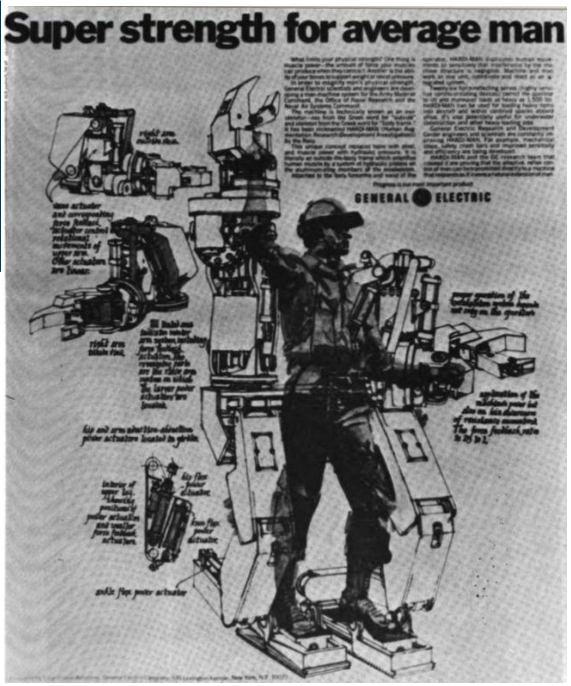
Intent detection and somatosensory feedback

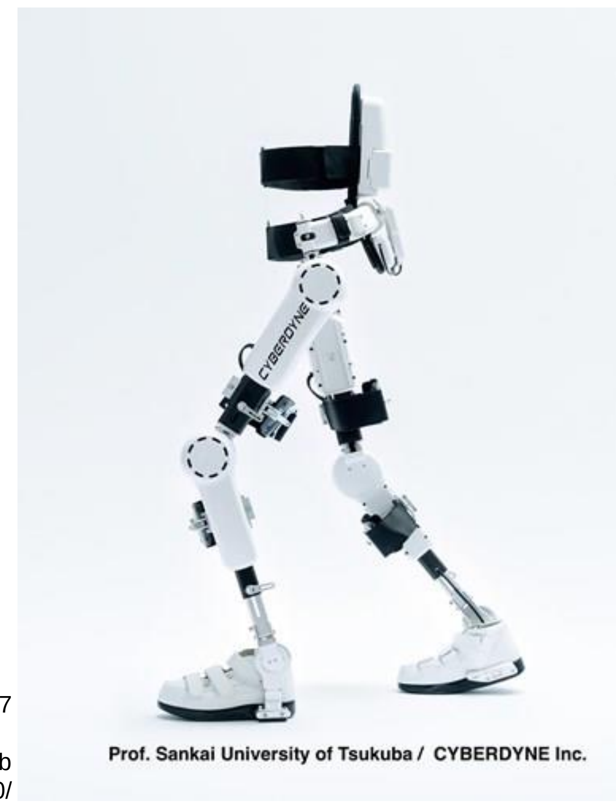
#5: Intention Detection

Claudio CASTELLINI, Sabine THÜRAUF



The *Hardiman,* General Electric, 1965/1970.

http://cyberneticzoo.com/man-amplifiers/1966-69-g-e-hardiman-i-ralph-mosher-american/

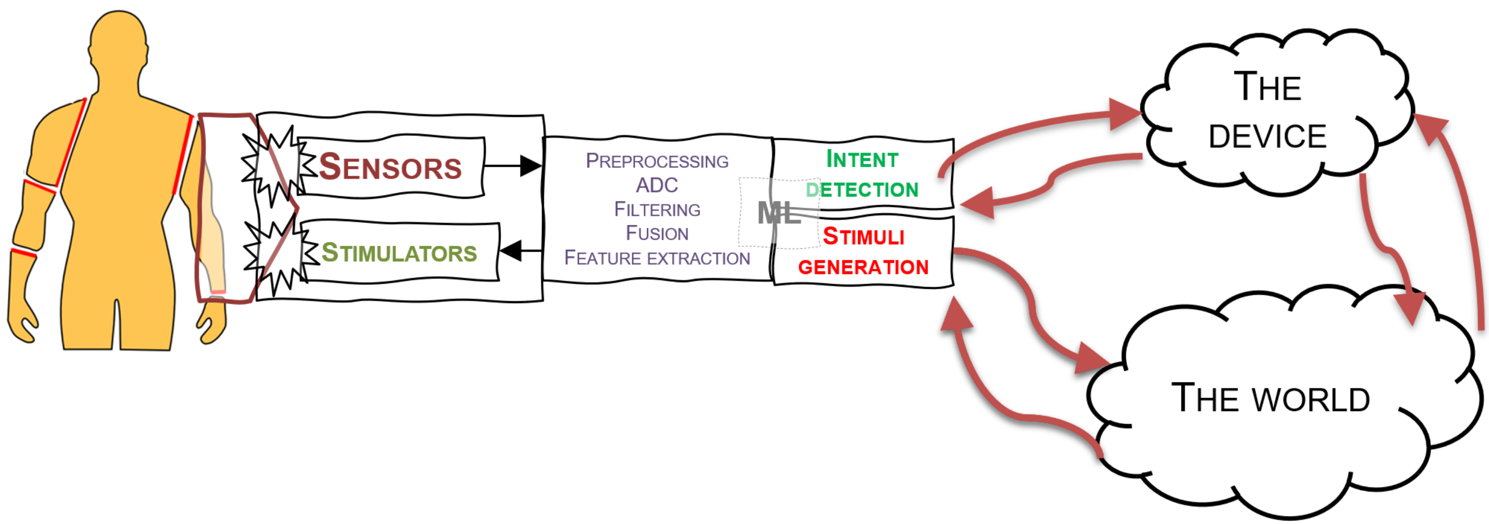


The *HAL Lumbar* by Cyberdyne, Inc., 2017

https://roboticsandautomationnews.com/2017/10/02/cyberdyne-launches-new-version-of-its-exoskeleton/14360/

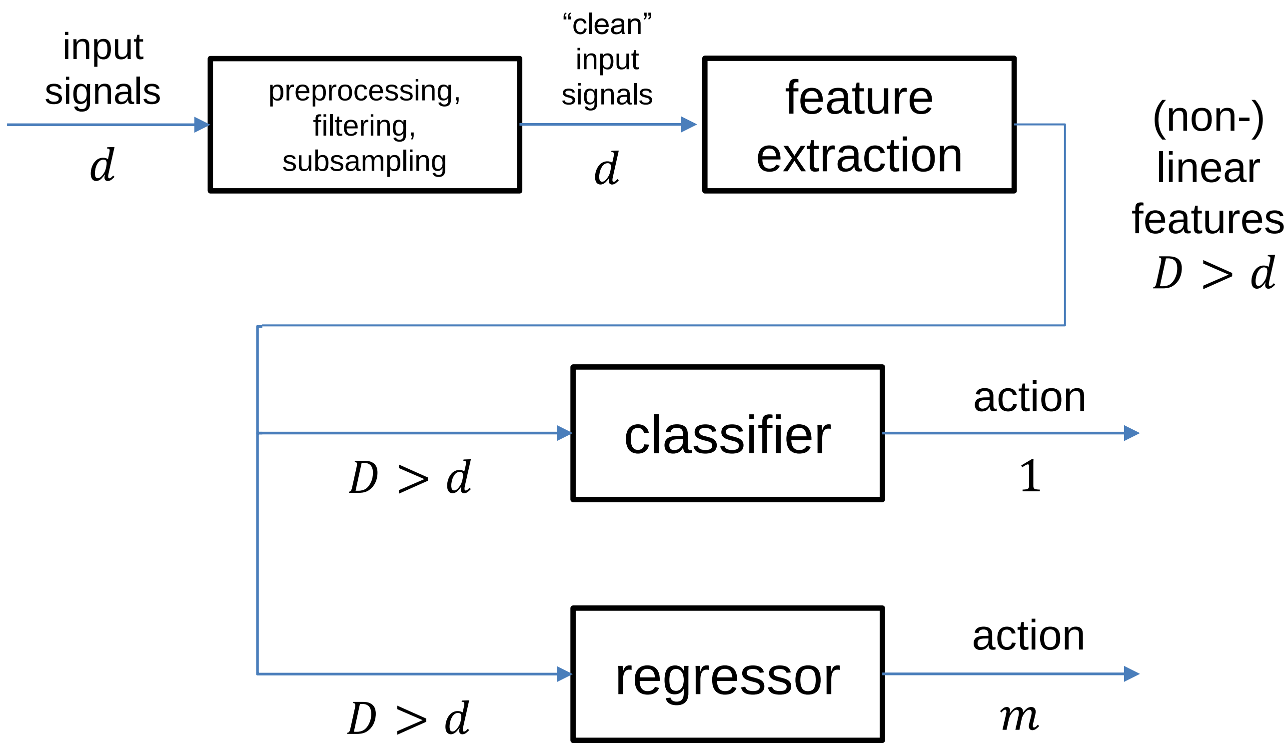
Theory Recap

Recall the control pipeline for prostheses



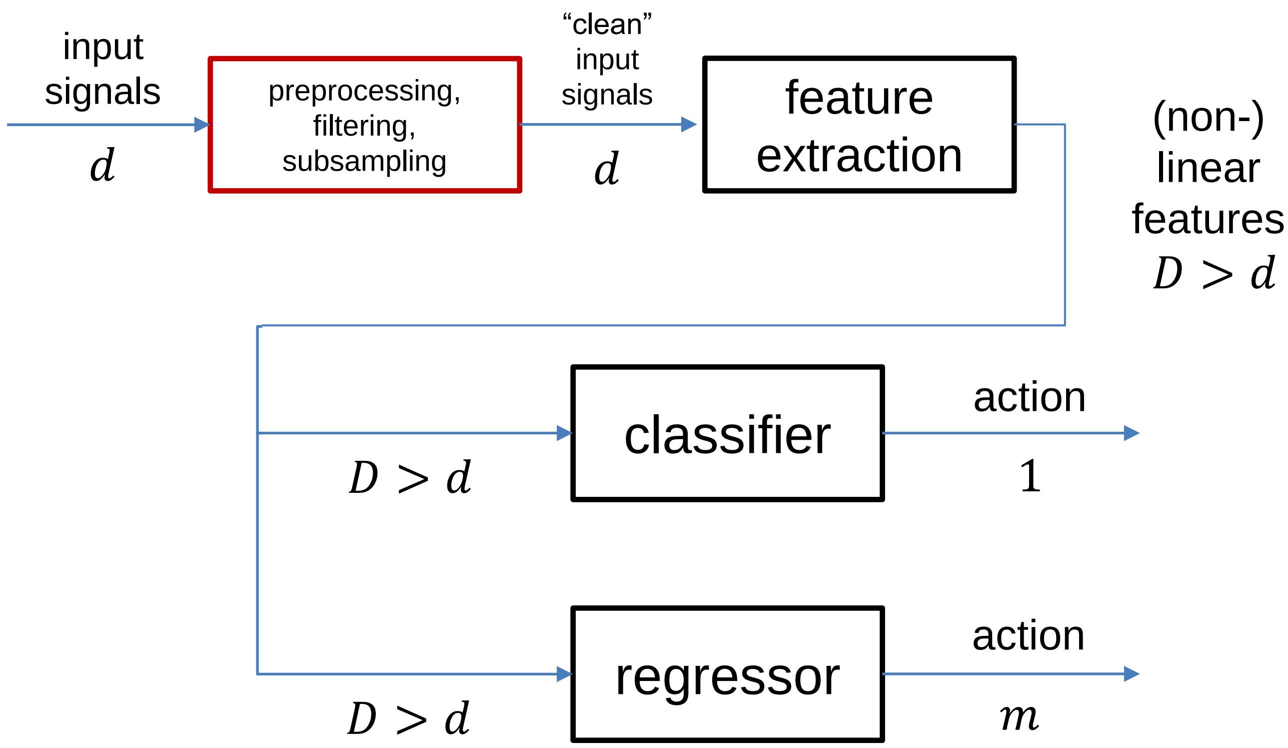
Theory Recap

Recall the control pipeline for prostheses



Theory Recap

Recall the control pipeline for prostheses



1. Pre-Processing – Example filtering

Input signal might be very noisy, have outliers, etc.

Task 1: How can we filter the signal to gain a „clean“ input signal?

1. Pre-Processing – Example filtering

Input signal might be very noisy, have outliers, etc.

Task 1: How can we filter the signal to gain a „clean“ input signal?

Task 2: Apply a sliding median filter with a window size of 5 for the following data set:

Time	1	2	3	4	5	6	7	8
Signal	0.5	3.4	2.1	1.5	4.2	-2.0	7.3	2.1

1. Pre-Processing – Example filtering

Input signal might be very noisy, have outliers, etc.

Task 1: How can we filter the signal to gain a „clean“ input signal?

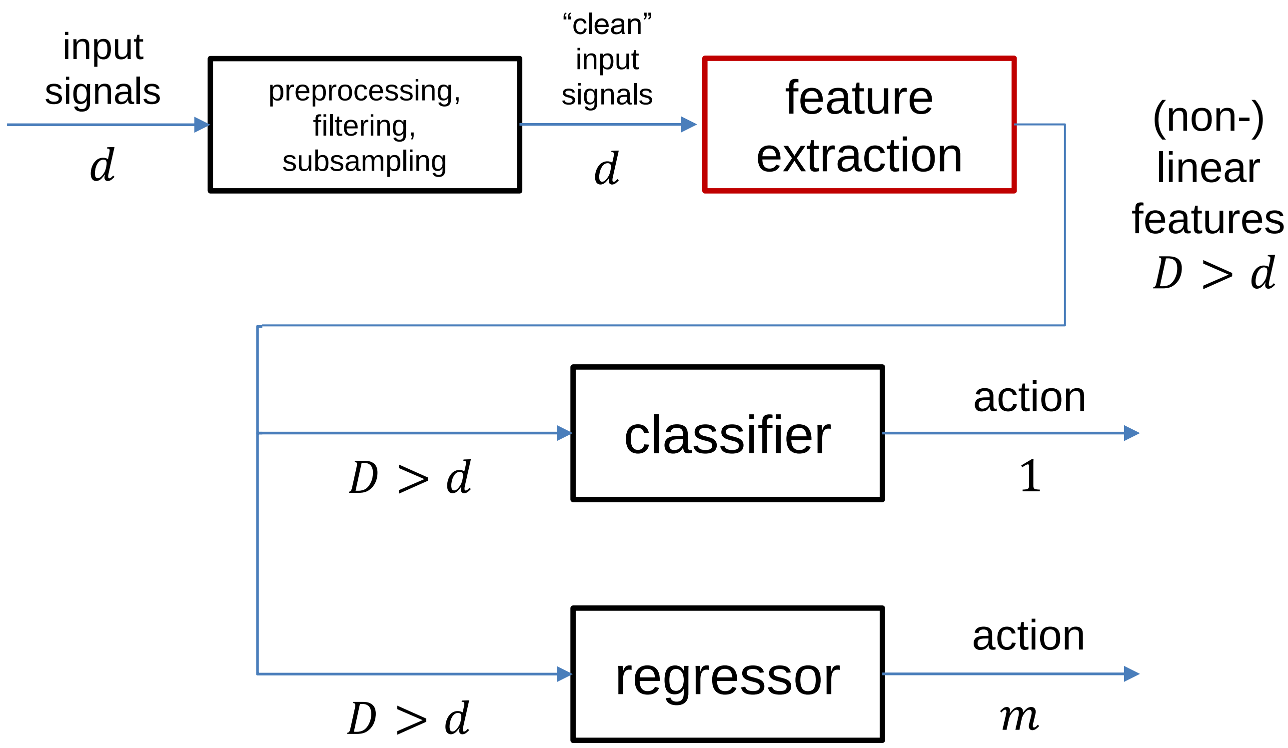
Task 2: Apply a sliding median filter with a window size of 5 for the following data set:

Time	1	2	3	4	5	6	7	8
Signal	0.5	3.4	2.1	1.5	4.2	-2.0	7.3	2.1
Mean			2.1	2.1	2.1	2.1		

0.5, 1.5, 2.1, 2.1, 2.1, 2.1, 2.1, 2.1

Theory Recap

Recall the control pipeline for prostheses



2. Feature Extraction

What are features?

Flower:

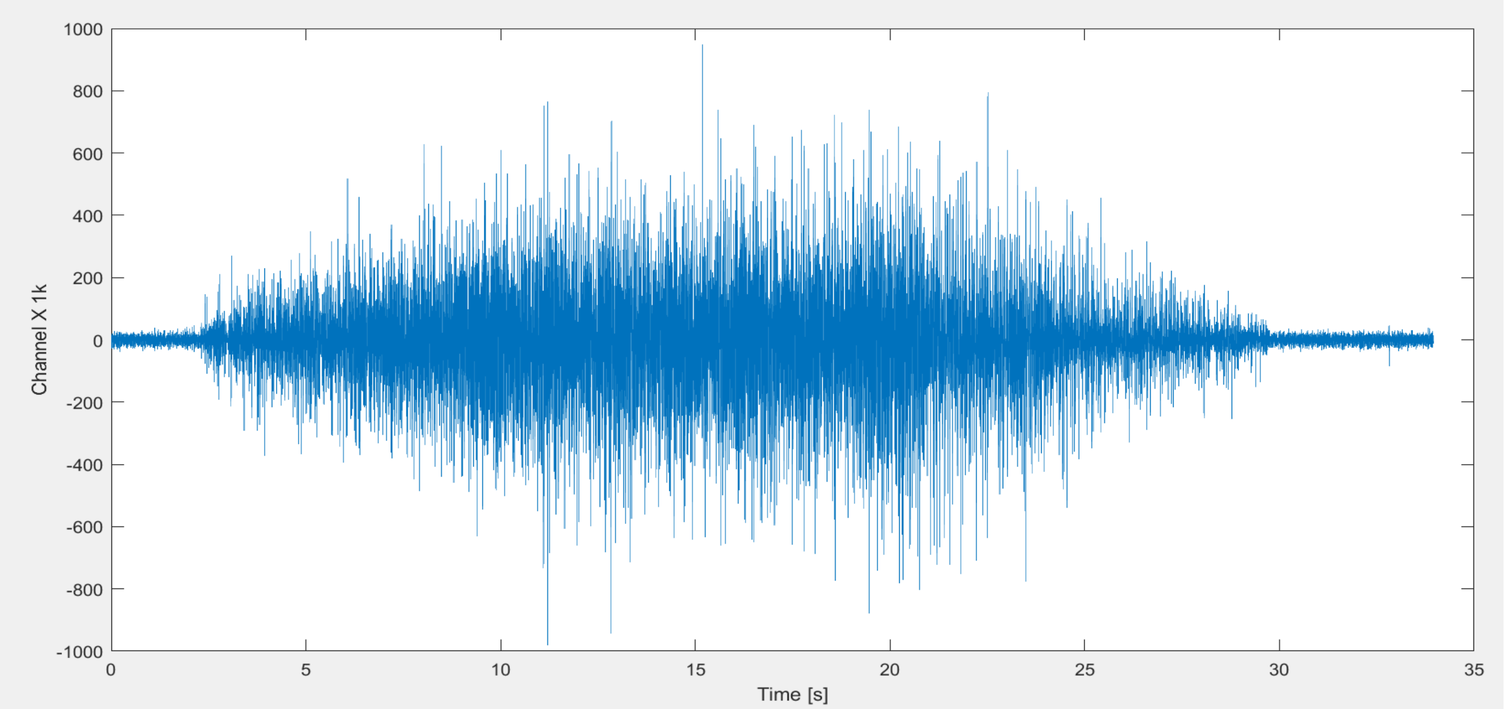
- Color
- Size
- Number of petals
- …

Image

- Size
- Color channels
- …

2. Feature Extraction

What are features?



2. Feature Extraction

1. Features are not always that obvious like in the flower image
2. Relevant features need to be selected
3. Features can be extracted from a signal using different algorithms

Eigenvalues/Eigenvectors and Co-Variances
are often a good starting point!

2. Feature Extraction

For input $\mathrm{X} = \left[x_1^T \ \dots x_{\mathrm{n}}^T\right]^T \in \mathbb{R}^{\mathrm{n} \times d}$ with $x_i = \left[x_{1,i} \ \dots x_{\mathrm{m},i}\right]^T i \in [1, \dots, n]$

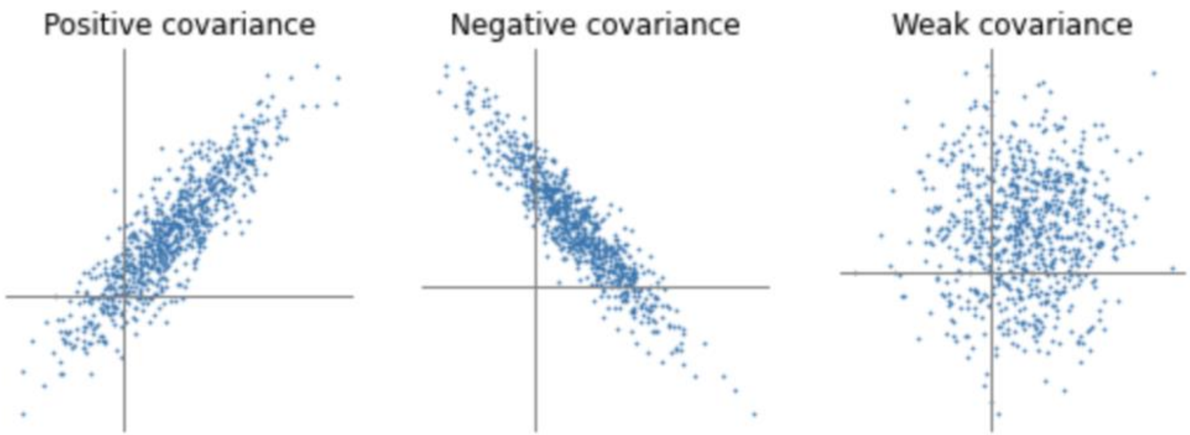
Standardization $\widetilde{x_{i,j}} = \frac{x_{i,j} - \overline{x_i}}{\sigma}$

Covariance two features/channels $\sigma(x_1, x_2) = \frac{\sum_{k=1}^{m}(x_{k,1} - \overline{x_1})(x_{k,2} - \overline{x_2})}{m}$

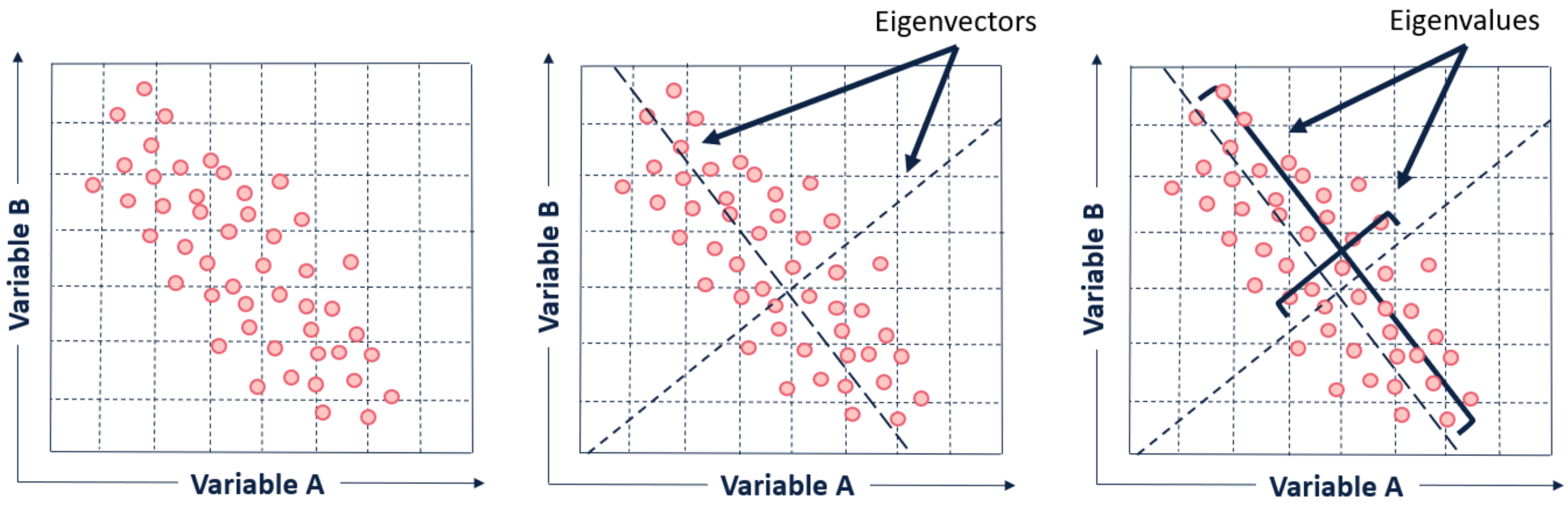
Covariance matrix of two features $C_{1,2} = \begin{pmatrix} \sigma(x_1, x_1) & \sigma(x_1, x_2) \\ \sigma(x_2, x_1) & \sigma(x_2, x_2) \end{pmatrix}$

Eigenvalues & -vectors $Ax = \lambda x \Leftrightarrow Ax - \lambda x = 0 \Leftrightarrow (A - \lambda I) \cdot x = 0$

2. Feature Extraction

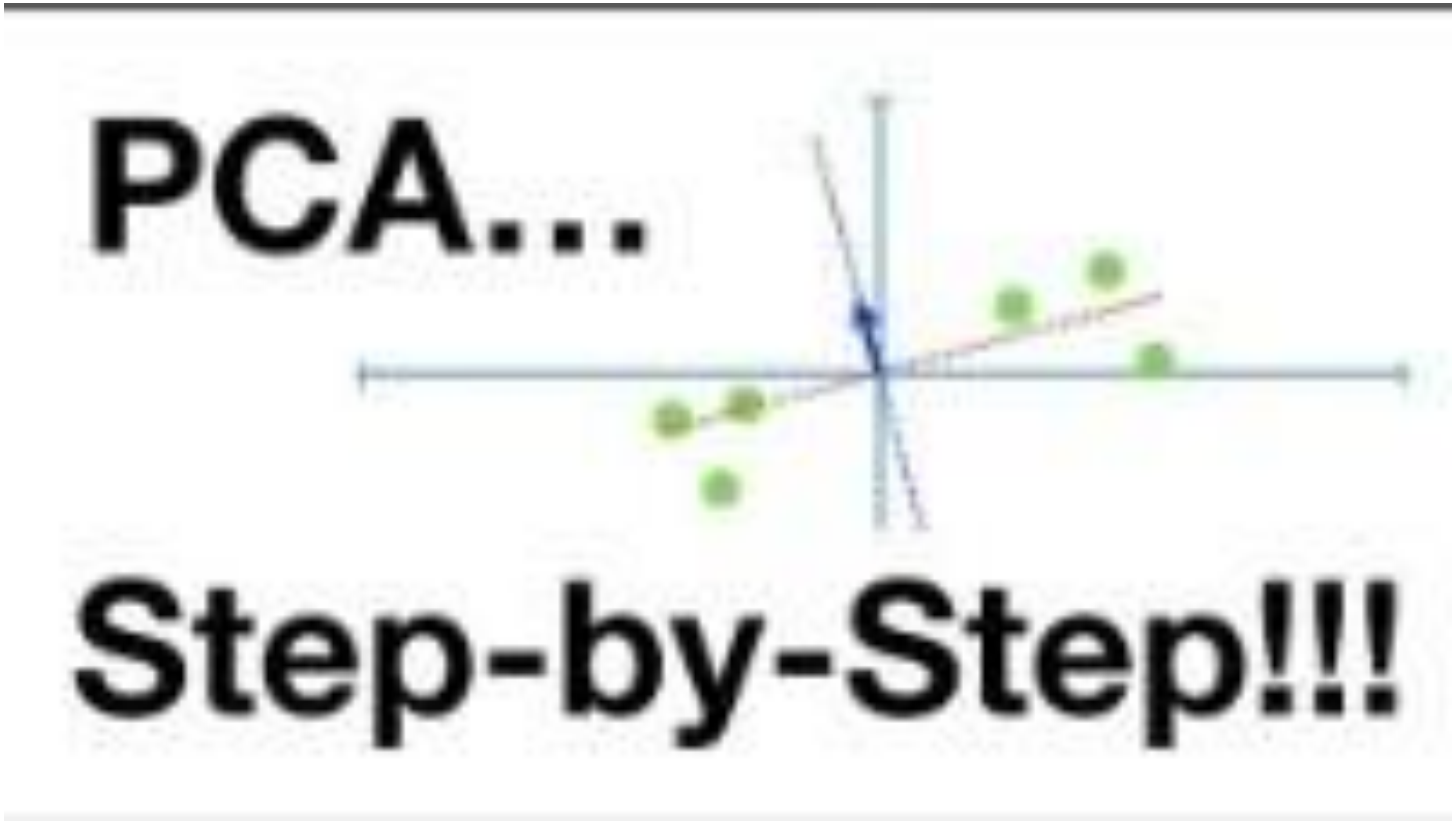


2. Feature Extraction



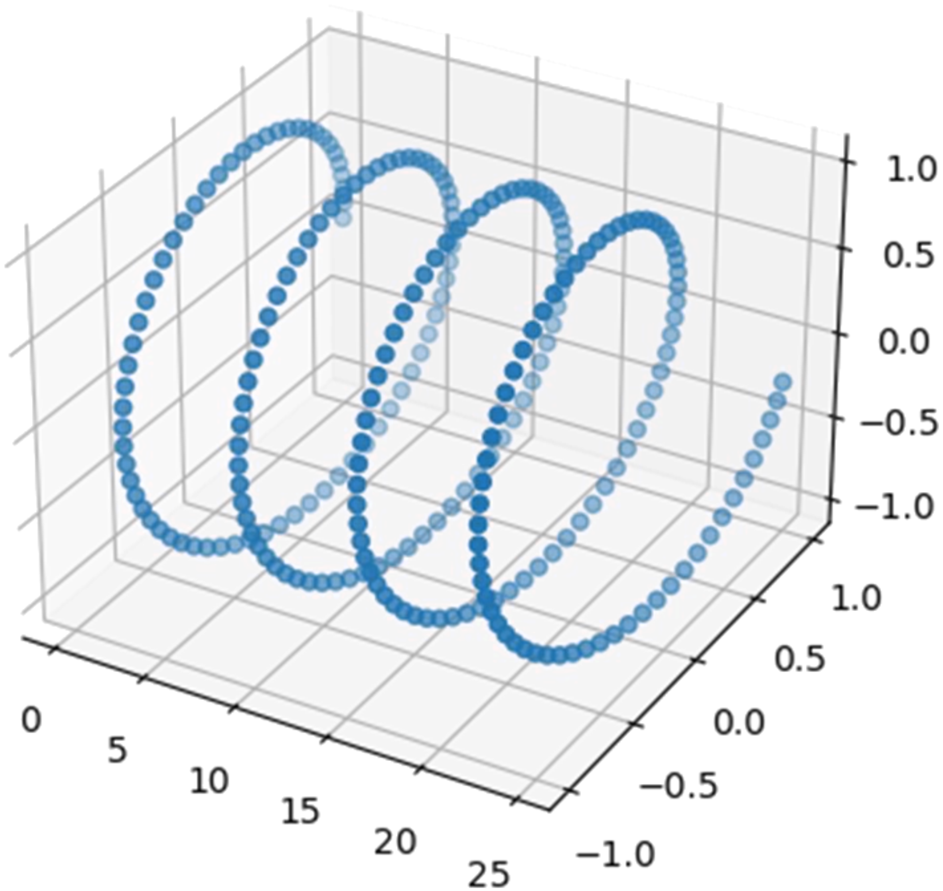
2. Feature Extraction

Principal Component Analysis (PCA)



https://www.youtube.com/watch?v=FgakZw6K1QQ

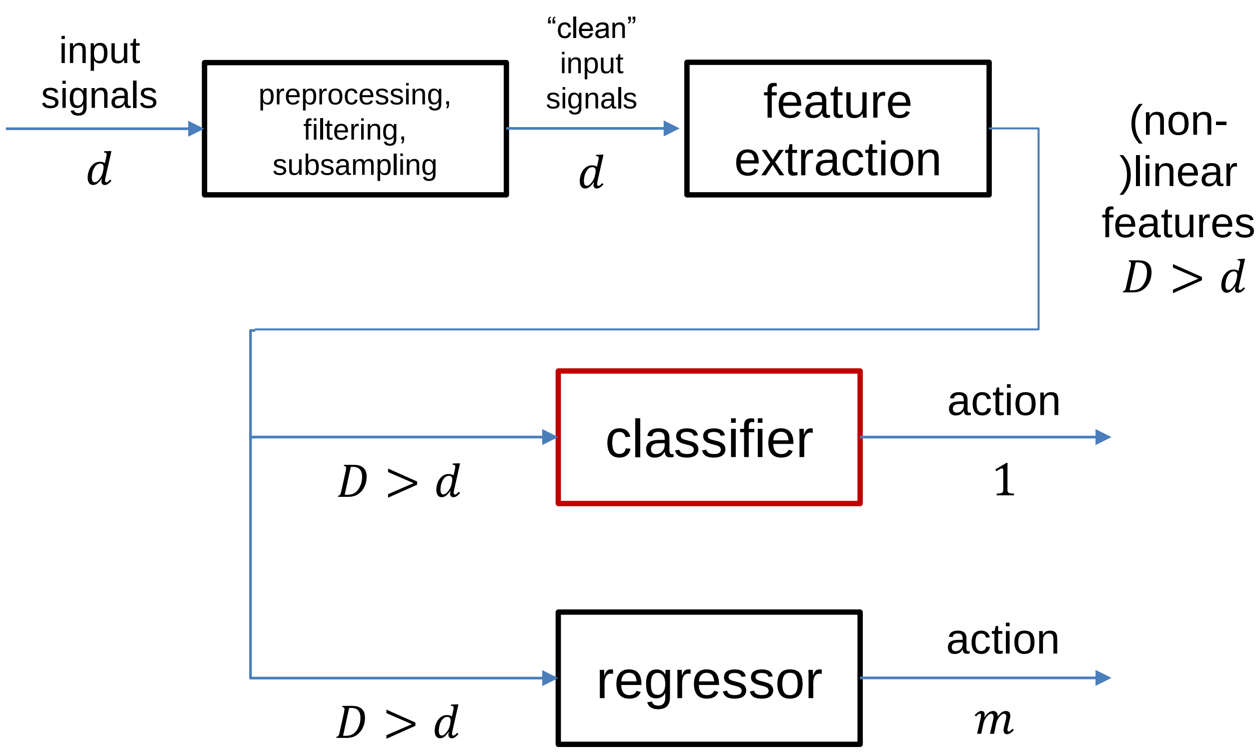
2. Feature Extraction



https://datagy.io/matplotlib-3d-scatterplot/

Theory Recap

Recall the control pipeline for prostheses



3. Classification

How do we classify our features?

- Decision Trees
- K-Nearest Neighbours (k-NN)
- Support Vector Machines (SVM)

3. Classification

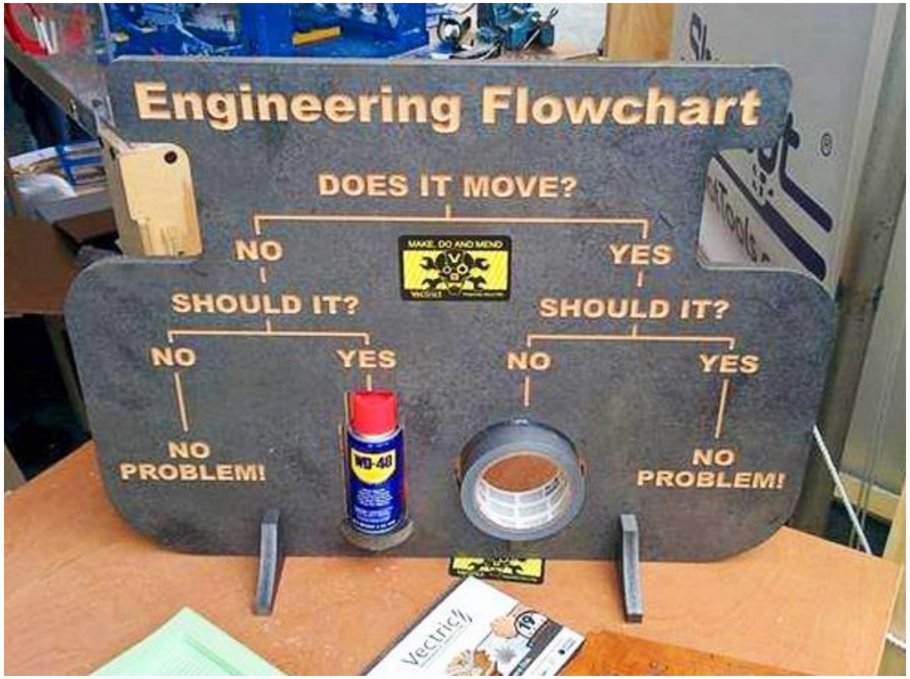
How do we classify our features?

- Decision Trees

3. Classification

How do we classify our features?

- Decision Trees

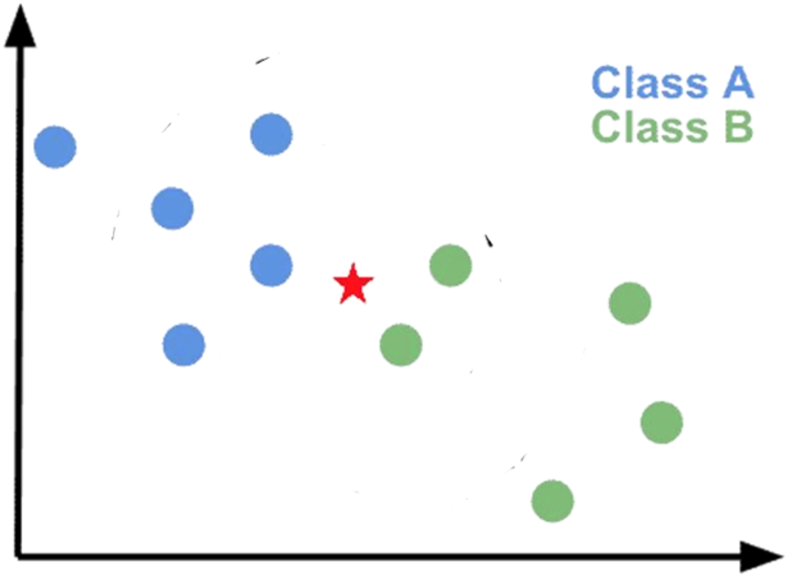


https://www.caliberdesign.co.nz/wp-content/uploads/2015/11/engineering-flowchart.jpg

3. Classification

How do we classify our features?

- Decision Trees
- K-Nearest Neighbours (k-NN)

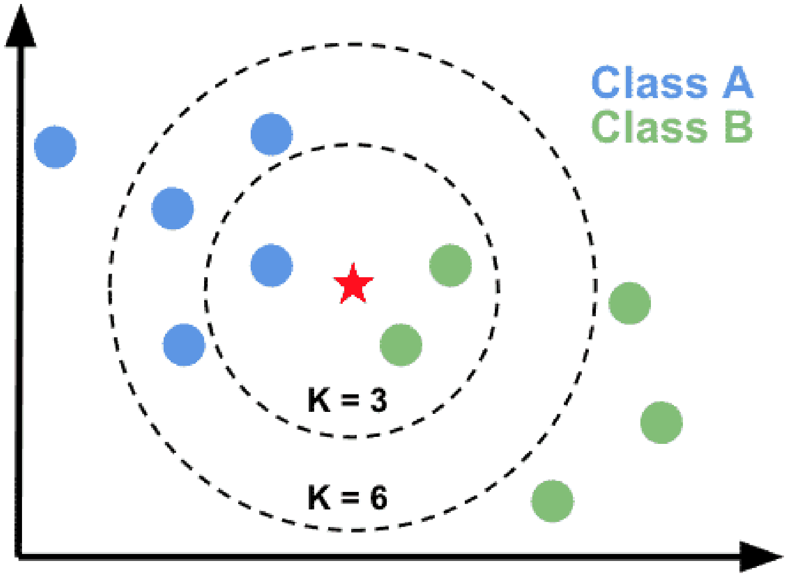


https://www.jcchouinard.com/wp-content/uploads/2021/08/image-8.png

3. Classification

How do we classify our features?

- Decision Trees
- K-Nearest Neighbours (k-NN)



https://www.jcchouinard.com/wp-content/uploads/2021/08/image-8.png

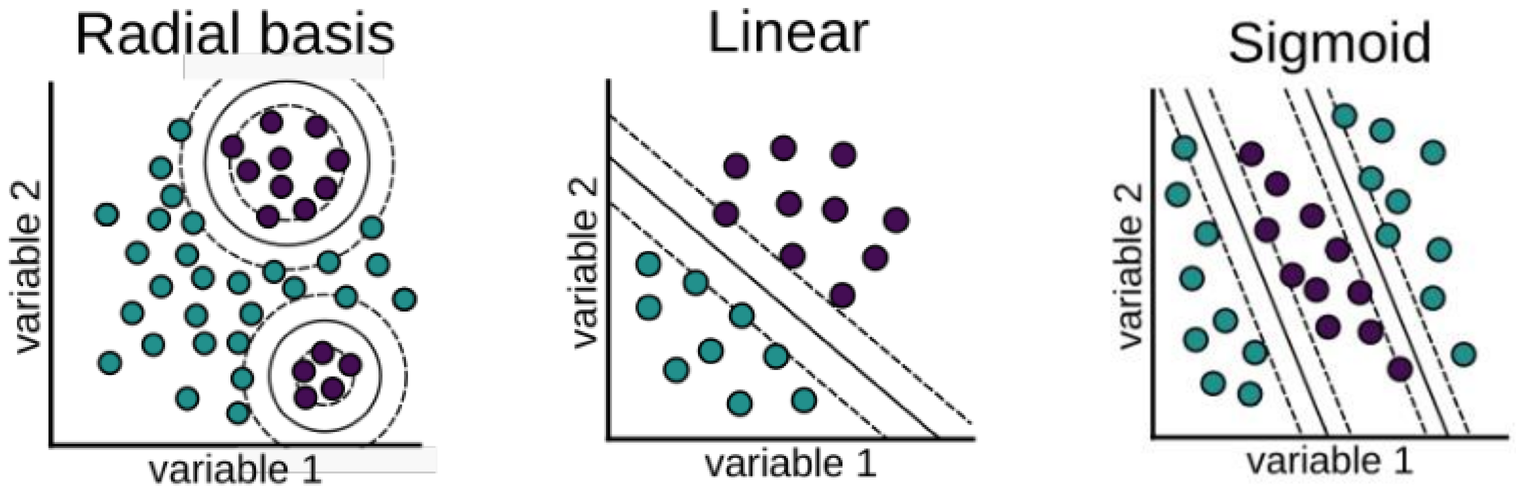
3. Classification

How do we classify our features?

- Decision Trees
- K-Nearest Neighbours (k-NN)
- Support Vector Machines (SVM)

3. Classification

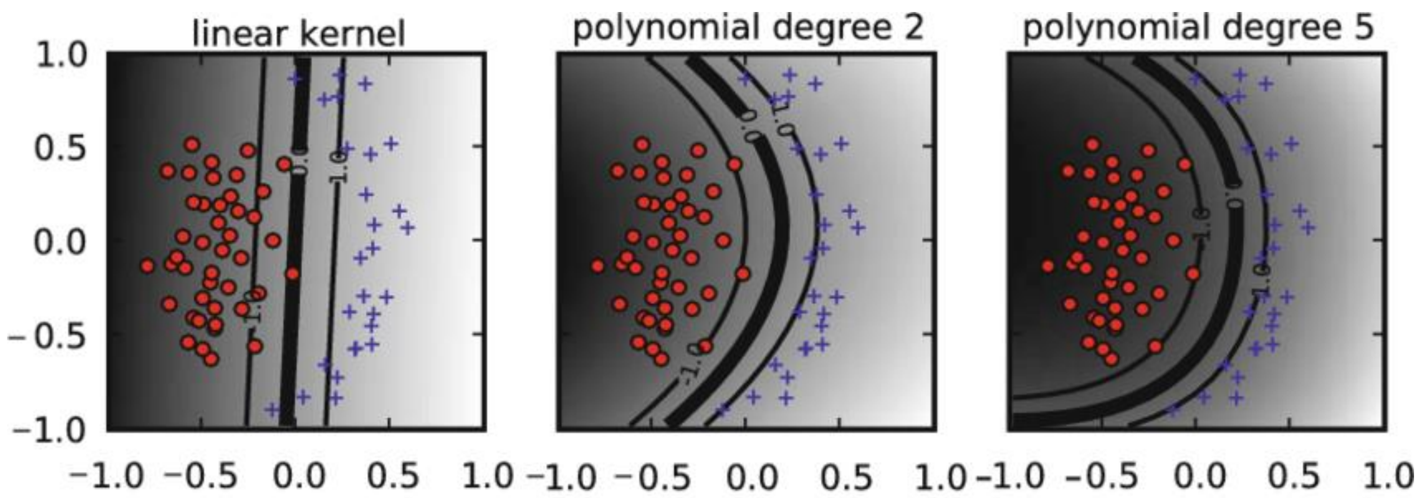
How do we classify our features?



https://medium.com/geekculture/kernel-methods-in-support-vector-machines-bb9409342c49

3. Classification

How to prevent overfitting?

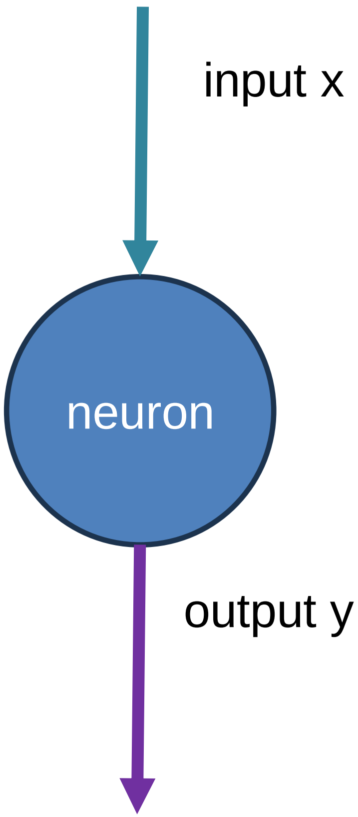


https://www.researchgate.net/profile/Asa-Ben-Hur/publication/41896604/figure/fig10/AS:667041182785544@1536046419481/The-effect-of-the-degree-of-a-polynomial-kernel-Higher-degree-polynomial-kernels-allow.png

3. Classification

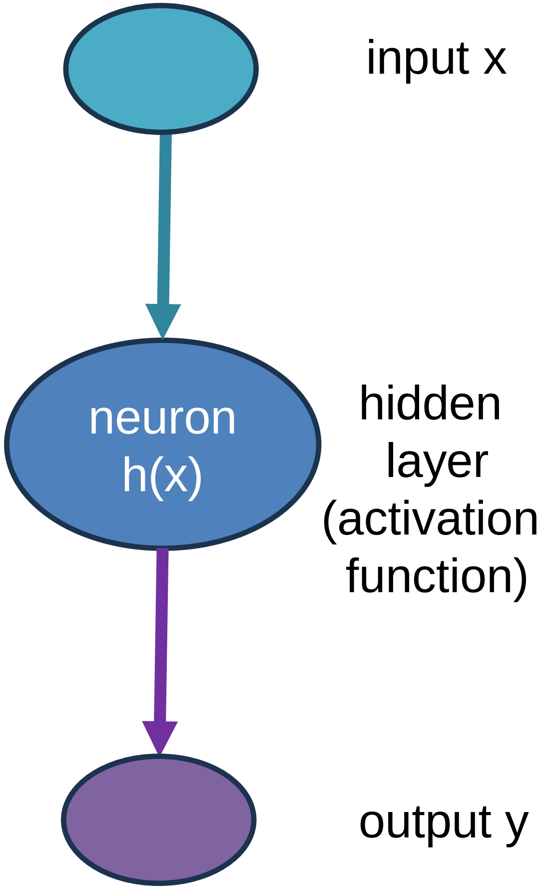
How do we classify our features?

- Decision Trees
- K-Nearest Neighbours (k-NN)
- Support Vector Machines (SVM)
- Neural Networks



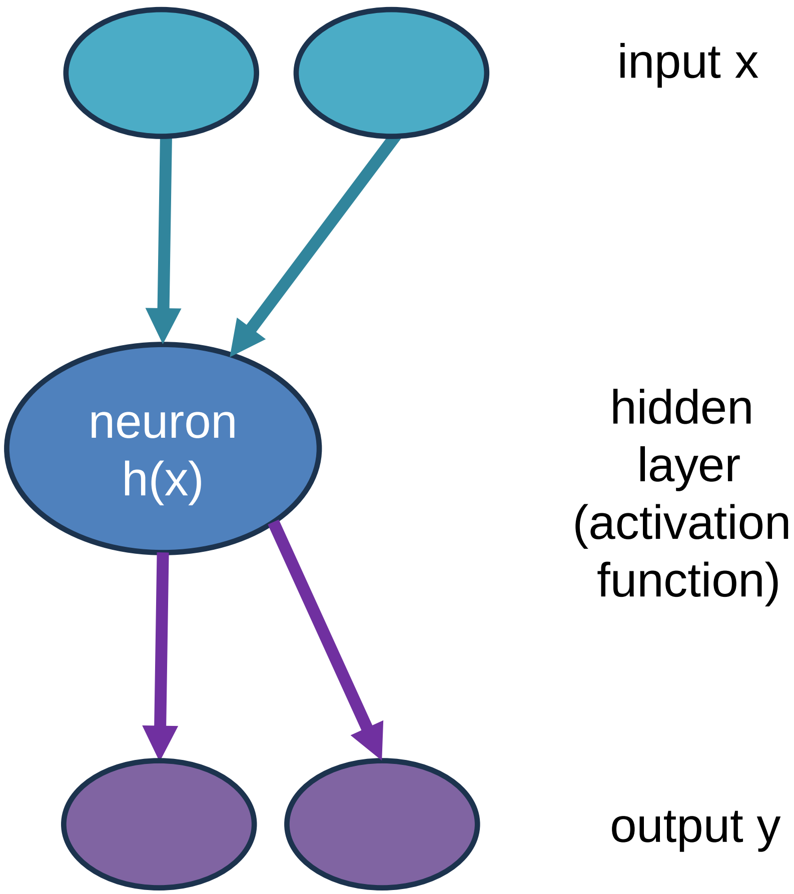
4. Neural networks

$$y = h(x)$$



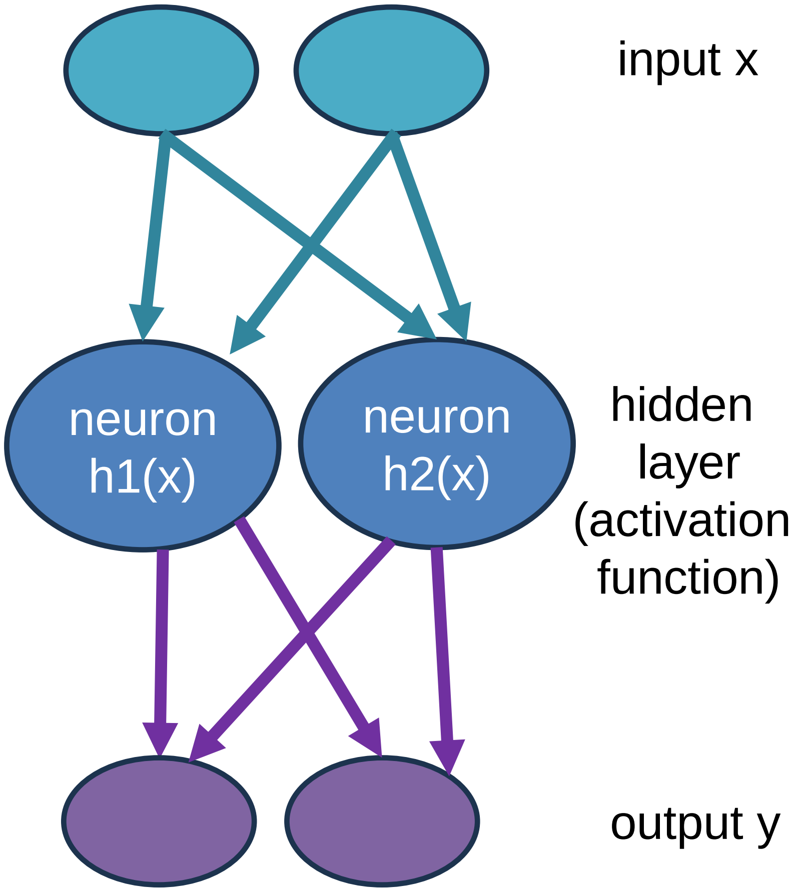
4. Neural networks

$$h(x_1, x_2) \rightarrow (y_1, y_2)$$



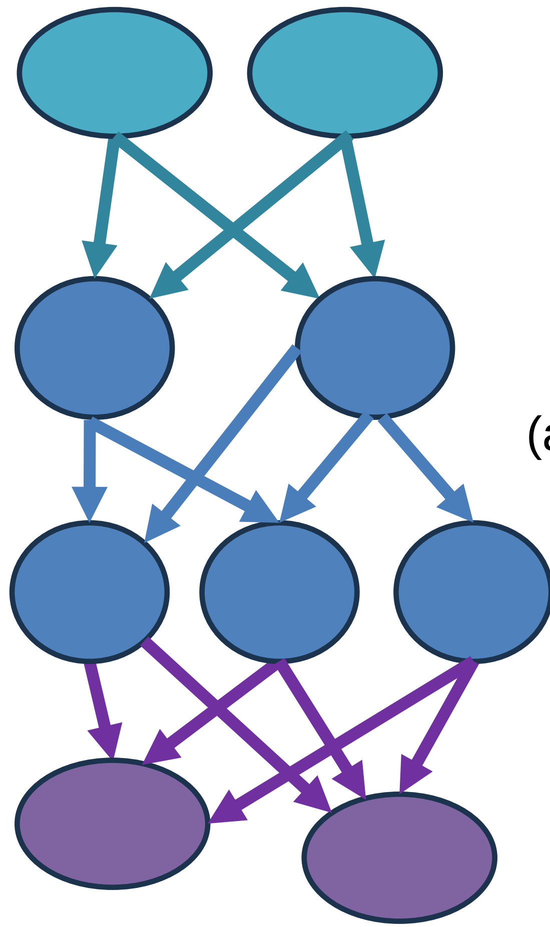
4. Neural networks

$$h_1(x_1, x_2) + h_2(x_1, x_2) \rightarrow (y_1, y_2)$$

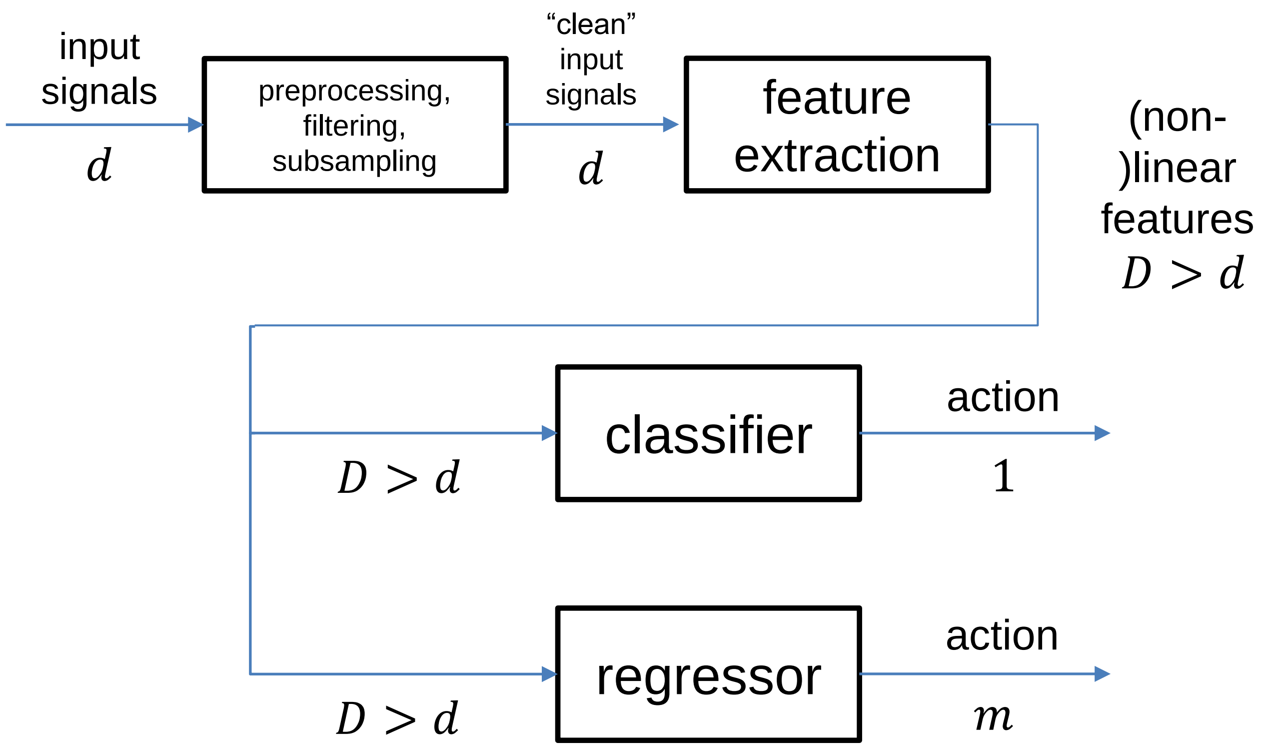


4. Neural networks

- There can be multiple hidden layers
- At least 3 layers ▸ deep learning



Summary


input signals
d
preprocessing, filtering, subsampling
"clean" input signals
d
feature extraction
(non-)linear features
$D > d$
classifier
$D > d$
action
1
regressor
$D > d$
action
m