CHALMERS
UNIVERSITY OF TECHNOLOGY

MACHINE LEARNING AND NEURAL NETWORKS

PART ONE

Bastiaan Bruinsma

February 28, 2022

Chalmers University of Technology

The Programme

- Today
  - How do we define Machine Learning?
  - Neural Networks and their role
  - Training Neural Networks
  - Explainability
  - Convolutional Neural Networks

- Thursday
  - Shallow and Deep Networks
  - Deep Q-Learning
  - Autoencoders
  - Natural Language Processing
  - Generative Adversary Networks

Machine Learning

A Definition

Arthur Samuel (1959)

"Field of study that gives computers the ability to learn without being explicitly programmed"

Tom M. Mitchell (1997)

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E"

An Example

Set-Up: A spam filter “learns” which e-mails you mark as “spam” and uses this to improve itself

- Task (T)
  - Classify e-mails as SPAM or NOT-SPAM
- Experience (E)
  - Labelling e-mails as SPAM or NOT-SPAM
- Performance (P)
  - The fraction of e-mails correctly classified as SPAM or NOT-SPAM

Rule-Based AI

Rule based AI has:

- A set of RULES given by humans (IF-THEN statements)
- A set of FACTS

Examples:

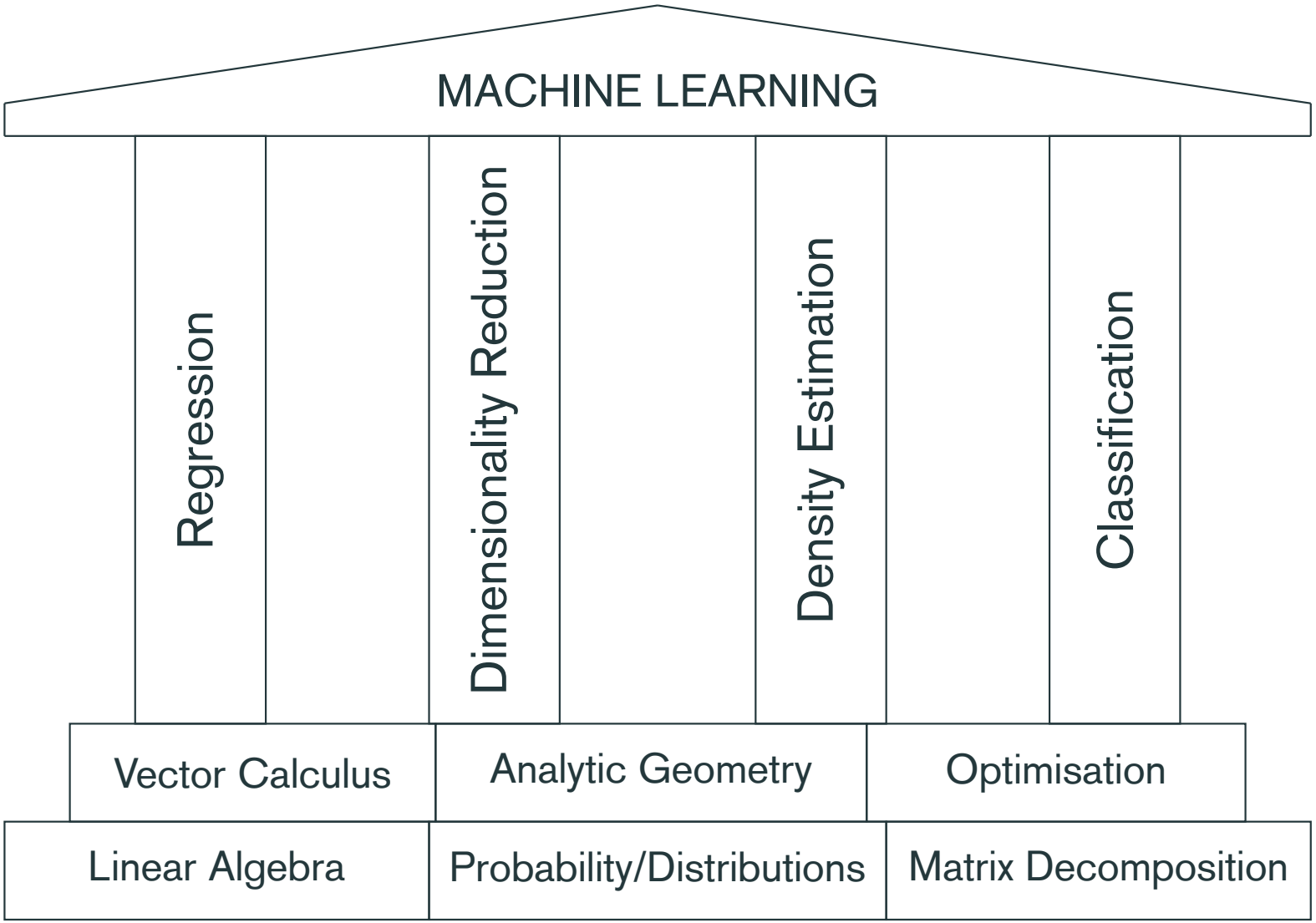
- Recognizing pictures
- Diagnoses
- Play games

Rule-Based AI

Differences:

Machine Learning	Rule-Based AI
Probabilistic	Deterministic
Evolve and Adept	Static
Scaleable	Not Scalable (Yet)
Much Data required	(Almost) No Data required
Mutable	Immutable

Foundations of Machine Learning (Deisenroth,Faisal, and Ong (2020))



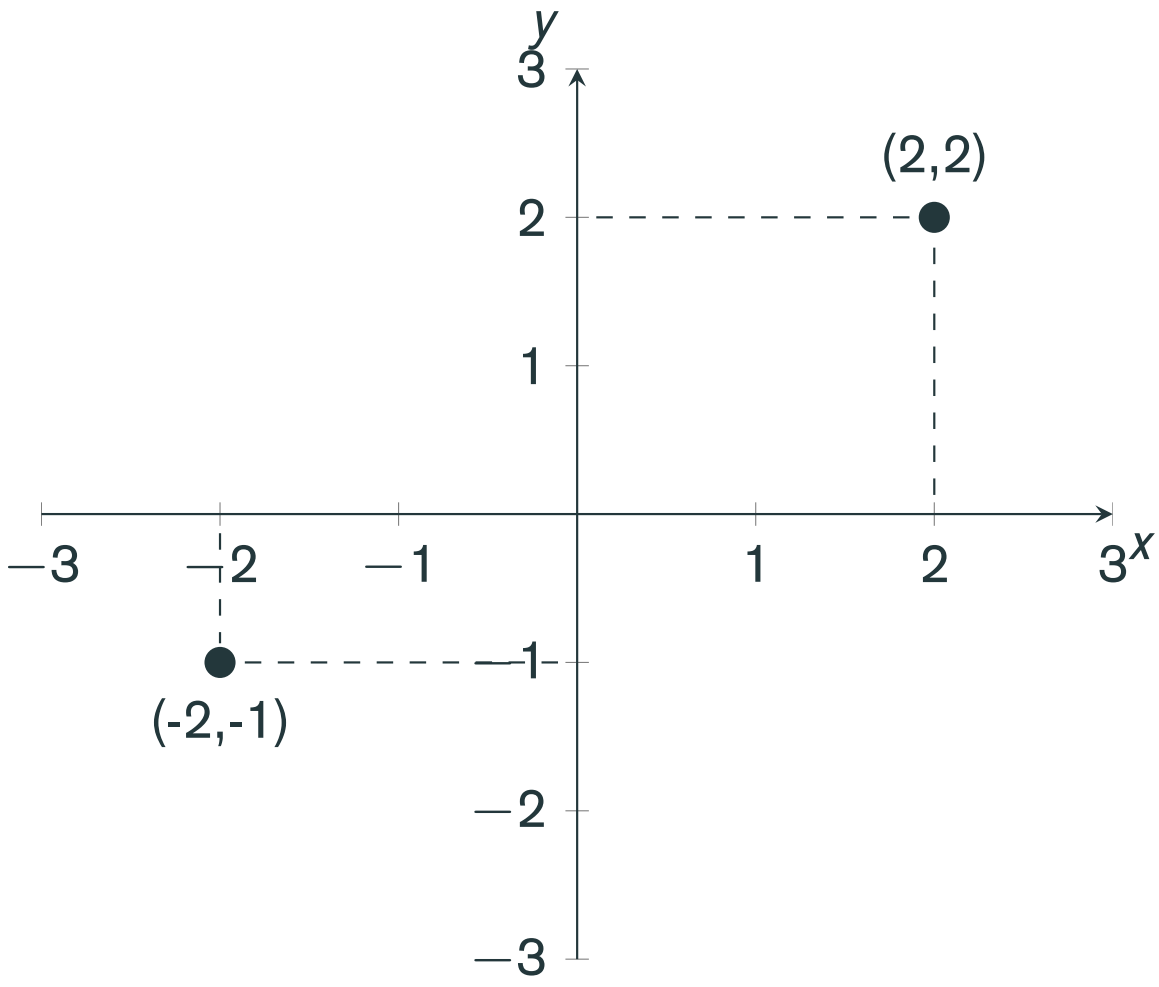
Linear Algebra

- Vectors
- Matrices
- Linear Equations

$$a_{m1}x_1 + \cdots + a_{mn}x_n = b_m$$

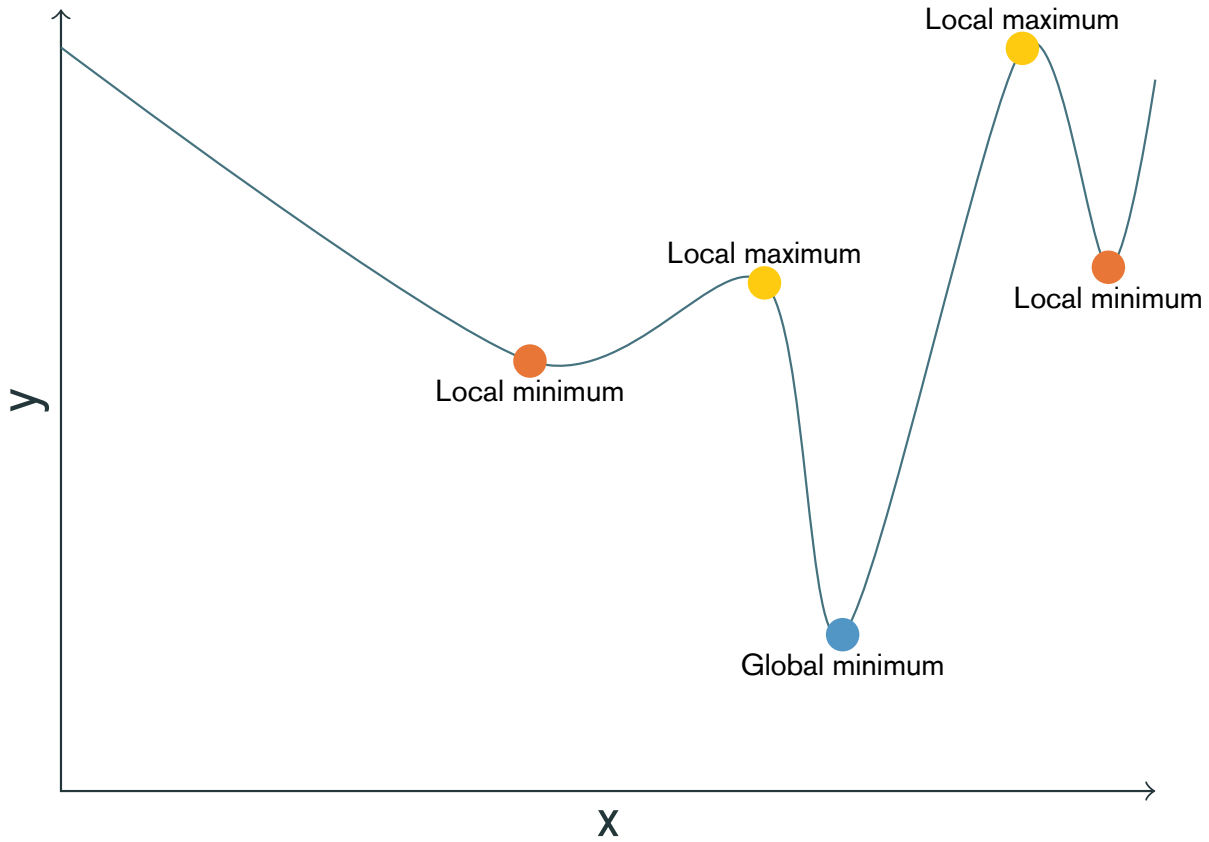
$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{bmatrix} \times \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix}$$

Analytic Geometry



Optimisation

- Minimize (or maximize) a function
- Constained of Unconstained



Vector Calculus

- Change, Limits and Functions
- Rates of change
- Derivatives
- Partial derivatives (keeping other variables constant)

$$\textit{derivative} = \frac{\text{change in y}}{\text{change in x}} = \frac{\Delta y}{\Delta x}$$

Probability/Distributions

- The study of uncertainty
- What model acurately depicts the data-generating process

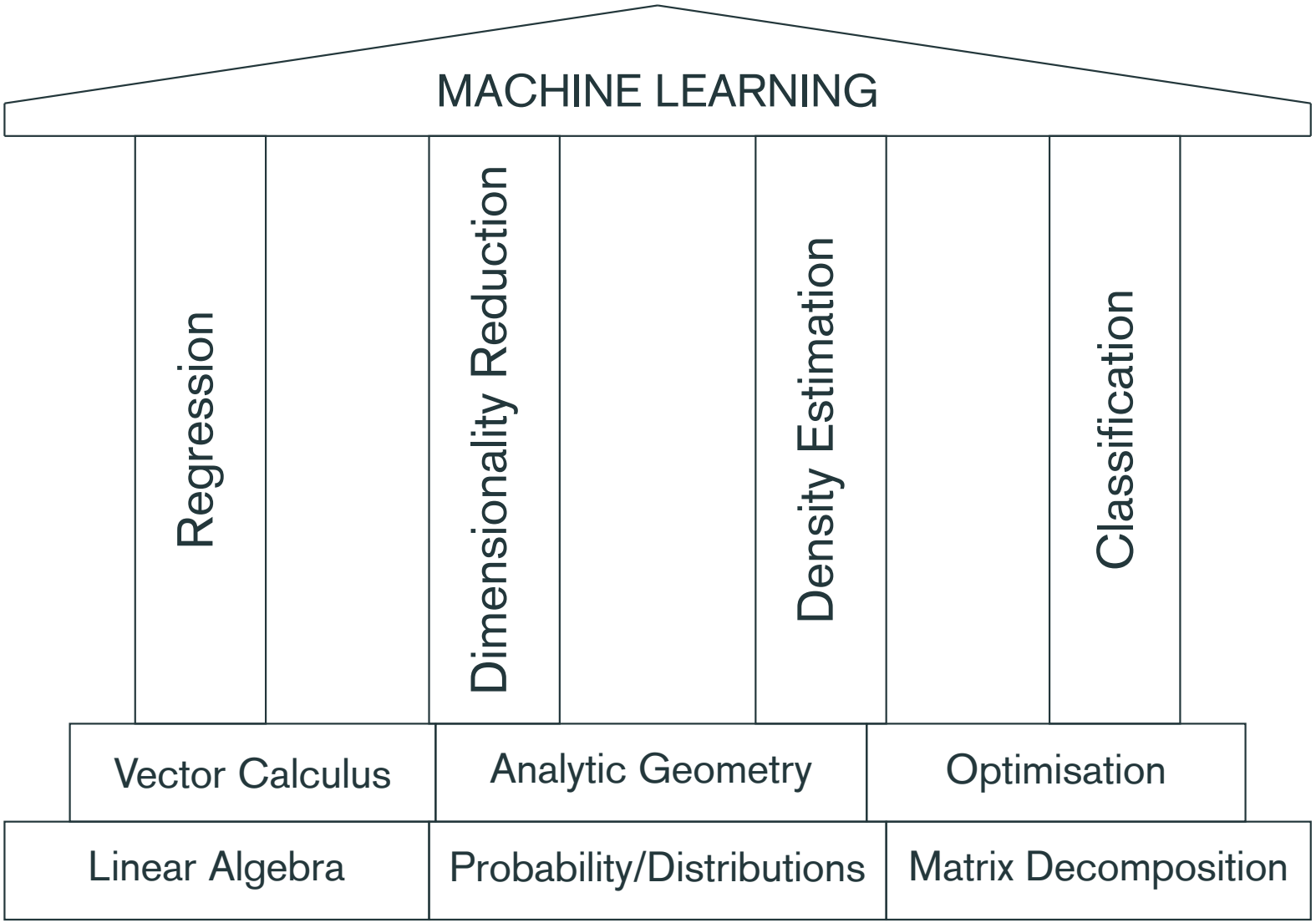
$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Matrix Decomposition

- Breaking up matrices into their constituent parts
  - Lower–upper (LU) decomposition
  - Eigenvalue decomposition
  - Singular value decomposition
  - ...

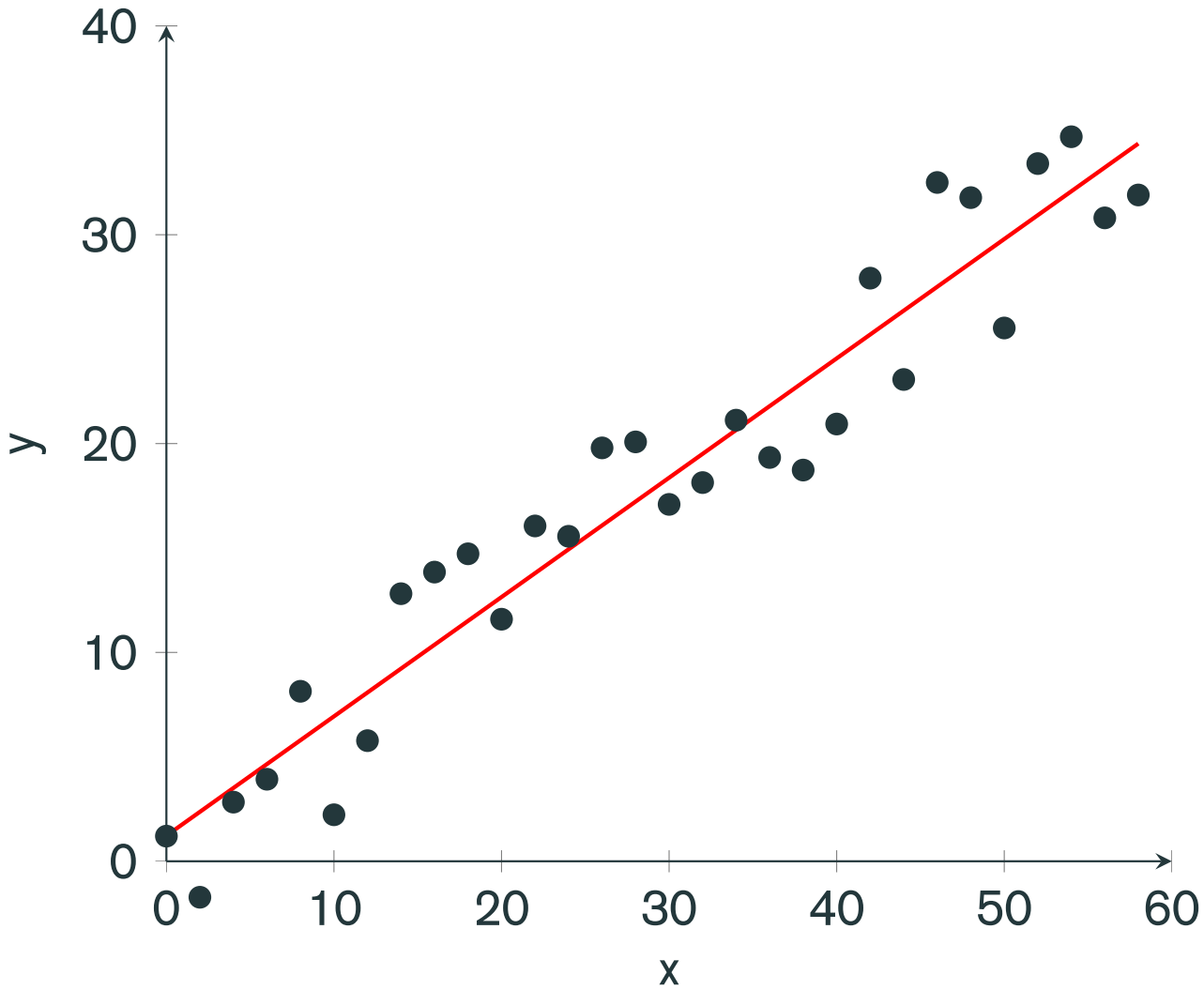
$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} \ell_{11} & 0 & 0 \\ \ell_{21} & \ell_{22} & 0 \\ \ell_{31} & \ell_{32} & \ell_{33} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

Foundations of Machine Learning (Deisenroth,Faisal, and Ong (2020))



Regression (Deisenroth, Faisal, and Ong (2020)

Regression problem: observed noisy function values from which we wish to infer the underlying function that generated the data)



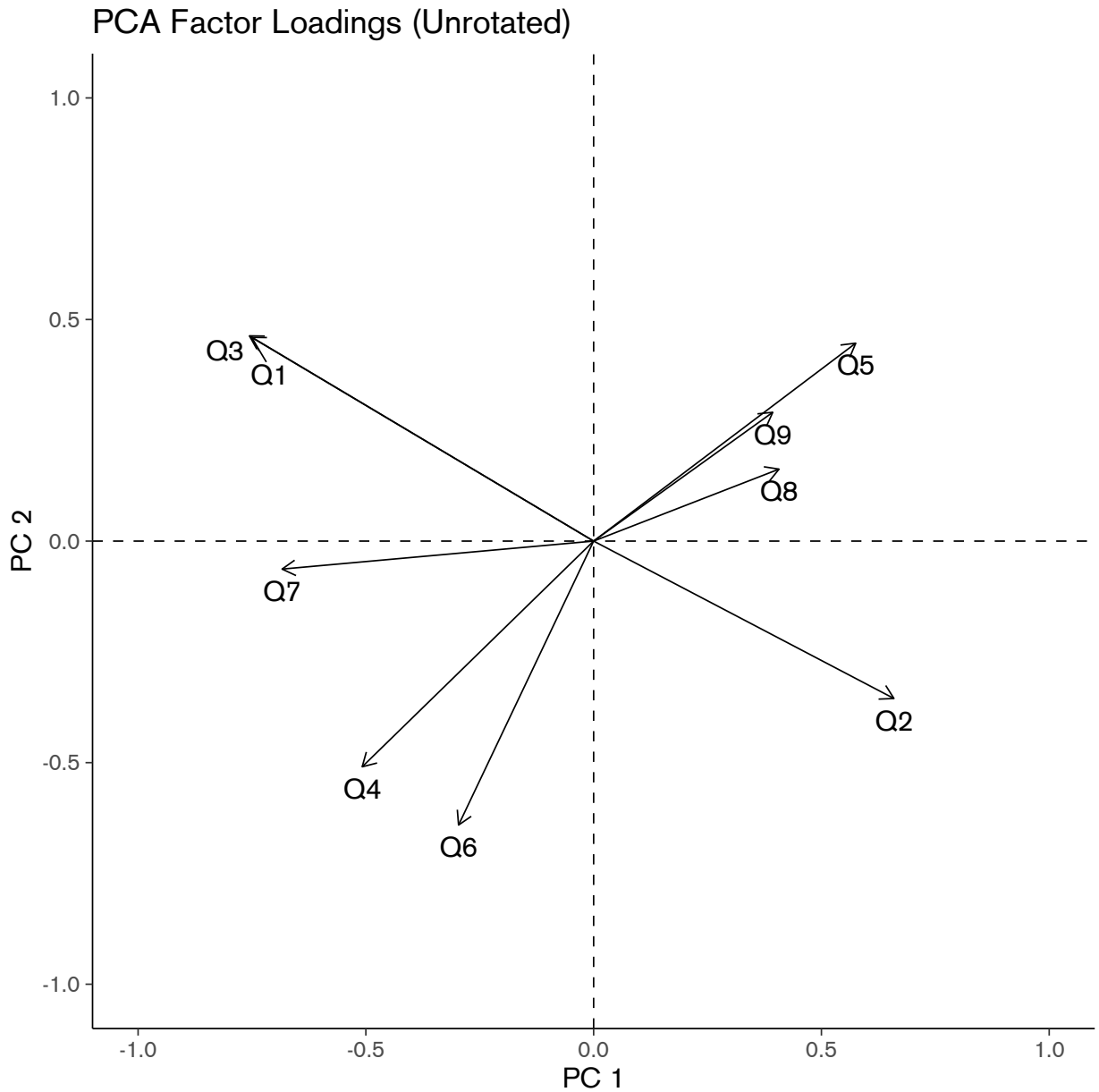
Regression

- Choice of the model (type) and the parametrization
  - What is a good model for the data?
  - What are good parameters?
- Finding good parameters
  - Loss/objective functions
  - What is “good”?
- Overfitting and model selection
  - When the model does not generalise (it fits “too good”)
- Relationship between loss functions and parameter priors
- Uncertainty modeling
  - What is the probability we found this model by chance?

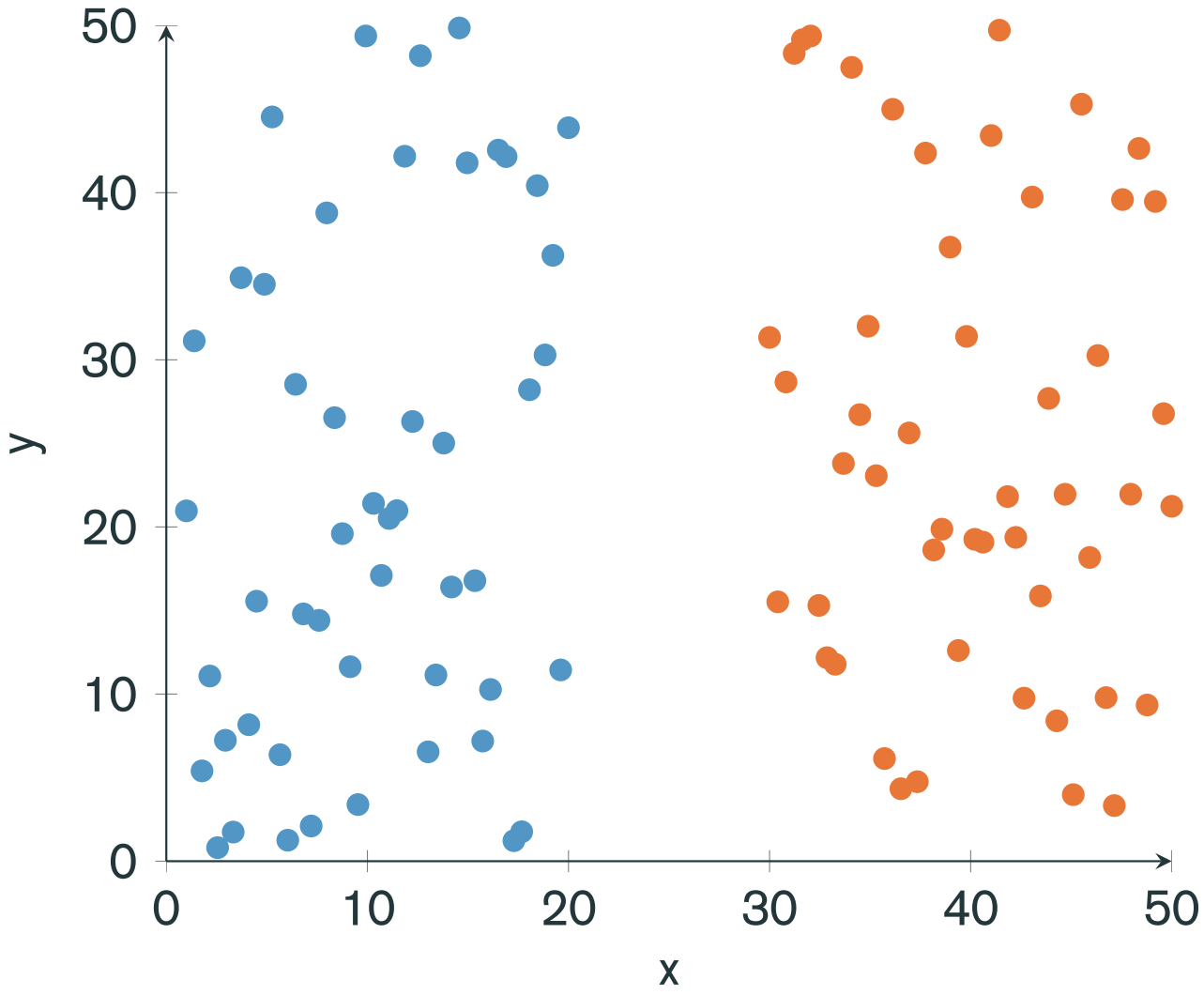
Dimensionality Reduction

- Both are *dimension reduction* methods
- They transform a set of observed variables (that we see and measure) into a smaller set of latent variables (that we cannot see or measure)
- Useful if:
  - . . . we deal with *big data*
  - . . . we deal with data compression
  - . . . we want to understand image formats
  - . . . we want to gain an overview of our data
  - . . . we want to visualize our data (try a 25-D graph!)

Dimensionality Reduction



Density Estimation



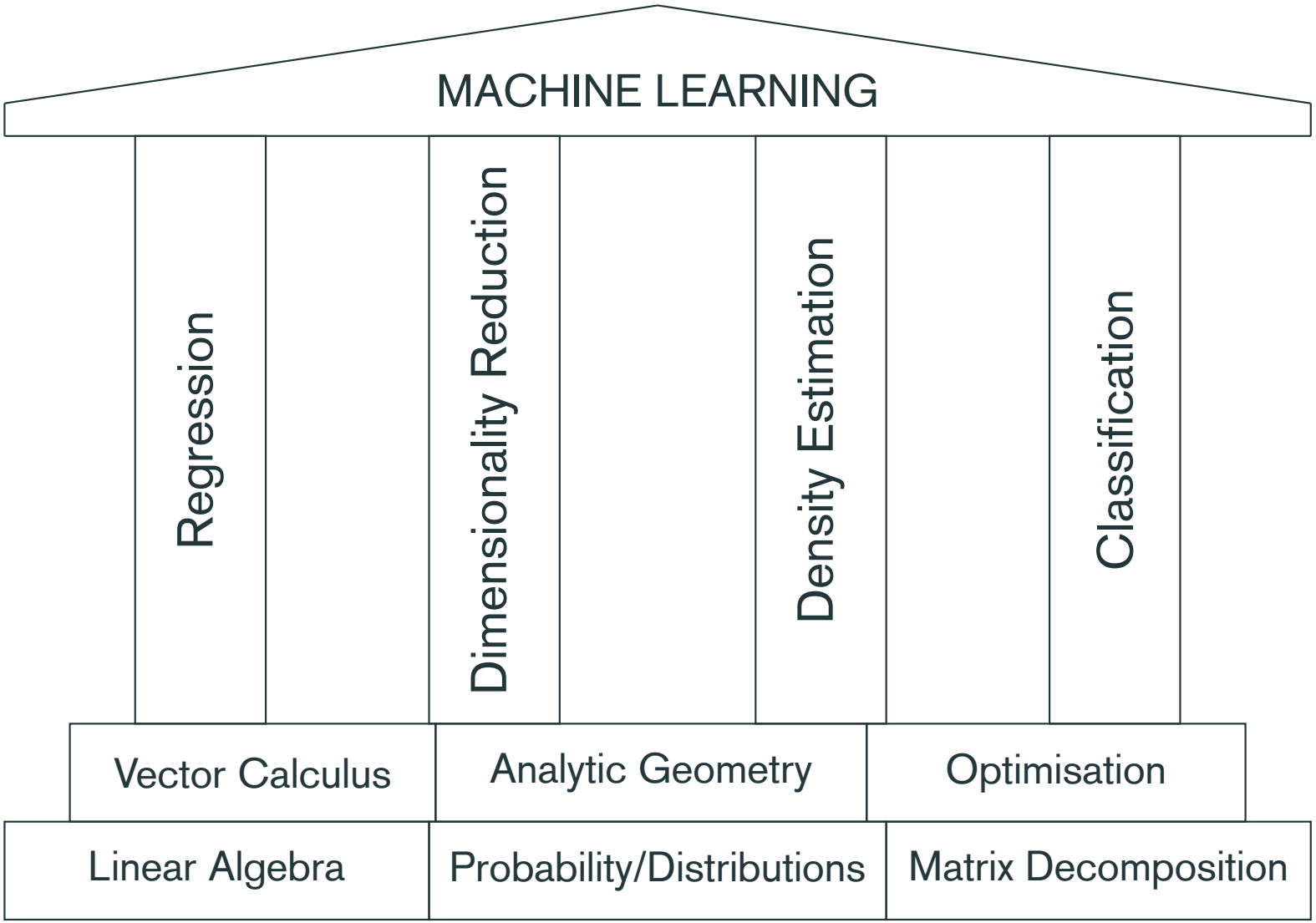
Density Estimation

- Esimate what the (unknown) underlying probability density function is
- Not all methods might work (multiple clusters?)
- Gaussian mixture model
  - Probabilistic model that assumes data comes from a mixture of a finite number of Gaussian distributions with unknown parameters

Classification

- What class do our data belong to?
- This refers to binary (or categorical) classification
- Examples
  - Support Vector Machines
  - k-Nearest Neighbour
  - Etc.

Foundations of Machine Learning (Deisenroth,Faisal, and Ong (2020))



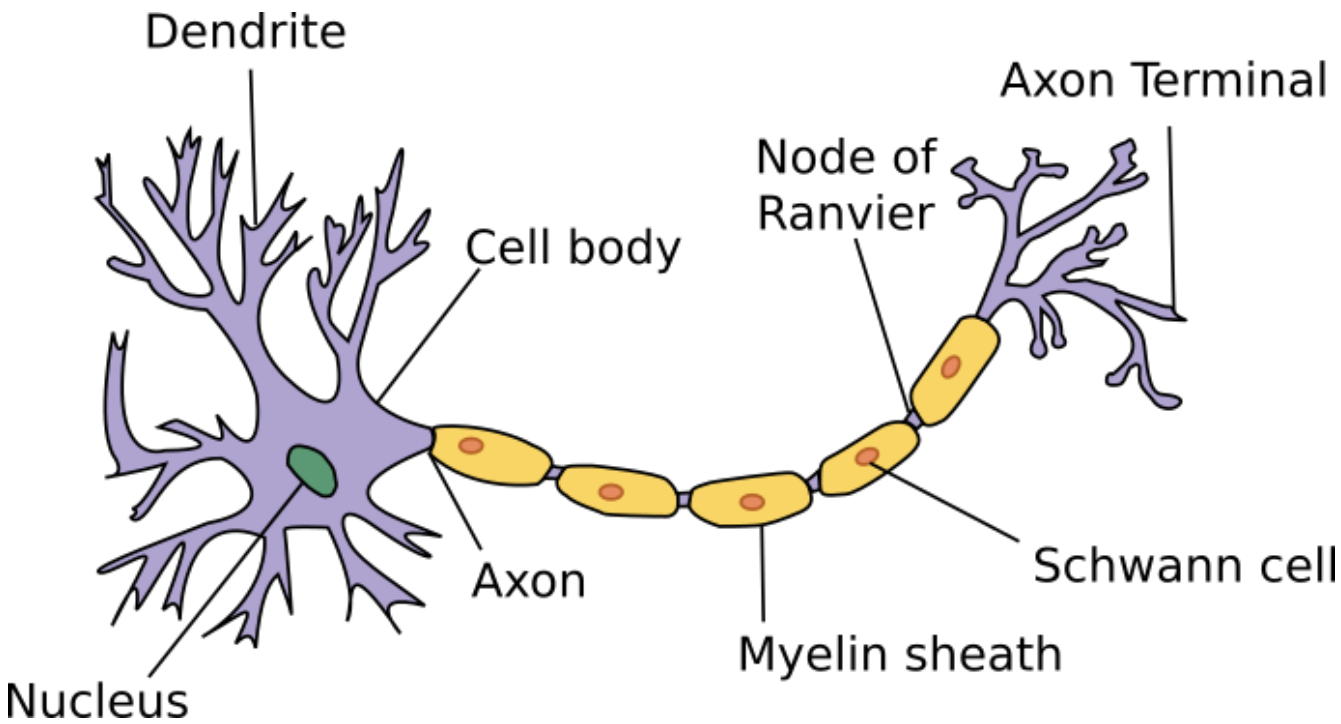
Types of Machine Learning

1. Supervised Learning
   - All our data are labeled (x_i, y_i)
2. Unsupervised Learning
   - All our data are not-labeled (x_i)
3. Reinforcement Learning
   - A new sample $(x_i, a_i, r(a_i))$appears every time that we take an action

Neural Networks

Neural Networks

- Neural Networks are models for programs!
- In our body neural networks work with neurons
  - Relay information, sense environment, induce actions
  - The *perceptron* is its computational model



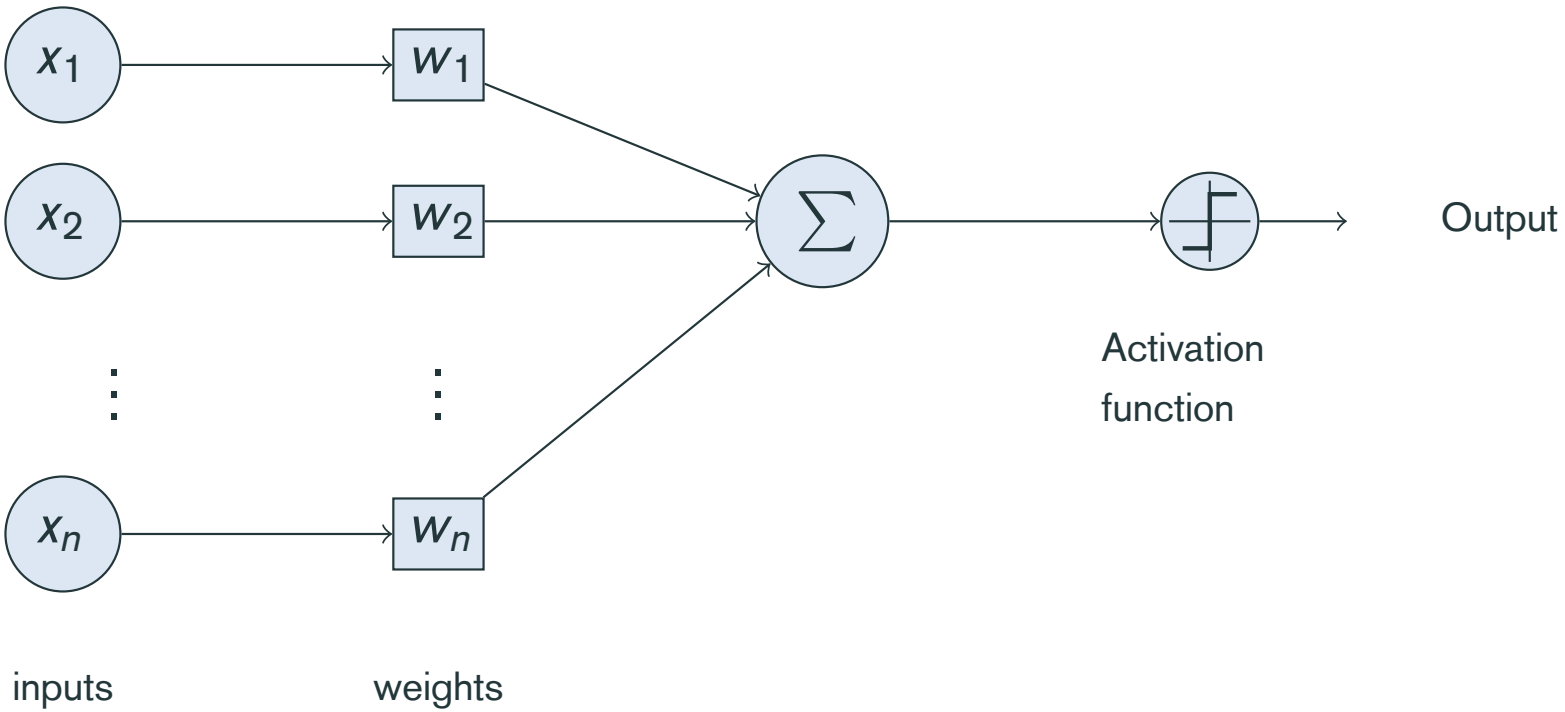
Andrew Ng

"A single neuron in the brain is an incredibly complex machine that even today we don't understand. A single 'neuron' in a neural network is an incredibly simple mathematical function that captures a minuscule fraction of the complexity of a biological neuron. So to say neural networks mimic the brain, that is true at the level of loose inspiration, but really artificial neural networks are nothing like what the biological brain does"

The perceptron is a computational model of a natural neuron:

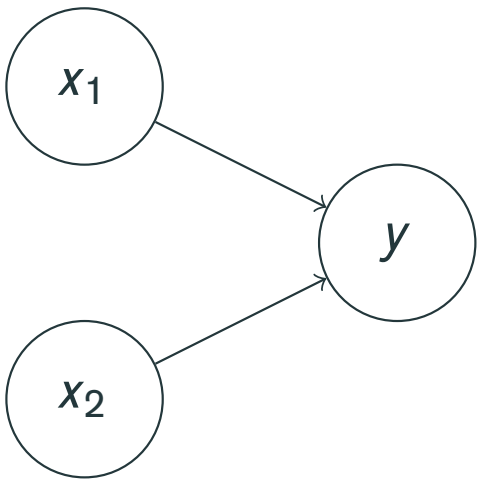
$$y = f\left(\sum_{i=1}^{N} w_i x_i + b\right)$$

- Only predicts values $y \in 0, 1$ (binary classification and logical gates)



Perceptron: Expressive Power

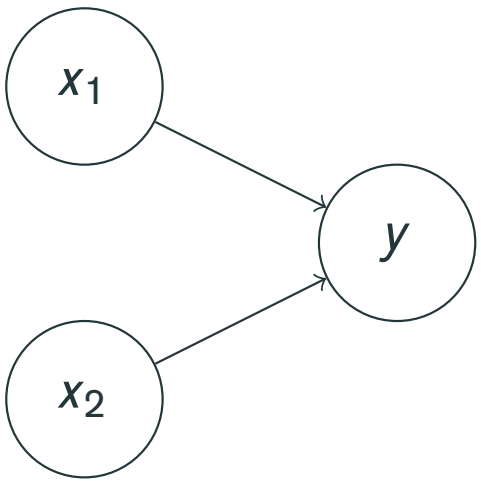
AND



connections = 1

$y = -2$

OR

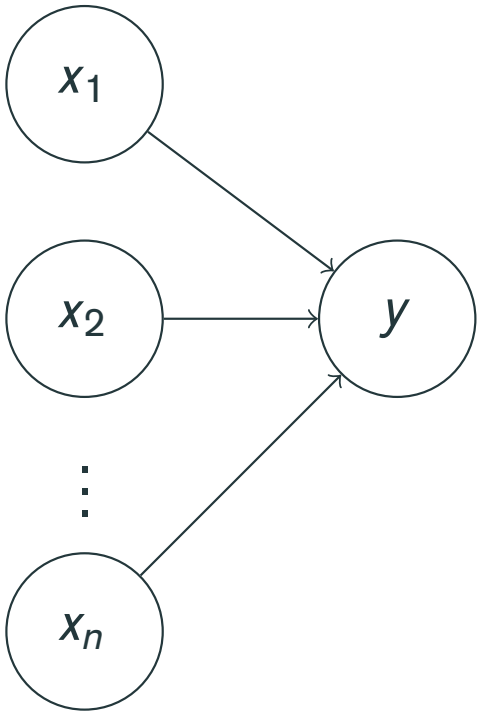


connections = 1

$y = -1$

Perceptron: Expressive Power

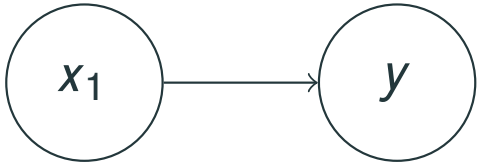
MOST



connections = 1

$y = -n$

NOT



connections = -1

$y = 0$

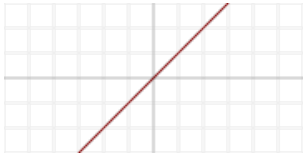
Neural Networks

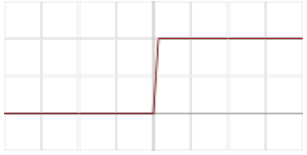
Definition

A neural network (or neural net or network or net consists of:

- an architecture a directed graph with a set of neurons and edges that specify how the neurons are connected
- a set of parameters weights of all the connections and biases of all the neurons

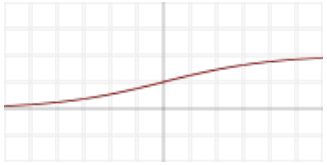
Activation Functions

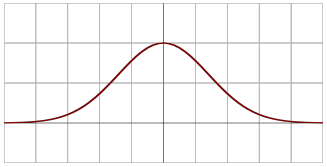
Identity  $f(x) = x$

Binary 

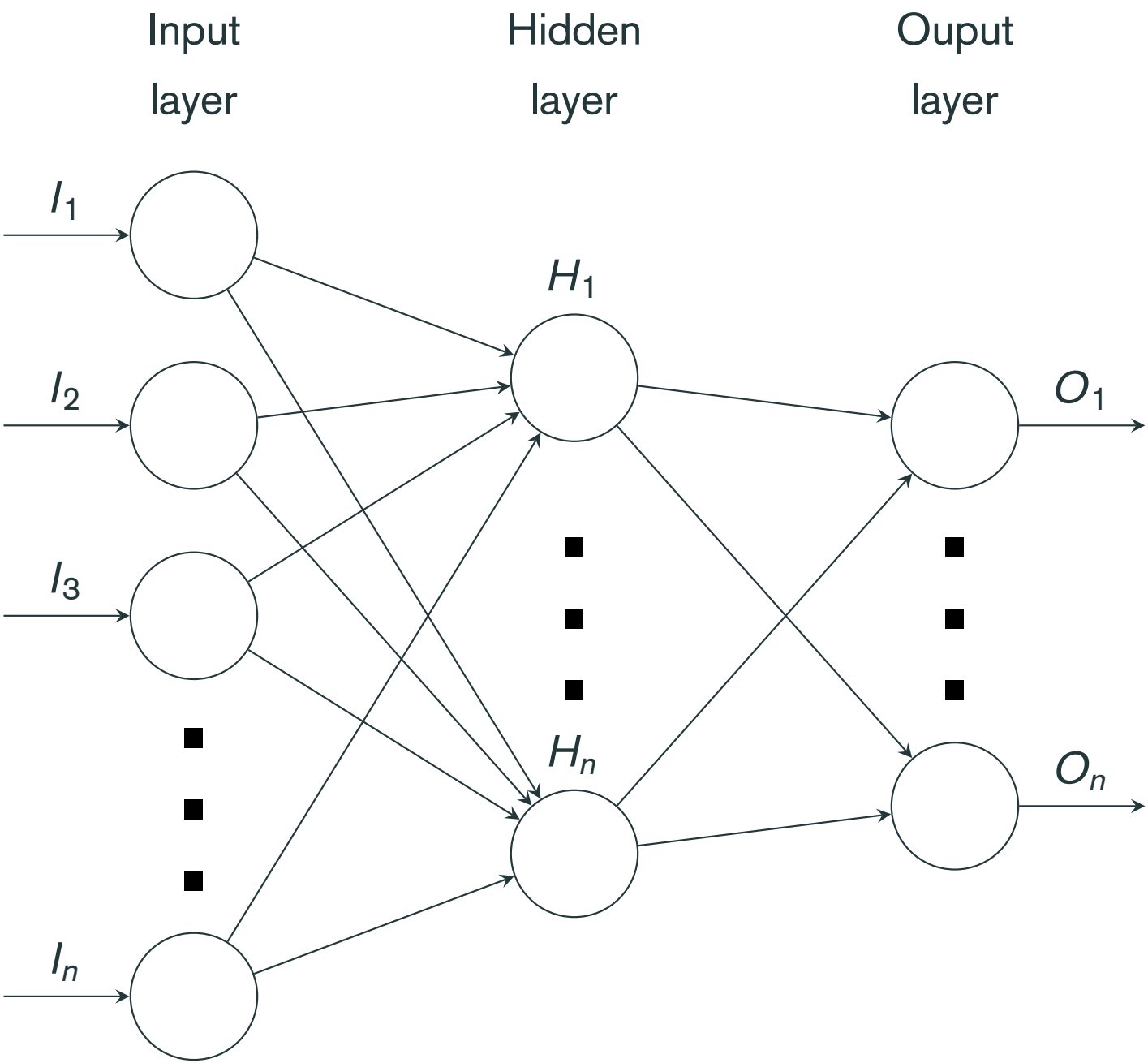
$f(x) = 1, \text{if x} >= 0$

$f(x) = 0, \text{if x} < 0$

Logistic  $f(x) = \frac{1}{1+e^{-x}}$

Gaussian  e^{-x^2}

Neural Networks

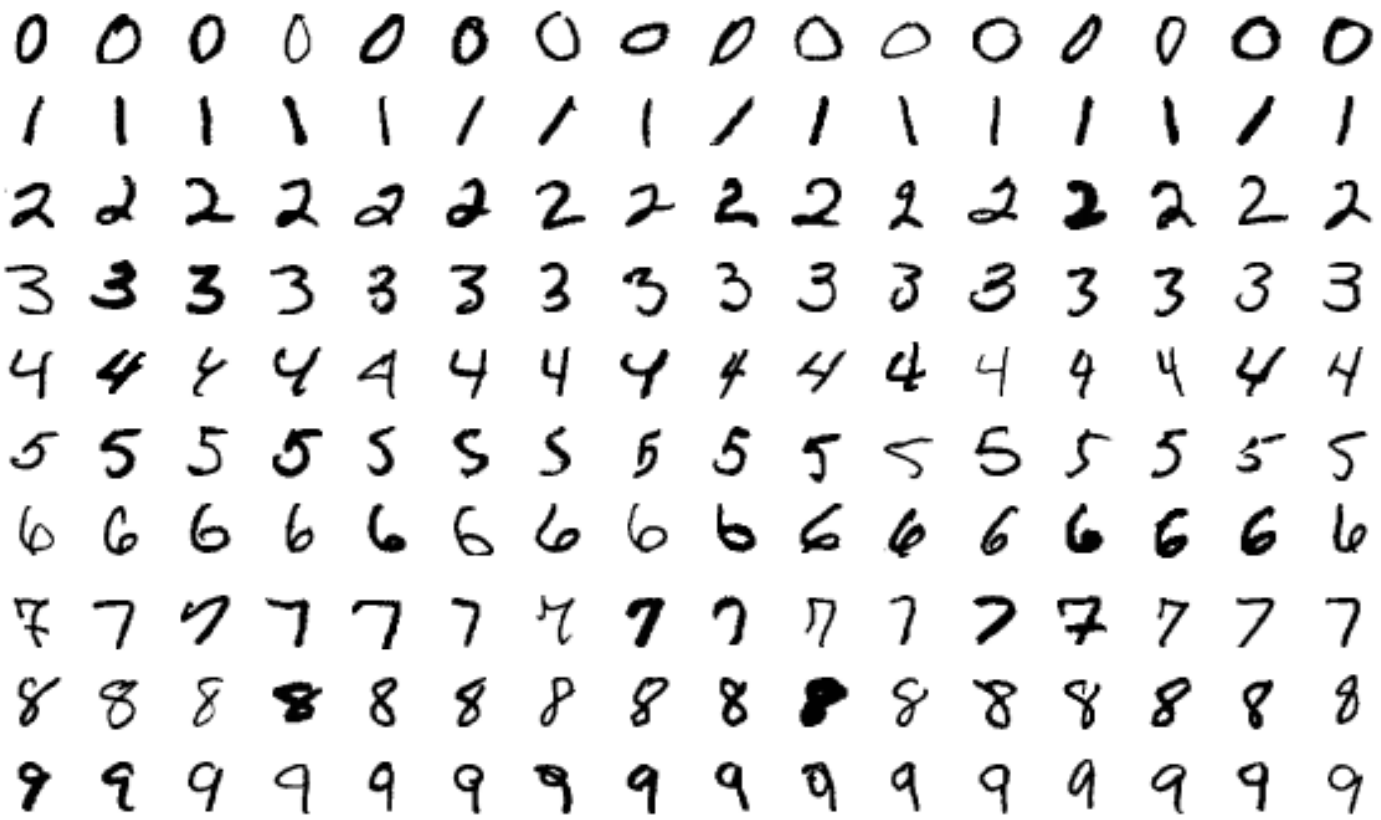


Neural Networks

- the i_j are called input nodes
- the h_j are called hidden nodes
- the o_j output nodes

- Any neuron of any network at any time will compute a value. This value is called the neuron's activation
- Feed forward networks are loop free. In any feed forward network, any input vector produces a well defined output vector

MNIST Data Set



- The MNIST data set consists of 70000 scanned images of handwritten digits (written by 250 different people)
- Each image consists of 28 by 28 pixels in greyscale with 0.0 representing white and 1.0 representing black

MNIST Data Set

- Input layer: $28 \times 28 = 784$ input neurons
- Hidden layer: 15 sigmoid neurons
- Output layer: 10 sigmoid neurons representing 0 through 9
- The most “active” output neuron is the answer

Expressive Power

- Hornik's universal approximation theorem shows that feedforward NNs can approximate any (bounded) mathematical function
- See: Hornik (1991) Approximation capabilities of multilayer feed forward networks. Neural Networks, 4(2), 251 257
- This is true even with a single hidden layer
- Yet
  - the theorem does not say how many hidden units we need
  - it doesn't say how the network should be trained
- although a single hidden layer is sufficient in theory, in practice it can be better to have several hidden layers

Training Feedforward Neural Networks

- Training a NN consists of finding the weights and bias in the layers
- So how do we find those weights and bias?
- What if we did not set the parameters (weights and biases) manually and instead tried to do it automatically somehow?
- We could use training data of the form (desired output) for specifying what the network should do
- What if we just set the parameters randomly at the start and then try to adjust them little by little (local search) so that the network becomes increasingly better on the training data?
- This idea of gradual improvement turns out to work!

Turning it into an optimization problem

- An input x is a $28 \times 28 = 784$ dimensional real valued vector
- The desired output (or label) of x is a 10 dimensional vector $y(x)$ For example, if x is a n image of a 3 then
  $y(x) = (0, 0, 0, 1, 0, 0, 0, 0, 0, 0)$
- This is called one hot encoding
- An example among many possibilities of a cost function (or *loss* function or error function) for a given data set of size n:
  $C(w, b) \equiv \frac{1}{2n} \Sigma_x \parallel y(x) - \text{actual output}(x) \parallel^2$
  where $\parallel \ldots \parallel$ is the Euclidean norm. This is the quadratic cost function a.k.a. the mean squared error or the MSE

Cost function

- Note that $C(w, b)$ is non negative
- It is essentially an average of the error on the data set
- Furthermore, the cost $C(w, b)$ becomes small, i.e., $C(w, b) \approx 0$, precisely when $y(x)$ is approximately equal to the output, for all training inputs x
- We want to minimize C, i.e. find (thousands of) weights and biases that make the cost as small as possible

Other Issues

- How explainable/interpretable/transparent are networks?
- Convolutional Neural Networks
  - Weakness of our network: If we shift a digit slightly left/right and/or up/down, our present network will not necessarily recognize it anymore
  - Convolutional networks can handle this
  - Today, convolutional networks are used in most networks for image recognition
  - Key idea: spotting patterns in images (or other data)

CHALMERS
UNIVERSITY OF TECHNOLOGY

MACHINE LEARNING AND NEURAL NETWORKS

PART ONE

Bastiaan Bruinsma

February 28, 2022

Chalmers University of Technology