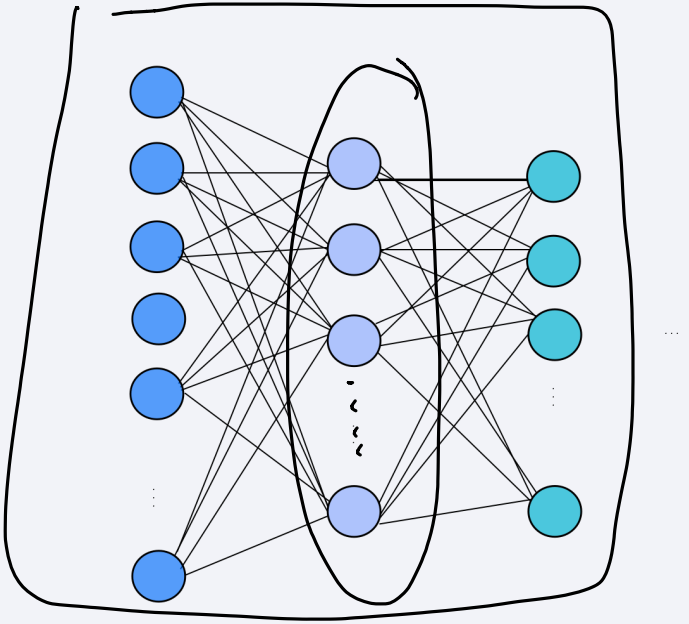
Krizhevsky et al."Imagenet classification with deep convolutional neural networks." Communications of the ACM 60.6 (2017): 84-90.

Neural networks

The success of neural networks



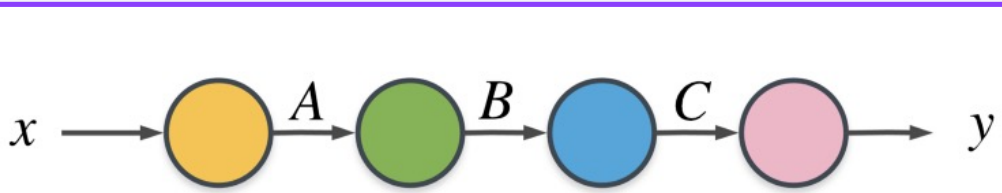
Hornik et al. "Multilayer feedforward networks are universal approximators." Neural networks 2.5 (1989): 359-366.



Michael Nielsen - https://www.youtube.com/watch?v=Ijqkc7OLenI

Neural networks

- 1970 & 1986: Backpropagation, Seppo Linnainmaa & Geoffrey Hinton



$$\frac{\partial y}{\partial x} = \frac{\partial y}{\partial C} \times \frac{\partial C}{\partial B} \times \frac{\partial B}{\partial A} \times \frac{\partial A}{\partial x}$$

Python

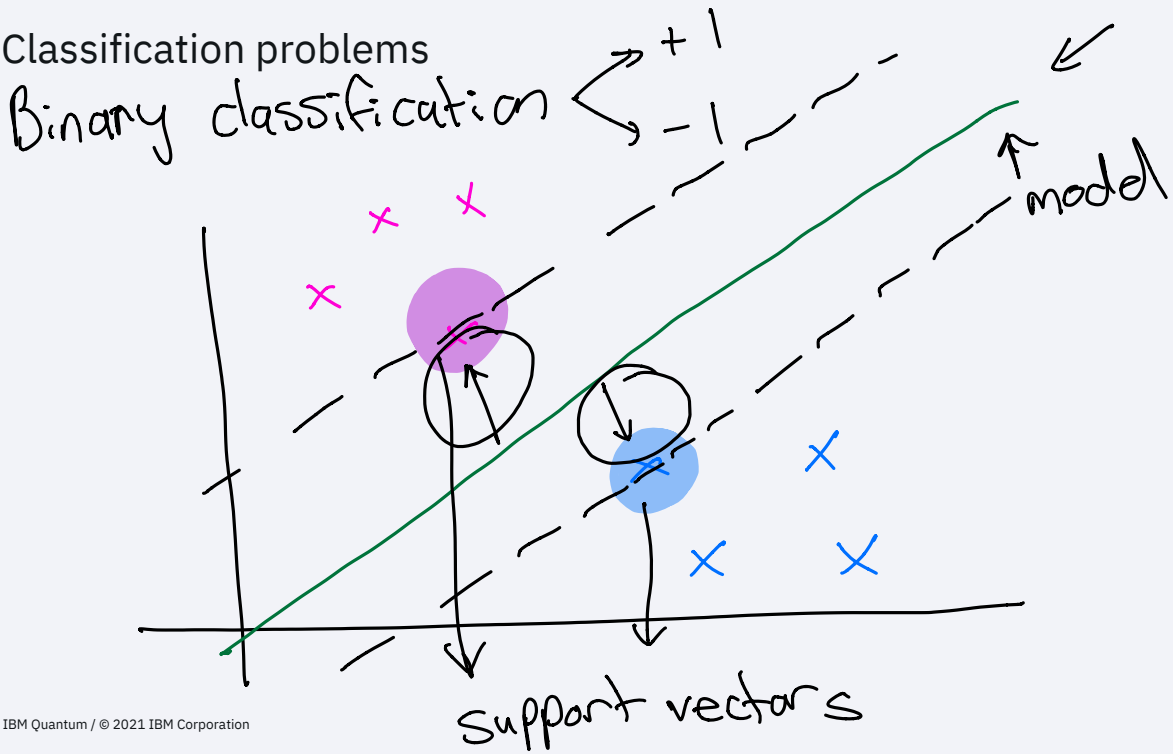
Qiskit

Support vector machines

Support vector machines

Qiskit

Classification problems



Support vector machines

Qiskit

Classification problems

Linear classifier

$y_i \in [-1, 1]$

$f(x) = \theta^T \vec{x} + b$

$\theta \in \mathbb{R}^d$

d-dimensional

→ quadratic opt.

- Primal formulation

Support vector machines

Qiskit

Classification problems

$$f(x) = \theta^T x + b$$

- Dual formulation

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i (x_i^T x) + b$$

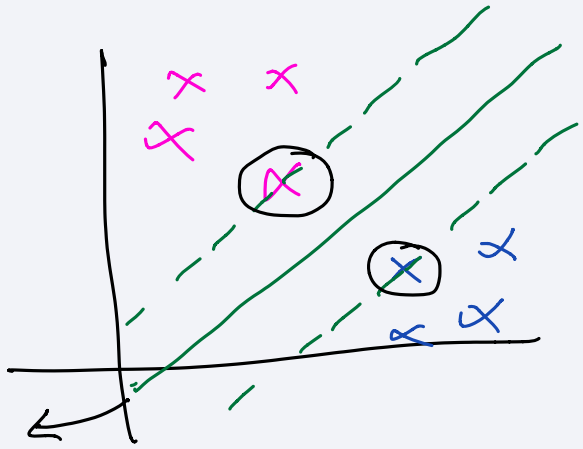
inner product

x_i x

Why?

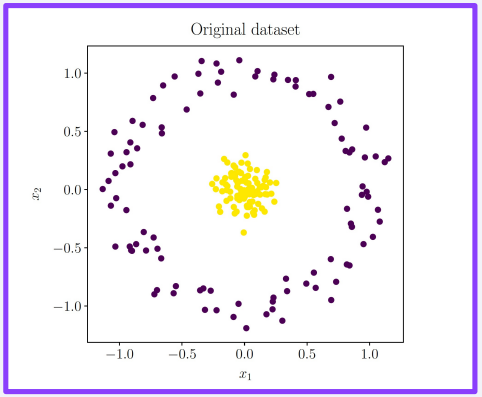
most $\alpha_i = 0$

$\#\alpha_i \neq 0 \rightarrow$ #support vectors



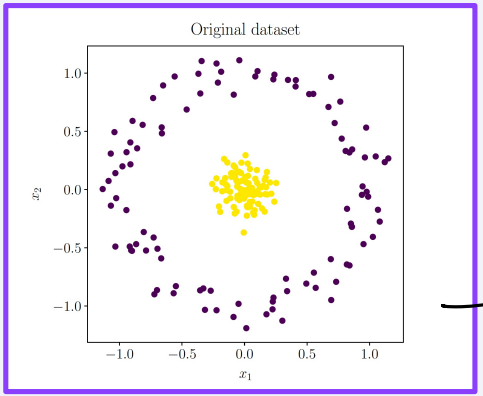
Support vector machines

Classification problems – data not linearly separable?

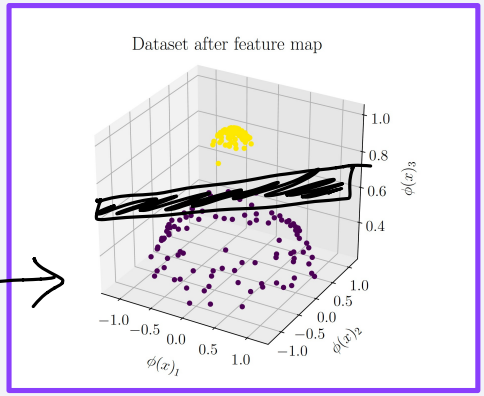


Support vector machines

Classification problems – data not linearly separable? Apply a **feature map**



ϕ



2-d

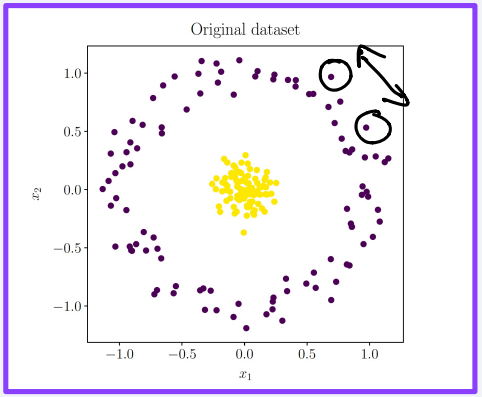
3d

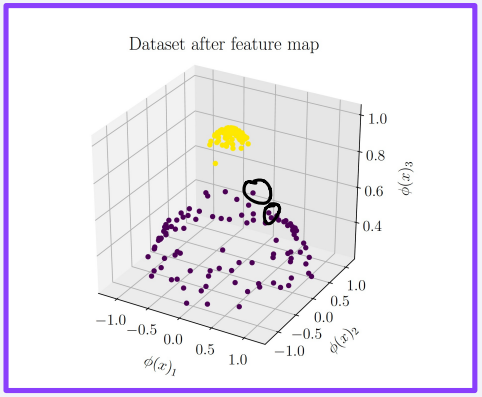
$\vec{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

$\phi(\vec{x}) = \begin{bmatrix} x_1 \\ x_2 \\ x_1 x_2 \end{bmatrix}$

Support vector machines

Classification problems – data not linearly separable? Apply a **feature map**





$x_i^T x_j$ similarity

$\phi(\vec{x})$

$\phi(x_i)^T \phi(x_j)$ similarity

Support vector machines

Qiskit

Classification problems

$$f(x) = \sum_i \alpha_i y_i (x_i^T x) + b$$

$$f(x) = \theta^T x + b$$

$$f(x) = \sum_i \alpha_i y_i (\phi(x_i)^T \phi(x)) + b$$

inner product

$$f(x) = \theta^T \phi(x) + b$$

kernel

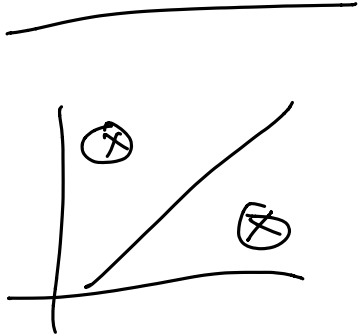
Recap

Linear models

$\hat{y} = \theta^T x + b$

Neural networks

Support vector machines



Quantum machine learning

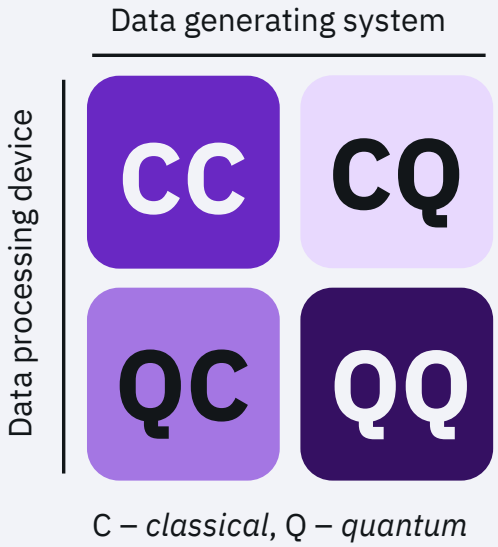


Image credit: *Maria Schuld and Francesco Petruccione. Supervised learning with quantum computers. Vol. 17. Springer, 2018.*

Quantum machine learning

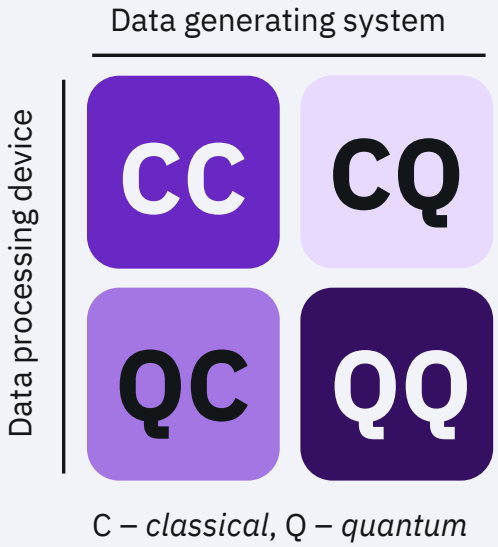
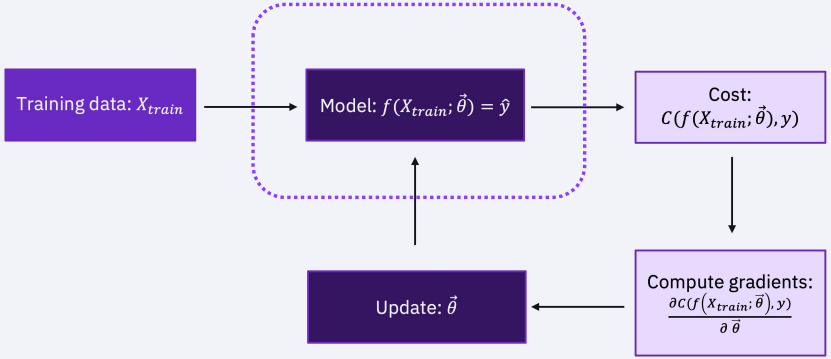


Image credit: *Maria Schuld and Francesco Petruccione. Supervised learning with quantum computers. Vol. 17. Springer, 2018.*



For the next lectures

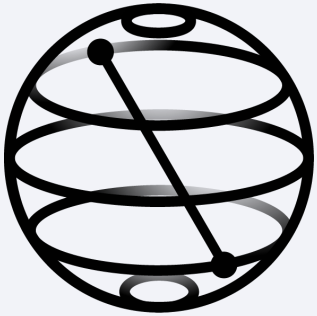
Qiskit

For the next lectures, brush up on classical ML and intro to QC concepts.

In particular:

- Linear algebra basics (vectors, matrices, tensors)
- Dirac notation (bra-ket notation, measurement operators)
- Circuit notation (gates)
- Cost function minimization (backpropagation, gradient descent)
- Support vector machines (data, feature maps, kernels)

Thank you!



Amira Abbas

@AmiraMorphism