IMPERIAL

Machine Learning for Neuroscience

ML4NS

Summary and Overview

Payam Barnaghi
Department of Brain Sciences &
School of Convergence Science in Human and Artificial Intelligence
Imperial College London
January 2025

1

1

From Day 1: What you are going to learn in this module

IMPERIAL

- Primary machine learning topics and concepts
- Hands-on skills on how to:
  - Decide and choose suitable models
  - Design machine learning tasks/experiments
  - Choose and apply suitable metrics to evaluate the models
  - Identify and resolve common issues such as overfitting/underfitting

2

2

From Day 1: By the end of this module

IMPERIAL

- You should be able to:
  - Construct machine learning experiments
  - Create and run different probabilistic/statistical learning and deep learning models
  - Read and interpret existing work and assess their applicability/validity [to different use cases]
  - Be familiar with common concepts such as training/test, cross-validation, performance evaluation metrics
  - Learn common issues and pitfalls in developing ML models
  - Obtain practical experience in developing ML models and running (simple) experiments.

3

3

Data and representations

IMPERIAL

CCXVI divided by VI?

216/6?

ChatGPT
CCXVI divided by VI is equal to XXXIX, which is 39 in Roman numerals. Here's the breakdown:

CCXVI (216 in Roman numerals) divided by VI (6 in Roman numerals) equals XXXIX (39 in Roman numerals).

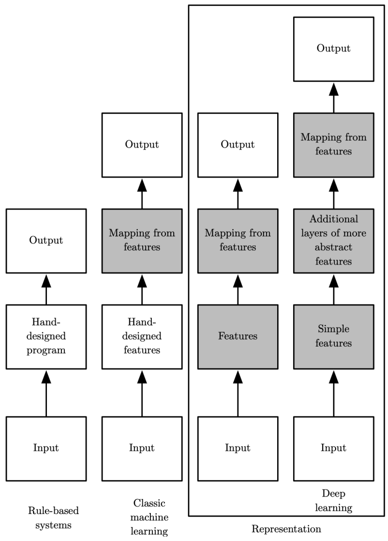
via: Goodfellow et al., Deep learning

4

4

Categories of learning and intelligent systems

IMPERIAL



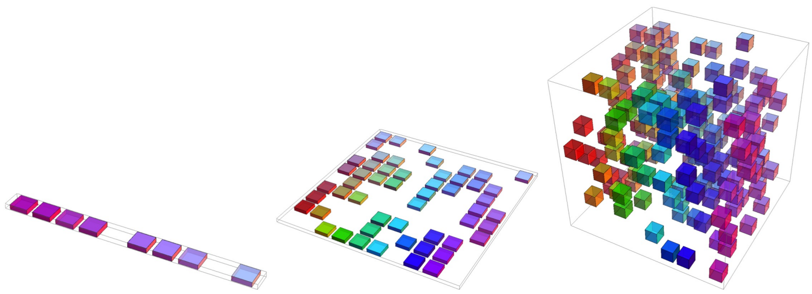
Source: Deep Learning, Ian Goodfellow et al, MIT press.

5

5

Curse of dimensionality

IMPERIAL



Overall, for d dimensions with v values we need $O(v^d)$ regions and examples

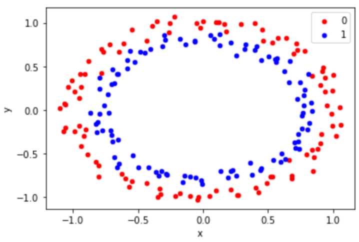
Source: Deep Learning, Ian Goodfellow et al, MIT press.

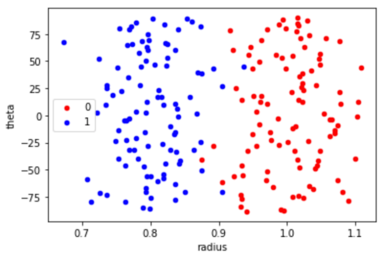
6

6

Data/sample representation

IMPERIAL





(x,y)

$$r=\sqrt{x^2+y^2}$$

$$\theta=\tan^{-1}\left(\frac{y}{x}\right).$$

<Code>

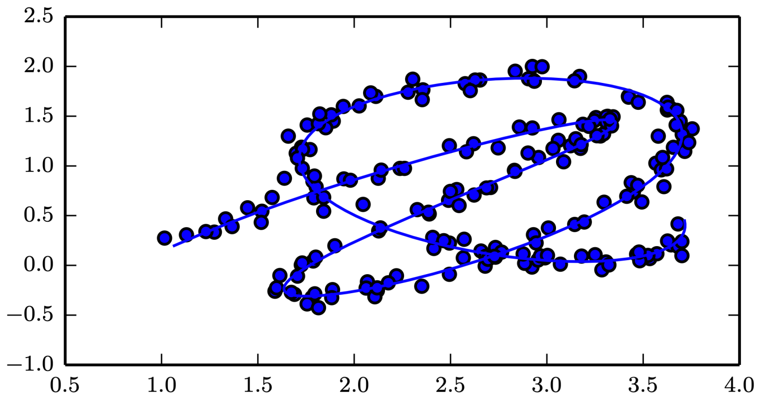
https://github.com/PBarnaghi/Basic-ML-Code/

7

7

Manifold learning

IMPERIAL



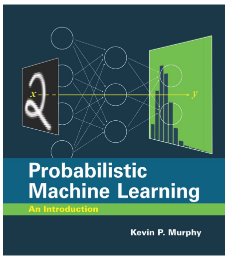
Source: Deep Learning, Ian Goodfellow et al, MIT press.

8

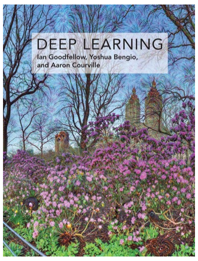
8

Reference and sources

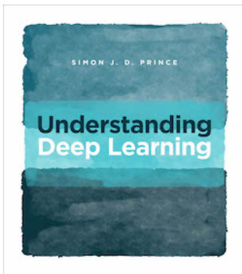
IMPERIAL



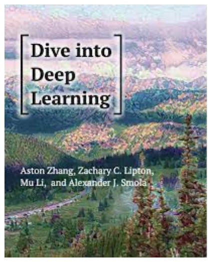
Probabilistic Machine Learning
Kevin P. Murphy, MIT Press



Deep Learning
Goodfellow et al., MIT Press



Understanding Deep Learning,
Simon J.D. Prince, MIT Press



Dive into Deep Learning
Aston Zhang et al.

Preprints:
Probabilistic Machine Learning: An Introduction, https://probml.github.io/pml-book/book1.html
Deep learning: https://www.deeplearningbook.org
Understanding Deep Learning, https://udlbook.github.io/udlbook/
Dive into deep learning, https://d2l.ai/index.html

9

9

Linear models - training

IMPERIAL

$$\alpha * Age + \beta * SleepQuality + \theta * CognitiveTestScore = Risk\_score$$

The question is how to learn the coefficients (weights in ML terms)?

Using training data.

10

10

How to evaluate your LR model

IMPERIAL

- Mean Absolute Error (MAE)
- Mean Squared Error (MSE)
- Root Mean Squared Error (RMSE)

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$$

11

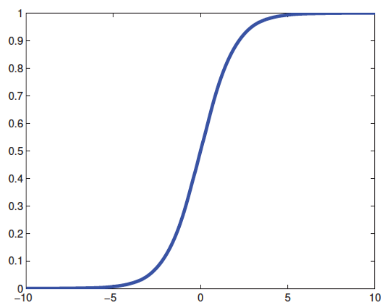
11

Logistic regression

IMPERIAL

- The process of applying a linear combination of the inputs, as before, but then we pass this through a logistics function, is called logistic regression due to its similarity to linear regression (although it is a form of classification, not regression!).

$$\text{sigm}(\eta) \triangleq \frac{1}{1+\exp(-\eta)} = \frac{e^\eta}{e^\eta + 1}$$

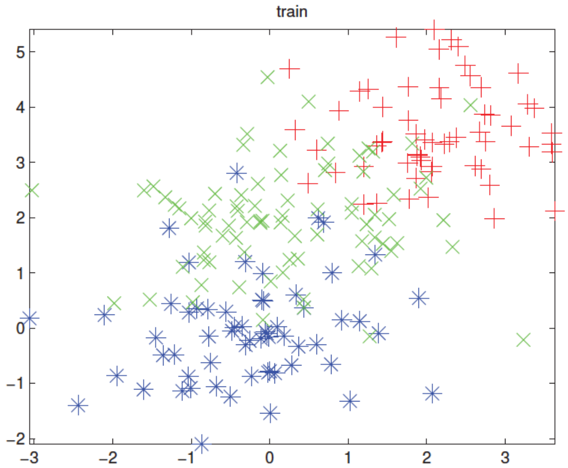


12

12

K-nearest neighbours - example

IMPERIAL



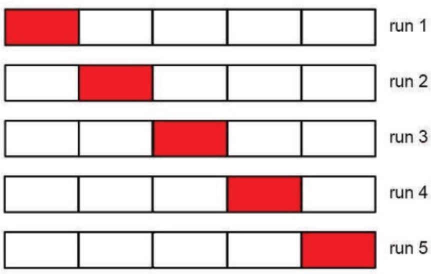
13

13

Cross validation

IMPERIAL

- The idea is simple: we split the training data into K folds; then, for each fold $k \in \{1, \ldots, K\}$, we train on all the folds but the k'th, and test on the k'th, in a round-robin fashion.



14

14

Cross validation - in Scikit learn

IMPERIAL

– https://scikit-learn.org/stable/modules/cross_validation.html#cross-validation

`sklearn.model_selection`: Model Selection

User guide: See the Cross-validation: evaluating estimator performance, Tuning the hyper-parameters of an estimator and Learning curve sections for further details.

Splitter Classes

`model_selection.GroupKFold`([n_splits])	K-fold iterator variant with non-overlapping groups.
`model_selection.GroupShuffleSplit`([...])	Shuffle-Group(s)-Out cross-validation iterator
`model_selection.KFold`([n_splits, shuffle, ...])	K-Folds cross-validator
`model_selection.LeaveOneGroupOut`()	Leave One Group Out cross-validator
`model_selection.LeavePGroupsOut`(n_groups)	Leave P Group(s) Out cross-validator
`model_selection.LeaveOneOut`()	Leave-One-Out cross-validator
`model_selection.LeavePOut`(p)	Leave-P-Out cross-validator
`model_selection.PredefinedSplit`(test_fold)	Predefined split cross-validator
`model_selection.RepeatedKFold`(*[, n_splits, ...])	Repeated K-Fold cross validator.
`model_selection.RepeatedStratifiedKFold`(*[, ...])	Repeated Stratified K-Fold cross validator.
`model_selection.ShuffleSplit`([n_splits, ...])	Random permutation cross-validator
`model_selection.StratifiedKFold`([n_splits, ...])	Stratified K-Folds cross-validator.
`model_selection.StratifiedShuffleSplit`([...])	Stratified ShuffleSplit cross-validator
`model_selection.StratifiedGroupKFold`([...])	Stratified K-Folds iterator variant with non-overlapping groups.
`model_selection.TimeSeriesSplit`([n_splits, ...])	Time Series cross-validator

Splitter Functions

`model_selection.check_cv`([cv, y, classifier])	Input checker utility for building a cross-validator.
`model_selection.train_test_split`(*arrays[, ...])	Split arrays or matrices into random train and test subsets.

15

15

L1 Regularisation (Lasso)*

IMPERIAL

– Lasso is an acronym for Least Absolute Shrinkage and Selection Operator.

– Lasso regression is a regression model that uses $\ell 1$ regularisation.

– In LASSO, we modify the optimisation function and add a coefficient, which is calculated based on the square of weights (parameters).

$$\sum_{i=1}^{n} (\hat{y}_i - y_i)^2 + \lambda \sum_{i=1}^{n} |\beta_i|$$

16

16

L2 regularisation (Ridge)*

- Ridge regression sum of squared weight values as the penalty term in the optimisation function.

$$\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 + \lambda \sum_{i=1}^{n} {\beta_i}^2$$

Lasso and Ridge

- The key difference between these two techniques is that lasso shrinks the less important feature's coefficient to zero, thus, removing some features altogether.
- In other words, $\ell 1$ regularisation works well for feature selection in case we have a huge number of features.

- Ridge reduces the complexity of the model by shrinking the coefficient (penalising higher weights).

Bayes Rule

IMPERIAL

Likelihood

Prior

Posterior

$$p(A|B) = \frac{p(B|A).\,p(A)}{p(B)}$$

Marginalisation

Naïve Bayes

IMPERIAL

- We call it Naïve because it assumes that all the features are independent. In practice, this is not always true – but this assumption allows us to apply the Bayesian theorem:

$$P(L \mid \text{features}) = \frac{P(\text{features} \mid L)P(L)}{P(\text{features})}$$

All in one place

IMPERIAL

$$Precision = \frac{TP}{TP + FP}$$

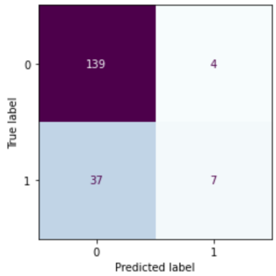
$$F_1 = 2 \times \frac{Precision \times Recall}{Precision\ + Recall}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Sensetivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

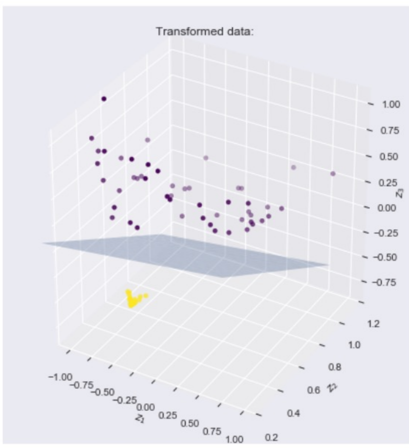


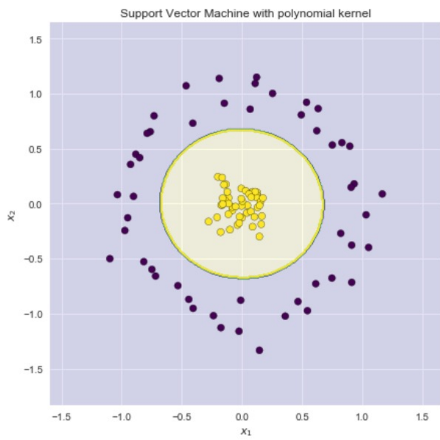
21

21

Example – applying polynomial kernel

IMPERIAL





Source: https://xavierbourretsicotte.github.io/Kernel_feature_map.html

22

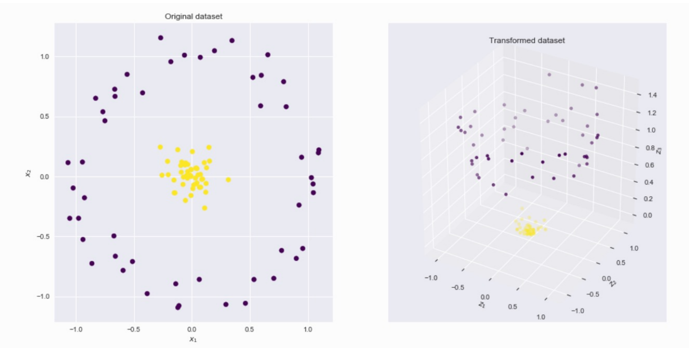
22

A polynomial kernel

IMPERIAL

$$\phi(x_1, x_2) = (z_1, z_2, z_3) = (x_1, x_2, x_1^2 + x_2^2)$$

```
def feature_map_1(X):
    return np.asarray((X[:,0], X[:,1], X[:,0]**2 + X[:,1]**2)).T
```



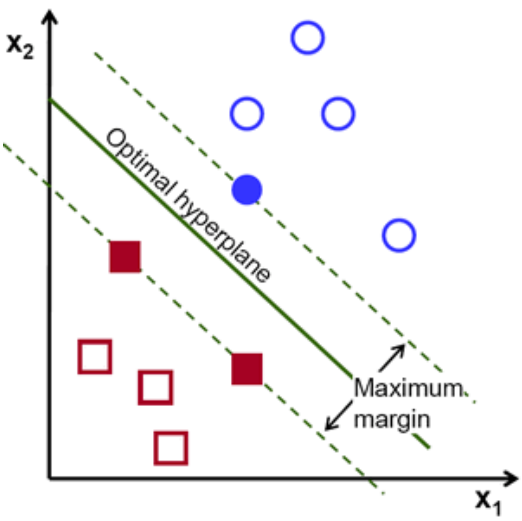
Source: https://xavierbourretsicotte.github.io/Kernel_feature_map.html

23

23

Example: SVM for a linearly separable set of 2d-points

IMPERIAL



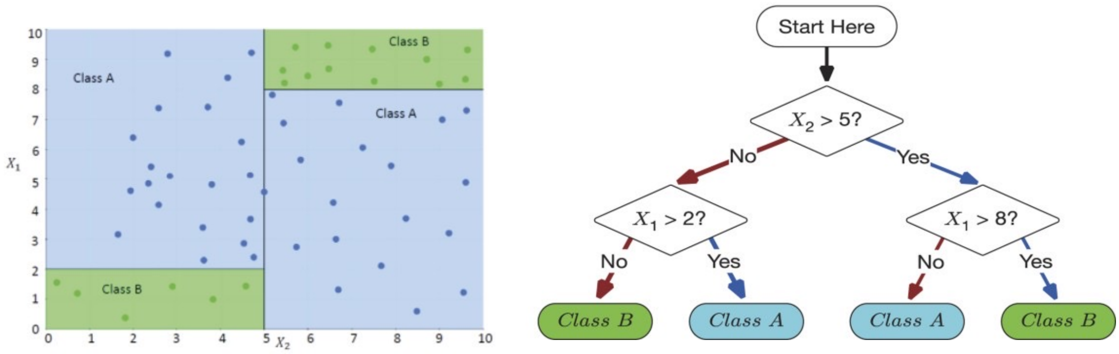
Source: https://vovkos.github.io/doxyrest-showcase/opencv/sphinx_rtd_theme/page_tutorial_introduction_to_svm.html

24

24

Decision Making Using DTs

IMPERIAL



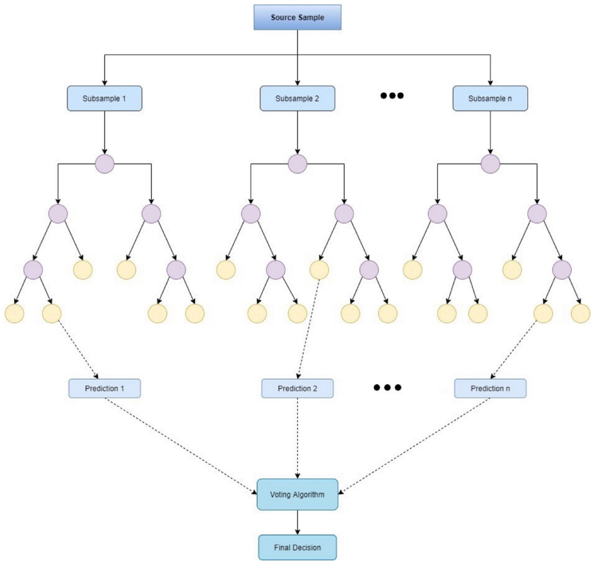
Source: Valdes, G., Luna, J., Eaton, E. *et al.* MediBoost: a Patient Stratification Tool for Interpretable Decision Making in the Era of Precision Medicine. *Sci Rep* 6, 37854 (2016). https://doi.org/10.1038/srep37854

25

25

Structure of Random Forests

IMPERIAL



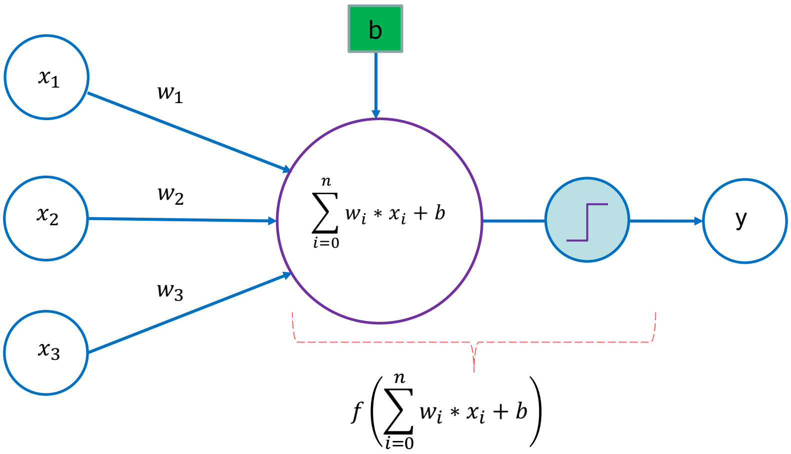
Source: Malebary, S.J., Khan, Y.D. Evaluating machine learning methodologies for identification of cancer driver genes. *Sci Rep* **11**, 12281 (2021). https://doi.org/10.1038/s41598-021-91656-8

26

26

NN with an activation function

IMPERIAL



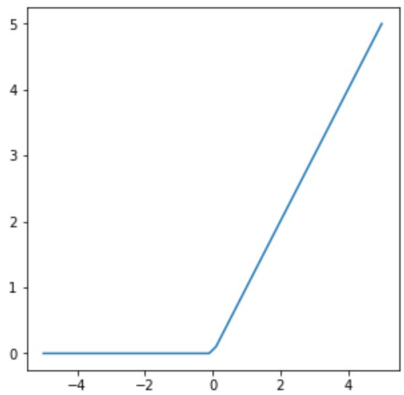
27

27

Rectified Linear Units (ReLU)

IMPERIAL

```
def relu(z):
    return np.maximum(0,z)
```



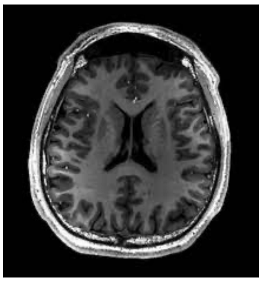
28

28

Images a vectors

IMPERIAL

- Using images by *flattening* the images means overlooking the spatial relation between pixels.
- This deeply unsatisfying approach could be a very simple solution in order to feed the resulting one-dimensional vectors through a fully connected MLP or other probabilistic models.



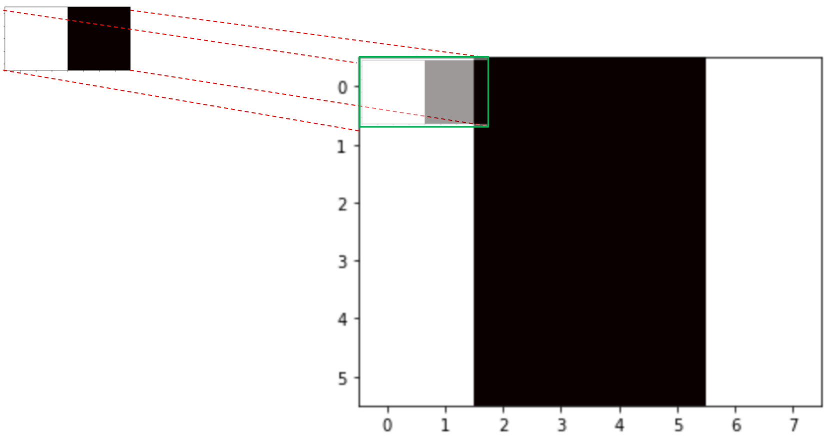
29

29

Example – edge detection: kernel

IMPERIAL

```
K = torch.tensor([[1.0, -1.0]])
```



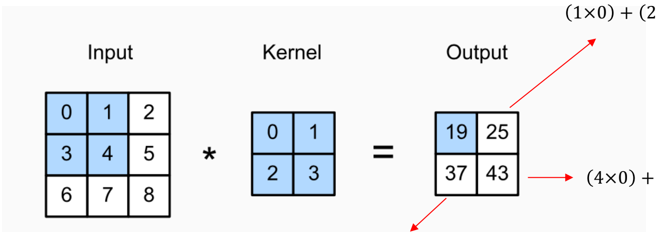
30

30

CNN kernels

IMPERIAL

$(1\times0) + (2\times1) + (4\times2) + (5\times3) = 25$



$(4\times0) + (5\times1) + (7\times2) + (8\times3) = 43$

$(3\times0) + (4\times1) + (6\times2) + (7\times3) = 37$

Two-dimensional cross-correlation operation. The shaded portions are the first output element as well as the input and kernel tensor elements used for the output computation:

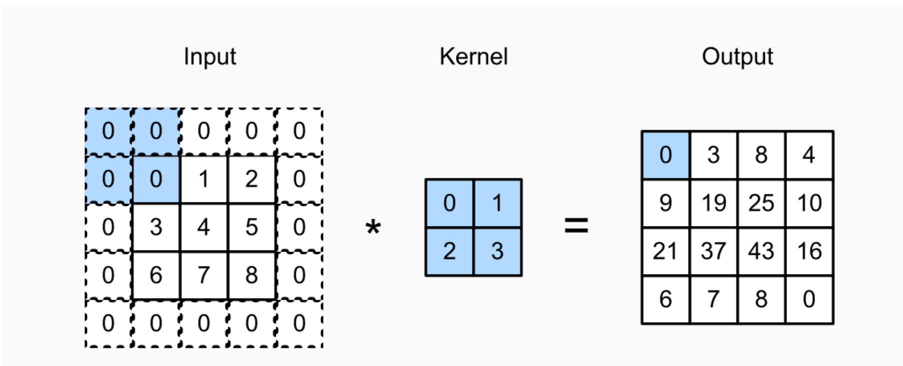
$(0\times0) + (1\times1) + (3\times2) + (4\times3) = 19.$

31

31

Padding: example

IMPERIAL



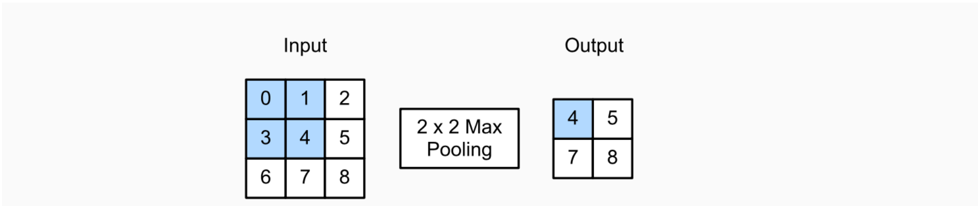
32

32

Pooling

IMPERIAL

- Like convolutional layers, *pooling* operators consist of a fixed-shape window that is slid over all regions in the input according to its stride, computing a single output for each location traversed by the fixed-shape window (sometimes known as the *pooling window*).

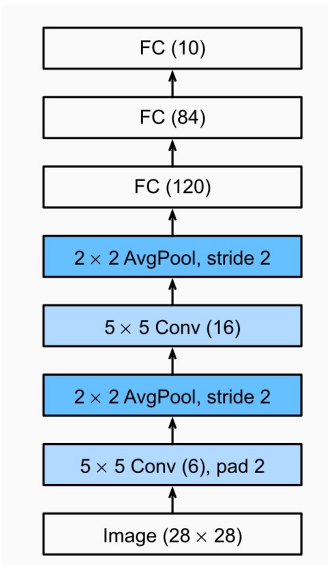


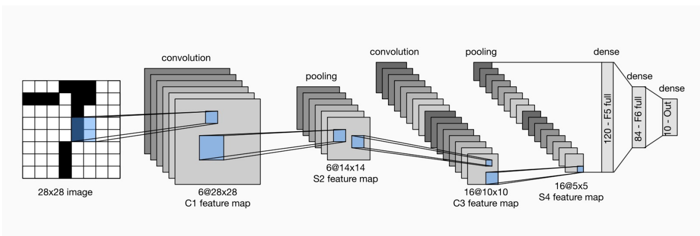
33

33

Compressed notion for LeNet-5

IMPERIAL



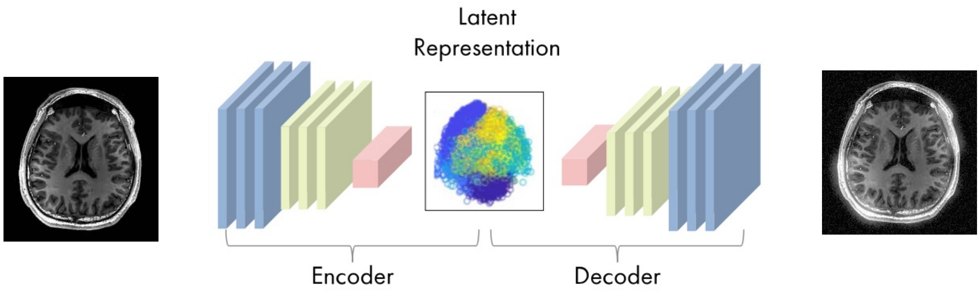


34

34

Autoencoder

IMPERIAL



Source for the middle image: https://uk.mathworks.com/discovery/autoencoder.html

35

35

ML deployments in real-world settings

IMPERIAL

- The systems are often used for decision support and have humans in the loop.
- However, the trustworthiness, reliability and robustness of the systems/models must be considered and investigated prior to the deployments.
- The users' perceptions of the system and appropriate training should be also considered.

36

36

Responsible and ethical machine learning

IMPERIAL

- Anticipate risks and potential issues
- Actively engage and think of solutions
- Work with domain experts and learn about the context and conditions of data collection, and the use of system/model.
- Create feedback loops (before and after deployment)
- Plan for update and maintenance

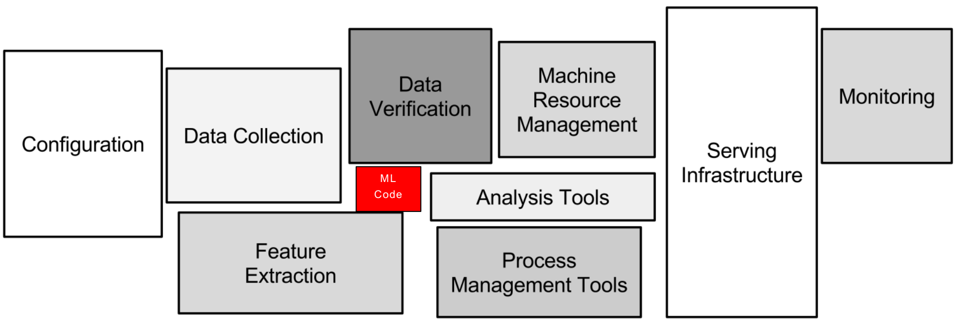
37

37

Hidden technical debt in real-world ML systems

IMPERIAL

- It is important to note that in several applications, only a small fraction of real-world ML systems comprise the ML code.



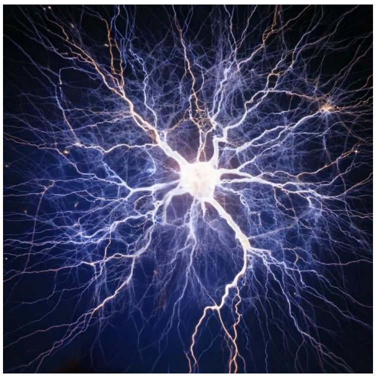
Source: Sculley et al., Hidden Technical Debt in Machine Learning Systems, NeurIPS, 2015.

38

38

IMPERIAL

Applications



39

39

IMPERIAL

Neuroscience inspired machine learning

nature communications

Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature > nature communications > articles > article

Article | Open access | Published: 22 January 2025

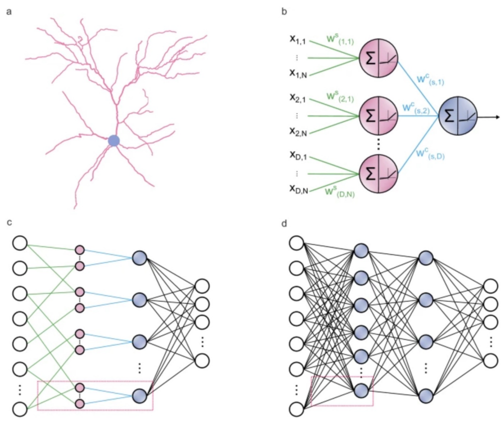
Dendrites endow artificial neural networks with accurate, robust and parameter-efficient learning

Spyridon Chavlis & Panayiota Poirazi

Nature Communications **16**, Article number: 943 (2025) | Cite this article

13 Altmetric | Metrics

Fig. 1: Schematic representation of the dendritic ANN (dANN) compared to a classical vanilla ANN (vANN).



Chavlis, S., Poirazi, P. Dendrites endow artificial neural networks with accurate, robust and parameter-efficient learning. *Nat Commun* **16**, 943 (2025). https://doi.org/10.1038/s41467-025-56297-9

40

40

Neuroscience inspired machine learning

IMPERIAL

"…build a Hodgkin–Huxley (HH) network with rich internal complexity, where each neuron is an HH model, and prove that the dynamical properties and performance of the HH network can be equivalent to a bigger leaky integrate-and-fire (LIF) network, where each neuron is a LIF neuron with simple internal complexity."


nature computational science

Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature > nature computational science > articles > article

Article | Published: 16 August 2024

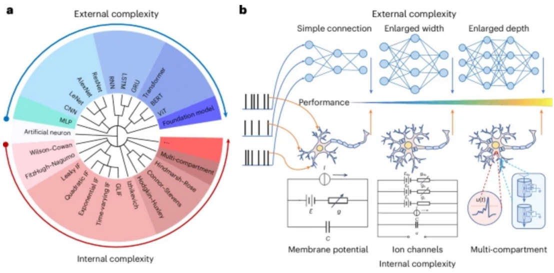
Network model with internal complexity bridges artificial intelligence and neuroscience

Linxuan He, Yunhui Xu, Weihua He, Yihan Lin, Yang Tian, Yujie Wu, Wenhui Wang, Ziyang Zhang, Junwei Han, Yonghong Tian, Bo Xu & Guoqi Li

Nature Computational Science 4, 584–599 (2024) | Cite this article

4907 Accesses | 2 Citations | 147 Altmetric | Metrics

Fig. 1: The internal and external complexity of neurons and networks.



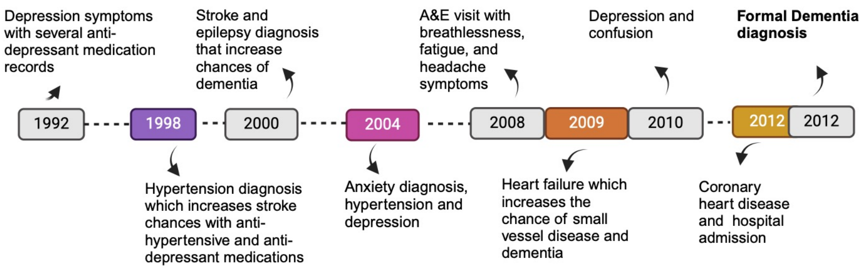
He, L., Xu, Y., He, W. *et al.* Network model with internal complexity bridges artificial intelligence and neuroscience. *Nat Comput Sci* **4**, 584–599 (2024). https://doi.org/10.1038/s43588-024-00674-9

41

41

Machine learning for neurodegenerative conditions

IMPERIAL



42

42

ML in Parkinson's Disease

IMPERIAL

nature medicine

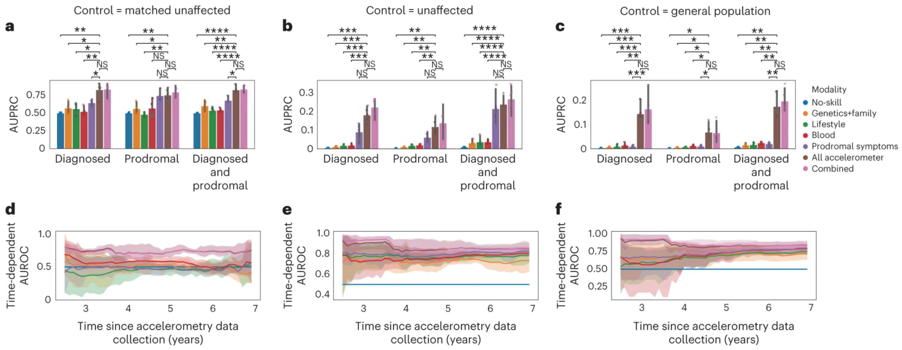
Explore content ⌄ About the journal ⌄ Publish with us ⌄

nature > nature medicine > articles > article

Article | Published: 03 July 2023

Wearable movement-tracking data identify Parkinson's disease years before clinical diagnosis

Ann-Kathrin Schalkamp, Kathryn J. Peall, Neil A. Harrison & Cynthia Sandor



chalkamp, AK., Peall, K.J., Harrison, N.A. *et al.* Wearable movement-tracking data identify Parkinson's disease years before clinical diagnosis. *Nat Med* **29**, 2048–2056 (2023). https://doi.org/10.1038/s41591-023-02440-2

43

43

ML in Dementia Care

IMPERIAL

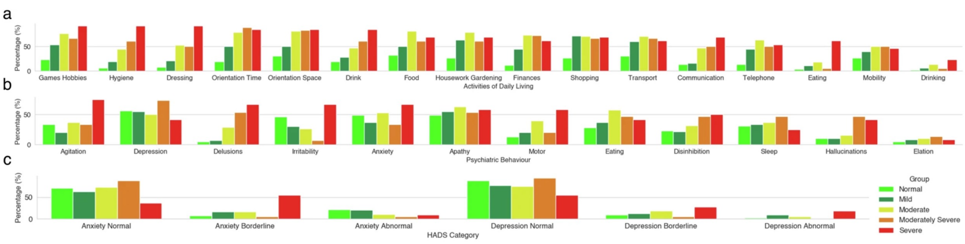
nature > communications medicine > articles > article

Article | Open access | Published: 10 January 2025

Longitudinal study of care needs and behavioural changes in people living with dementia using in-home assessment data

Chloe Walsh, Alexander Capstick, Nan Fletcher-Lloyd, Jessica True, CR&T Group, Ramin Nilforooshan & Payam Barnaghi

Communications Medicine **5**, Article number: 14 (2025) | Cite this article



Walsh, C., Capstick, A., Fletcher-Lloyd, N. *et al.* Longitudinal study of care needs and behavioural changes in people living with dementia using in-home assessment data. *Commun Med* **5**, 14 (2025). https://doi.org/10.1038/s43856-024-00724-3

44

44

IMPERIAL

ML in comorbidities associated with neurodegenerative conditions

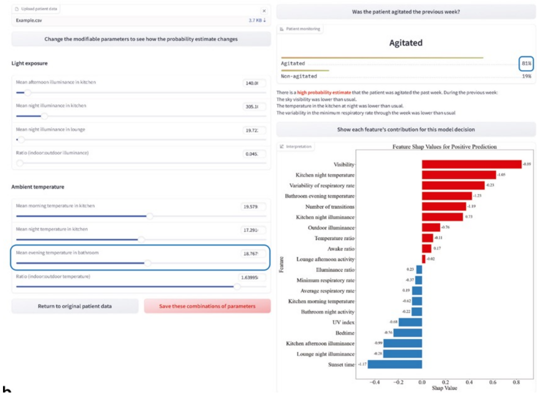
eClinicalMedicine
Part of THE LANCET *Discovery Science*

This journal · Journals · Publish · Clinical · Global health · Multimedia · Events · About

ARTICLES · Volume 80, 103032, February 2025 · *Open Access* · Download Full Issue

An interpretable machine learning tool for in-home monitoring of agitation episodes in people living with dementia: a proof-of-concept study

Marirena Bafaloukou [a,b,c] · Ann-Kathrin Schalkamp [a,b,c] · Nan Fletcher-Lloyd [a,b] · Alex Capstick [a,b] · Chloe Walsh [a,b,d] · Cynthia Sandor [a,c] · Samaneh Kouchaki [a,b,e] · CR&T Group [a,b] · Ramin Nilforooshan [a,b,d,e] · Payam Barnaghi [a,b,f] · Show less



Bafaloukou, Marirena *et al.* , An interpretable machine learning tool for in-home monitoring of agitation episodes in people living with dementia: a proof-of-concept study, eClinicalMedicine, Volume 80, 103032

45

45

IMPERIAL

Review Questions

46

46

Q1

IMPERIAL

- What machine learning libraries you have worked with before?

47

47

Q2

IMPERIAL

- What is the difference between L1 and L2 norms for regularisation and when you use them.

48

48

Q3

IMPERIAL

- If you have a dataset with an imbalanced number of samples, how do you deal with this problem?

49

49

Q4

IMPERIAL

- If we give you a set of data (imagine imaging data or electronic healthcare records) and ask you to design a machine learning model. Where and how would you start designing your model?

50

50

Q5

IMPERIAL

- If you are given a dataset, how do you decide which model would be the most efficient model to solve your problem?

51

51

Q6

IMPERIAL

- Suppose you are given a dataset to cluster. How do you decide what number of clusters would be the most suitable for the dataset?

52

52

Q7

IMPERIAL

- How would you ensure the model you designed is not overfitted?

53

53

If you have any questions

IMPERIAL

- Please feel free to arrange a meeting or email (p.barnaghi@imperial.ac.uk).
- My office: 928, Sir Michael Uren Research Hub, White City Campus.

54

54