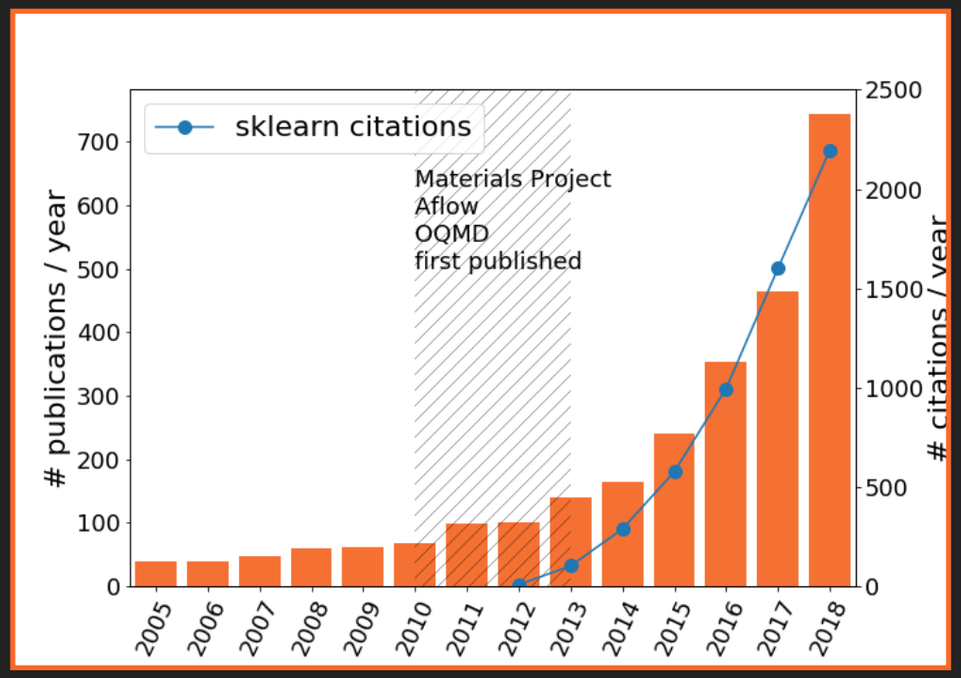
KEITH T. BUTLER

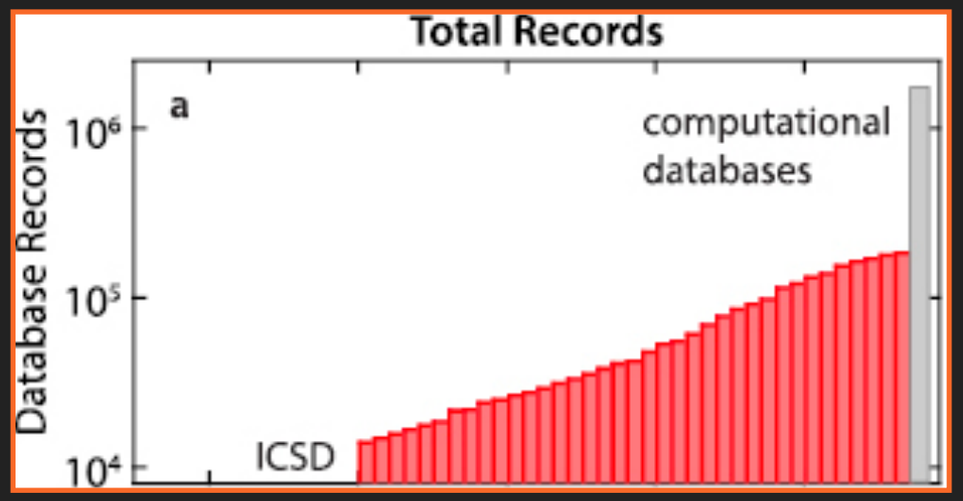
A QUICK START GUIDE TO MACHINE LEARNING

SCIML/STFC/RAL

- Scientific machine learning group
- Interact with Facilities/Academia/Industry

THE RISE OF MACHINE LEARNING





J. Phys. D: Appl. Phys. 52 013001

"What made deep learning take off was big data. ... The explosion of data is having an influence not just on science and engineering but also on every area of society."
Terry Sejnowski - Deep Learning Revolution

ACADEMIA / INDUSTRY IN AI

- These companies all have many, very large, private datasets that they will never make publicly available
- Each of these companies employs many hundreds of computer scientists with PhDs in Machine Learning and AI
- Their researchers and developers have essentially unlimited computing power at their disposal

SCIML/STFC/RAL

- National facilities are data rich
- Eg single time-resolved tomographic experiment = 100 TB

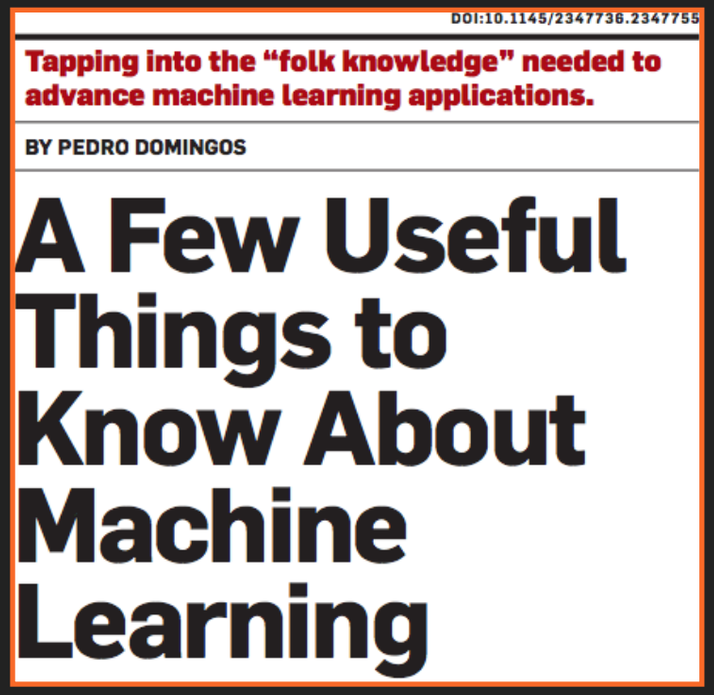
5 KEY QUESTIONS BEFORE GOING ML

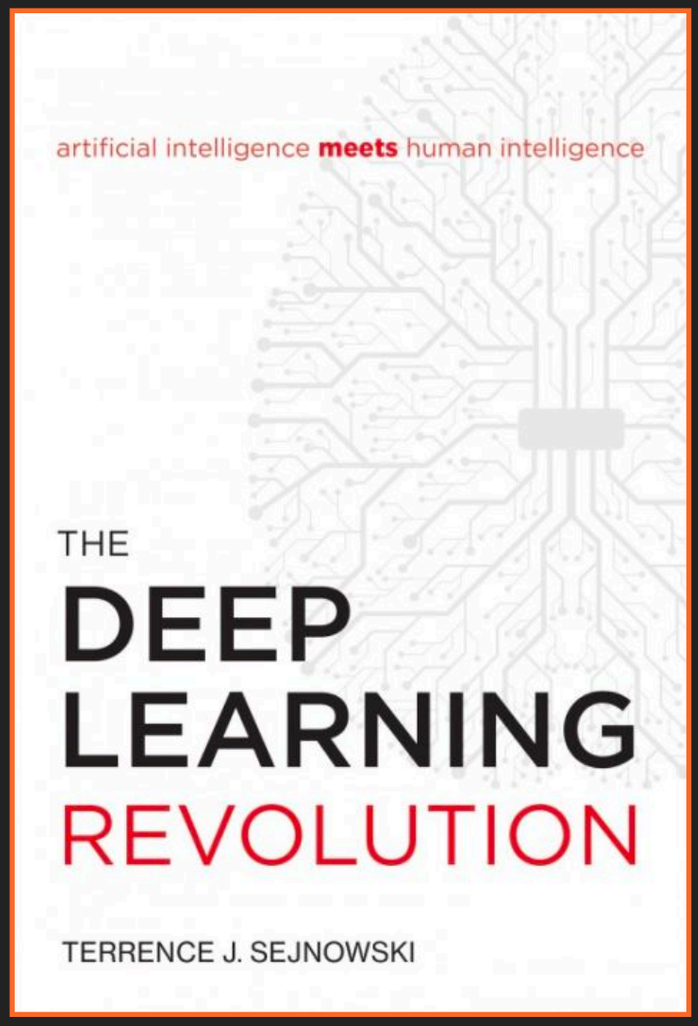
- What do I want to achieve?
- How much data do I have/can I get?
- What kind of data do I have?
- Do I care more about prediction or inference?
- What kind of hardware do I have?

TUTORIAL OVERVIEW

- An introduction to machine learning
  - Background definitions
  - Some traditional ML approaches
- Deep networks for materials science
  - CNNs for images and spectra
  - LSTMs for time series

SOME RECOMMENDED READING





"A Few Useful Things to Know About Machine Learning" by Pedro Domingos
“The Deep Learning Revolution" by Terry Sejnowski

INTRODUCTION TO MACHINE LEARNING

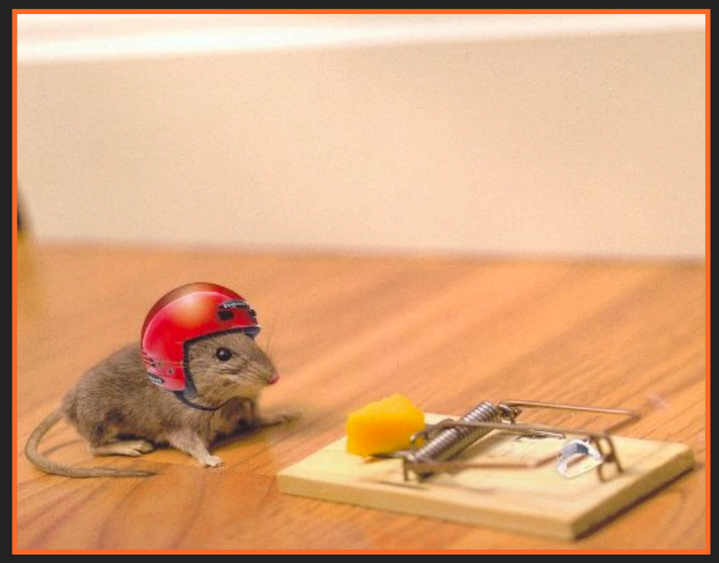
- Some definitions
  - Machine learning
- Some major issues
  - Generalisation
  - Representation
- Some popular algorithms (except neural nets!)
  - Regression
  - Bayes
  - Decision trees

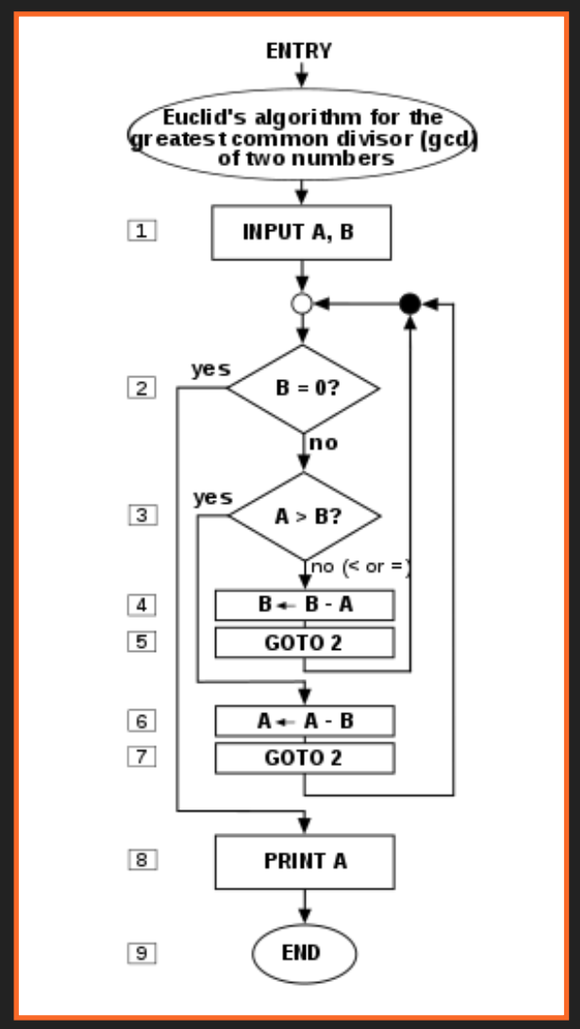
WHAT IS MACHINE LEARNING

ANIMALS LEARN FROM EXPERIENCE

MACHINES LEARN FROM ~~EXPERIENCE~~ DATA

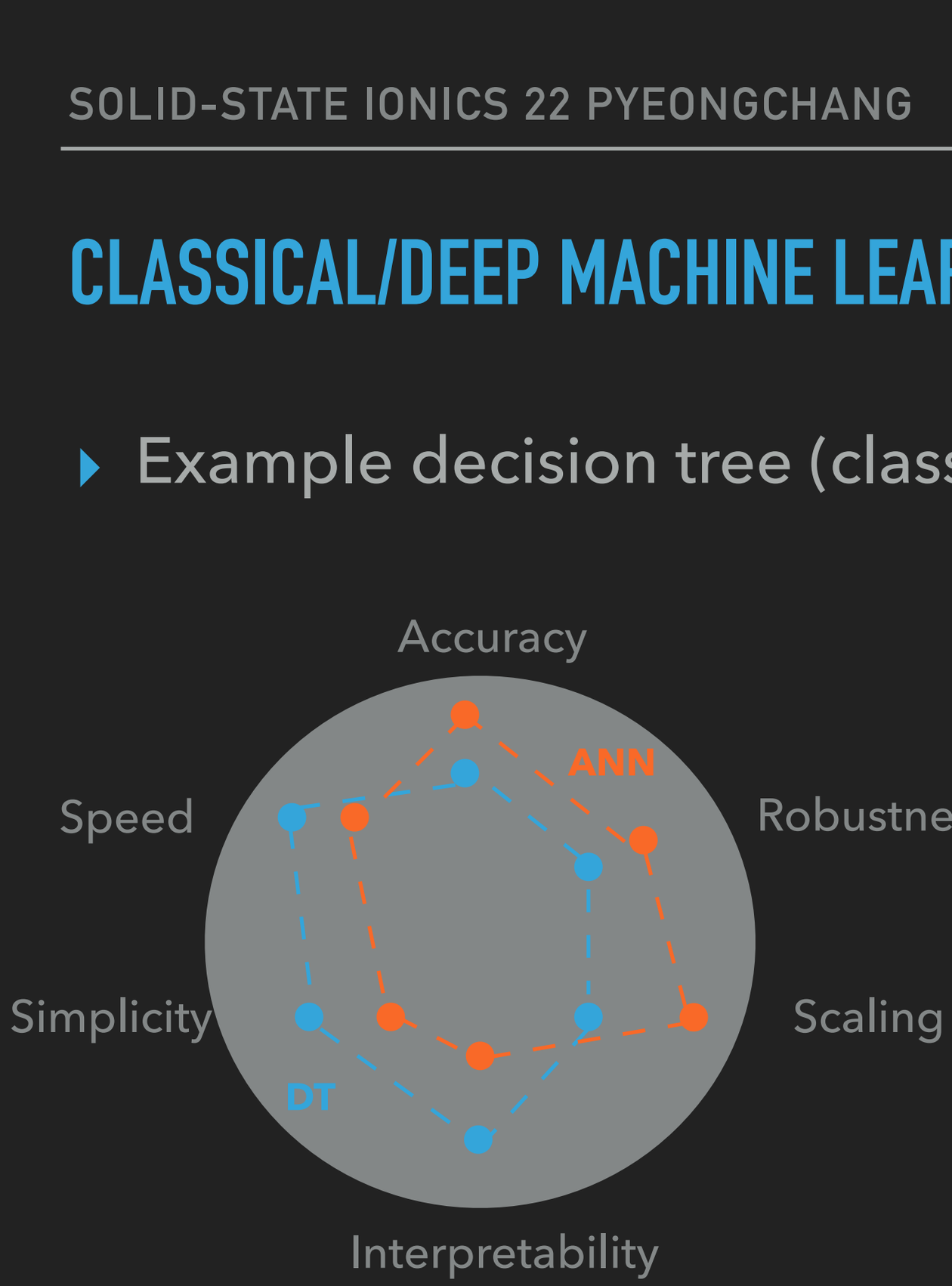
MACHINES ARE PROGRAMMED

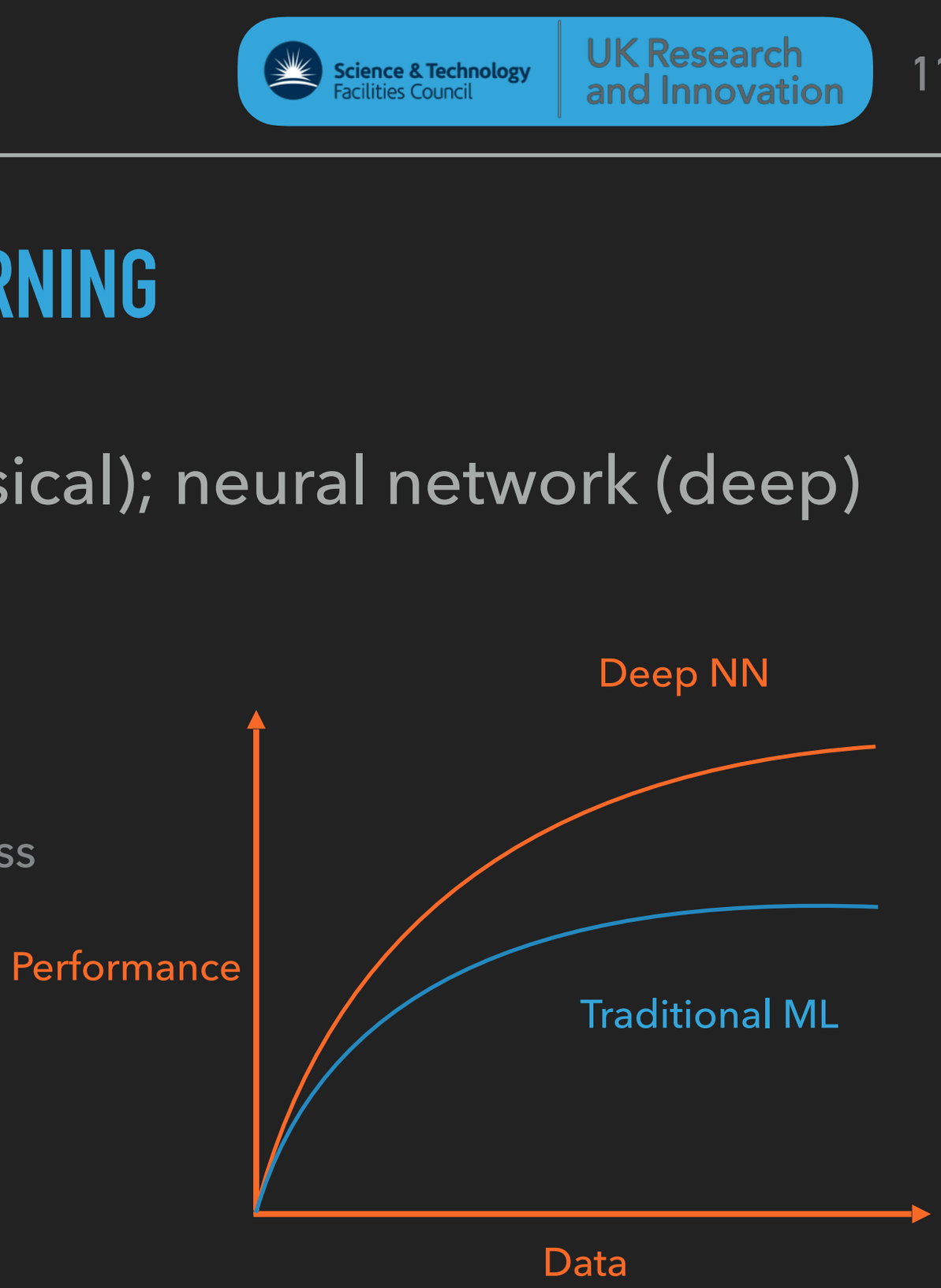




CLASSICAL/DEEP MACHINE LEARNING

- Example decision tree (classical); neural network (deep)





SUPERVISED/UNSUPERVISED

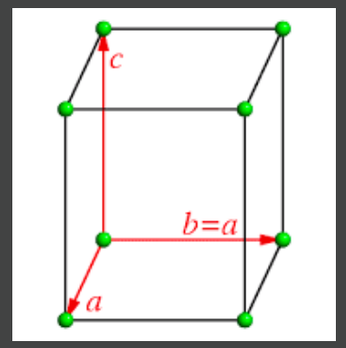
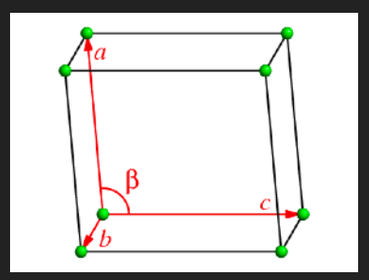
- Supervised learning
  - Labelled training data
    - Learn relations between input and label
- Unsupervised learning
  - Unlabelled training data
  - Learn relations between data points

CLASSIFICATION/REGRESSION

- Classification
  - Separate data points
  - Finite number of discreet classes
- Regression
  - (Usually) a single value
  - Infinite continuous variable

ONE-HOT ENCODING

- Classification problems
- Vector of length = number of categories
- Each element is the probability that the data represents a given class

Material	Ortho	Rhomb
	1	0
	0	1

PARAMETERS AND HYPER-PARAMETERS

- Parameters are part of the model
  - E.g. y = Bx + C; B and C are parameters
  - You do not set parameters
- Hyper-parameters control the learning process
  - E.g. number of parameters allowed
  - Type of optimiser
  - Learning rate
  - You do set hyper parameters

LEARNING

Learning =

Representation + Evaluation + Optimisation

- Representation
  - How we represent the knowledge.
  - This also chooses the set of possible classifiers.
  - Hypothesis space.
  - Eg. Neural network, decision tree …

LEARNING

Learning =

Representation + Evaluation + Optimisation

- Evaluation
  - Objective function or scoring function.
  - Distinguish good from bad classifiers.
  - NB need not be the same as the external function that the classifier is optimising.

LEARNING

Learning =

Representation + Evaluation + Optimisation

- Optimisation
  - Searches between classifiers.
  - Identifies the highest-scoring one.
  - Determines the efficiency of a learner.

LEARNING

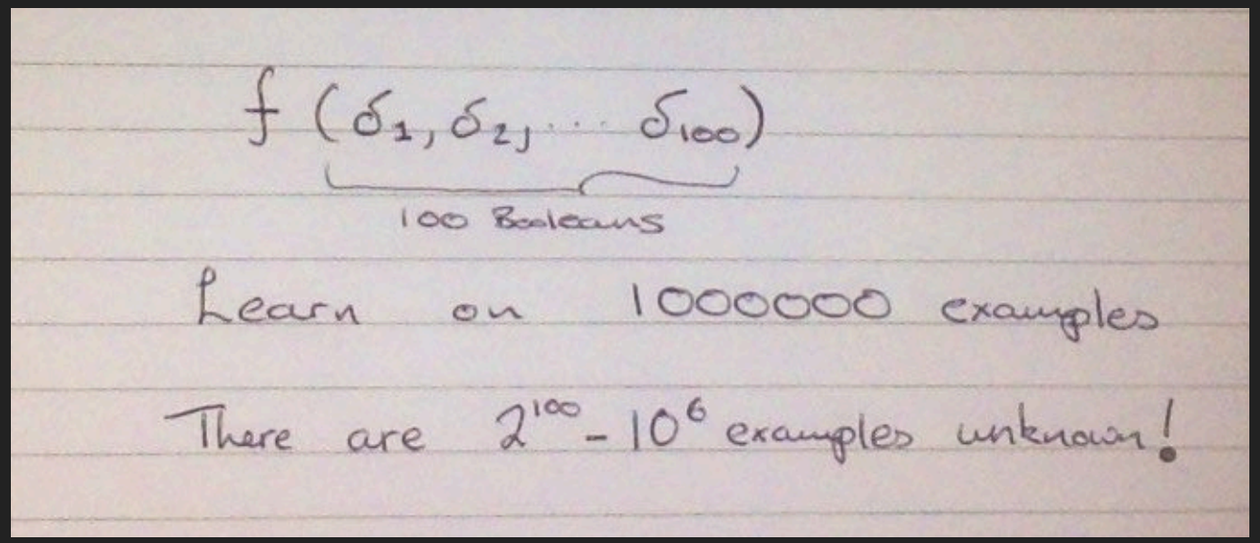
Learning =

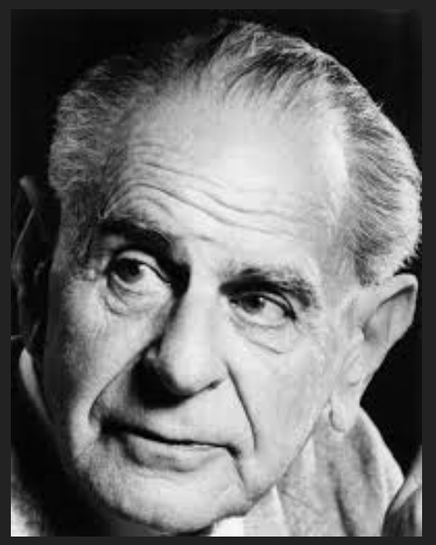
Representation + Evaluation + Optimisation

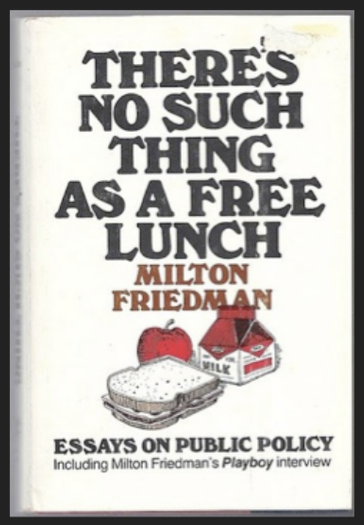
- Representation
  - How we represent the knowledge.
  - This also chooses the set of possible classifiers.
  - Hypothesis space.
  - Eg. Neural network, decision tree …

INDUCTION/REPRESENTATION

- Data alone is not enough – we need knowledge.

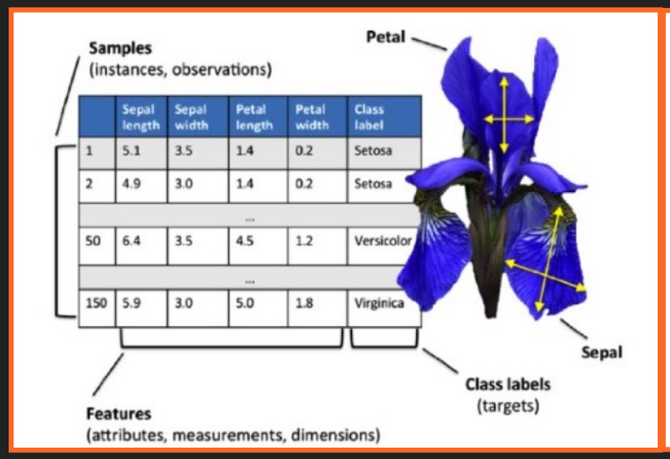




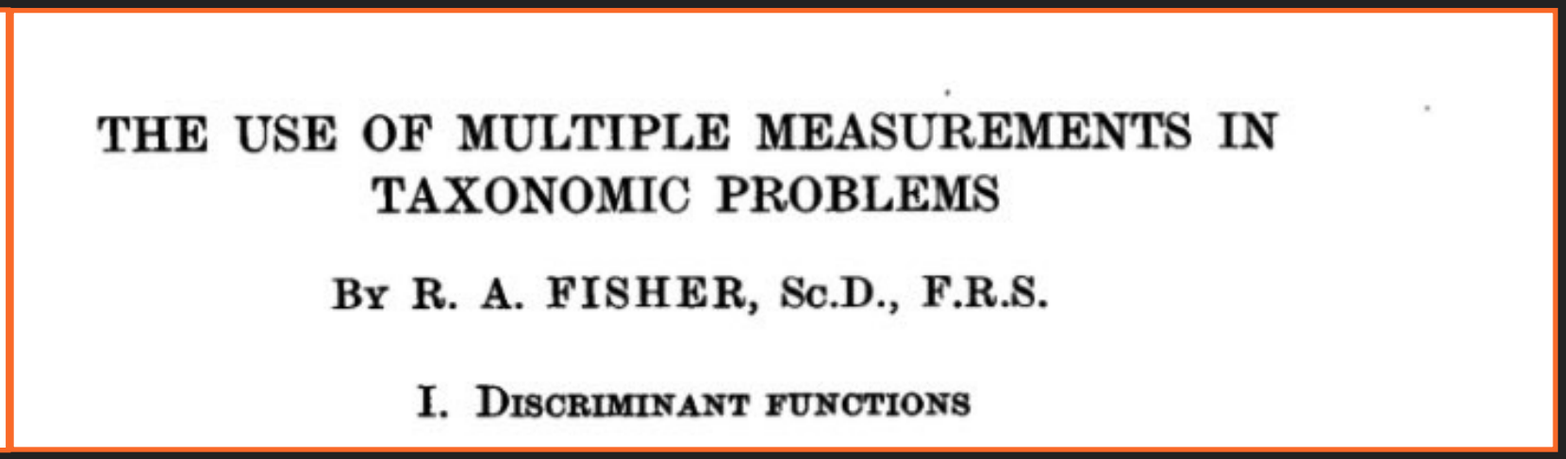


REPRESENTATION

- Data to knowledge
- THE biggest challenge - the most active
- Both data ***and*** model are forms of representation
- Requires domain + AI knowledge



	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica



THE USE OF MULTIPLE MEASUREMENTS IN TAXONOMIC PROBLEMS

By R. A. FISHER, Sc.D., F.R.S.

I. Discriminant functions

LEARNING TABLE

Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search
Hyperplanes	Likelihood	Branch-and-bound
Naive Bayes	Posterior probability	Continuous optimization
Logistic regression	Information gain	Unconstrained
Decision trees	K-L divergence	Gradient descent
Sets of rules	Cost/Utility	Conjugate gradient
Propositional rules	Margin	Quasi-Newton methods
Logic programs		Constrained
Neural networks		Linear programming
Graphical models		Quadratic programming
Bayesian networks		
Conditional random fields		

"A Few Useful Things to Know About Machine Learning" by Pedro Domingos

REPRESENTATIONS

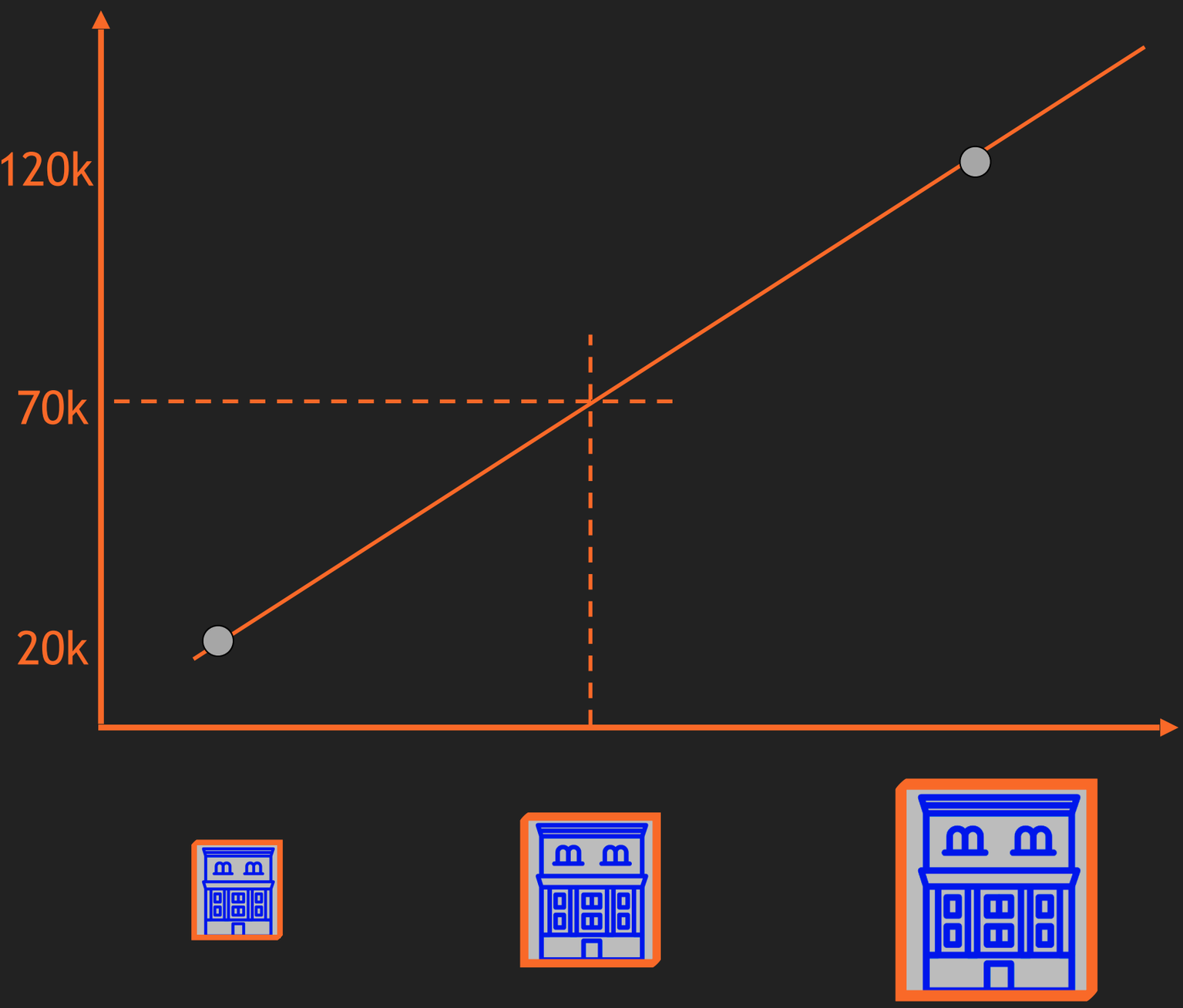
- Classical machine learning
  - Regression
  - Decision trees
  - K-nearest neighbour
- Deep learning
  - Neural networks
  - Convolutional neural networks etc…

LINEAR REGRESSION

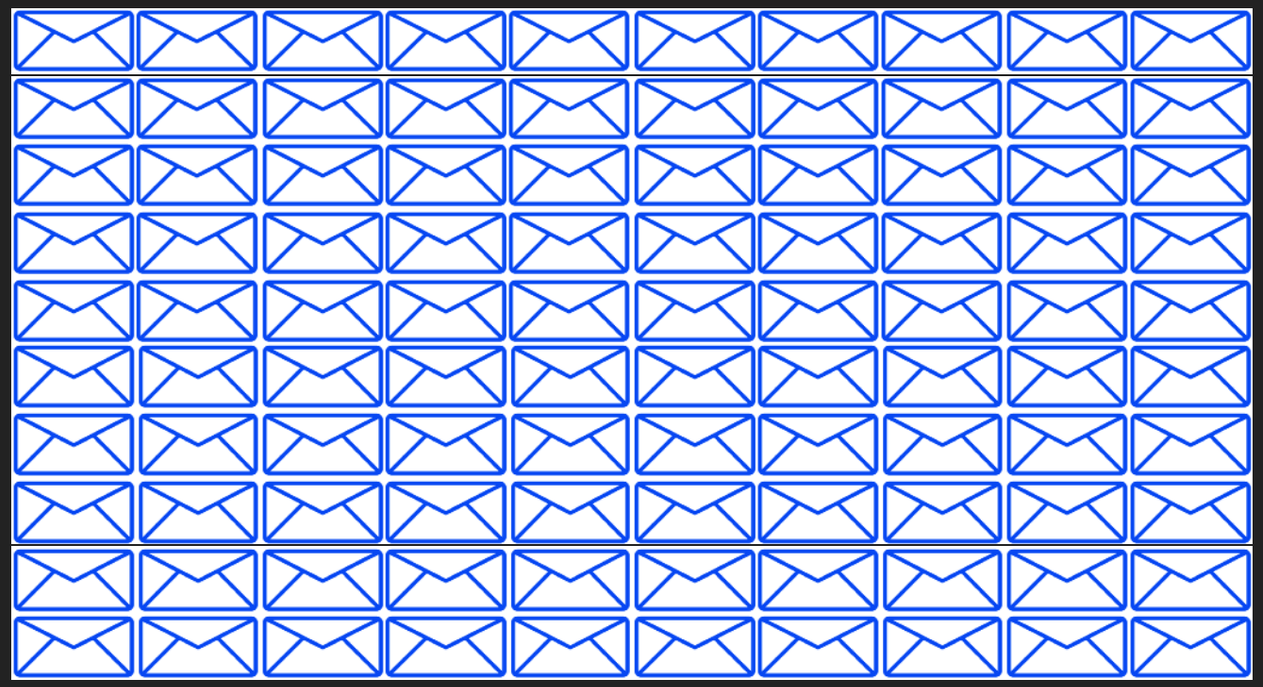
£20,000 ? £120,000

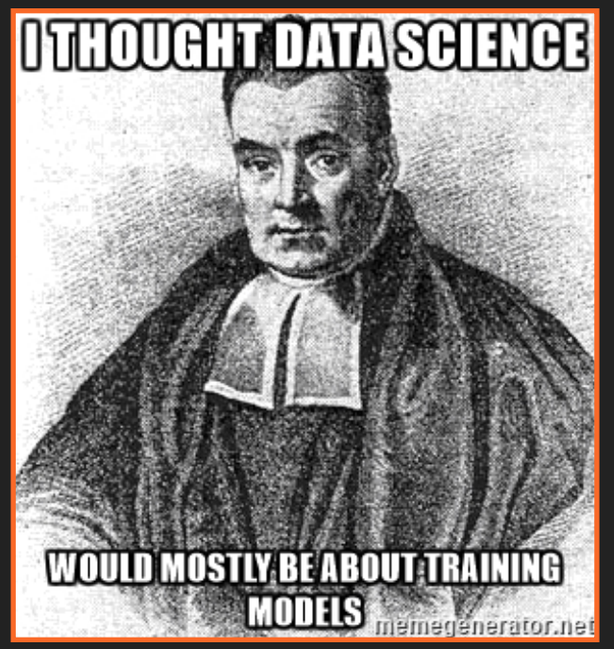
https://www.autodraw.com/

LINEAR REGRESSION



NAIVE BAYES

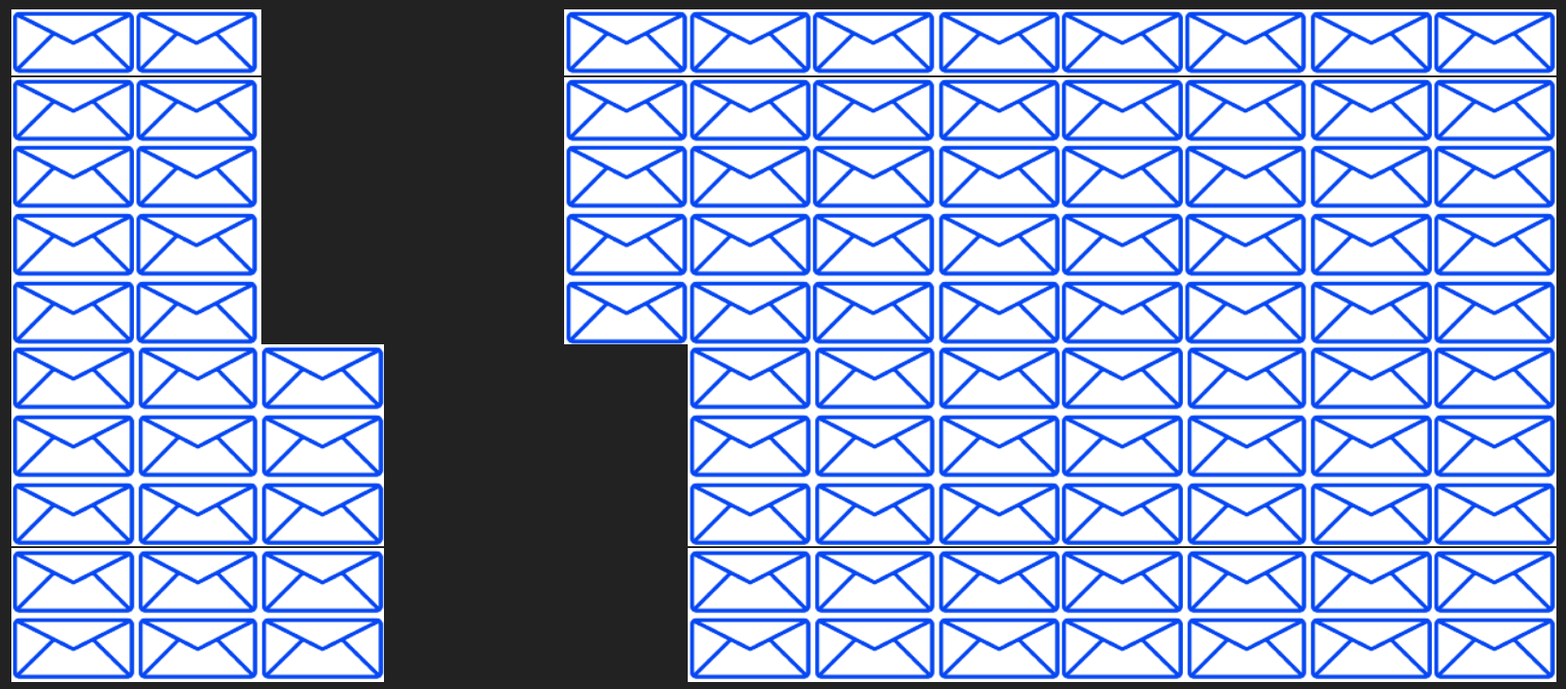



I THOUGHT DATA SCIENCE
WOULD MOSTLY BE ABOUT TRAINING MODELS
memegenerator.net

NAIVE BAYES

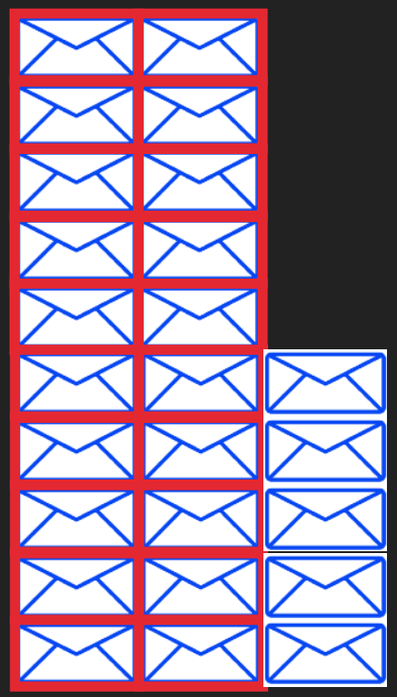
Spam

Not spam

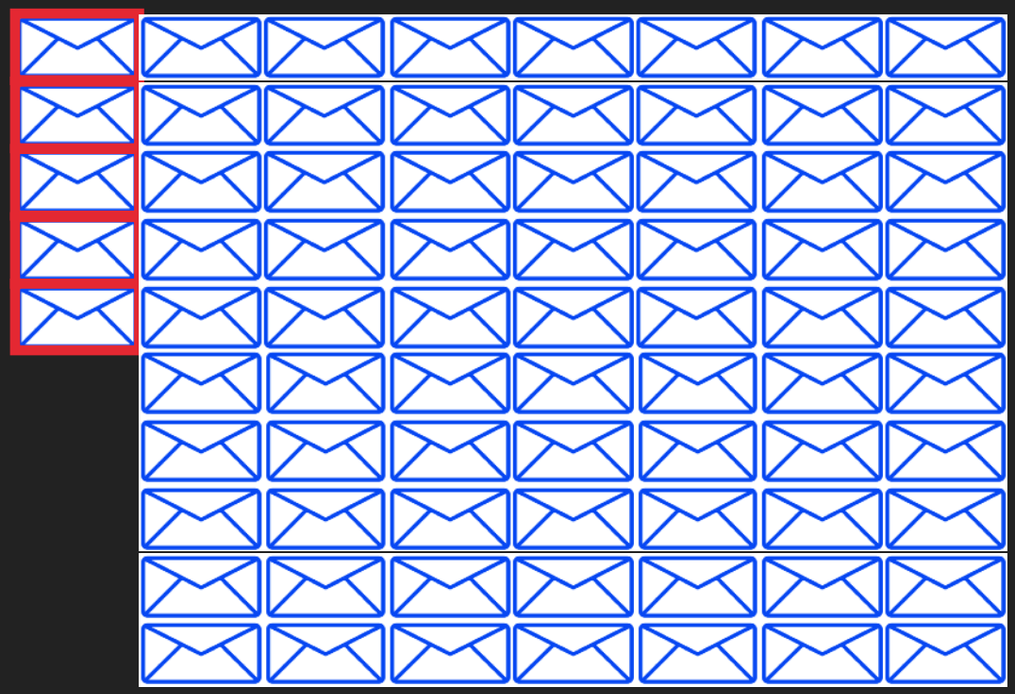


NAIVE BAYES

Spam



Not spam



Cheap

80% chance that an email with the word ‘cheap’ will be spam

NAIVE BAYES

CHEAP PREGNENCY TEST

Cheap → 80 %

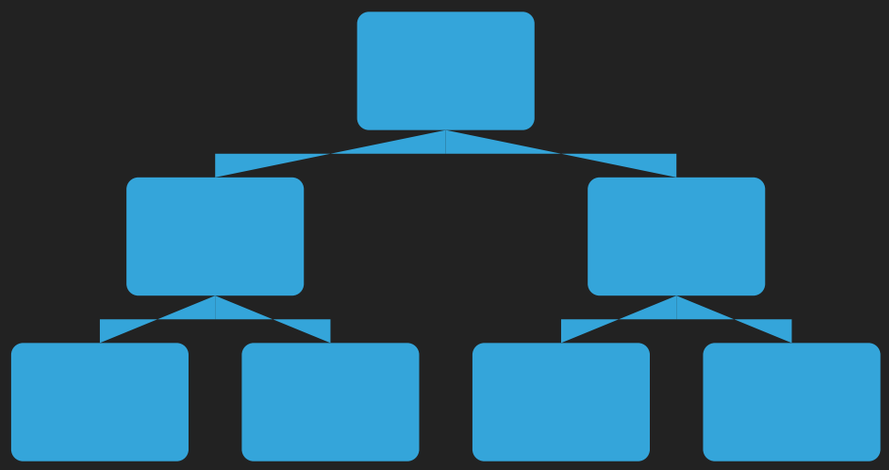
Typos → 60 %

Caps title → 90 %

99.2 %

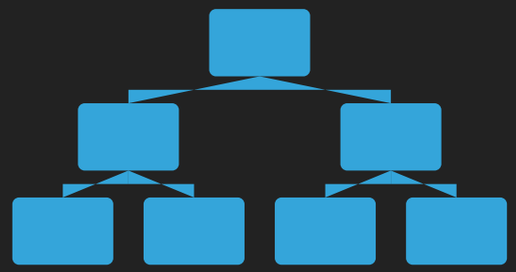
DECISION TREES

- Has a band gap Y/N



At each node, the space is split such that samples with similar labels are grouped together

DECISION TREES



For a trial split θ at node j…

The impurity of j is calculated for a trial split using an impurity function H()…

And θ are chosen in a greedy fashion…

$$Q_{left}(\theta) = (x, y)|x_f \leq t_j$$
$$Q_{right}(\theta) = Q \setminus Q_{left}(\theta)$$

$$C(Q, \theta) = \frac{n_{left}}{N_j} H(Q_{left}(\theta)) + \frac{n_{right}}{N_j} H(Q_{right}(\theta))$$

$$\theta^* = \underset{\theta}{\operatorname{argmin}} C(Q, \theta)$$

ENSEMBLE LEARNER

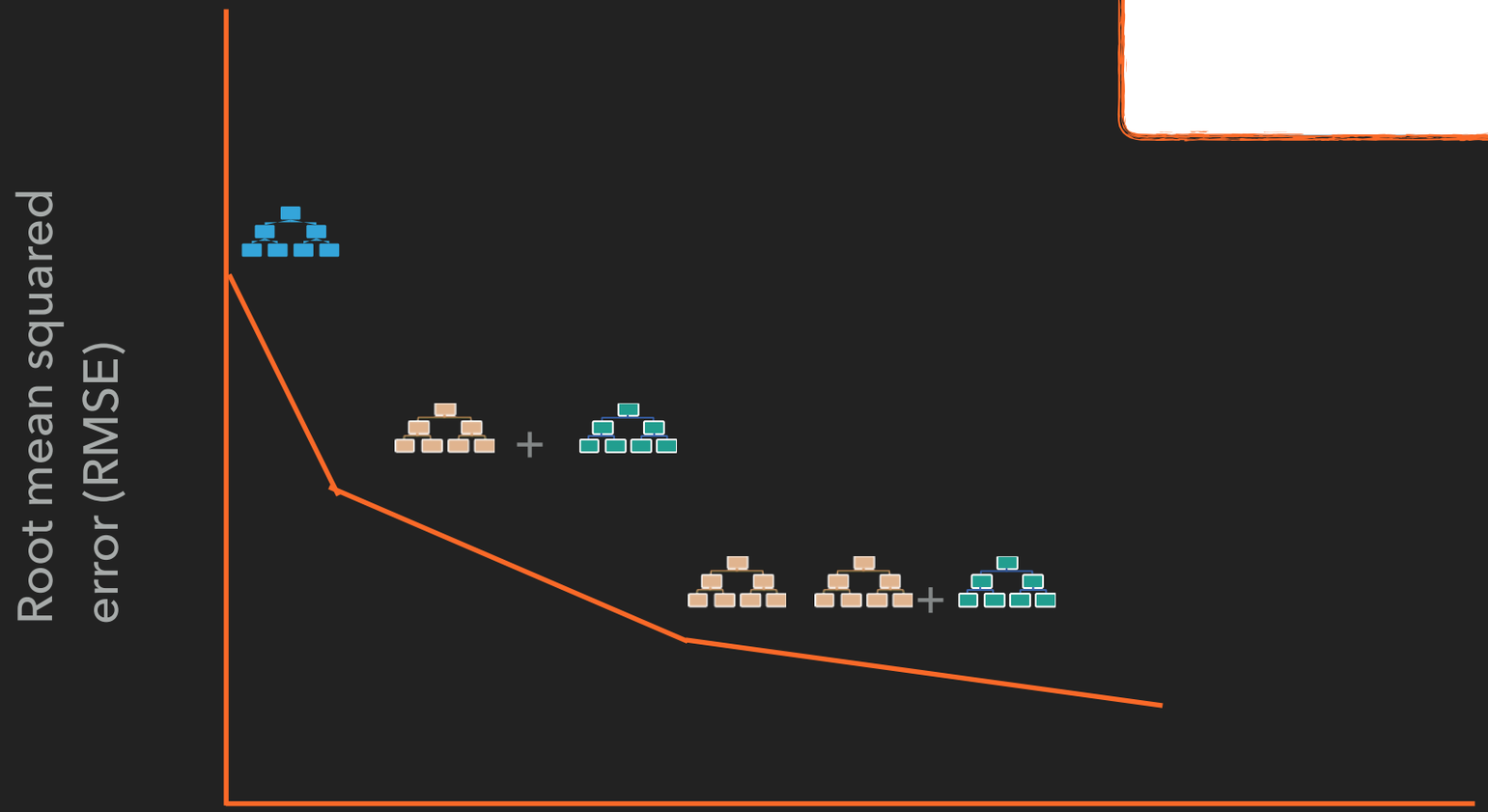
- Decision trees are weak learners
- A group of trees can overcome limitations
- Can optimise the group or chose randomly
  - Random forest
  - Gradient boosted

EXAMPLE: GRADIENT BOOSTED

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$$

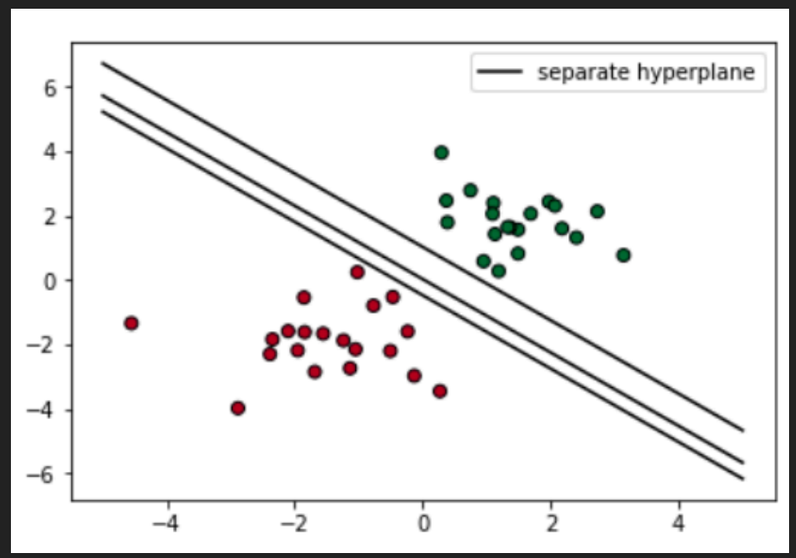
Learning rate

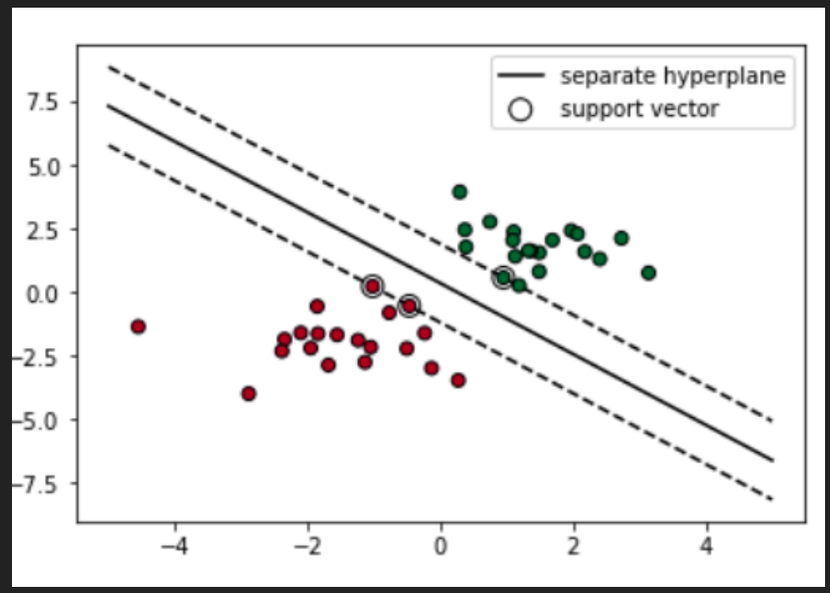
Sub model of errors



SUPPORT VECTOR MACHINES (CLASSIFICATION)

- SVMs seek to separate classes of observation
- Additional constraint of maximum margins
- Use a hyper-plane (a plane with one dimension less than the feature space)

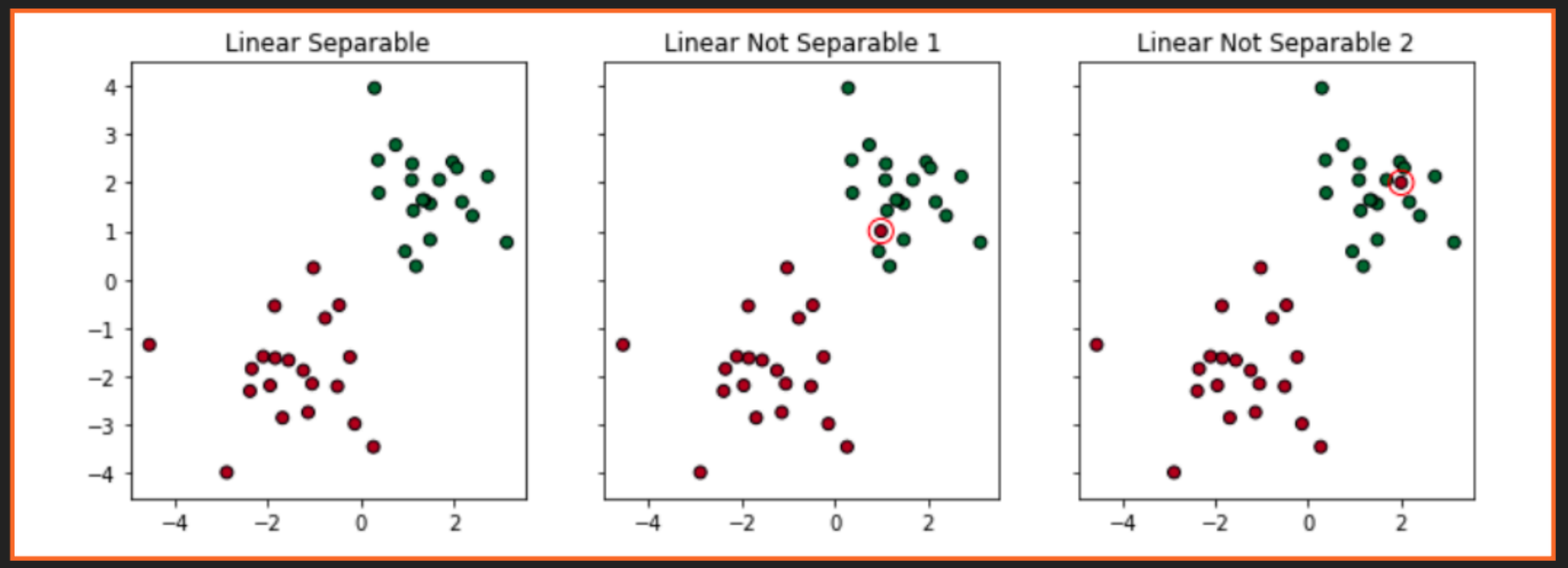




https://towardsdatascience.com/support-vector-machine-simply-explained-fee28eba5496

SVMS IN NON-LINEAR SEPARATIONS

- Classes not linearly separable in the feature space
- Soft margins
- Kernel trick

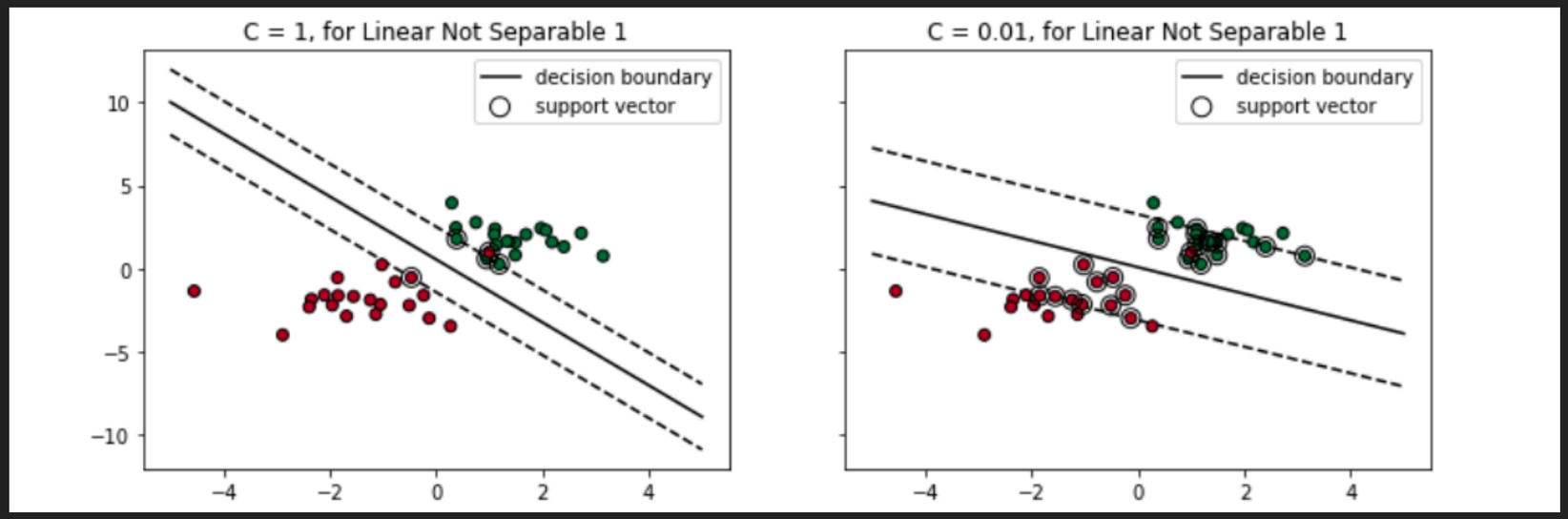


https://towardsdatascience.com/support-vector-machine-simply-explained-fee28eba5496

SVMS WITH SOFT MARGINS

- Tolerate a certain number of mis-classifications to maximise the margin
- Trade-off between mis-classification and margin width
  - Tolerance hyper-parameter determines the balance

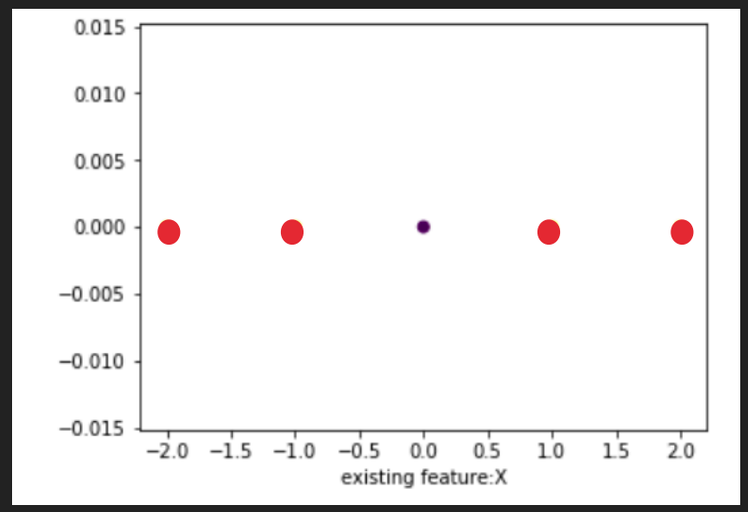
Classification is more important than margin



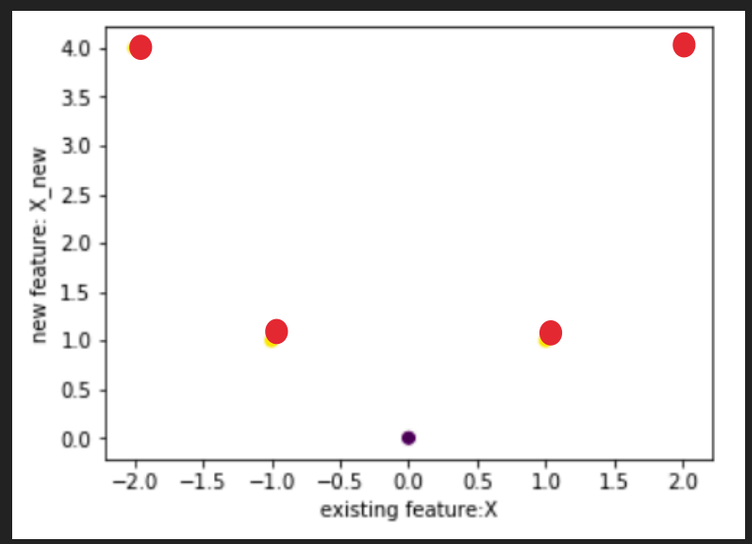
Margin is more important than classification

SVMS WITH THE KERNEL TRICK

- Combine and manipulate existing parameters to create new parameters
- Move the objects to a new dimensional space
- See if the classes are linearly separable in the new space

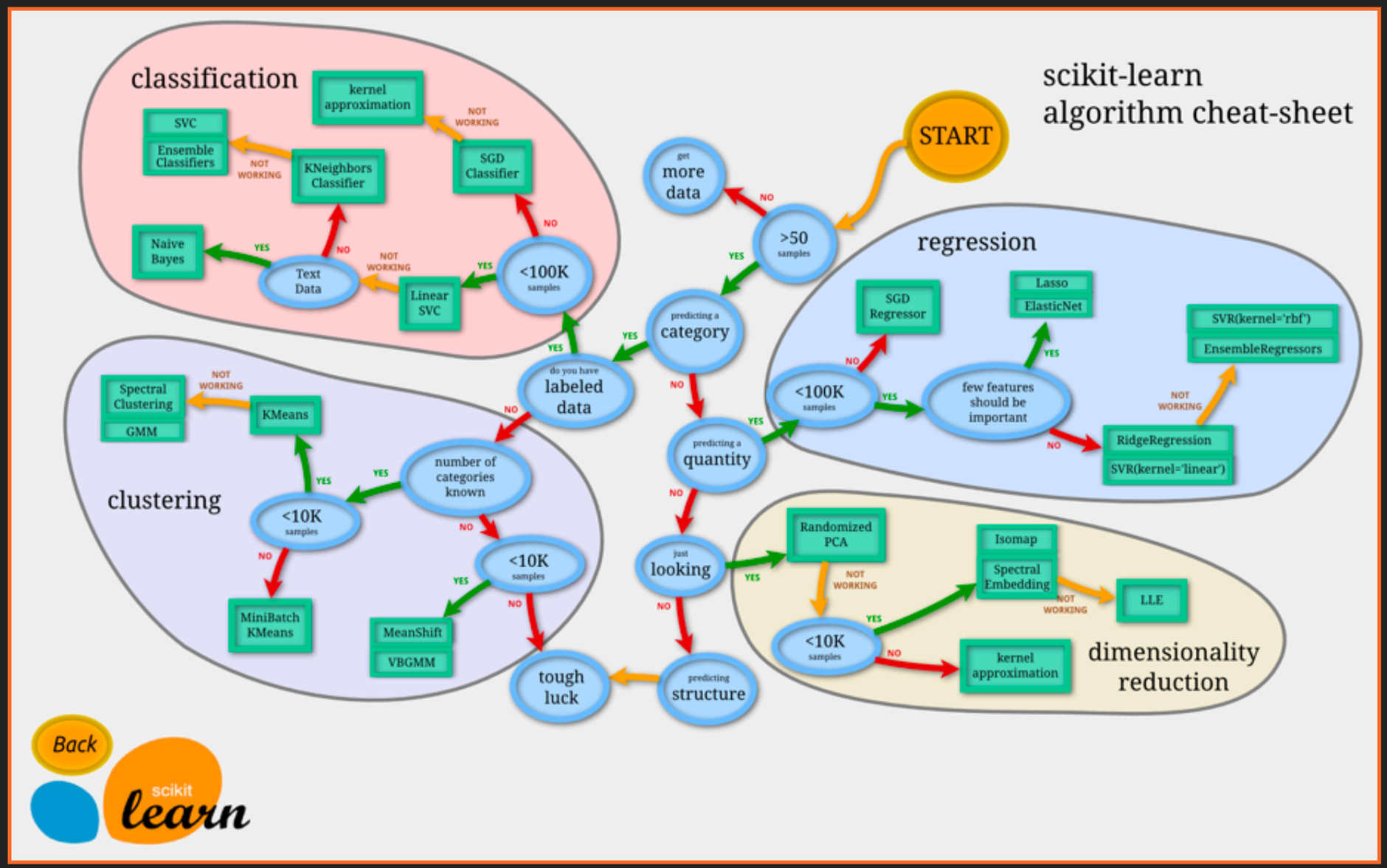


Not separable in standard space



Apply polynomial kernel => separable

CLASSICAL MODELS SUMMARY



NEURAL NETWORKS

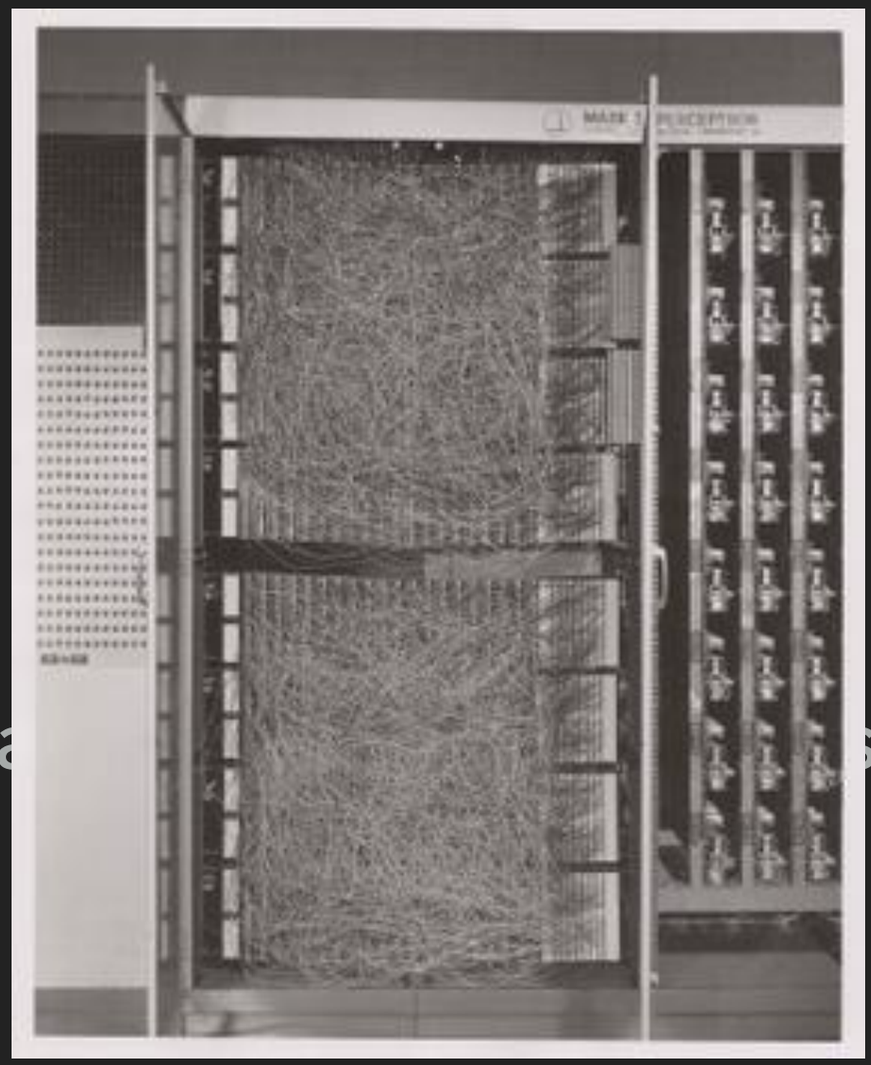
- A history of neural nets
  - Rise-Fall-Rise-Fall-Rise-?
- The elements of a network
  - Neurons, connections, optimisers
- Modern networks: CNNs
  - Image recognition, feature detection etc

THE PERCEPTRON

- Originally a device
- Intended for binary classification

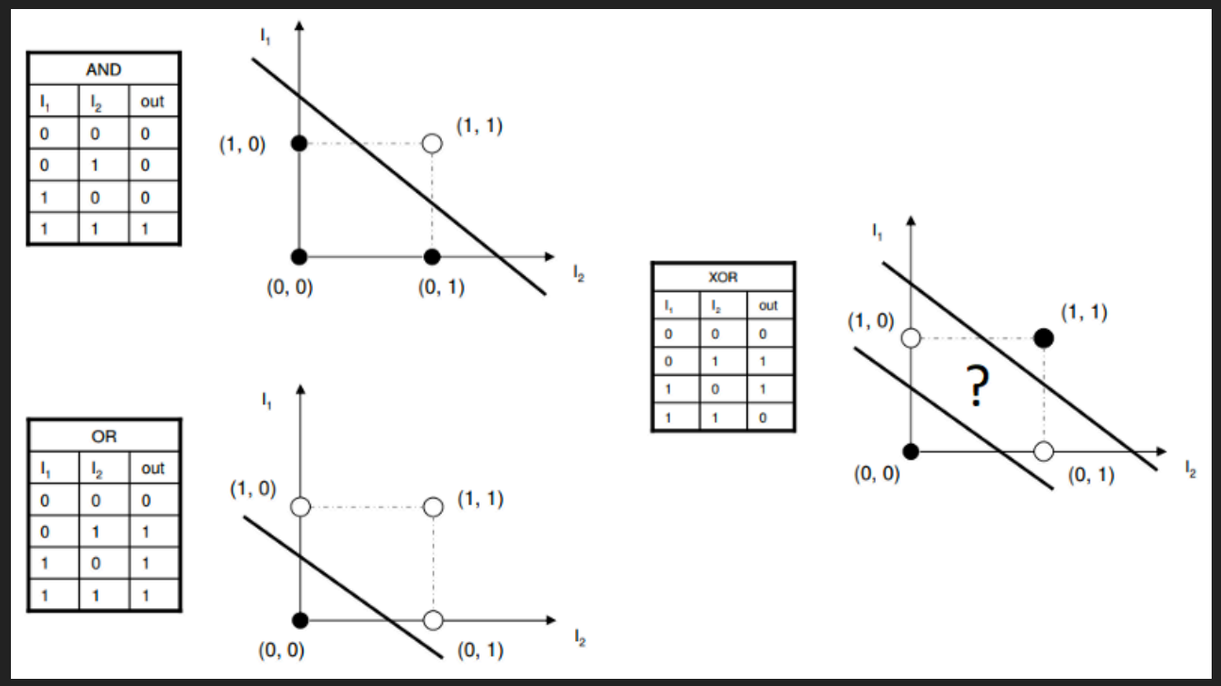
$$y = \phi(\sum_i w_i x_i + b) = \phi(\mathbf{w}^T \mathbf{x} + b)$$

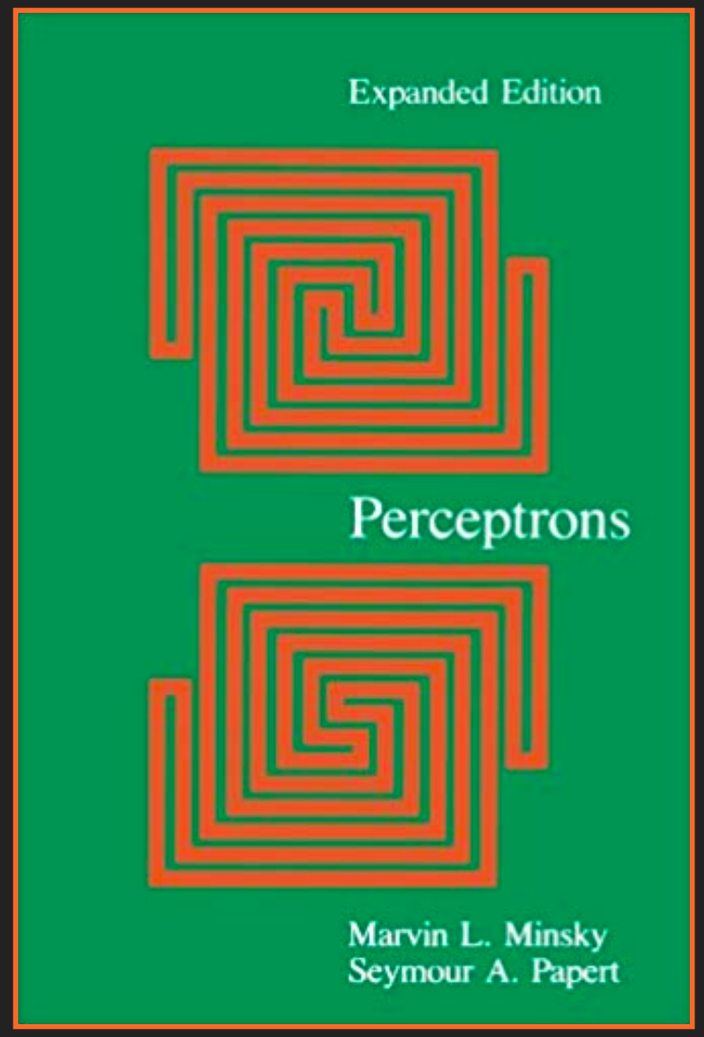
- Produces a single output from a ma… s and biases



THE FIRST FALL OF NEURAL NETWORKS

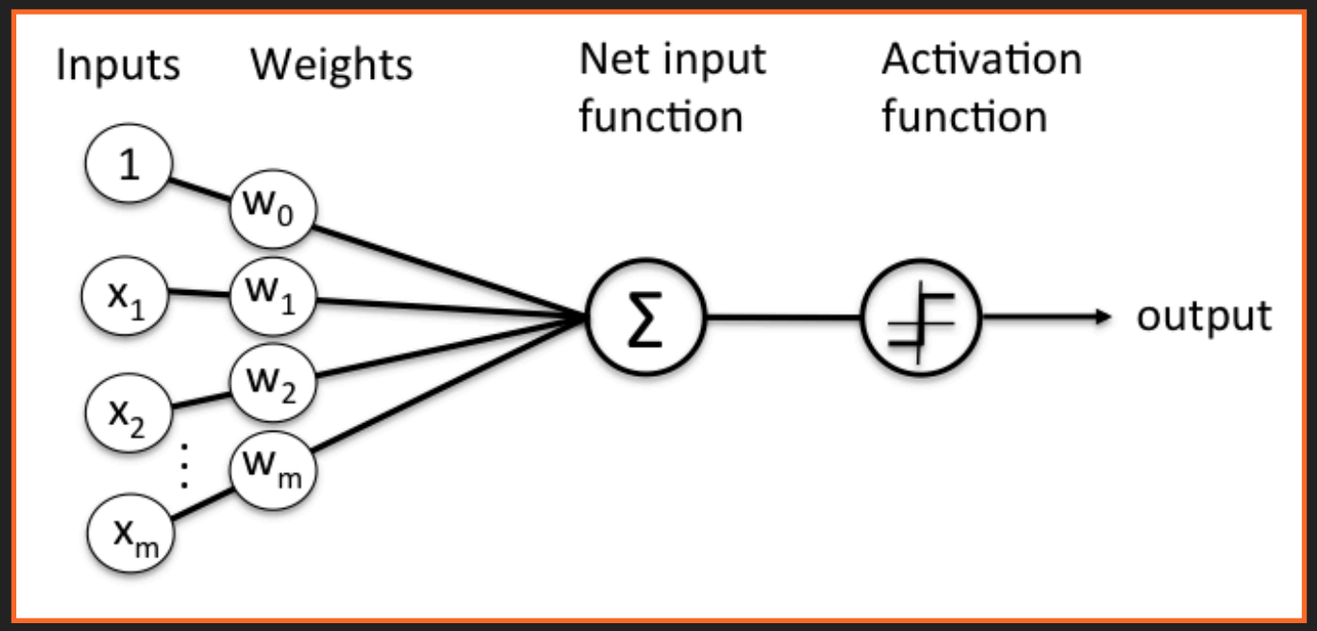
- Single layer
- Minsky and Papert showed they could not solve non-linear classification





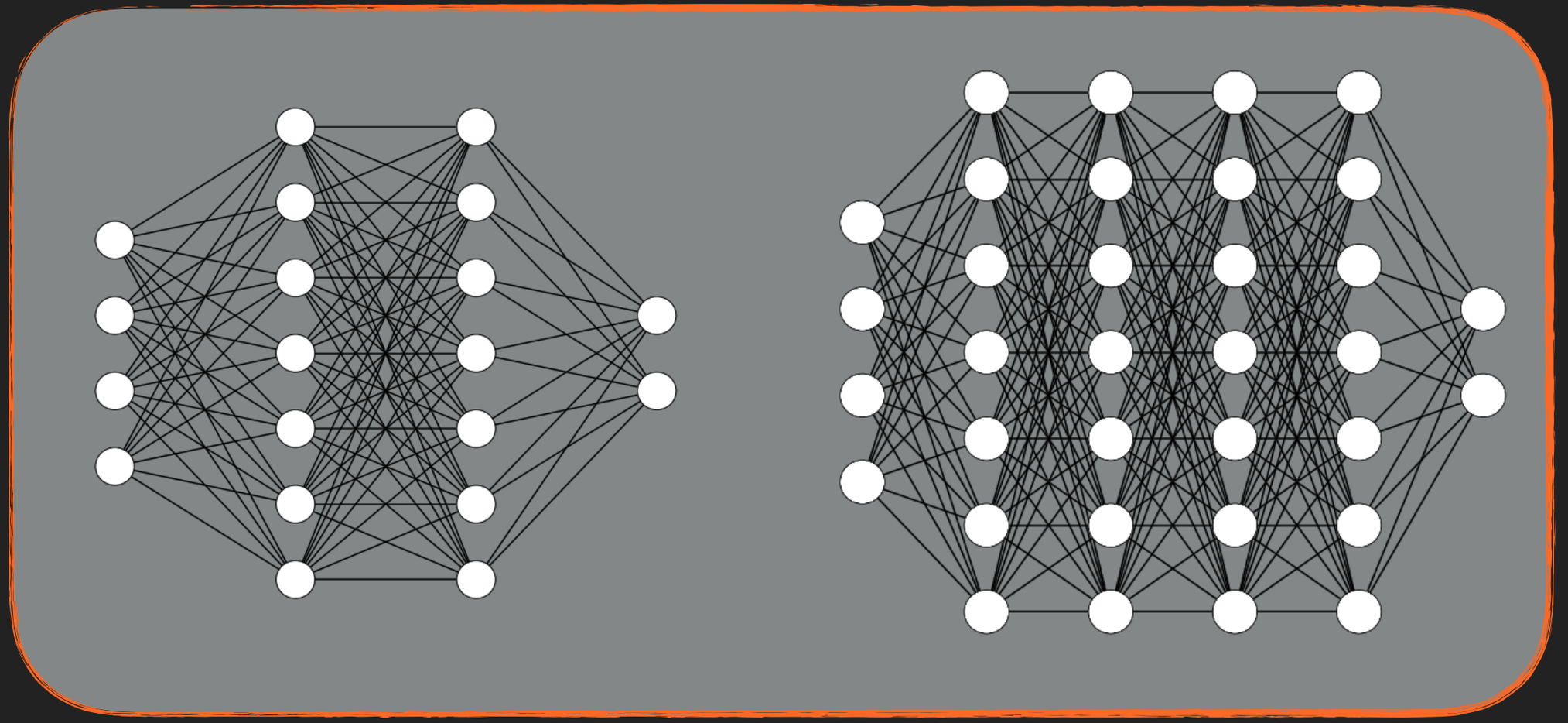
THE NEXT WAVE OF NEURAL NETS: 1980S

- Back propagation
- Now gradients could be used to minimise error
- Modifications back propagate through the network using the chain rule



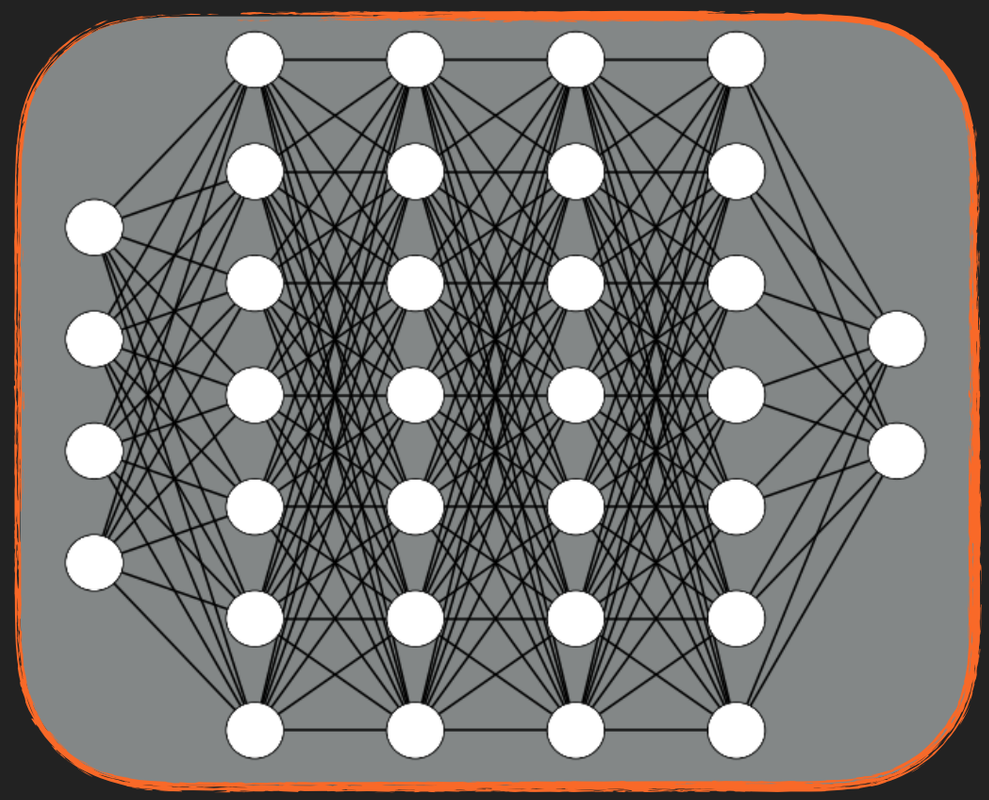
THE NEXT WAVE OF NEURAL NETS: 1980S

- Multi-layer perceptrons (MLPs)
- Can now solve non-linear problems



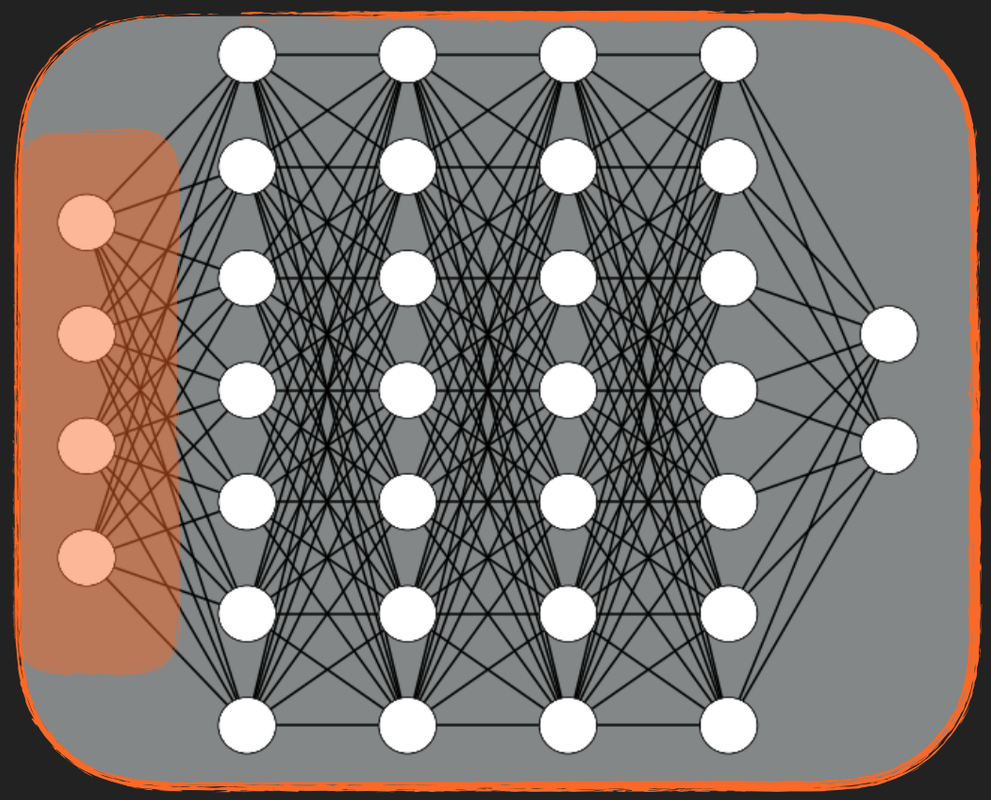
THE ELEMENTS OF A NEURAL NETWORK

- Input layer
- Hidden layers
- Output layer



THE ELEMENTS OF A NEURAL NETWORK

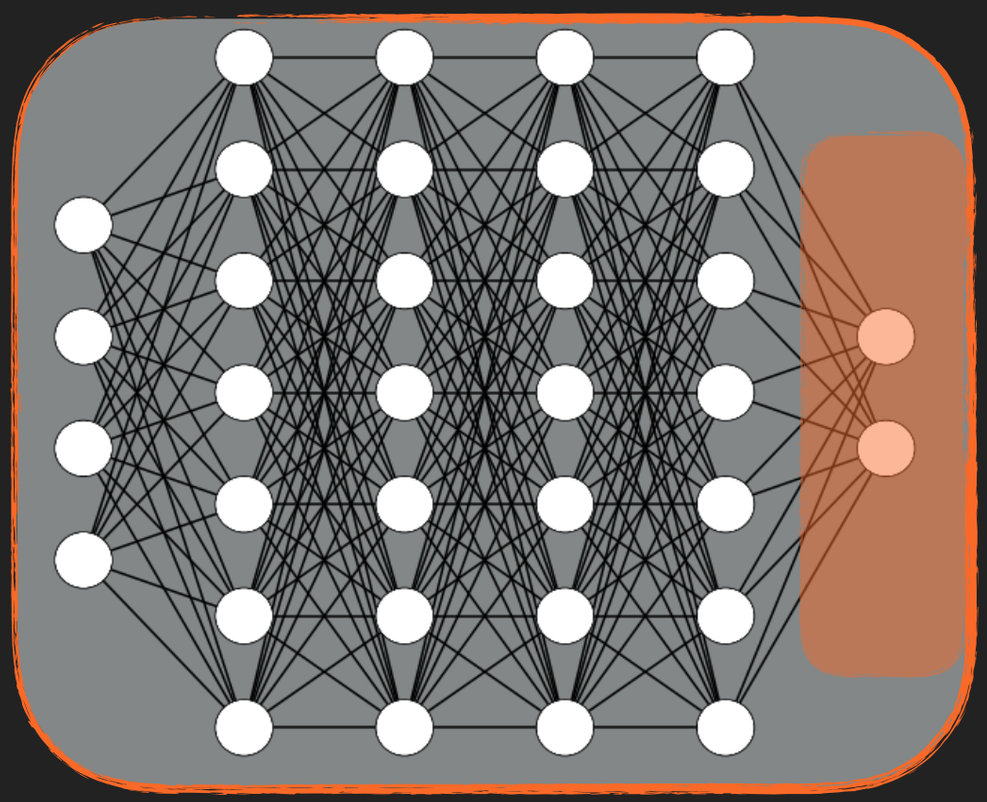
- **Input layer**
- Hidden layers
- Output layer



- Data structure
  - Features of the data

THE ELEMENTS OF A NEURAL NETWORK

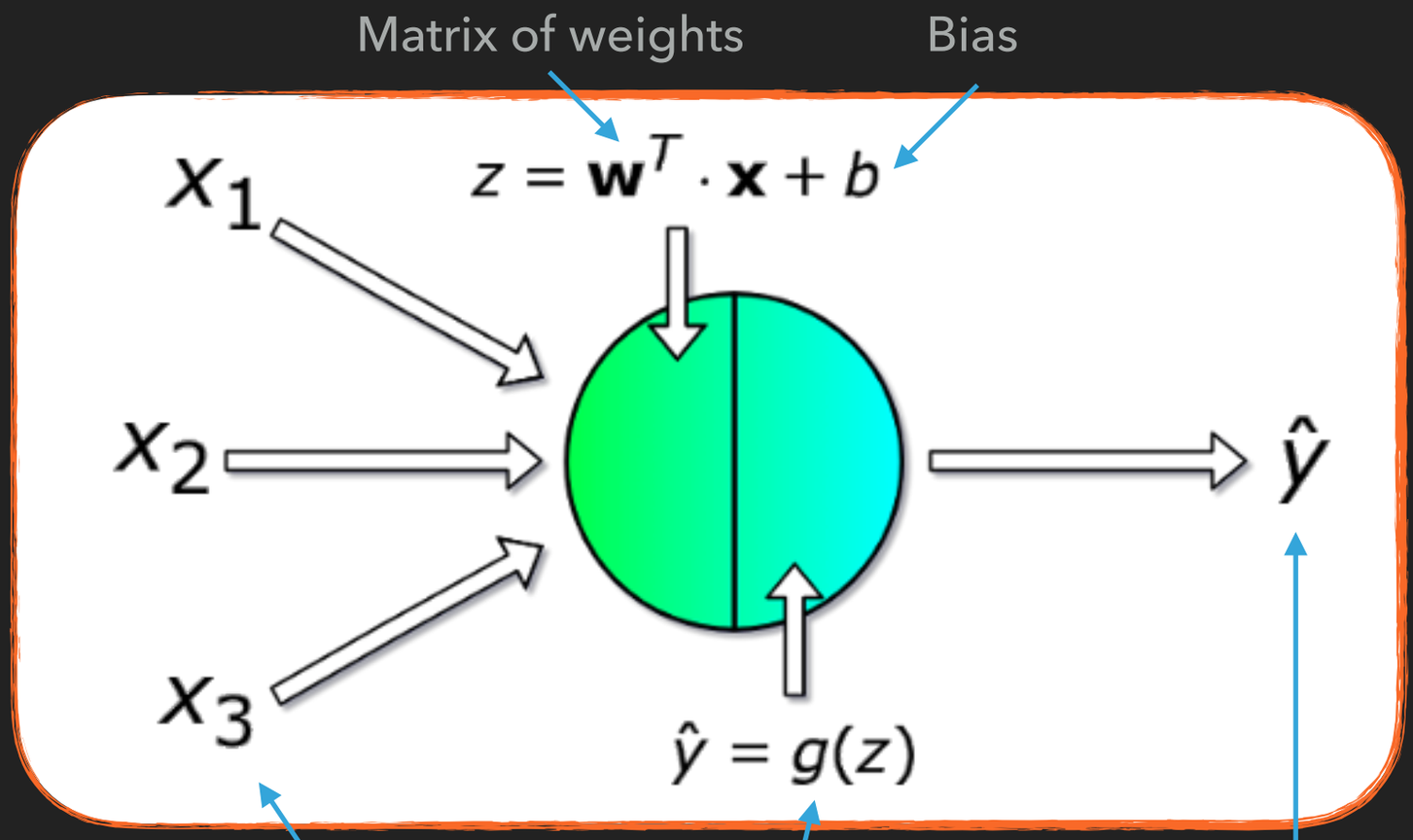
- Input layer
- Hidden layers
- **Output layer**

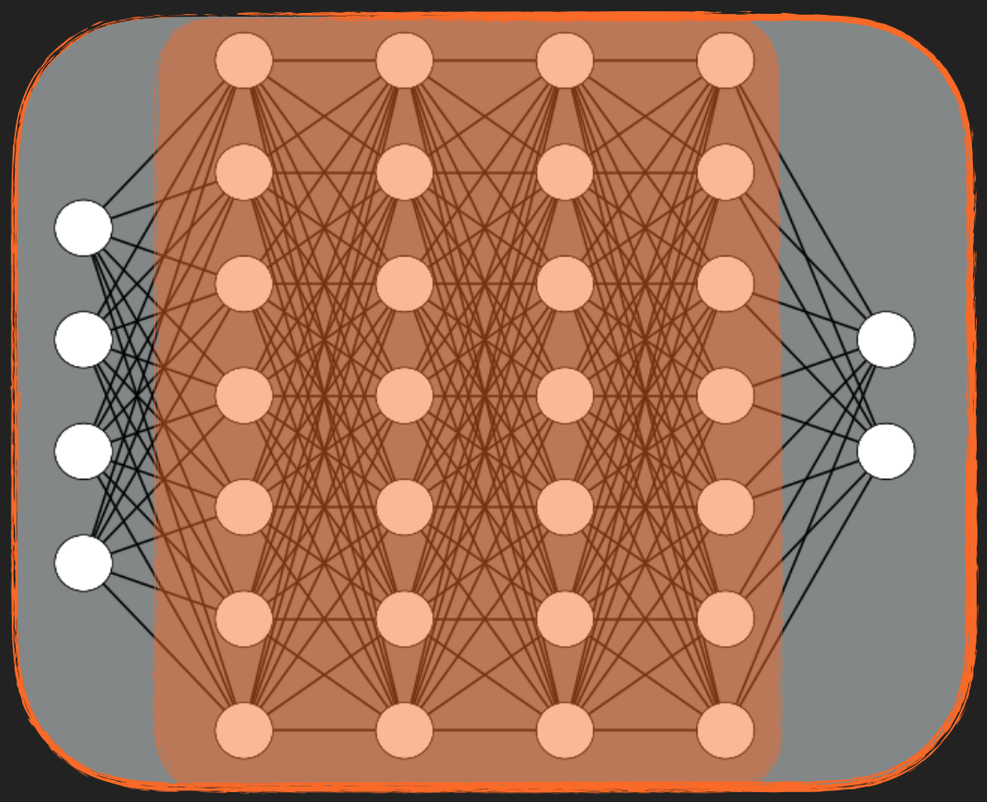


  - Regression
    - Single unit (usually)
  - Classification
    - Multiple units
    - One-hot encoding

HIDDEN LAYERS

- Neurons
- Connections



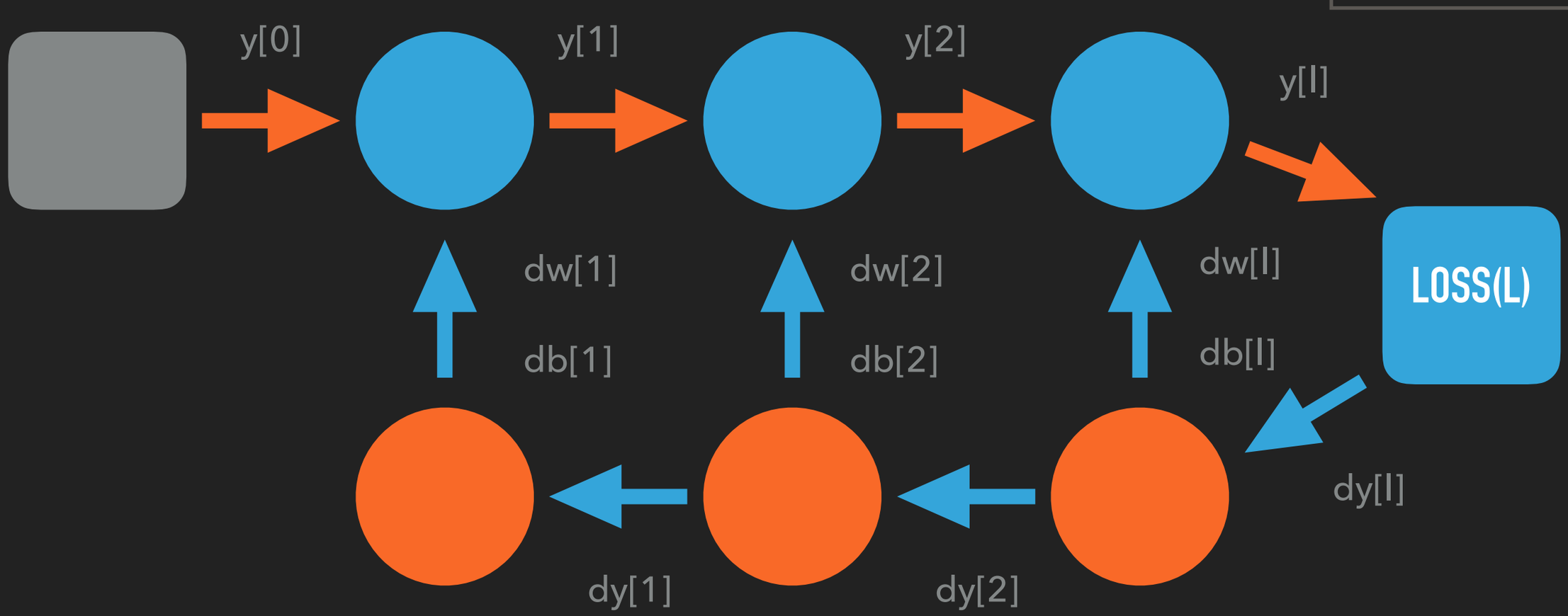


BACK PROPAGATION

- Learn from the loss

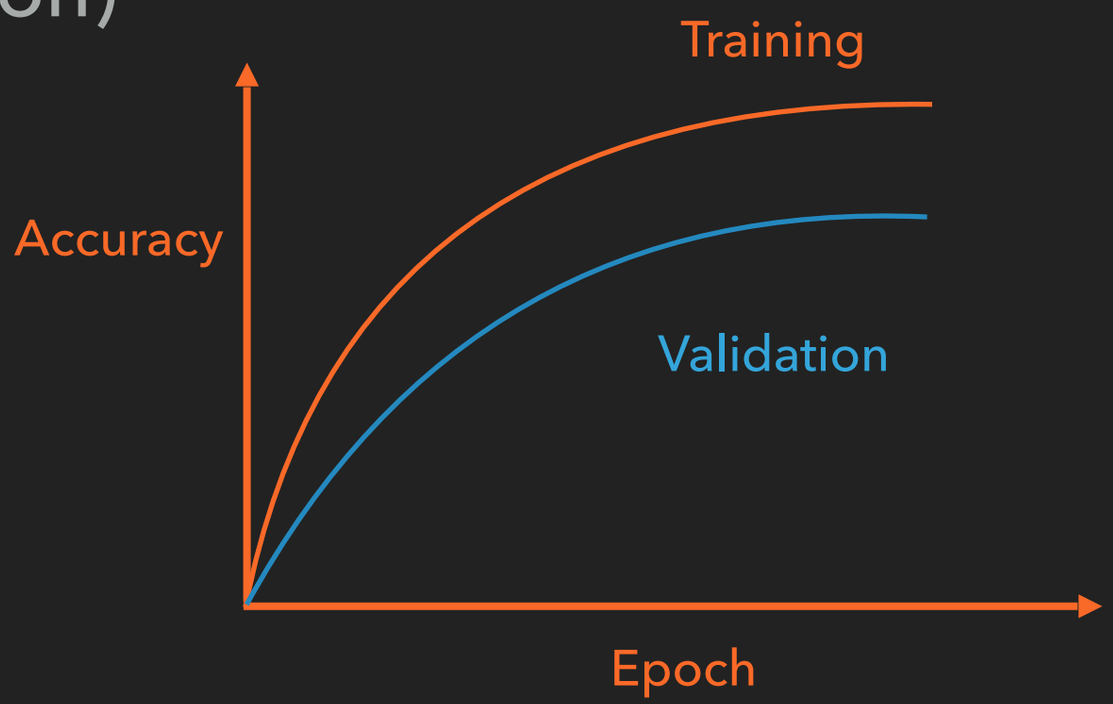
Notation:

dL/dy[l] => dy[l]



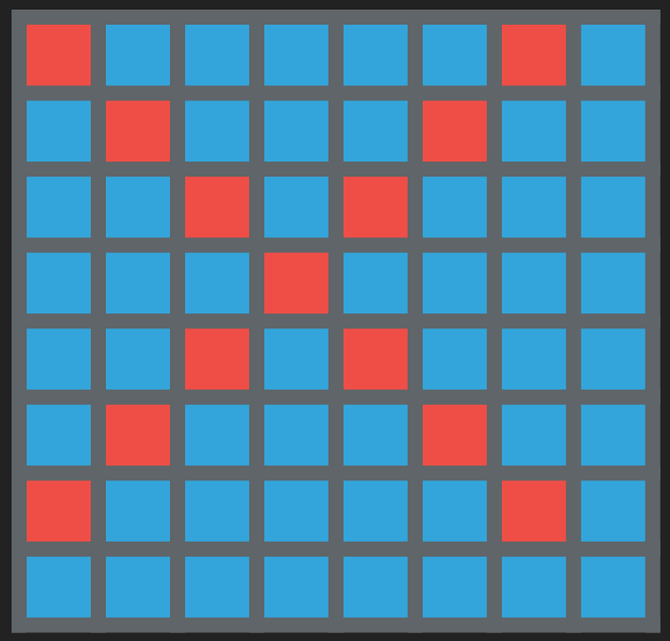
TRAINING

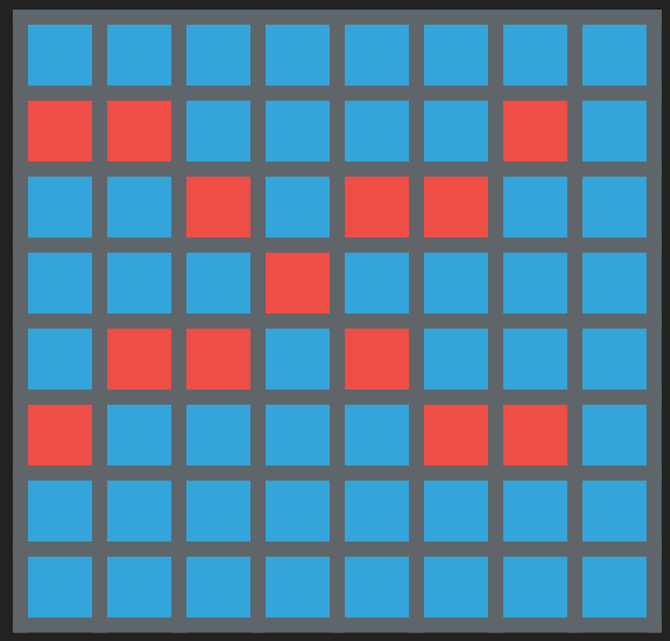
- Run back-prop until a criterion is met
- Loss functions
  - Cross-entropy (categorisation)
  - Mean average error (regression)
- Optimisers
  - Stochastic gradient descent
  - ADAM

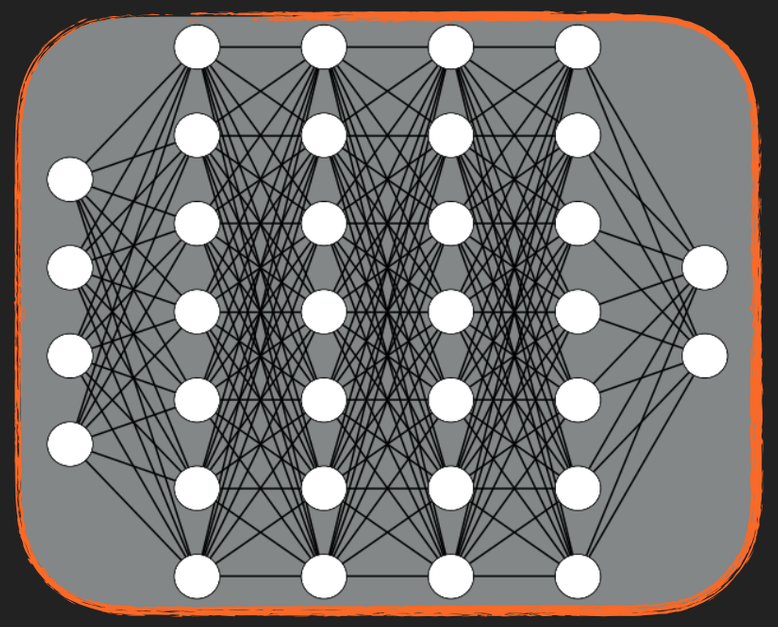


MLPS STRUGGLE WITH IMAGE RECOGNITION

- For a computer, literally these do not match
- The MLP has no real concept of the spatial relations
- Also, dense connections lead to parametric explosions for many pixel images

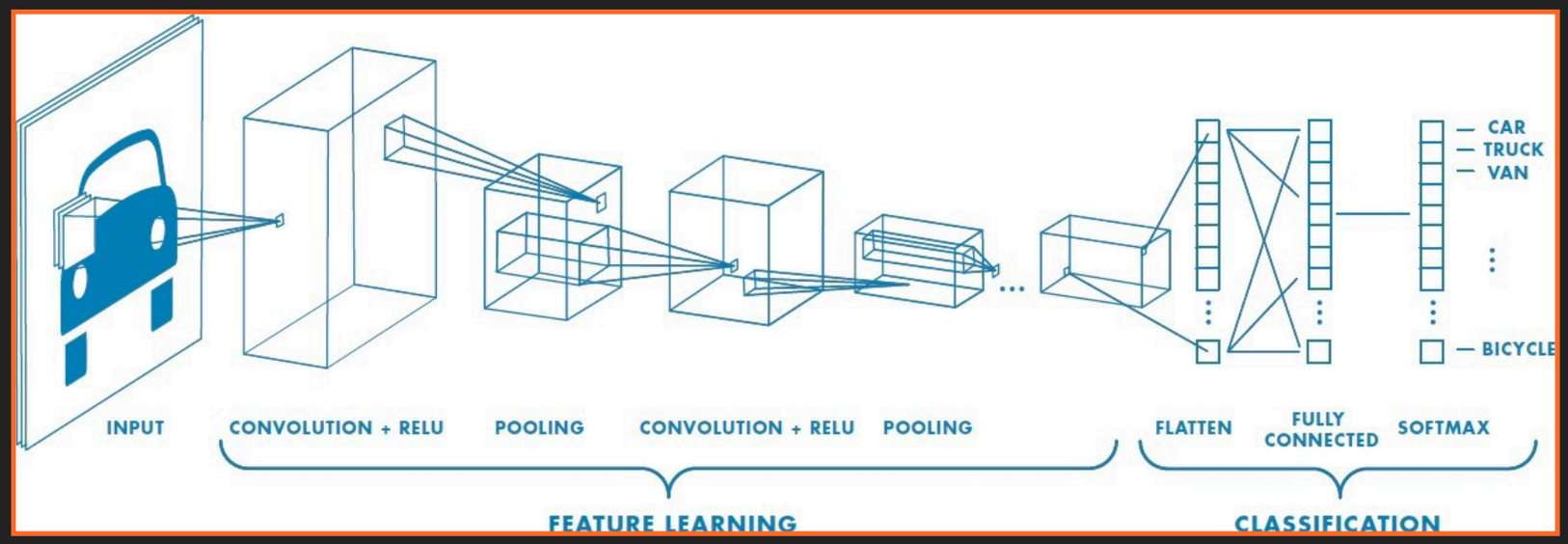


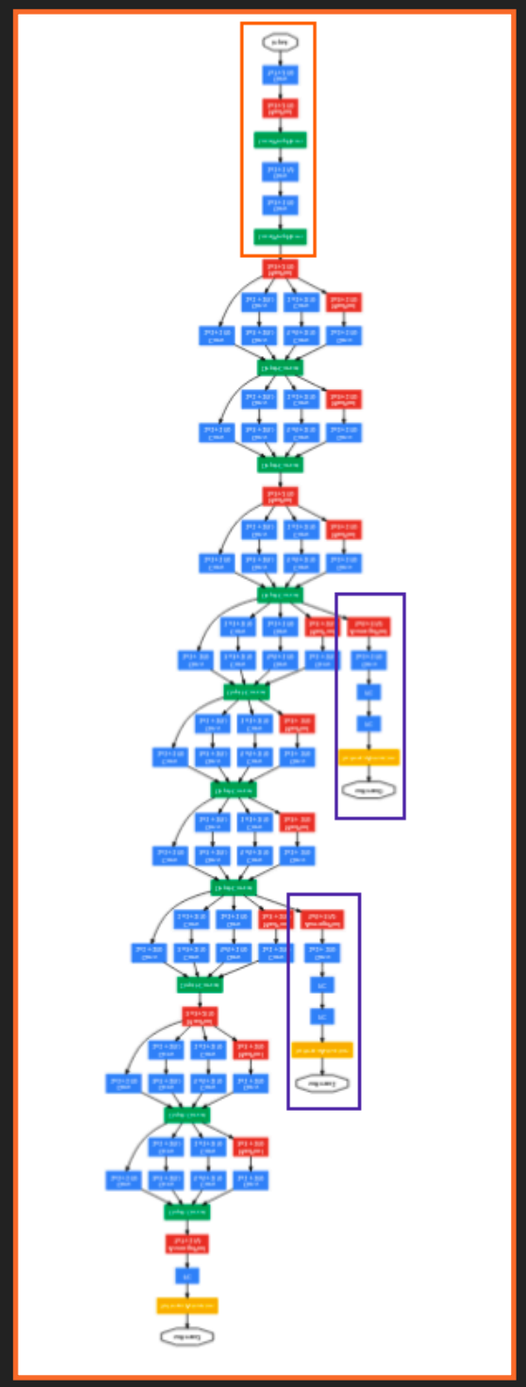




CONVOLUTIONAL NEURAL NETS (CNNS)

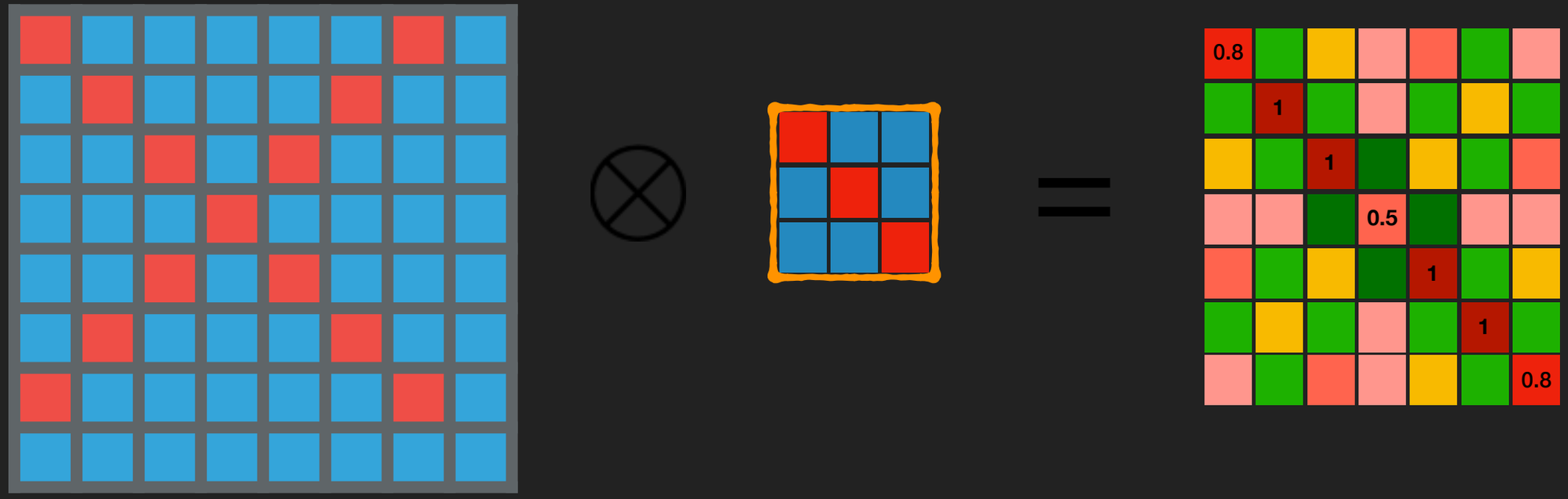
- Uses filters to pick out important features
- Compresses image information
- Is finally connected to a typical NN layer
- Successful CNNs are often very deep





HOW CNNS WORK

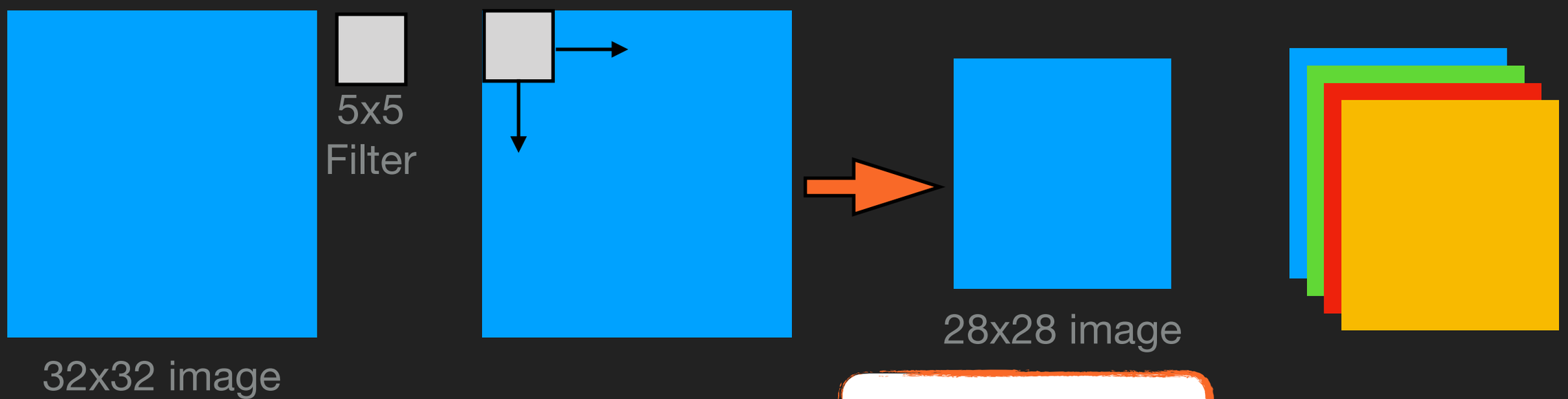
- Filters work to pick out features in the image



HOW CNNS WORK

$$\begin{bmatrix} x_{11} & x_{12} & \dots & x_{15} \\ x_{21} & x_{22} & \dots & x_{25} \\ \dots & \dots & \dots & \dots \\ x_{51} & x_{52} & \dots & x_{55} \end{bmatrix}$$

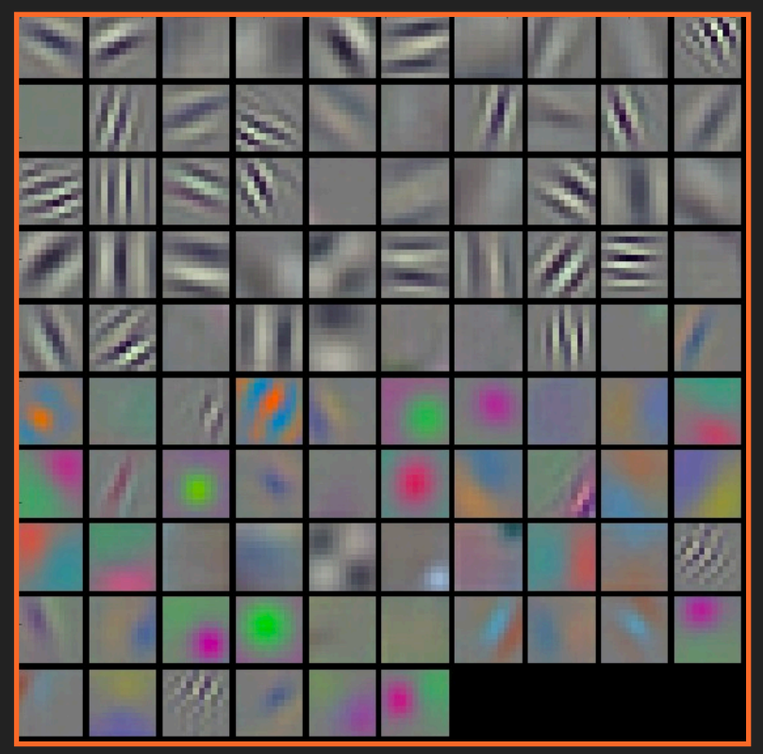
- Filter is a matrix
- Filter dot product with image to produce scalar
- A number of filters are added at each layer



$$I \cdot F = s$$

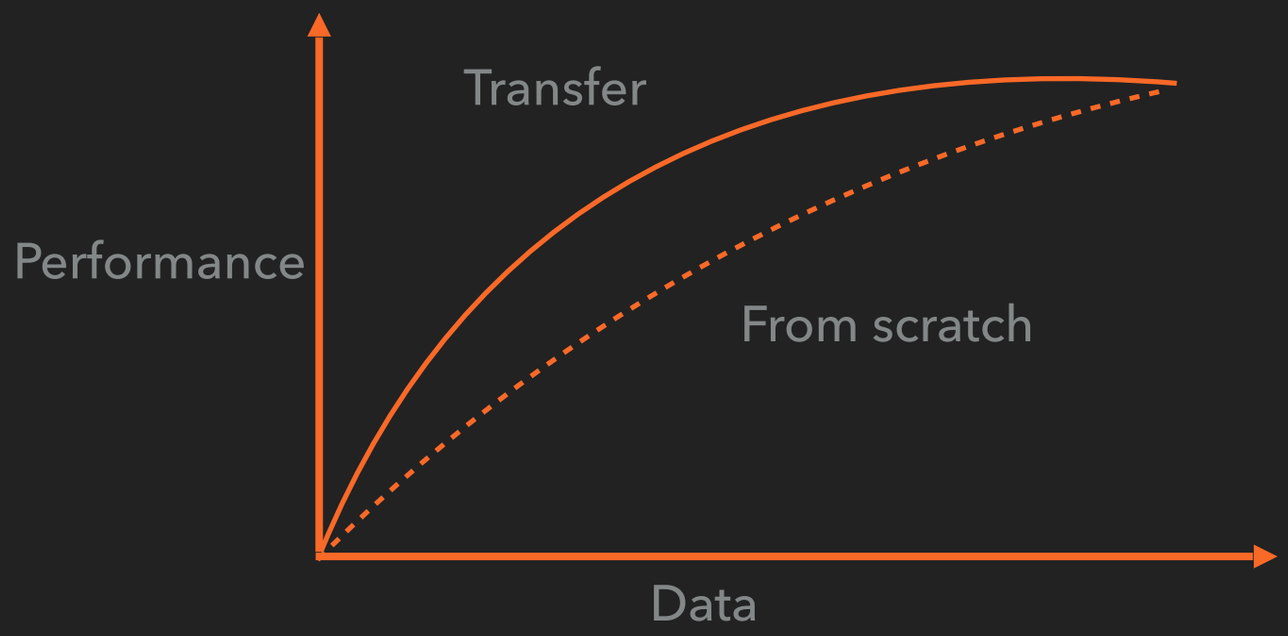
LOOKING AT THE FILTERS

- Visualise trained filters to see what they detect
- Example form the AlexNet network
- Filters pick up edges and colours



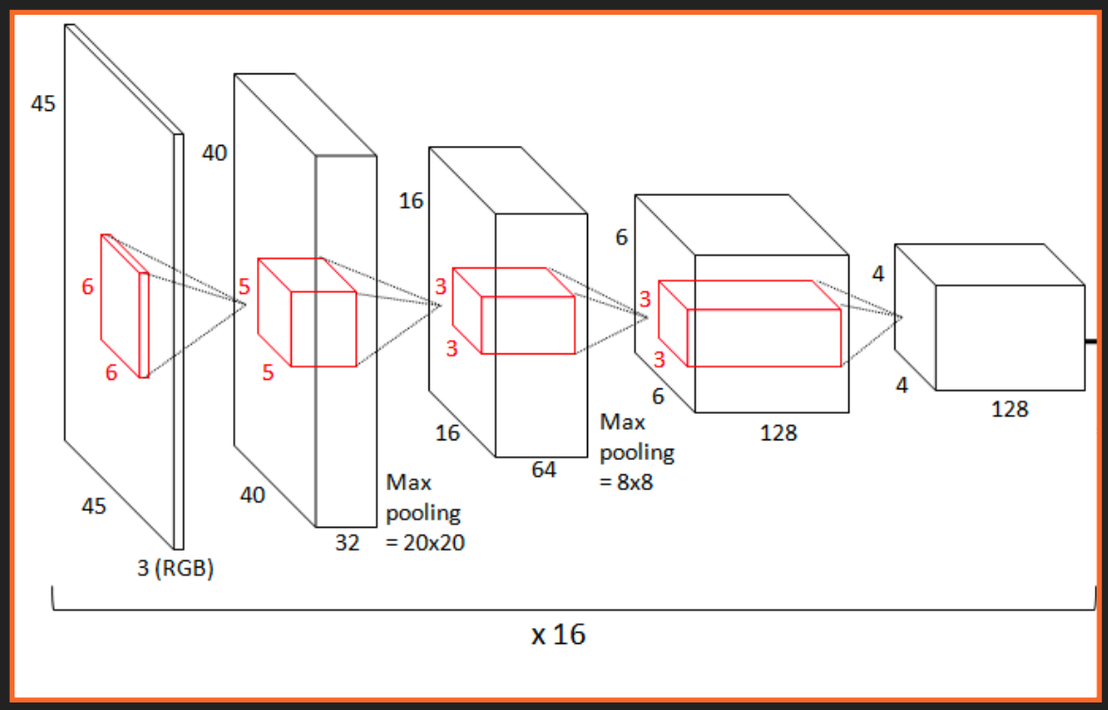
LOW DATA LEARNING OF CNNS

- Often existing feature maps will work for a new problem
- Can load existing models and weights
- Retrain on a small labelled dataset
- Transfer learning



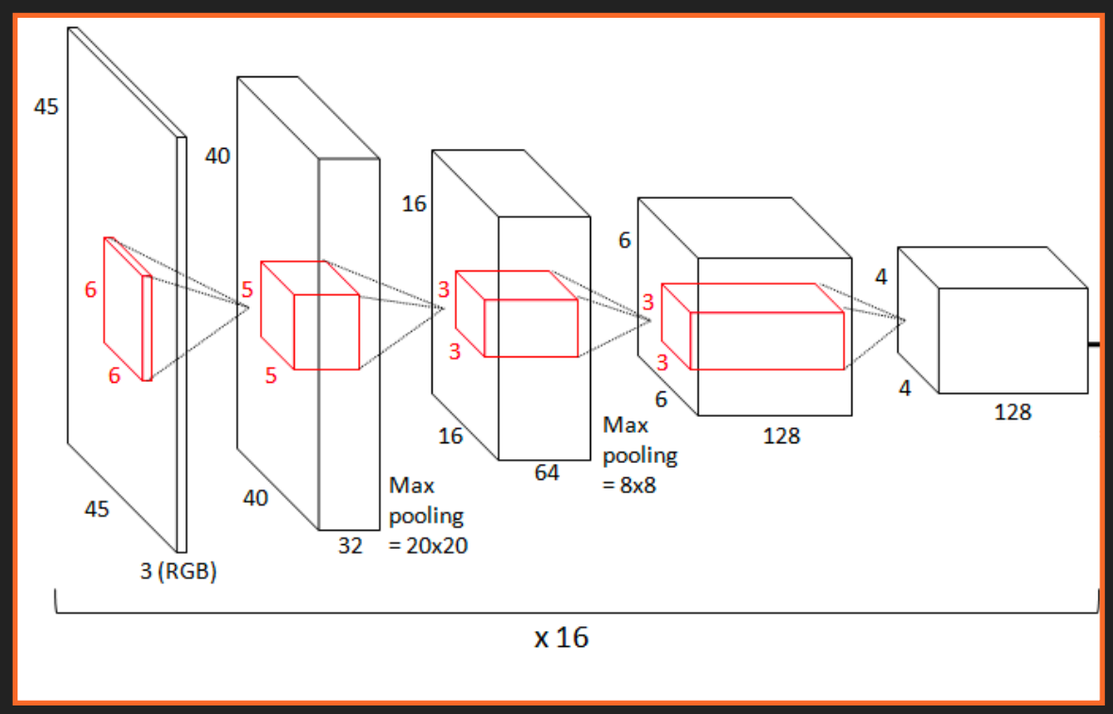
THREE WAYS TO USE A CNN

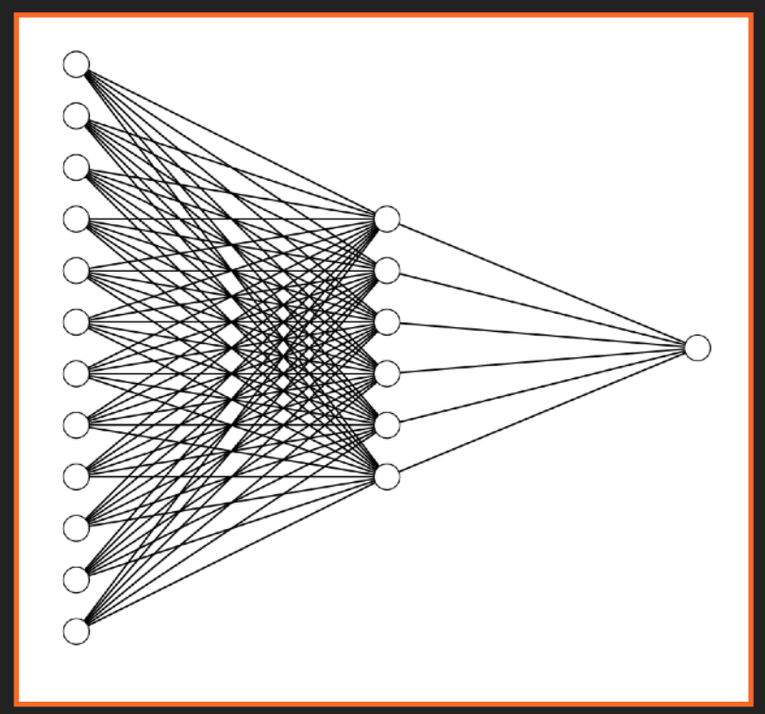
- Regression
- Classification
- Segmentation



THREE WAYS TO USE A CNN

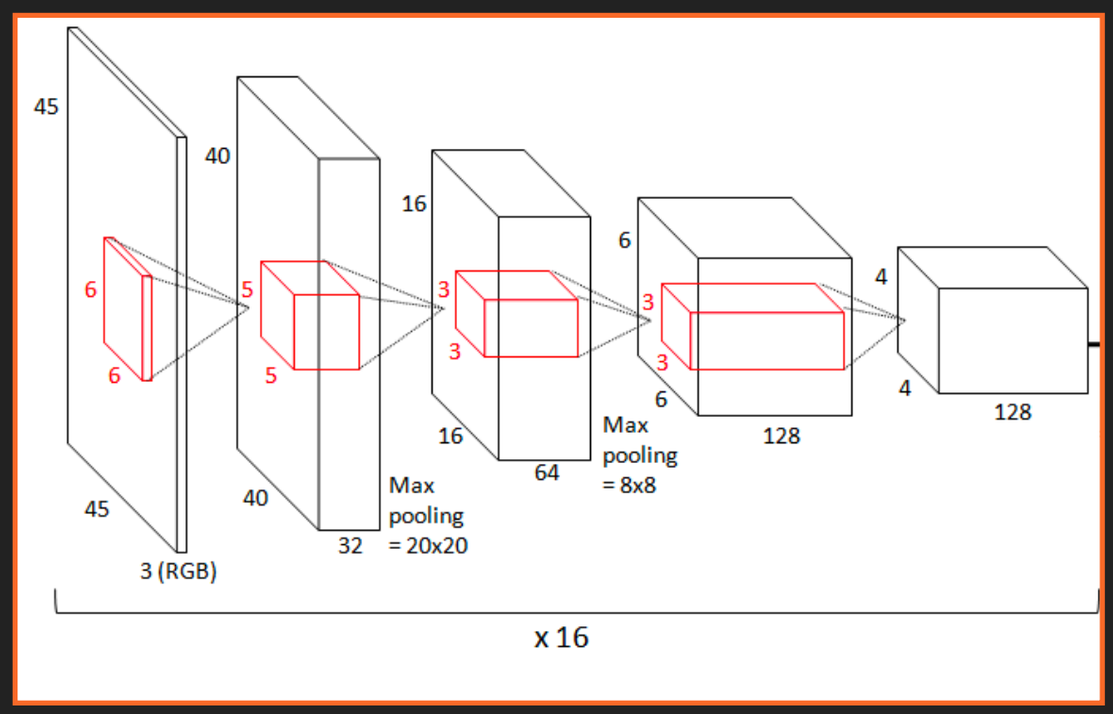
- **Regression**
- Classification
- Segmentation

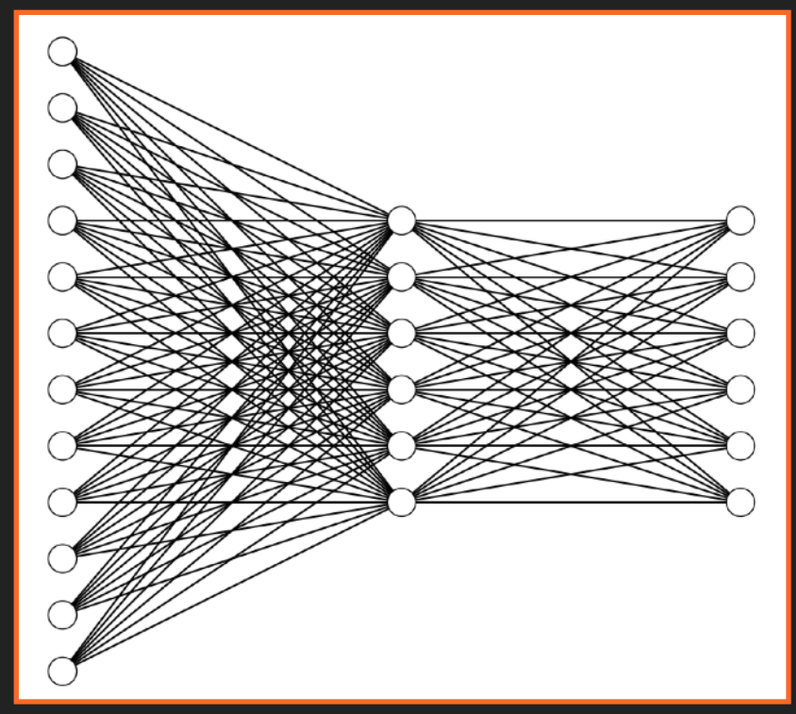




THREE WAYS TO USE A CNN

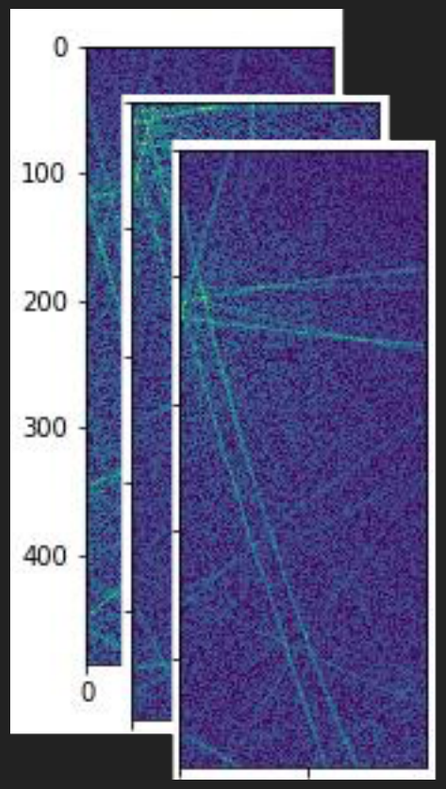
- Regression
- **Classification**
- Segmentation



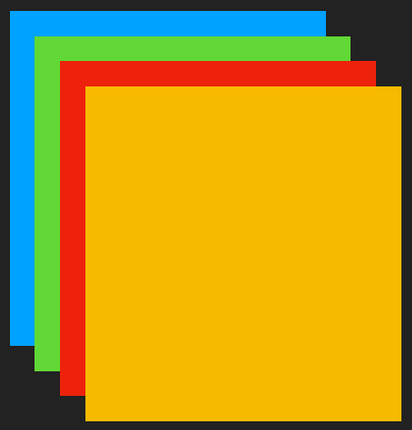


USING CNNS IN MATERIALS SCIENCE

- Images are compressed by filters
- Filters are updated to learn the important features of the image



Feature maps 32@486x194

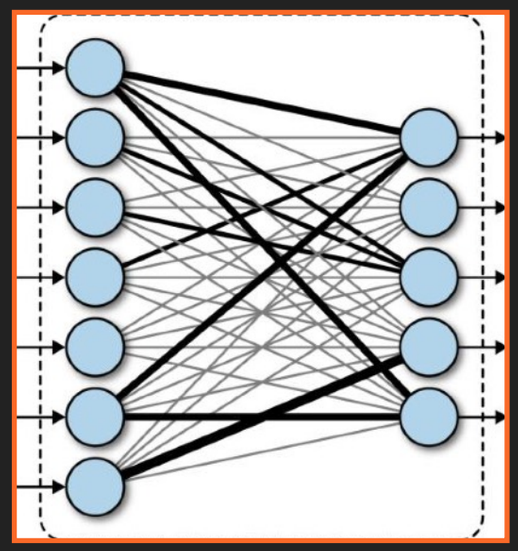


3x3 kernel

Feature maps 64@242x96

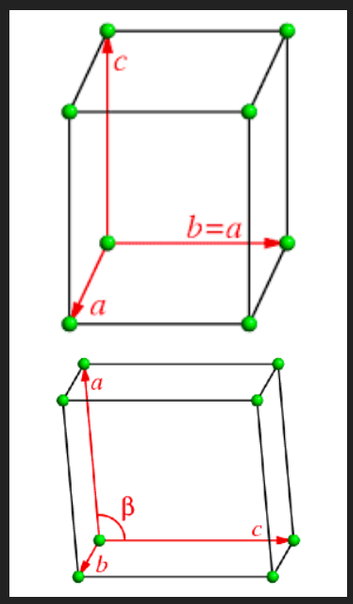
3x3 kernel

Fully connected Layers



16 nodes
8 nodes

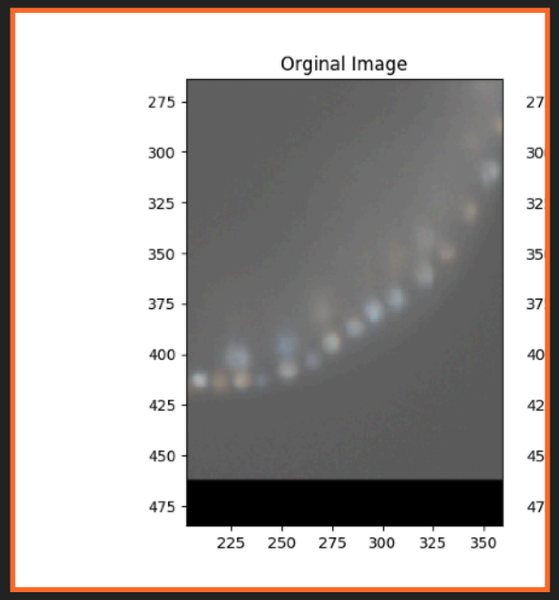
Identify lattices present

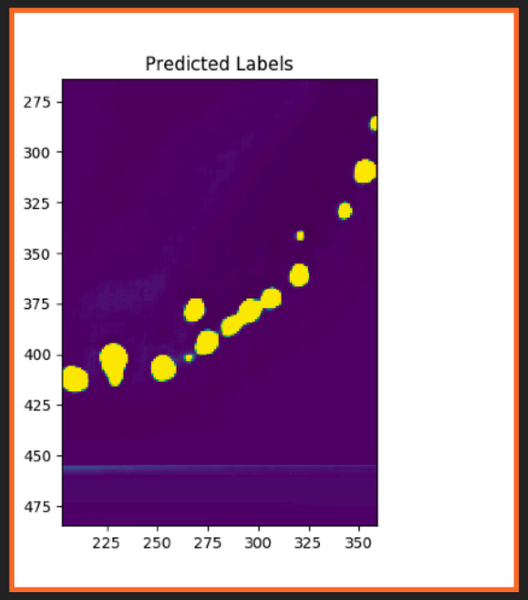


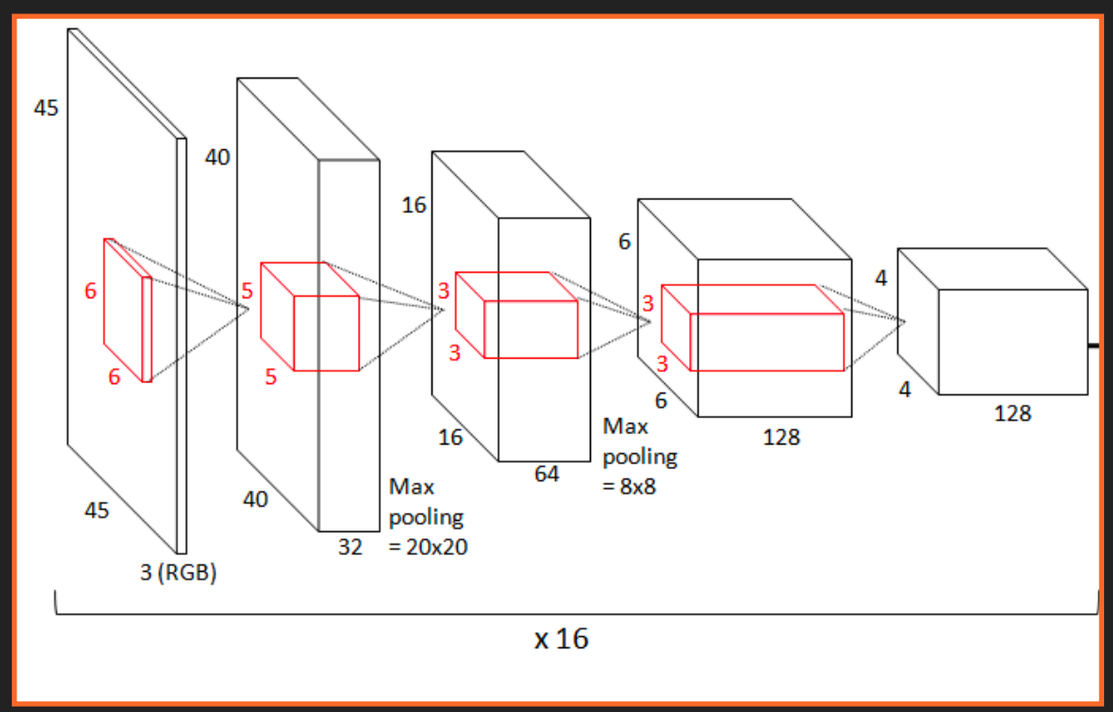
Butler, Proc. Royal Soc. A - Under Review

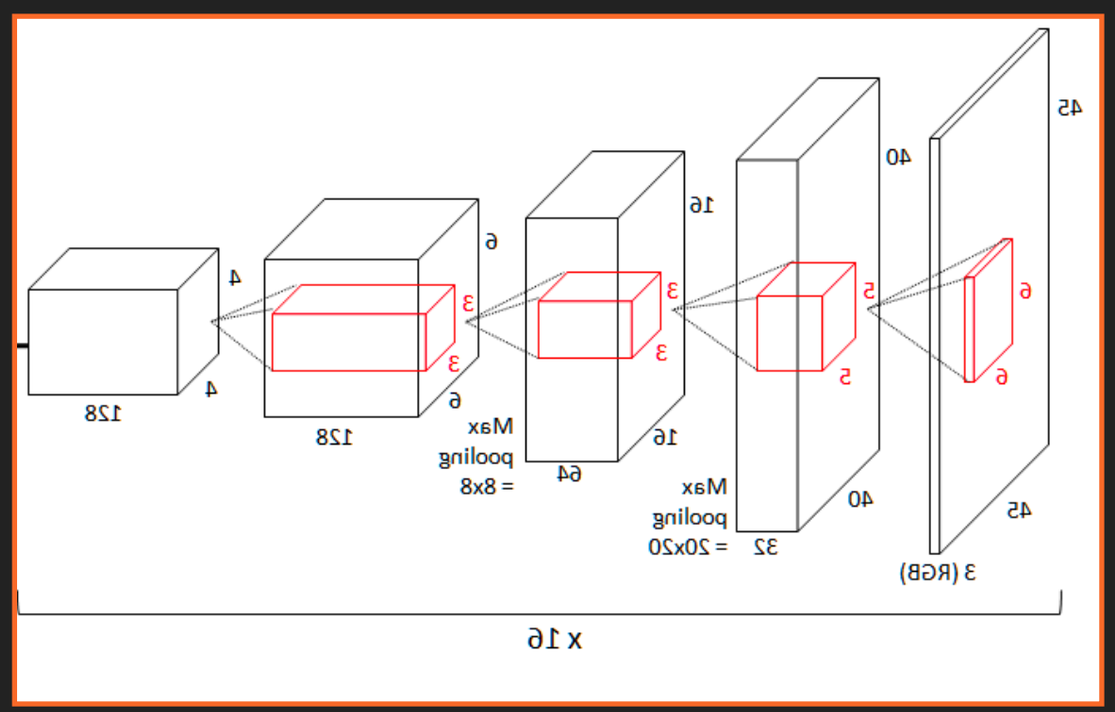
THREE WAYS TO USE A CNN

- Regression
- Classification
- **Segmentation**

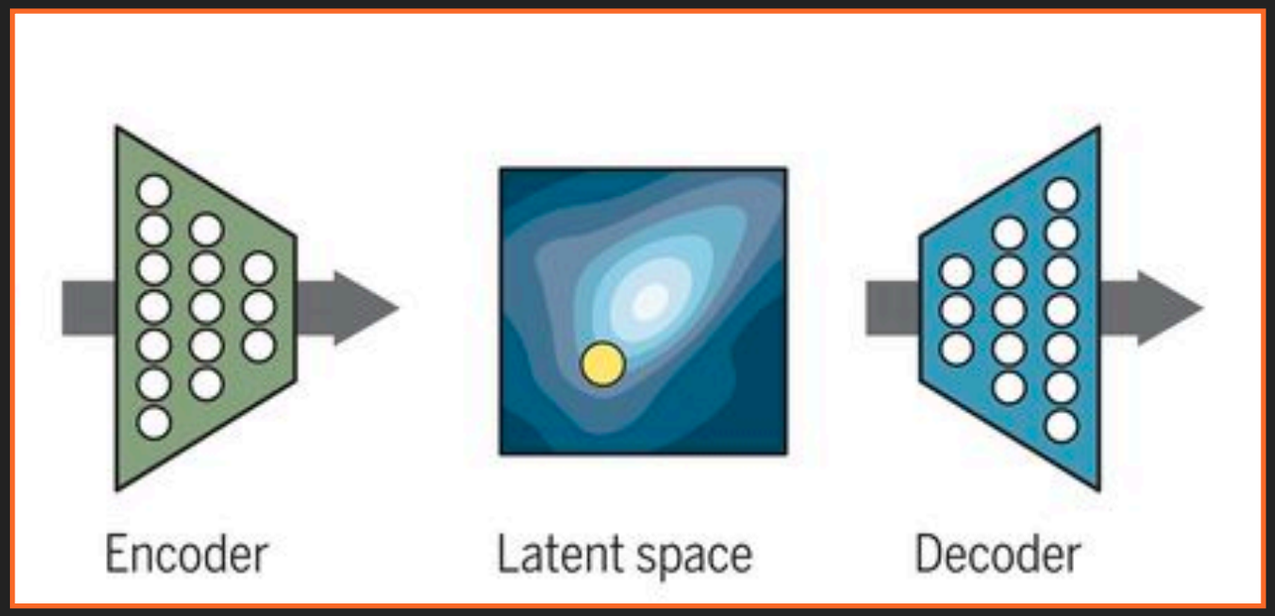


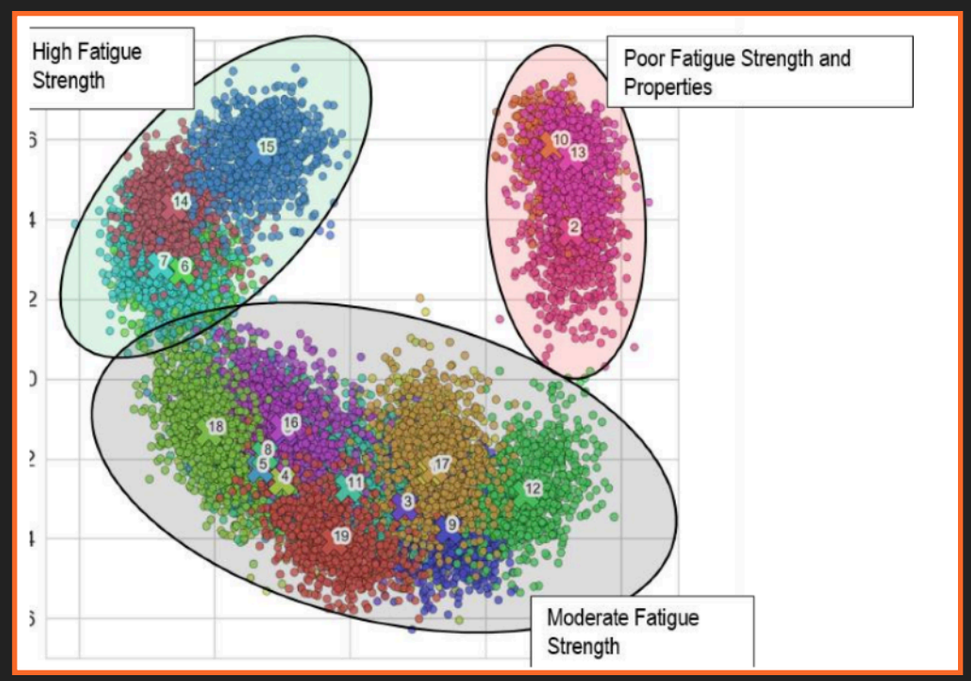






INVERT CNNS AS GENERATIVE MODELS



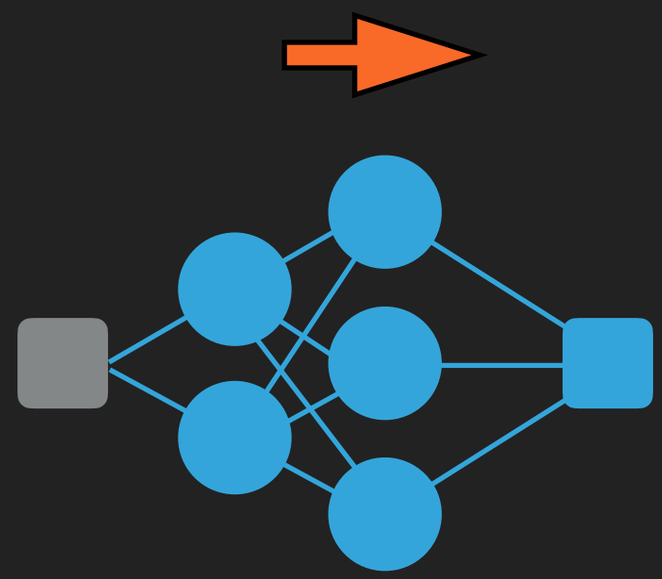


- Convolution compresses to a latent space
- Examine the latent space with PCA
- Explore latent space and invert the encoder to predict new systems

Science 361 360 2018
arXiv:1901.10281 2019

NEURAL NETWORKS FOR TIME SERIES DATA

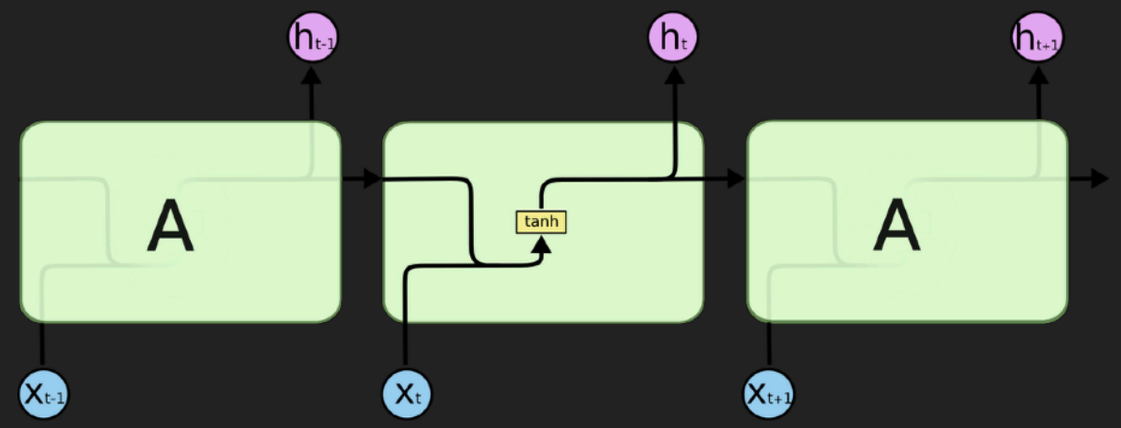
- Often algorithms are desired for predicting the next event based on a series of previous events
- Eg Pressure/temperature evolution, speech prediction …
- In this case standard NNs are not very useful due to a lack of ‘memory’



Feed forward network
Information never touches a node twice

RECURRENT NEURAL NETS

- Recurrent networks re-apply a representation of the state from the previous step
- This is combined with the new information to influence the outcome of the present step
- This gives the network memory - but only for one step

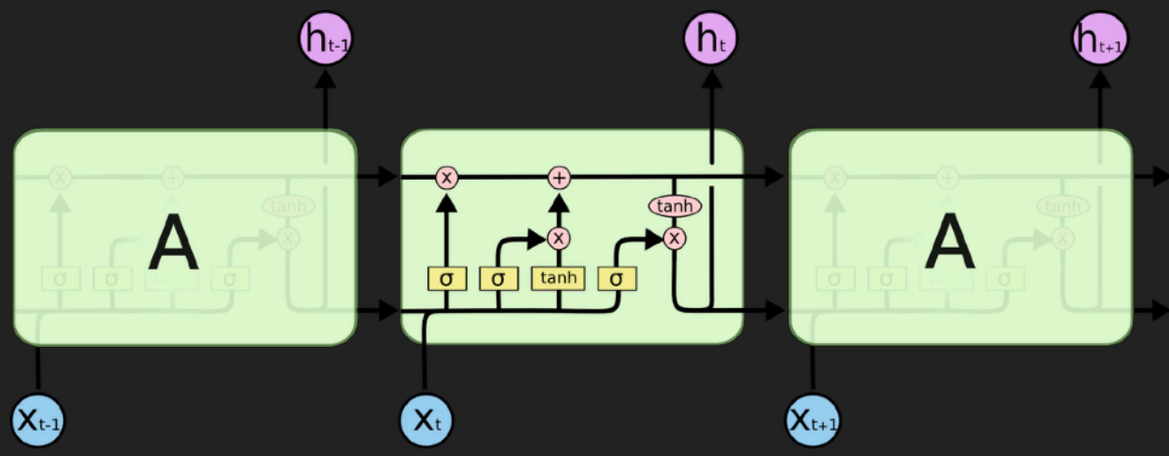


Recurrent network
Information is fed back to the node at the next step

http://colah.github.io/posts/2015-08-Understanding-LSTMs/

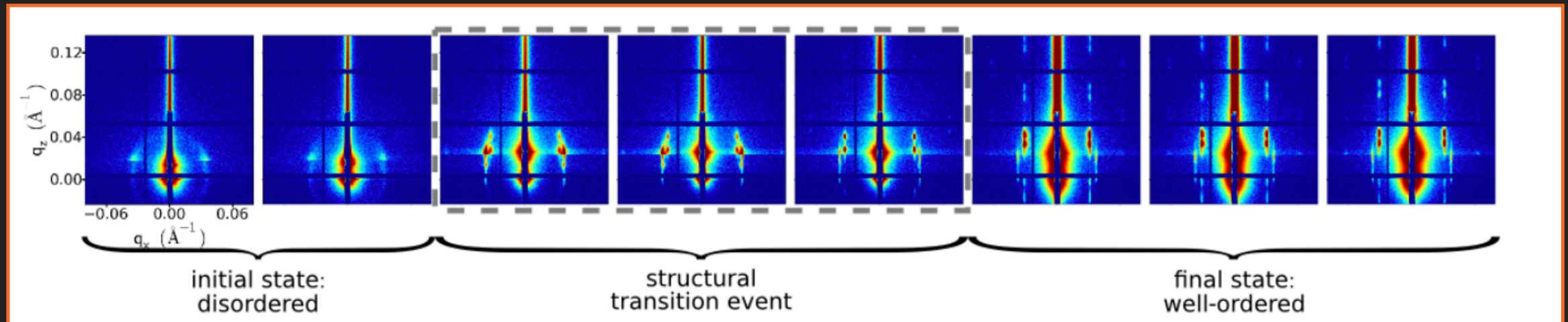
LONG SHORT TERM MEMORY NETWORKS

- LSTMs store representations in separate memory units
- These have three gates
  - Input - decides if a state should enter memory
  - Output - decides if memory should affect the current state
  - Forget - decides if memory should be dumped
- Very effective for time series problems



http://colah.github.io/posts/2015-08-Understanding-LSTMs/

LSTMS FOR MATERIALS CHARACTERISATION



- LSTM can predict the likelihood of a structural transition during operando measurement of a material
- Allows for optimisation of experiment and identification of the region of interest

https://doi.org/10.1145/3217197.3217204

LEARNING

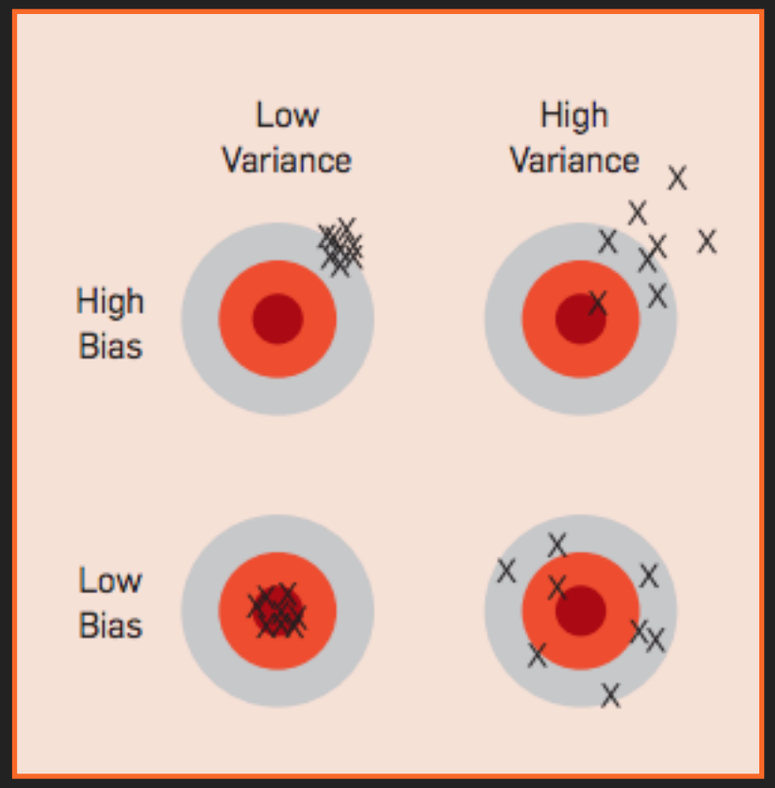
Learning =

Representation + Evaluation + Optimisation

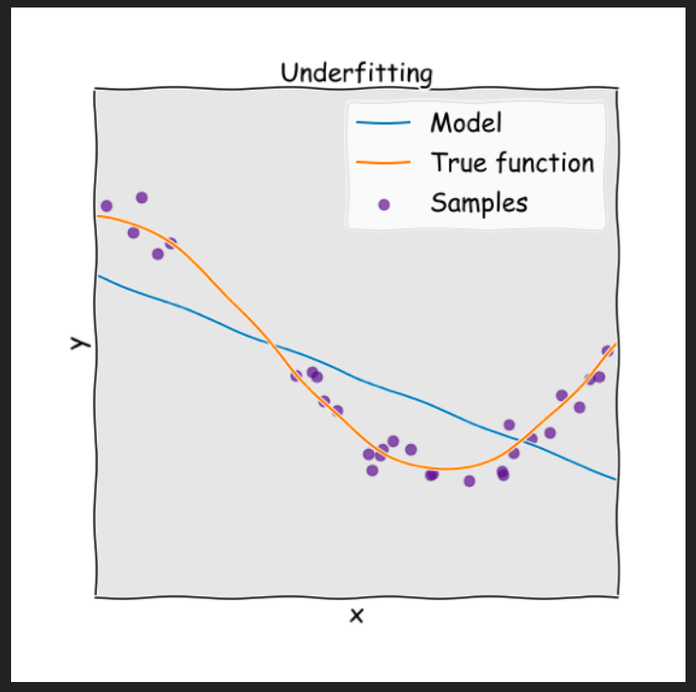
- Evaluation
  - Objective function or scoring function.
  - Distinguish good from bad classifiers.
  - NB need not be the same as the external function that the classifier is optimising.

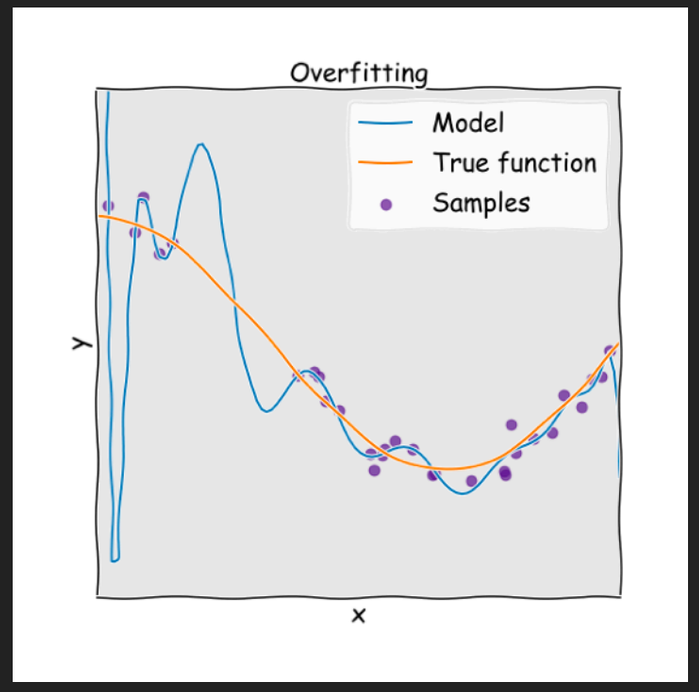
OVER/UNDERFITTING

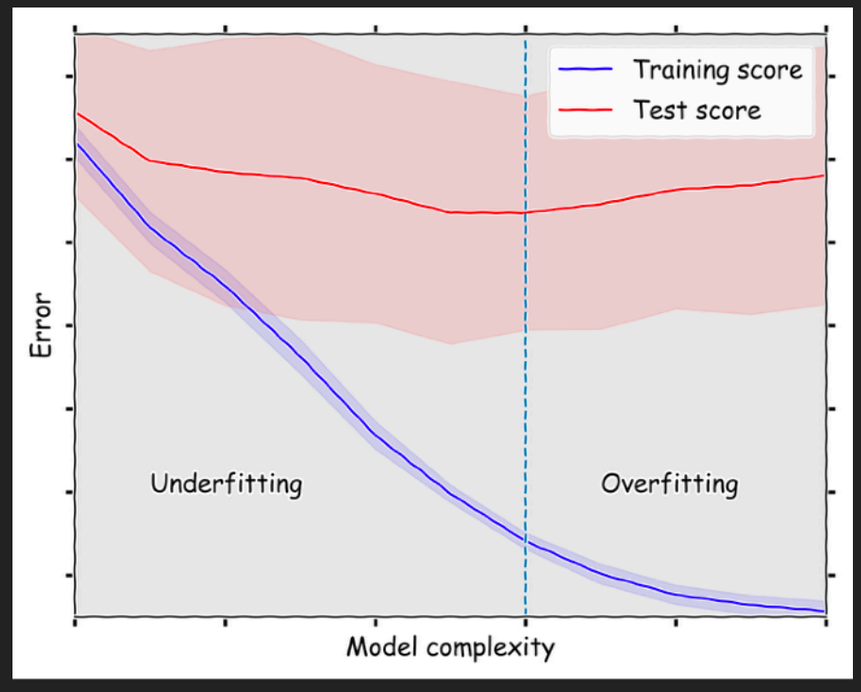
- Bias
  - Constantly learning the same wrong thing.
- Variance
  - Learn random things irrespective of real values.



UNDER/OVER-FITTING

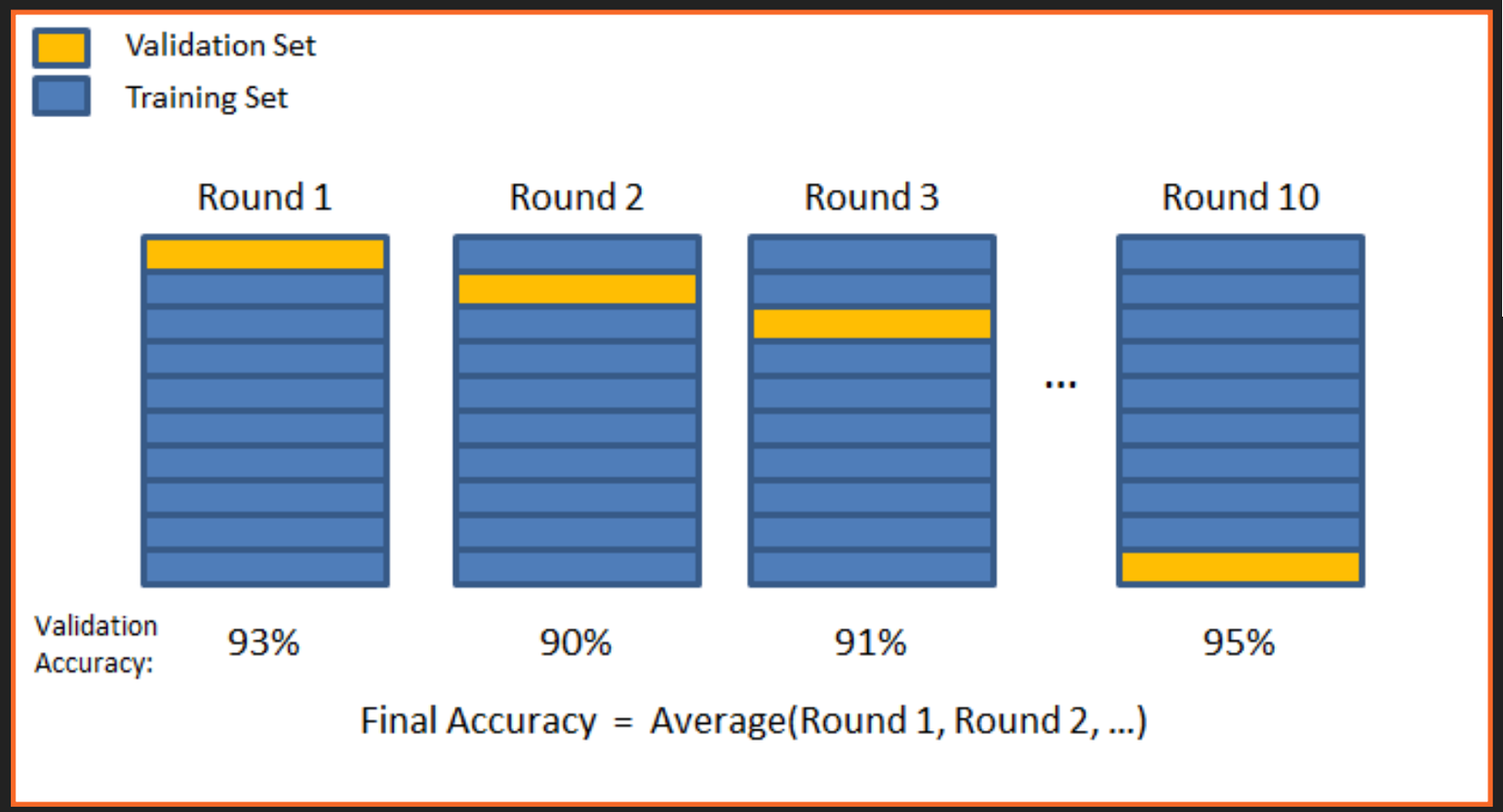


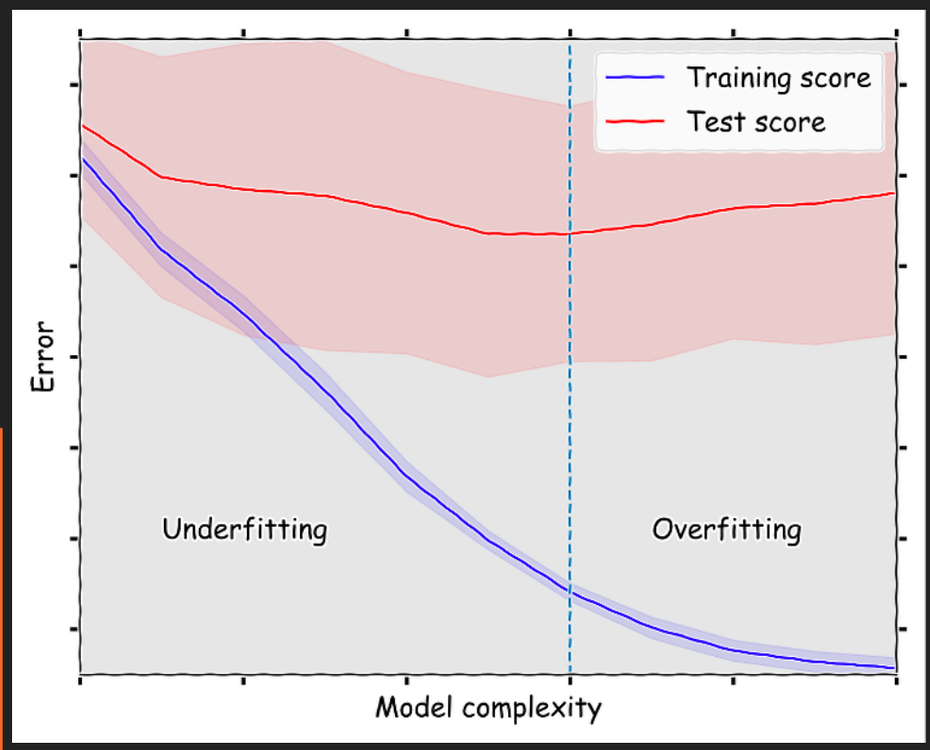




WAYS TO AVOID OVERFITTING

- n-fold cross validation
  - Ensure training/test splits





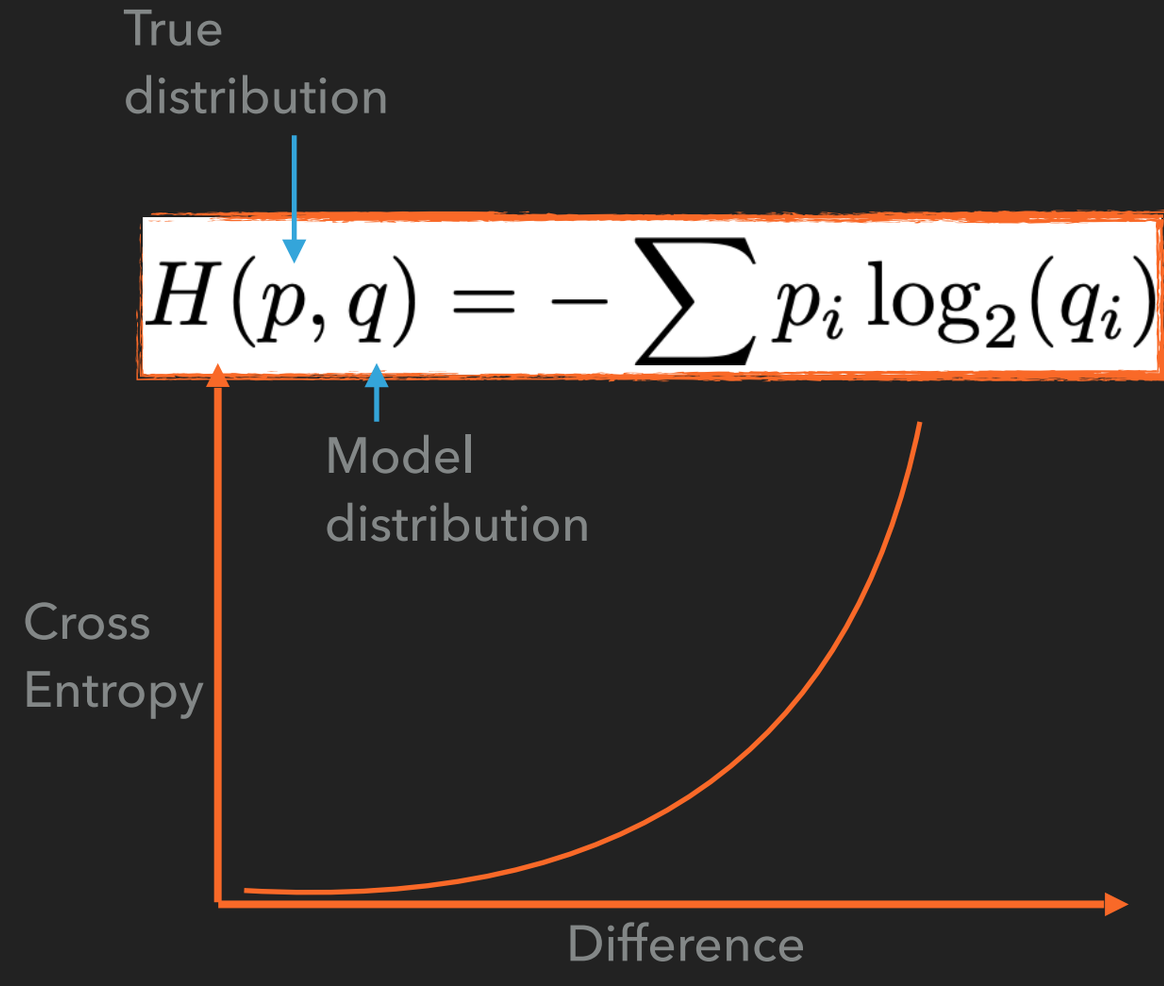
EVALUATION OF MODELS : LOSS FUNCTIONS

- Objective function = loss function = cost function
- Must faithfully represent the “goodness” of a model in a single number

LOSS FUNCTIONS 1: CROSS ENTROPY

- Used for classification problems
- Tells us how similar our model distribution is to the true distribution
- Penalises all errors, but especially those that are most inaccurate

0	0	1	0
0.15	0.25	0.5	0.1



LOSS FUNCTIONS 2: HINGE LOSS

- Used for classification
- Does not seek to reproduce the distribution of data
- 0 as long as the classification is correct

Label(+/-1)

$$L = max(0, 1 - t \cdot y)$$

Prediction

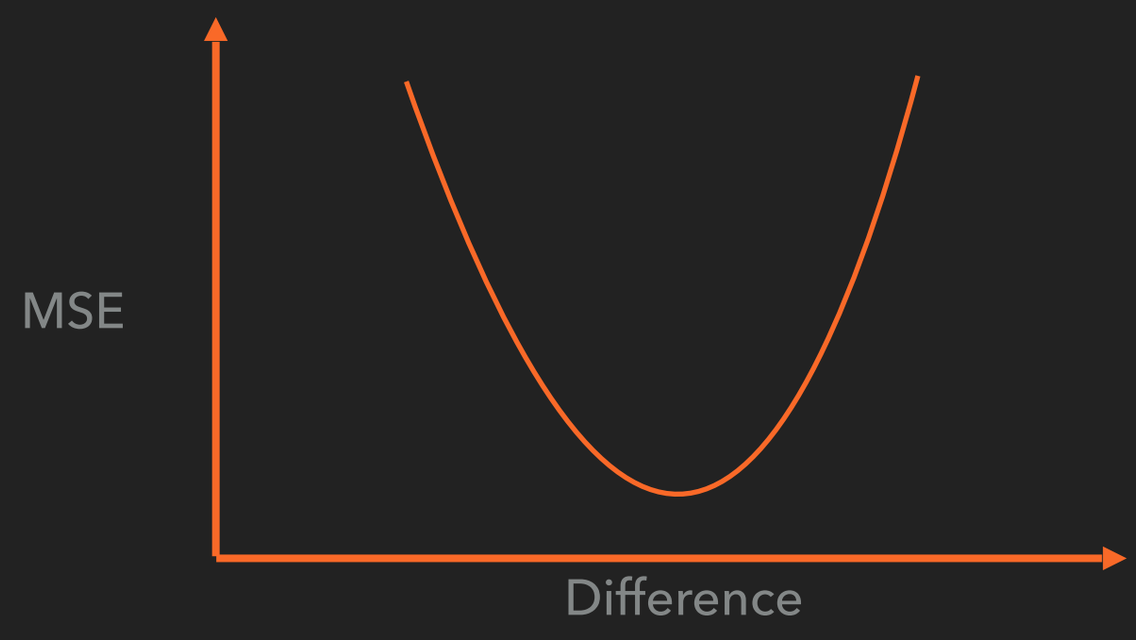
LOSS FUNCTIONS 3: MEAN SQUARED ERROR

- Used in regression
- Square endures a single minimum
- Avoids local minima trapping
- Easy to calculate

Prediction

$$MSE = \frac{1}{N}\sum(f_i - y_i)^2$$

Label

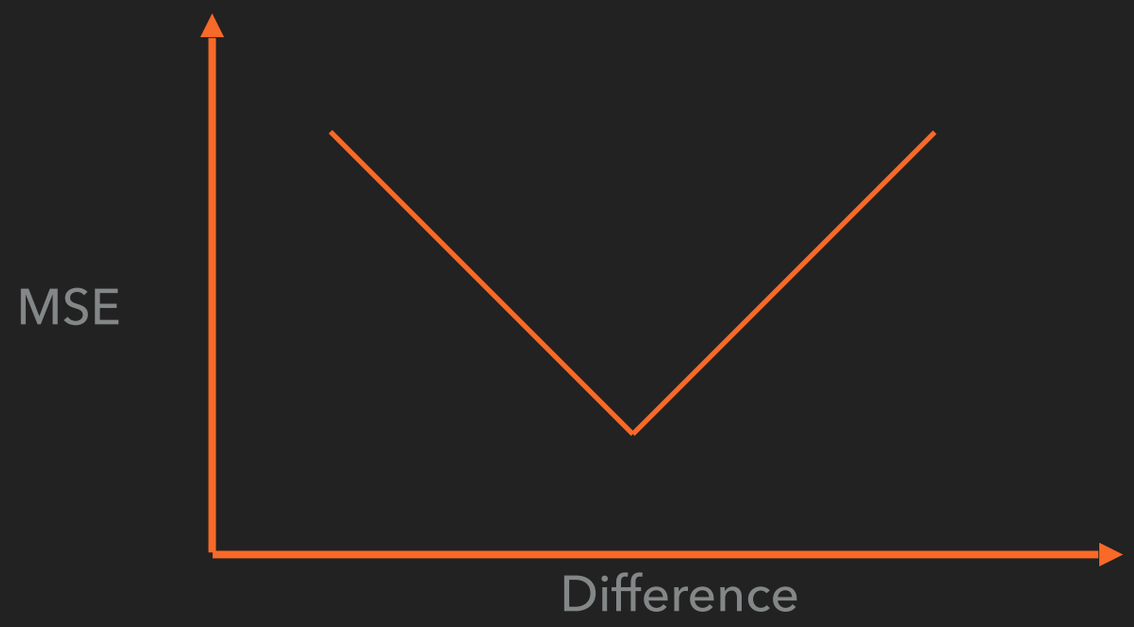


LOSS FUNCTIONS 4: MEAN ABSOLUTE ERROR

- Similar to MSE
- No quadric term
- More robust to outliers

$$MAE = \frac{1}{N}\sum|f_i - y_i|$$

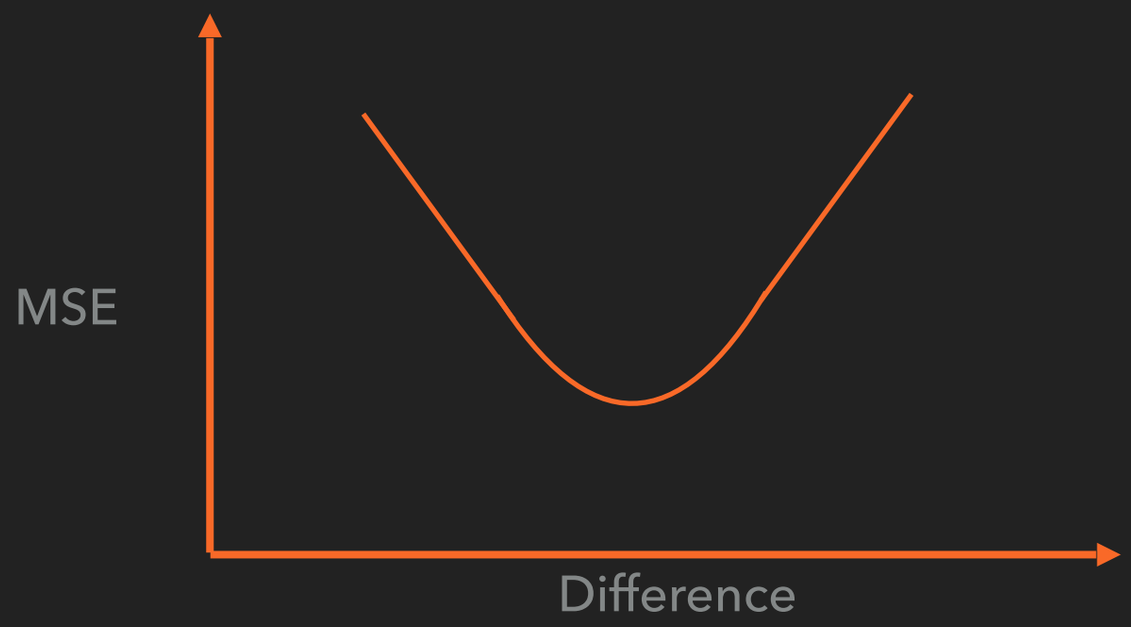
- MSE penalises large differences much more than MAE
- Large gradients close to zero - slow to optimise



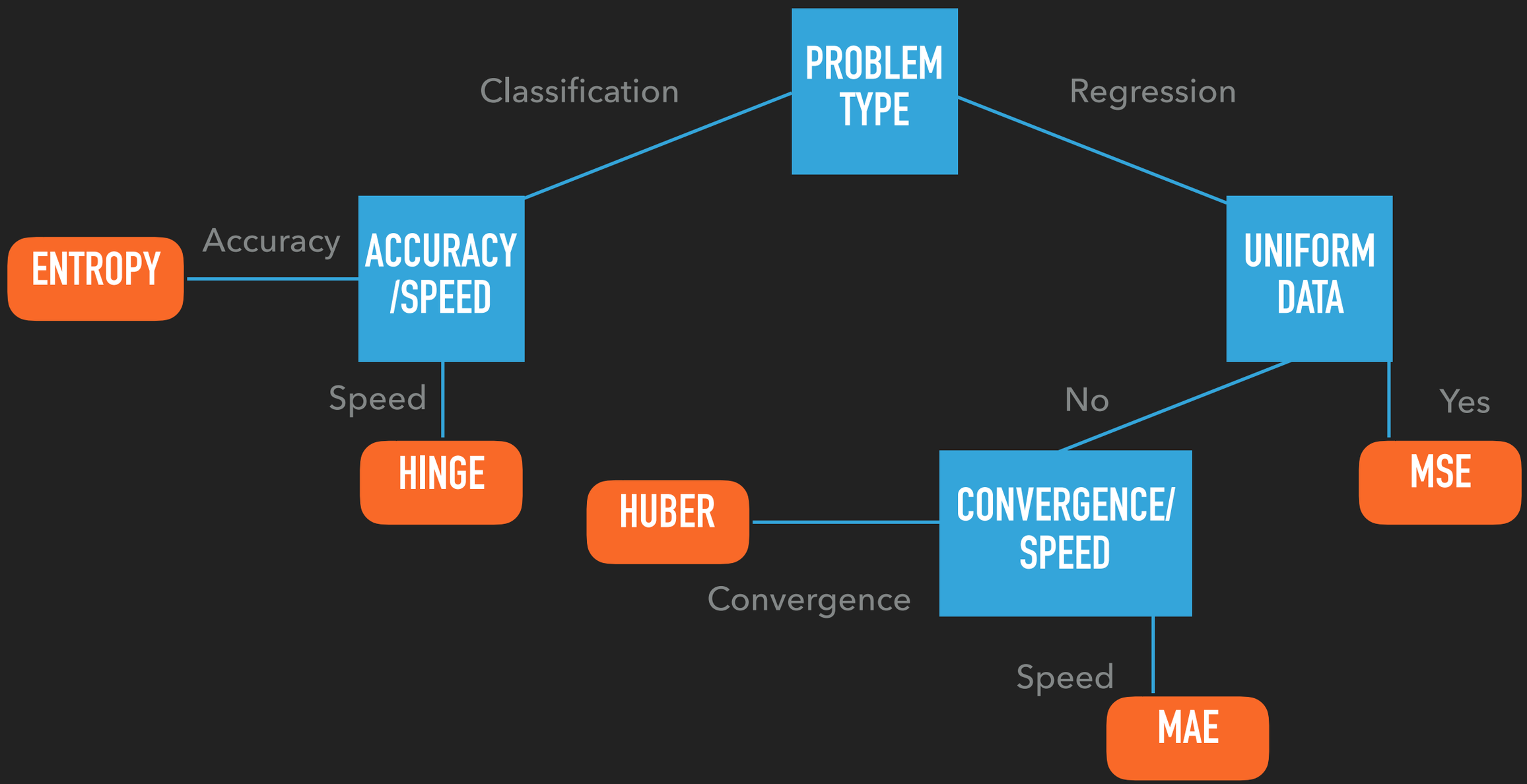
LOSS FUNCTIONS 5: HUBER LOSS

- Quadratic close to the minimum
- Linear far from the minimum
- Overcomes problems of MSE and MAE
- More expensive to calculate

$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{for} |y - f(x)| \le \delta, \\ \delta\,|y - f(x)| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$



CHOOSING A LOSS FUNCTION



LEARNING

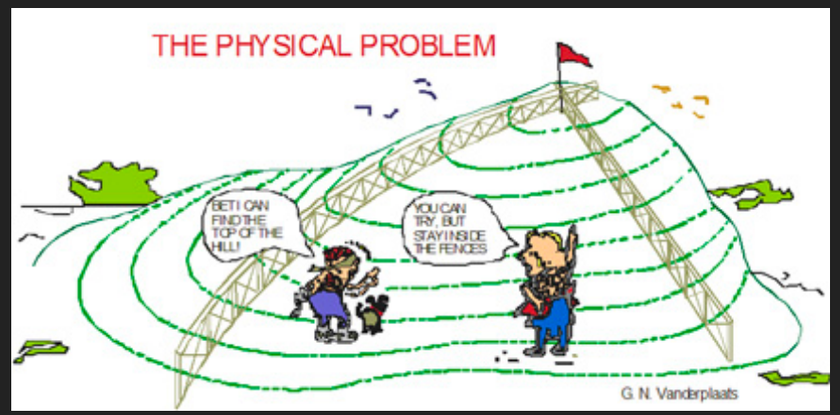
Learning =

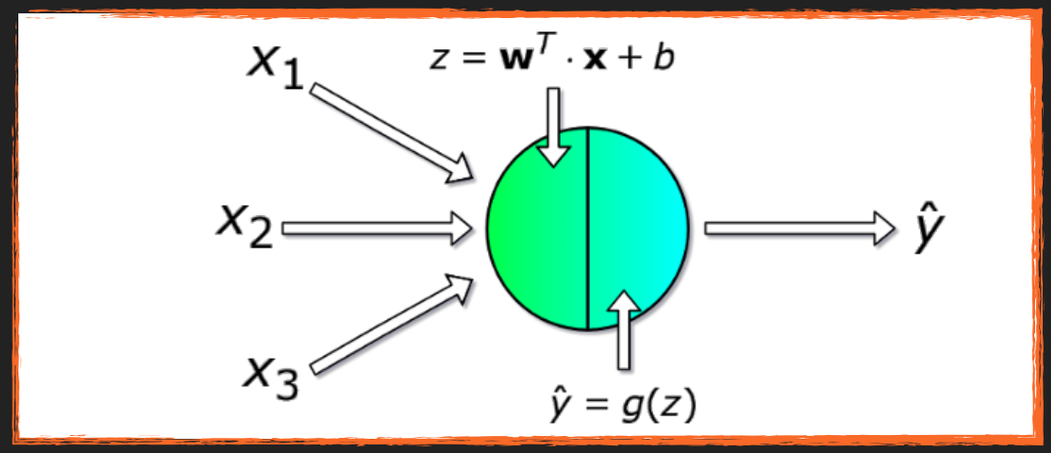
Representation + Evaluation + Optimisation

- Optimisation
  - Searches between classifiers.
  - Identifies the highest-scoring one.
  - Determines the efficiency of a learner.

OPTIMISERS

- Maximise/minimise an objective function
- In our case the loss function
- Updates the weight and biases



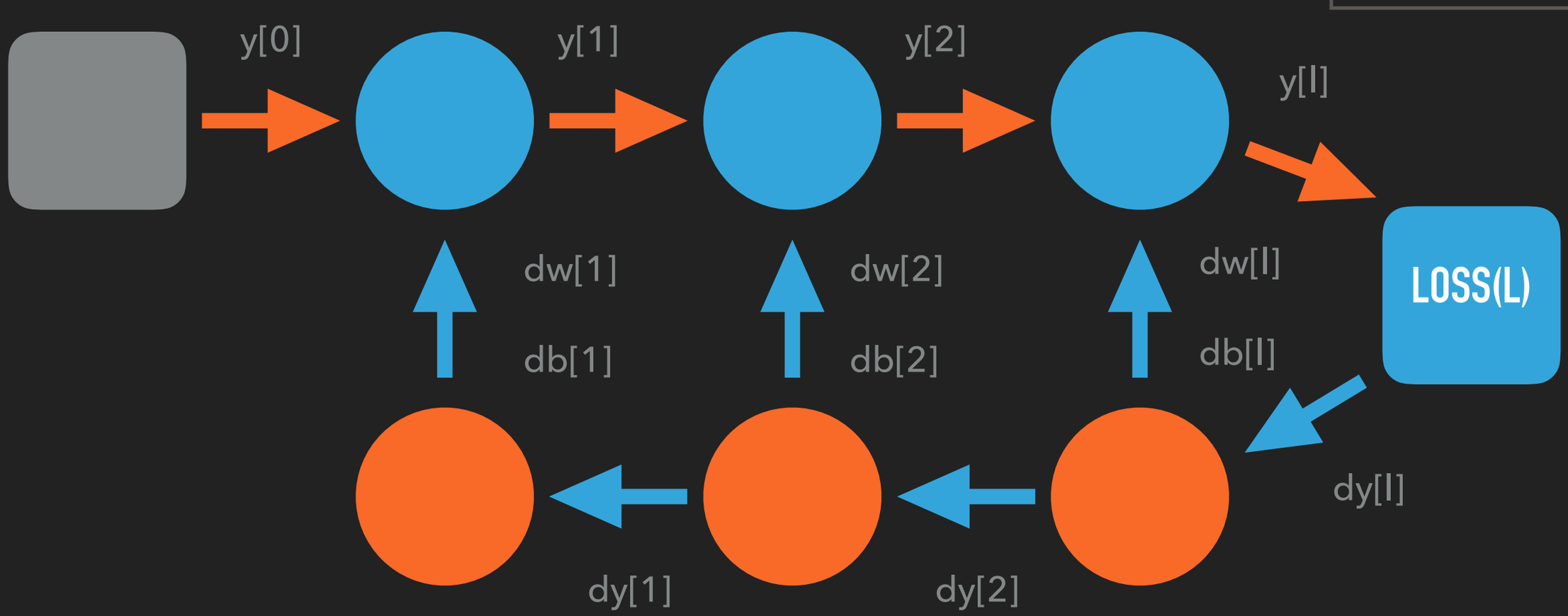


BACK PROPAGATION

- Learn from the loss

Notation:

dL/dy[l] => dy[l]



TYPES OF OPTIMISERS

- First order
  - Optimise with respect to the slope
  - Jacobin matrix

PRO: quick

CON: No curvature

- Second order
  - Use second order derivative to optimise
  - Hessian matrix

PRO: Curvature

CON: Slow

GRADIENT DESCENT

- Most common approach
- First order follow the gradient
- Calculate the loss on the full data set and then update the parameters
- Can be slow

$$\Theta_n = \Theta_{n-1} + \eta \nabla J(\Theta)$$

Gradient

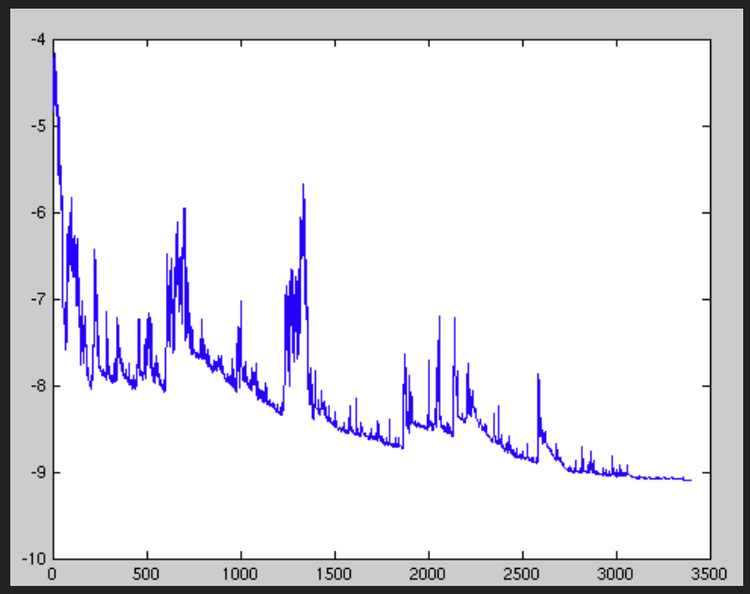
Parameters

Learning rate

Loss

STOCHASTIC/BATCH GRADIENT DESCENT

- Speed up gradient descent
- Calculate loss at each sample
- Quicker, but noisey
- Batch = middle ground, calculate loss at certain batch sizes (~50-256)
- Minibatch gradient descent very popular in NN training



Challenges: (i) choosing learning rate. (ii) single learning rate for all parameters. (iii) local minimum trapping.

MOMENTUM GRADIENT DESCENT

- Momentum
  - Include knowledge of previous update
    - Fewer oscillations, more stable
- Nesterov accelerated gradient
  - Also looks ahead

Momentum term

$$V_n = \gamma V_{n-1} + \eta \nabla J(\Theta)$$

$$\Theta_n = \Theta_{n-1} - V_n$$

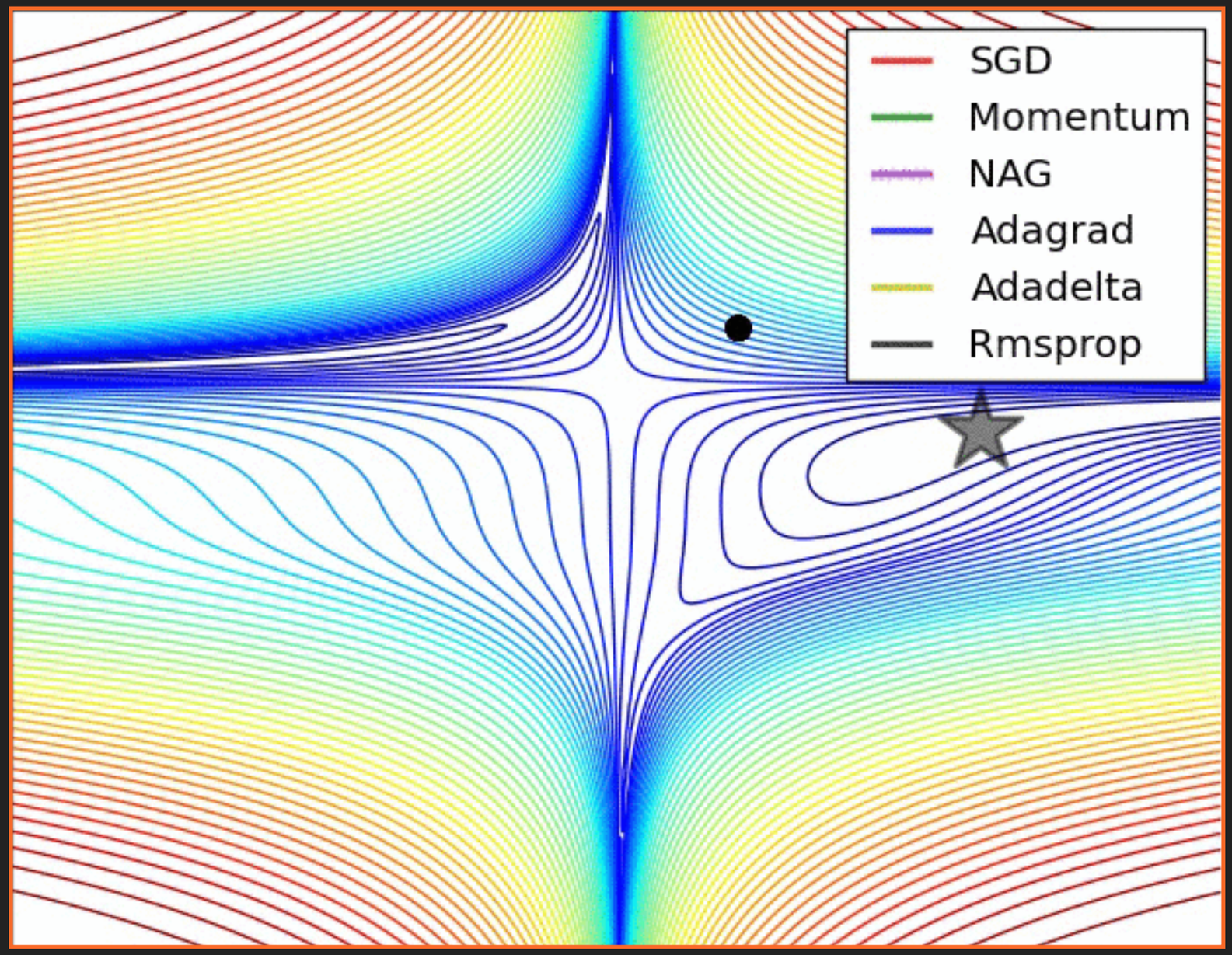
Parameters

Update

ADAPTIVE MOMENTUM

- Allows the learning rate to adjust for parameters
- Small updates for frequent parameters, large updates for sparse parameters
- Adagrad, Adadelta, Adam
- Adam is becoming the most popular method for NN optimisation

COMPARE



https://towardsdatascience.com/types-of-optimization-algorithms-used-in-neural-networks-and-ways-to-optimize-gradient-95ae5d39529f

USEFUL LINKS

- Blog on types of machine learning: https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/
- Open source reading on many ML issues: https://distill.pub/
- Information about back-propagation: https://www.youtube.com/watch?v=Ilg3gGewQ5U
- More on CNNs: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

SUMMARY

- Understanding your problem before diving in is critical
- Understand your data
- Traditional methods work well on well structured and characterised datasets
- CNNs are useful for analysis of patterns in visual data
- LSTMs are state of the art for time series data
- Many packages exist to assist with implementation
- Benchmarks are going to be important!

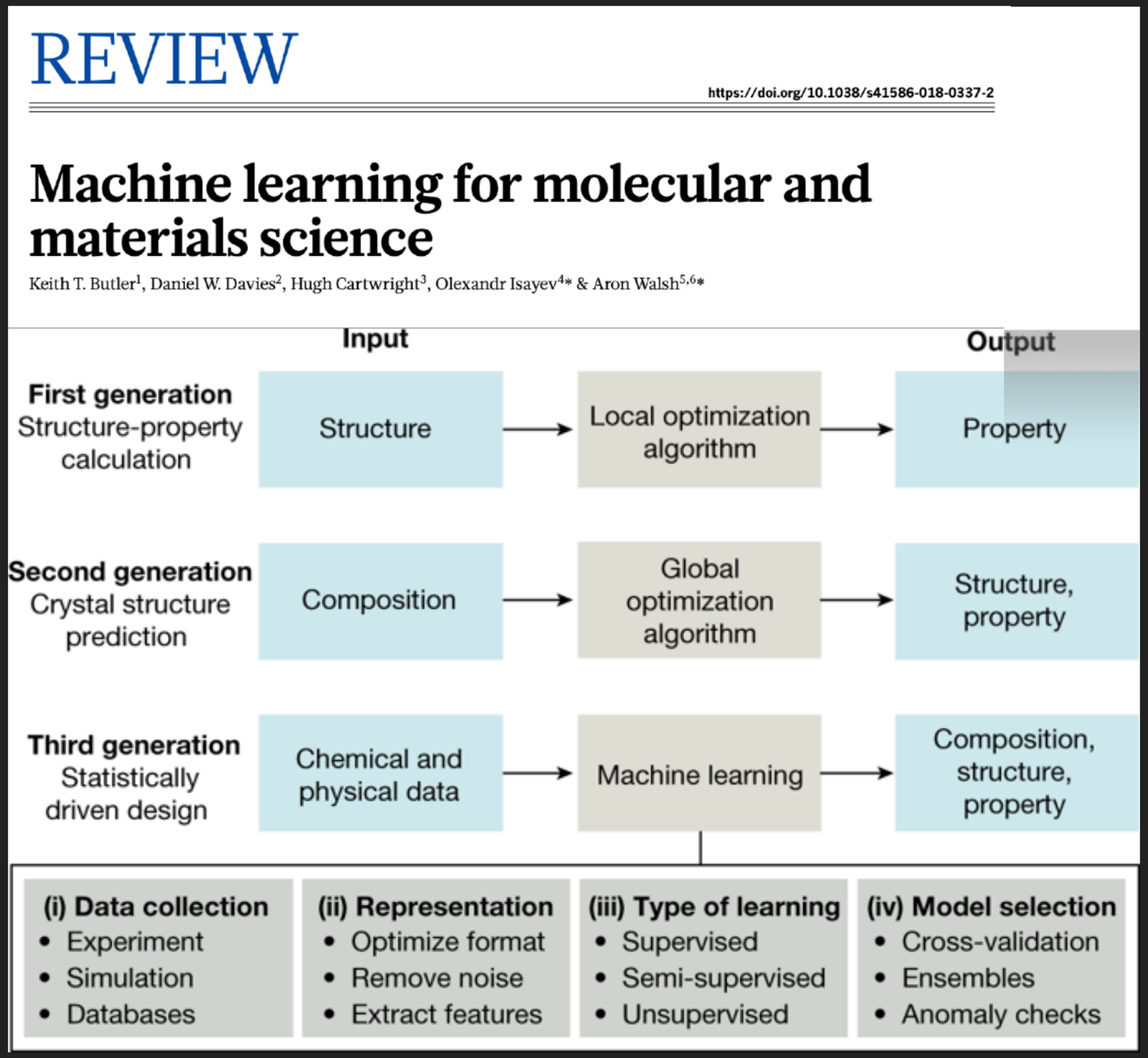
ACKNOWLEDGMENTS

- Tony Hey, Jeyan Thiyagalingam, Rebecca Mackenzie, Sam Jackson (SciML)
- Aron Walsh, Daniel Davies (Imperial College London)
- Toby Perring, Duc Le (ISIS Neutron and Muon Source)
- Gareth Nisbet, Steve Collins (Diamond Light Source)
- Alex Leung, Peter Lee (Research Complex at Harwell, UCL)

The Alan Turing Institute

Science & Technology Facilities Council | UK Research and Innovation

THANK YOU



REVIEW

https://doi.org/10.1038/s41586-018-0337-2

Machine learning for molecular and materials science

Keith T. Butler[1], Daniel W. Davies[2], Hugh Cartwright[3], Olexandr Isayev[4]* & Aron Walsh[5,6]*

	Input		Output
First generation Structure-property calculation	Structure	Local optimization algorithm	Property
Second generation Crystal structure prediction	Composition	Global optimization algorithm	Structure, property
Third generation Statistically driven design	Chemical and physical data	Machine learning	Composition, structure, property

(i) Data collection	(ii) Representation	(iii) Type of learning	(iv) Model selection
• Experiment • Simulation • Databases	• Optimize format • Remove noise • Extract features	• Supervised • Semi-supervised • Unsupervised	• Cross-validation • Ensembles • Anomaly checks

NATURE, 2018, 559, 547.

@keeeto2000

@ml_sci

keeeto.github.io

www.scd.stfc.ac.uk/Pages/Scientific-Machine-Learning.aspx