IFN509

Data Exploration and Mining

Week 11
Algorithms of Predictive Data Mining
Neural Networks

Professor Richi Nayak
r.nayak@qut.edu.au

School of Computer Science & Centre for Data Science
Faculty of Science
https://research.qut.edu.au/adm

Learning Objectives: Week 11

- Predictive Modelling Algorithms
  - Neural Networks Classification
    - Define a neural network.
    - Neural Network Architecture
    - Activation Functions
    - Training Process – Weight Optimization
    - Quick Intro: Deep Learning
  - Case-based reasoning classification
    - Nearest Neighbours: Classification

What Should You Do in Week 11?

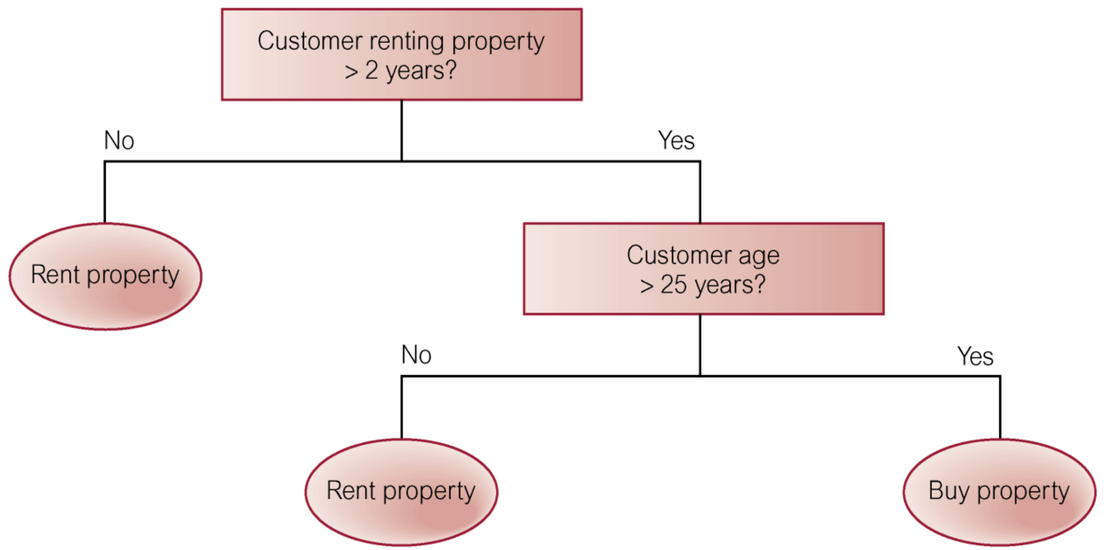
- Listen to the lecture recording and review the lecture slides on Neural Networks.
- Tutorial: Attempt the exercise questions related to the lecture on Regression
- Practical: Complete practical tasks on Regression
- Consult the Lecturer or Tutor if you have any questions related to the subject.
- Assessment Item 2
  - Team registration should have been finalised
  - Association mining: Should have finished
  - Clustering: Should have finished
  - Decision Tree: Should have finished
  - Regression: Should have been attempted

Neural Networks

Architecture

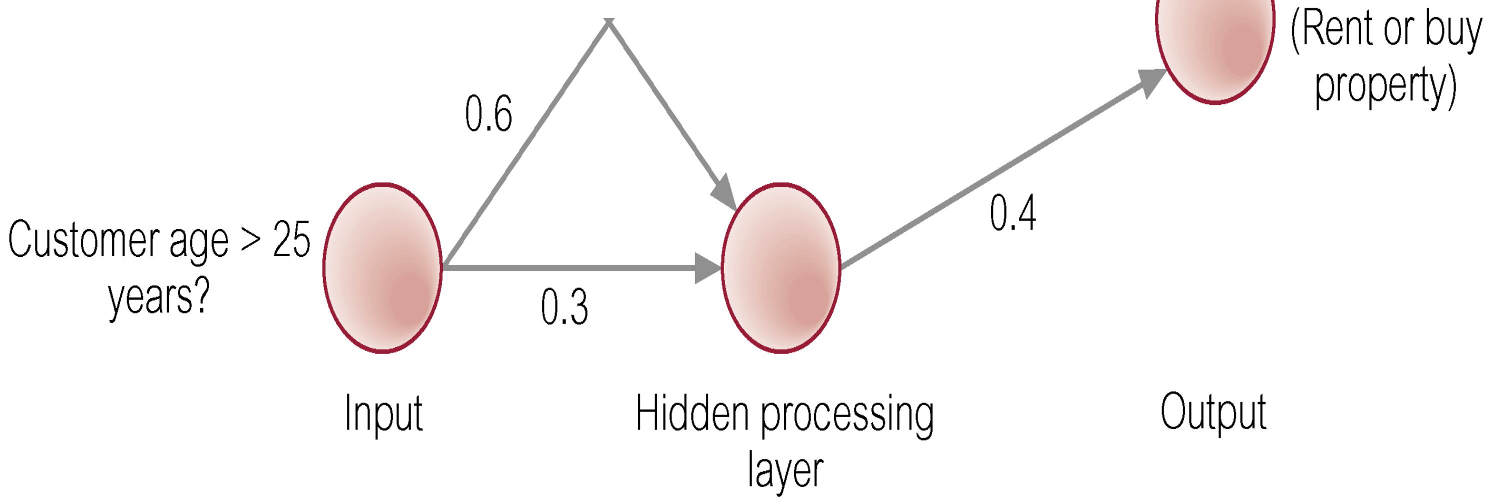
Activation Function

Example of Classification using Neural Networks

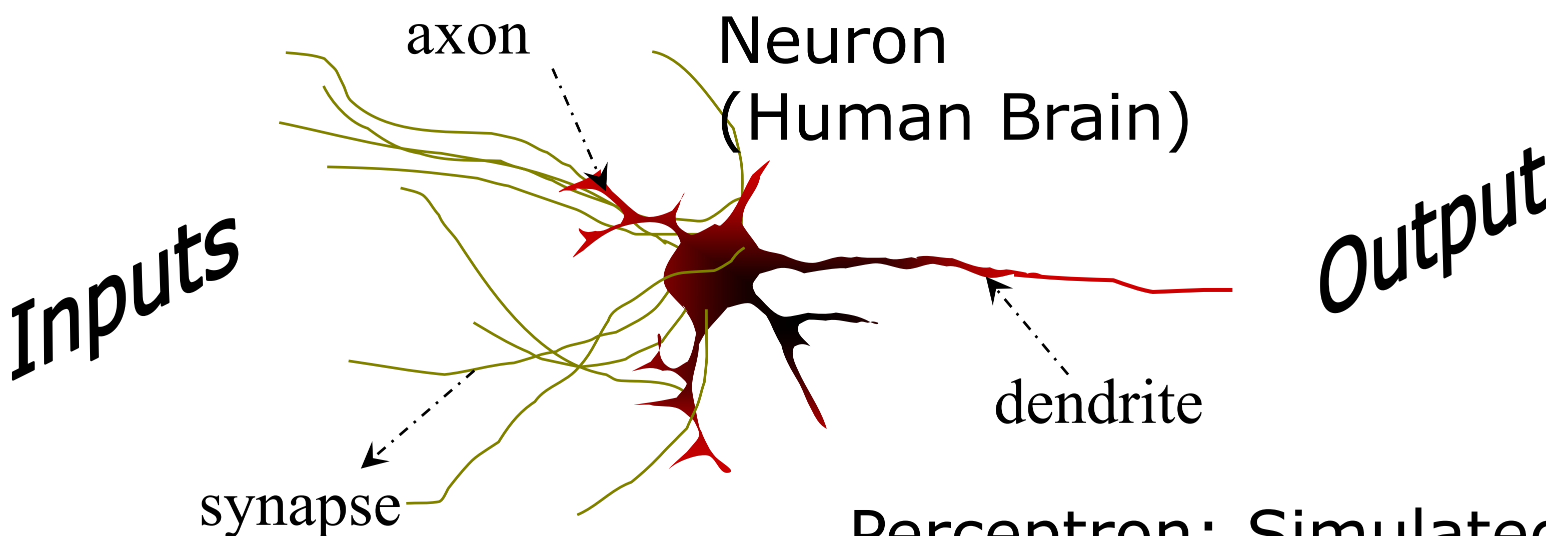


Inputs = Duration of renting,
Age and many othe
Target = Rent or Buy

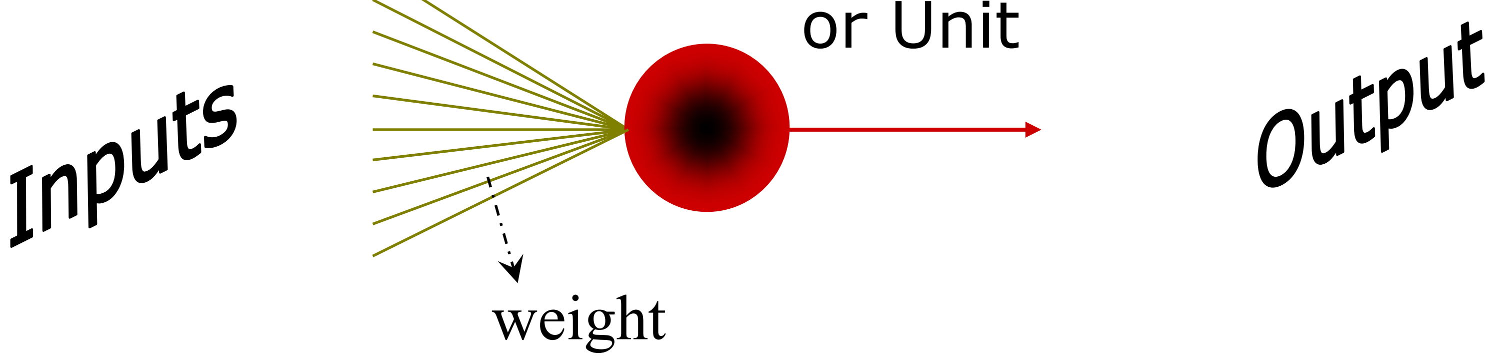
Root Node: Rent Property
Internal Node: Age
Decision Nodes: Rent or Buy



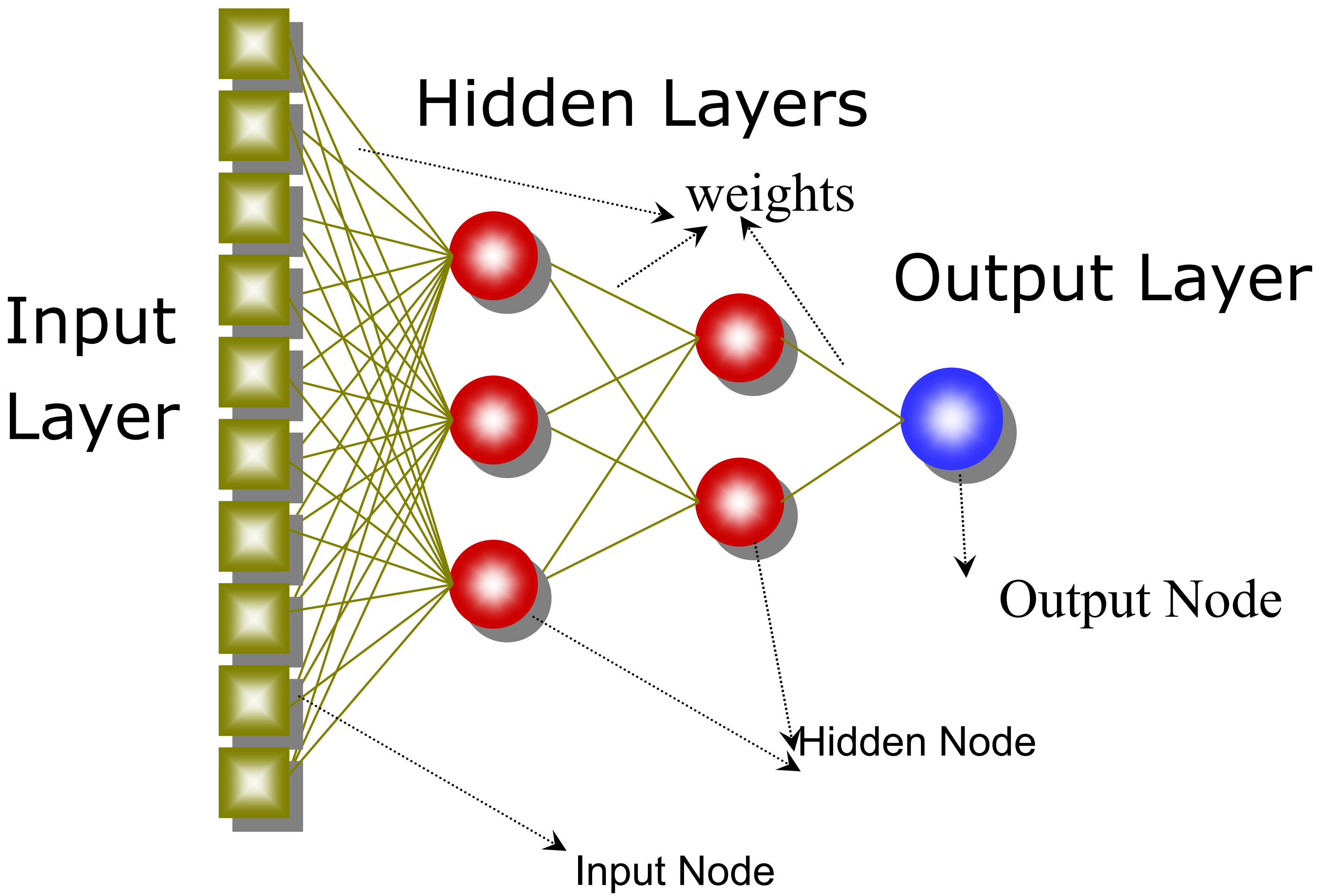
Artificial Neural Networks



Perceptron: Simulated Neural Network Node or Unit



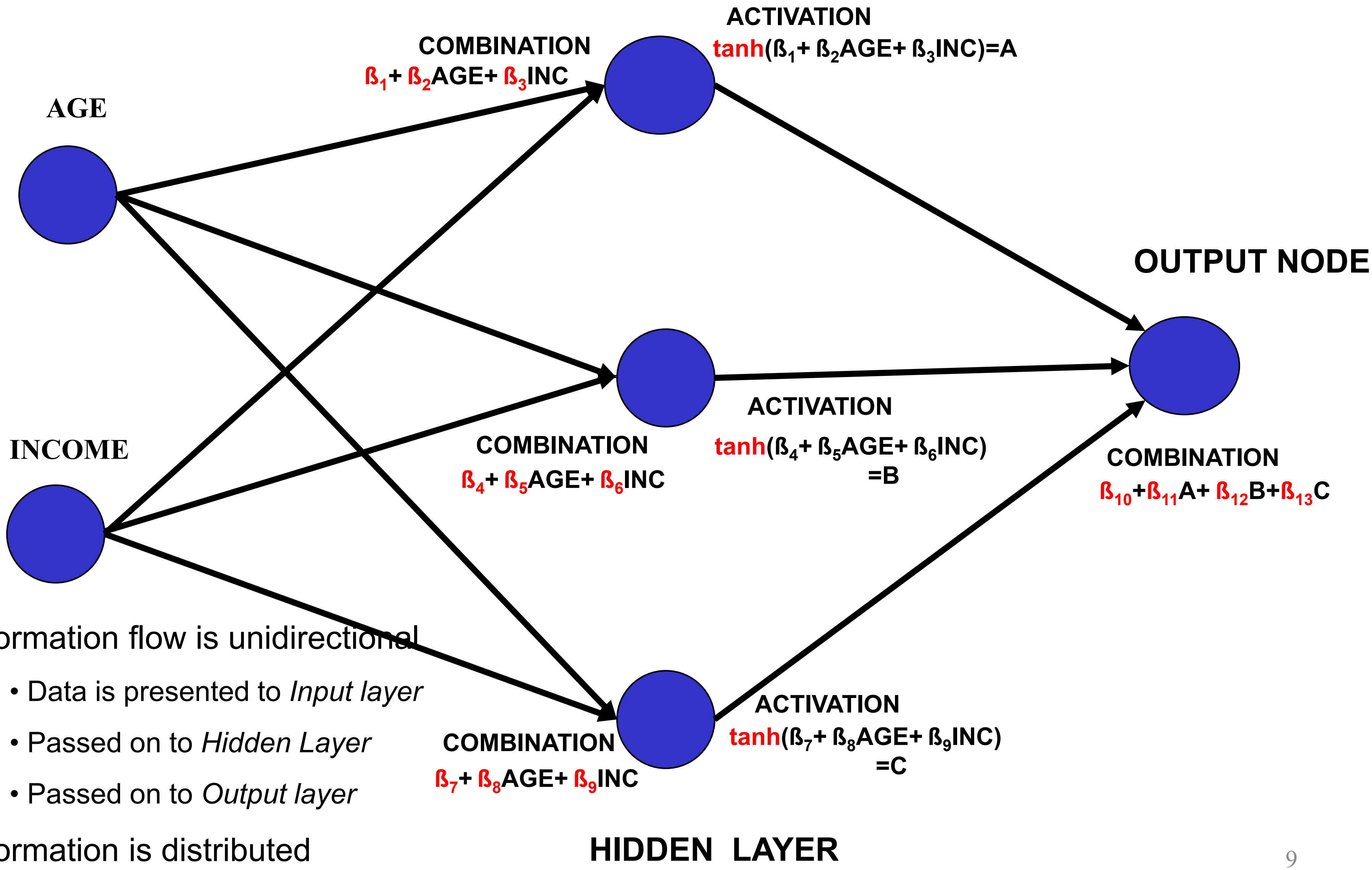
Multilayer Perceptron



Neural Nets Topology or Architecture

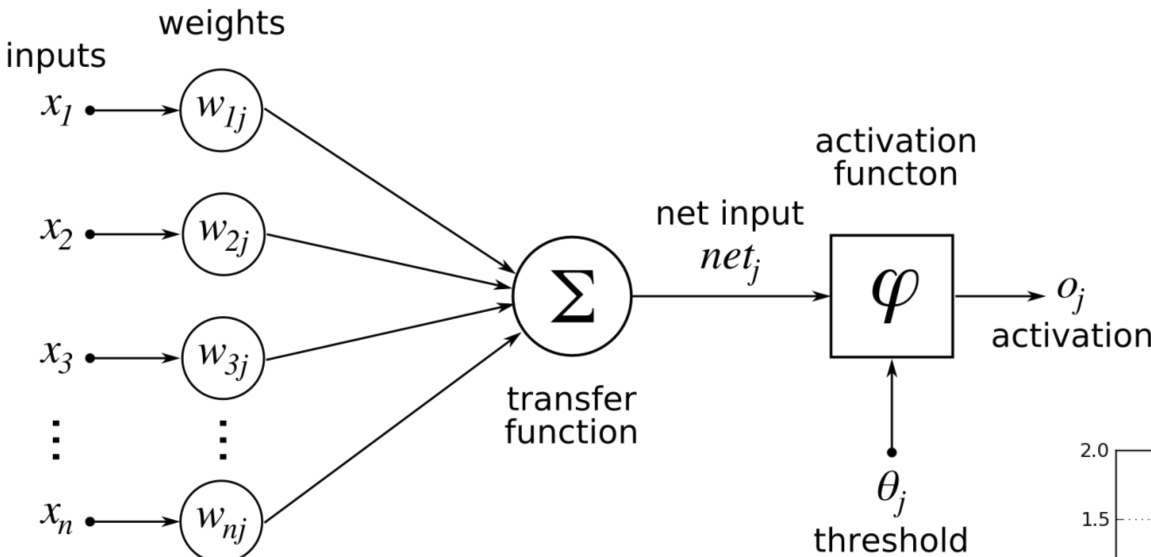
- Neural network techniques in general do not restrict the number of output nodes.
- There can be multiple outputs representing multiple simultaneous predictions (each prediction referring to a target attribute)
  - This is one way that neural nets differ from most other predictive techniques.
- The number of hidden nodes and layers is often increased with the number of inputs and the complexity of the problem.
- Too many hidden nodes can lead to overfitting and too few can result in models with underfitting (or poor accuracy).

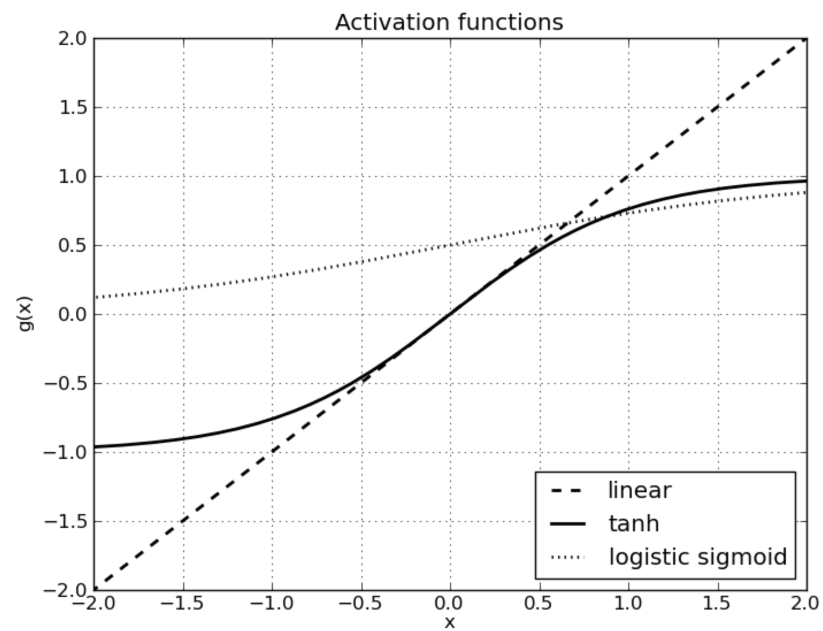
Information Processing



- Information flow is unidirectional
  - Data is presented to *Input layer*
  - Passed on to *Hidden Layer*
  - Passed on to *Output layer*
- Information is distributed
- Information processing is parallel

Activation Function

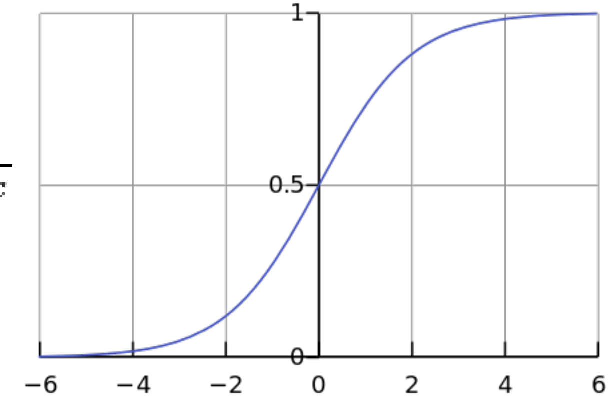


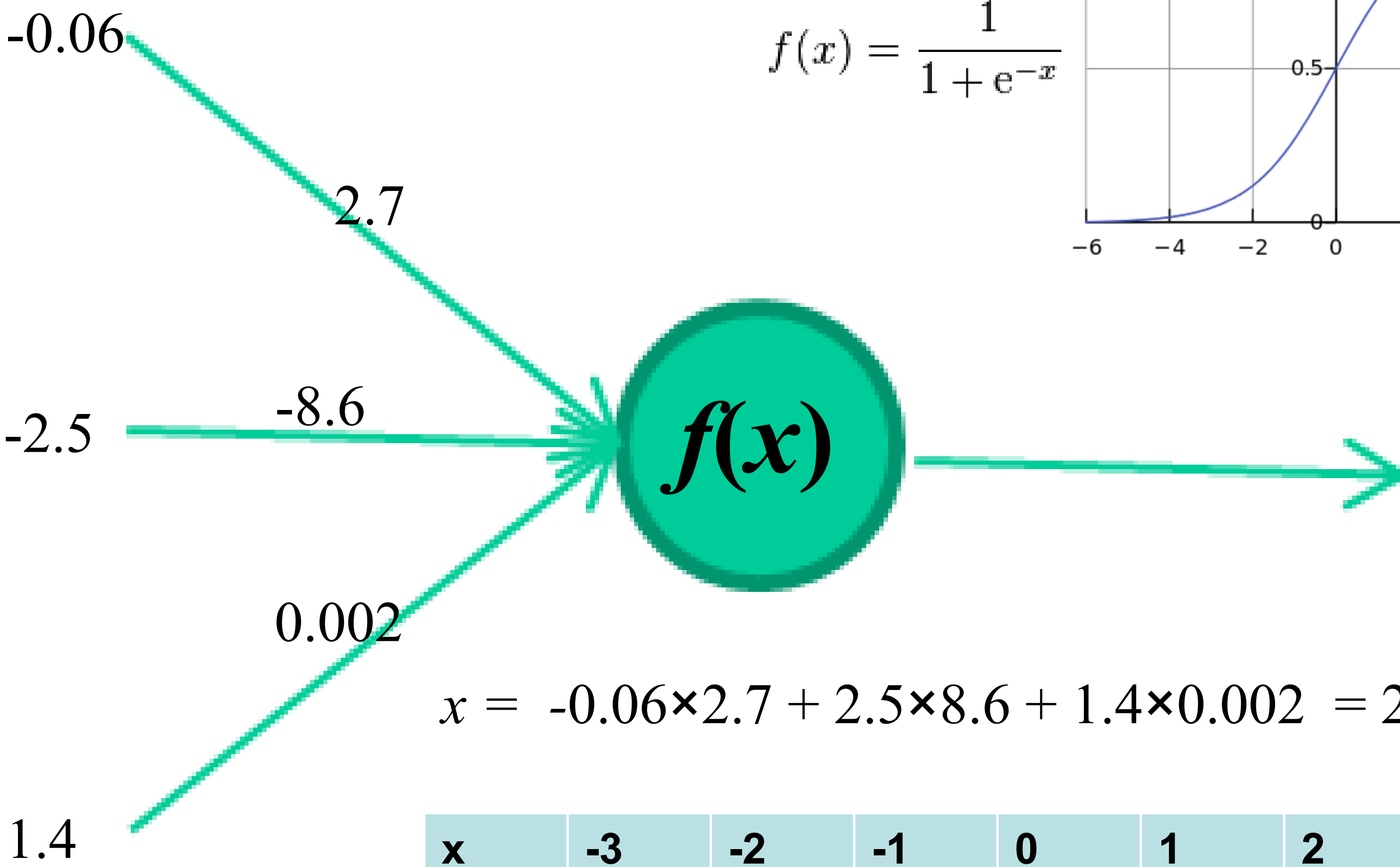


@https://en.wikibooks.org/wiki/Artificial_Neural_Networks/Activation_Functions

An Example: Logistic (sigmoid)

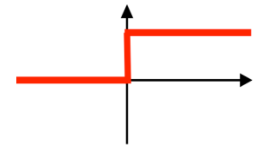
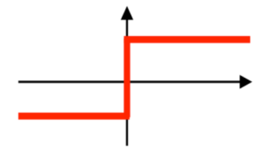
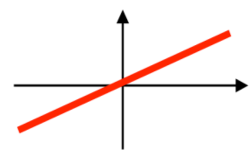
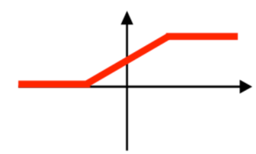
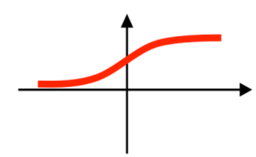
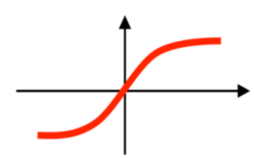
$$f(x) = \frac{1}{1+e^{-x}}$$





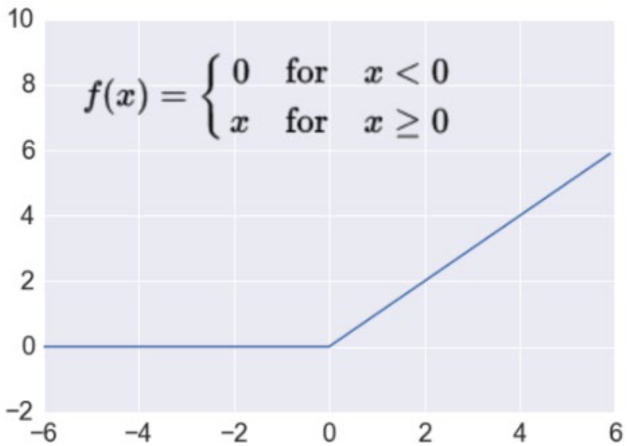
$$x = \text{-}0.06\times 2.7 + 2.5\times 8.6 + 1.4\times 0.002 = 21.34$$

x	**-3**	**-2**	**-1**	**0**	**1**	**2**	**3**
e^x	0.05	0.14	-.37	1	2.72	7.39	20.09

Activation function	Equation	Example	1D Graph
Unit step (Heaviside)	$\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Sign (Signum)	$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Linear	$\phi(z) = z$	Adaline, linear regression	
Piece-wise linear	$\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z) = \dfrac{1}{1 + e^{-z}}$	Logistic regression, Multi-layer NN	
Hyperbolic tangent	$\phi(z) = \dfrac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$	Multi-layer NN	

@http://pypr.sourceforge.net/ann.html

Activation: ReLU



http://adilmoujahid.com/images/activation.png

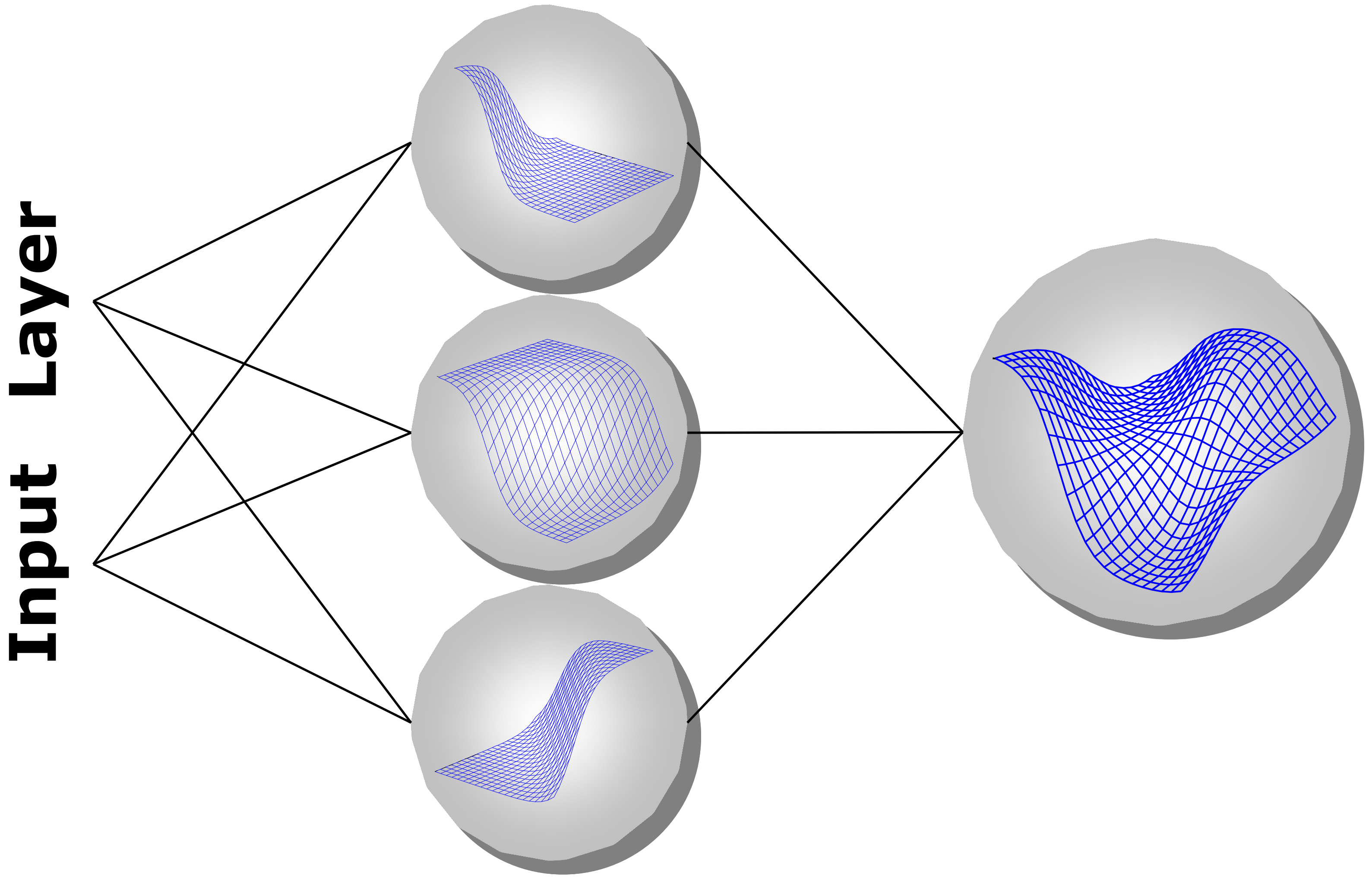
Takes a real-valued number and thresholds it at zero

$$R^n \rightarrow R_+^n$$

$$\mathrm{f}(x) = \max(0, x)$$

- Most Deep Networks use ReLU nowadays
- Less expensive operations
  - compared to sigmoid/tanh (exponentials etc.)
  - implemented by simply thresholding a matrix at zero
- More **expressive** than some other functions
- Trains much **faster**
  - accelerates the convergence of SGD – an optimization algorithm due to linear, non-saturating form

Training: Activation Function



Neural Networks

Training: Weight Optimization

Neural Networks Training (1)

- Finding the best combination of weights is a significant search problem.
- A number of techniques have been used to search for the best combination of weights.
  - The most common is a class of algorithms called gradient descent.
- A gradient descent algorithm starts with a solution (i.e a set of weights that have been randomly generated).
- Then an instance from the training set is presented to the neural network.
- The network (initially with random weights) is used to compute an output, the output is compared to the desired result (i.e. the value of the target attribute),
  - the difference, called the error, is computed.
- The weights are then altered so that if the same instance were presented again, the error would be less. This gradual reduction in error is the descent.

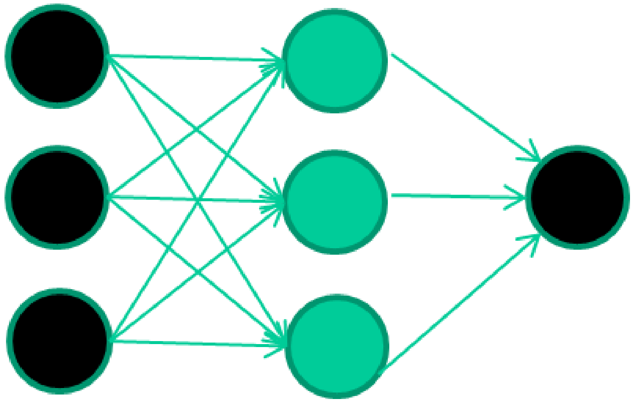
Neural Networks Training (2)

- The cycle is repeated for each instance in the training set, with small adjustments being made in the weights after each instance.
- When the entire training set has been processed, the process is repeated again.
- Each run through the entire training set is called an epoch.
- An ANN is usually trained on many epochs.
- Neural net algorithms use a number of different stopping rules to control when training ends.
- Common stopping rules include:
  - Stop after a specified number of epochs.
  - Stop when an error measure falls below a preset level.
  - Stop when the error measure has seen no improvement over a certain number of epochs.

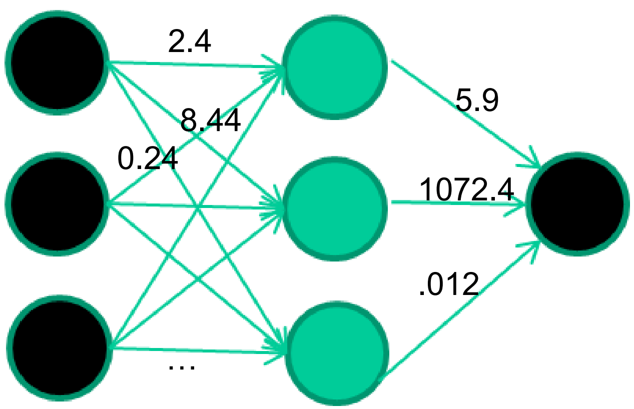
Neural Network Training: An Example

A dataset

Attributes			***Target/class***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0
etc …			



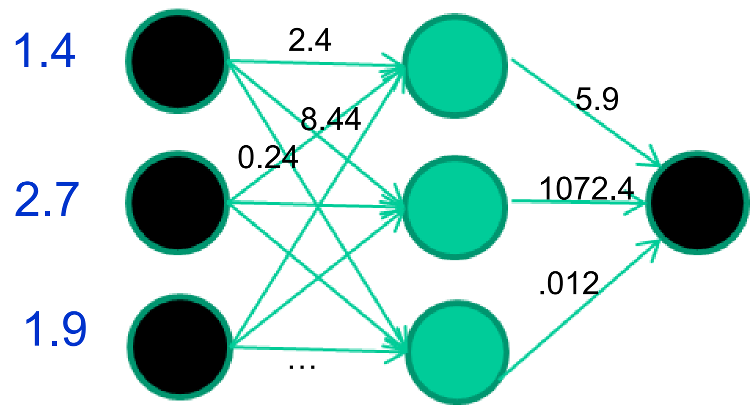
Initialise the network with random weights



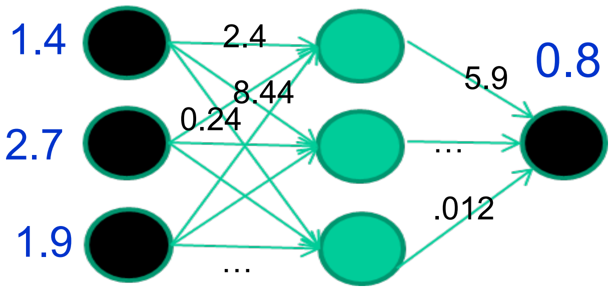
Present a training sample to the network

Training data

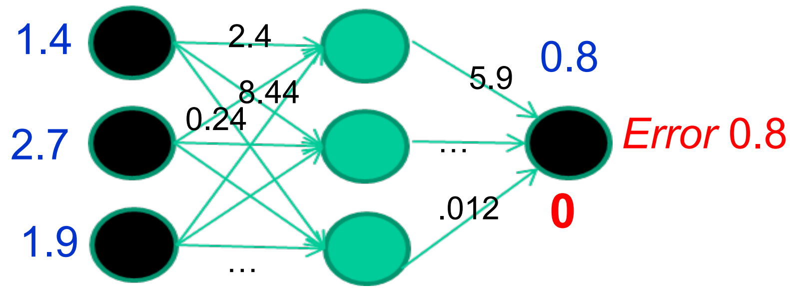
Attributes			***class***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0
etc …			



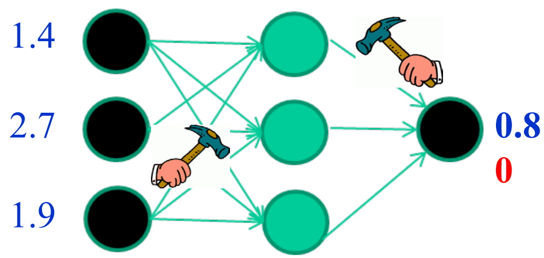
Feed it through the network to calculate output



Compare the estimated (or predicted) output with the expected (or target) output & calculate Error



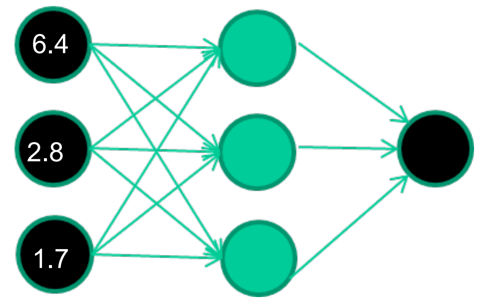
Adjust weights based on error



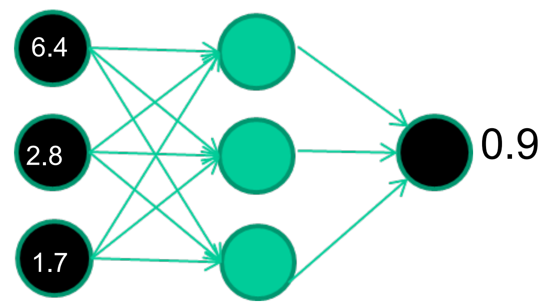
Present another training sample

Training data

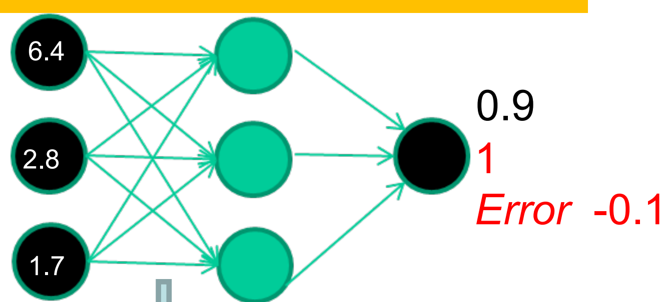
Fields			***class***
1.4	2.7	1.9	0
3.8	3.4	3.2	0
6.4	2.8	1.7	1
4.1	0.1	0.2	0
etc …			



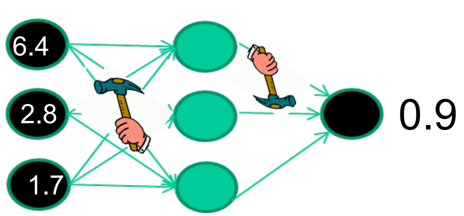
Feed the sample inputs through the network to calculate predicted output



Compare the estimated and expected outputs



Adjust weights based on error

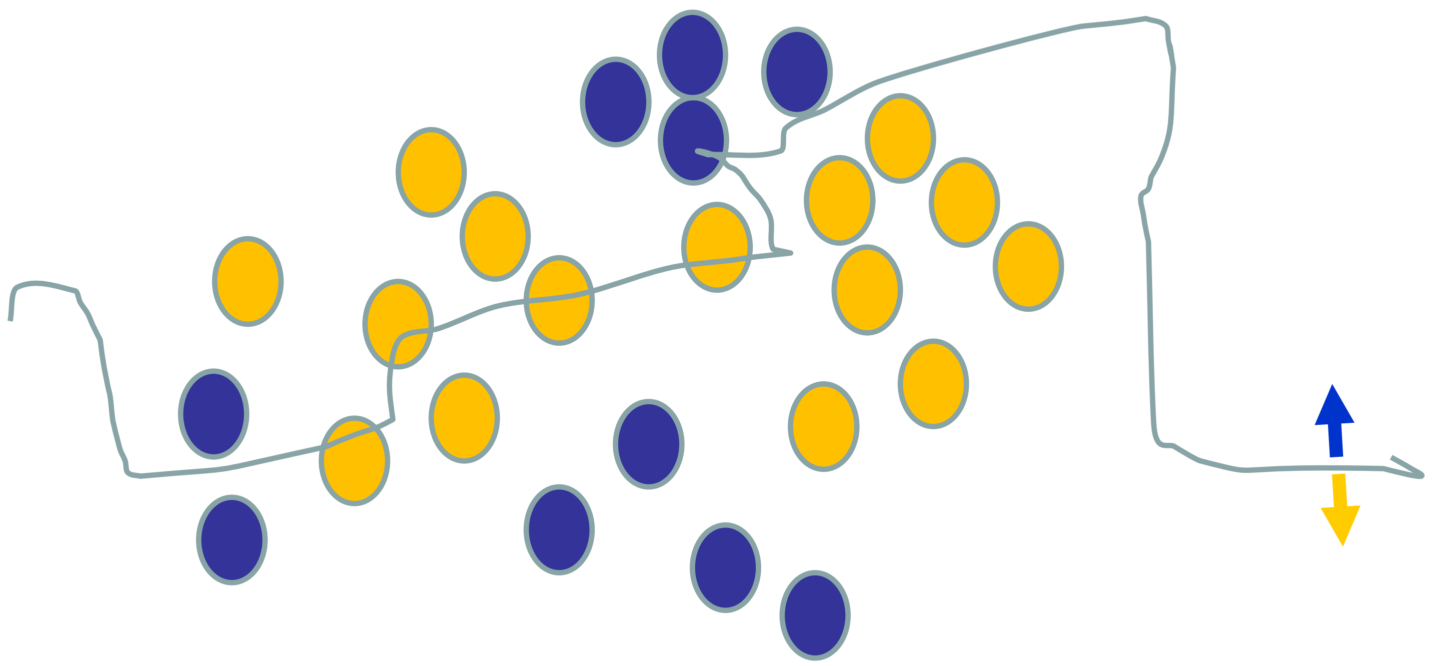


Repeat this process thousands maybe millions of times – each time taking a random training instance, and making weight adjustments

Algorithms for weight adjustment are designed to make changes that will reduce the error

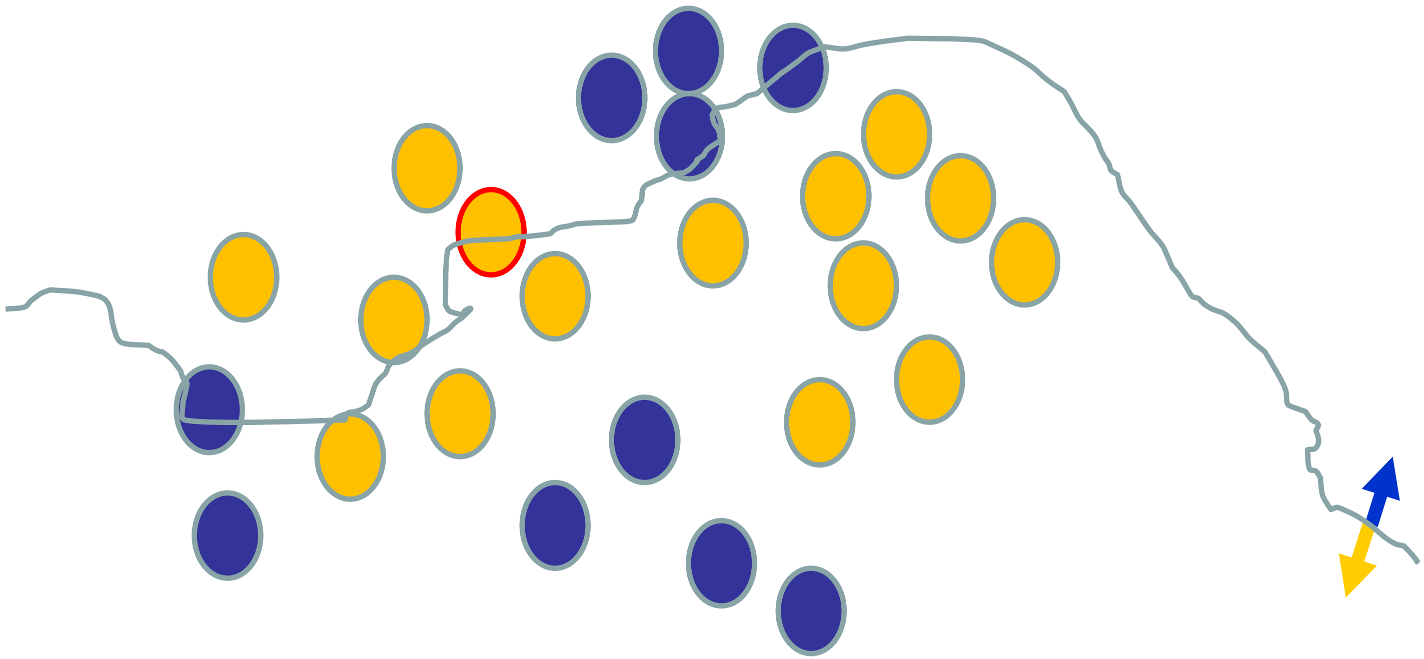
The decision boundary perspective…

Initial random weights



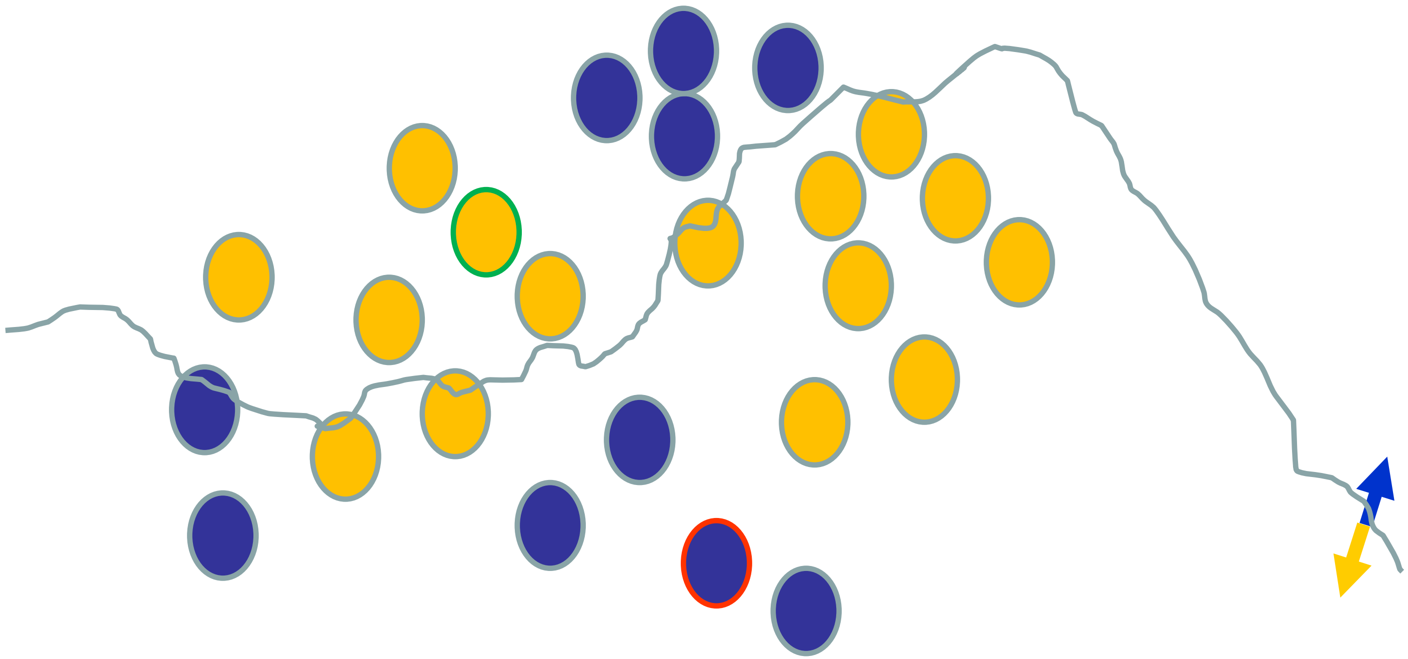
The decision boundary perspective…

Present a training instance / adjust the weights



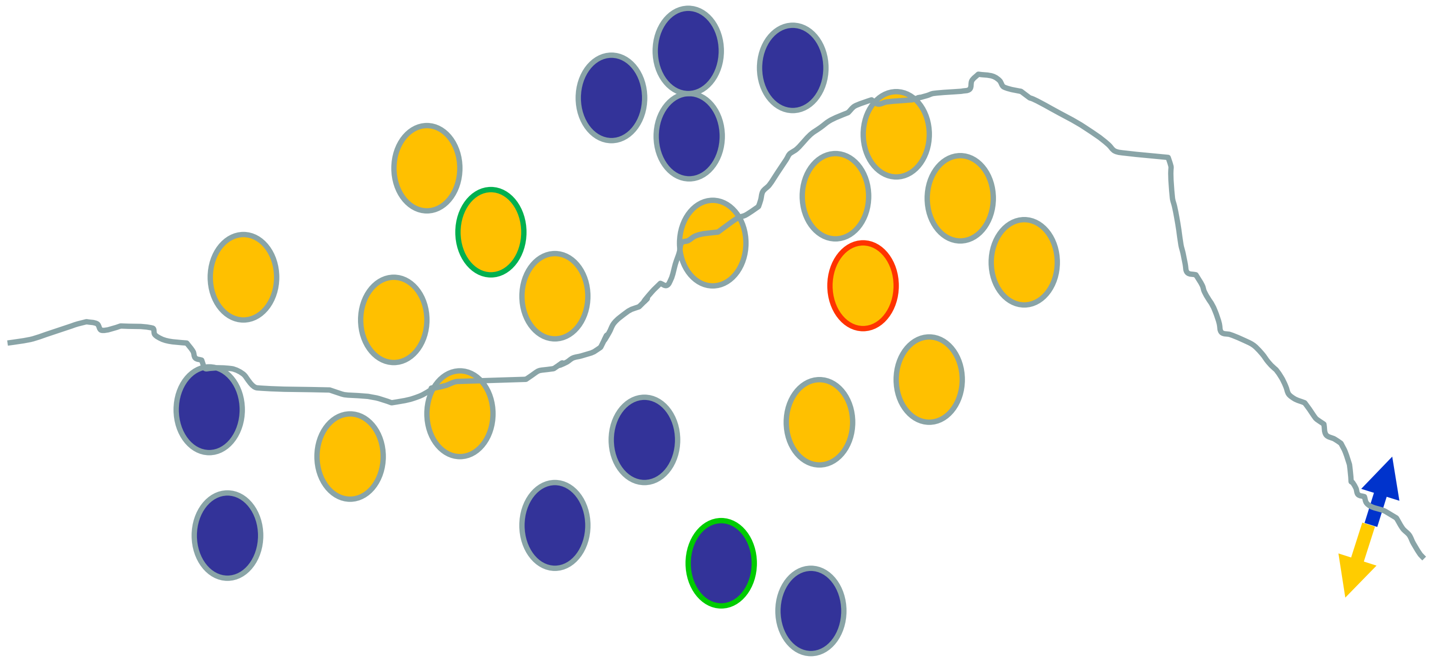
The decision boundary perspective…

Present another training instance / adjust the weights



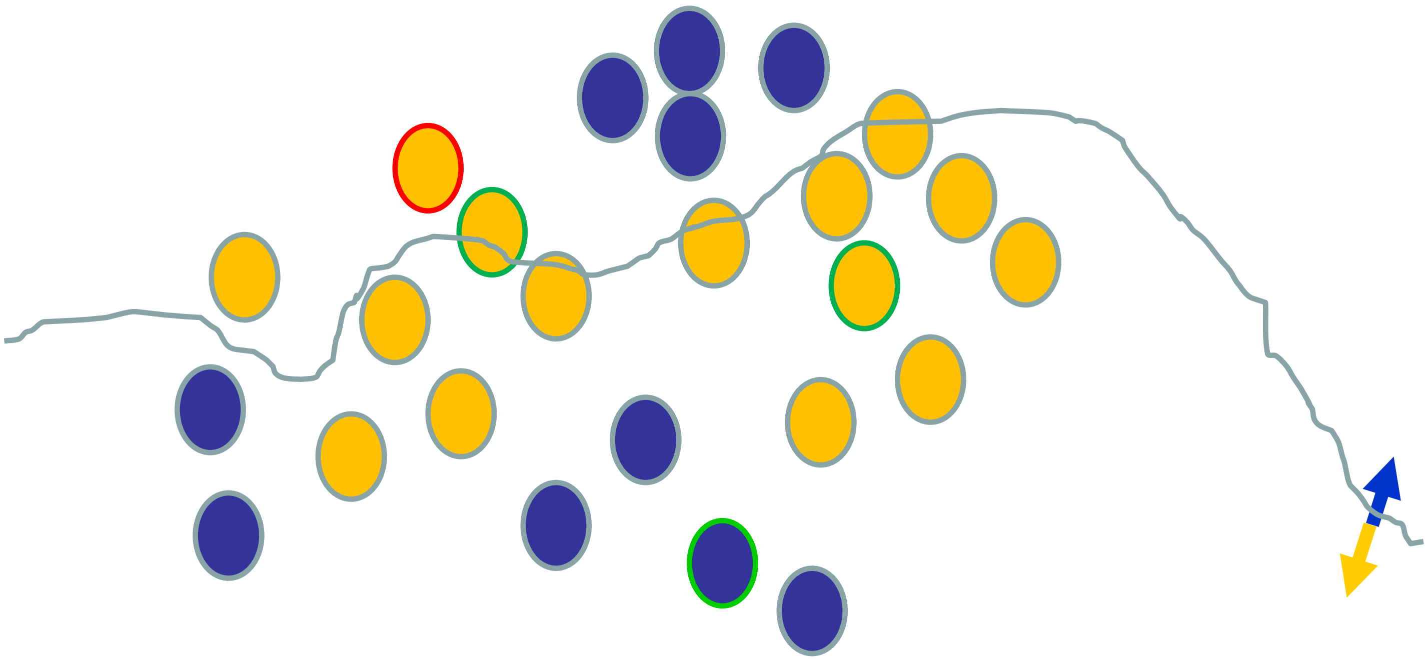
The decision boundary perspective…

Present another training instance / adjust the weights



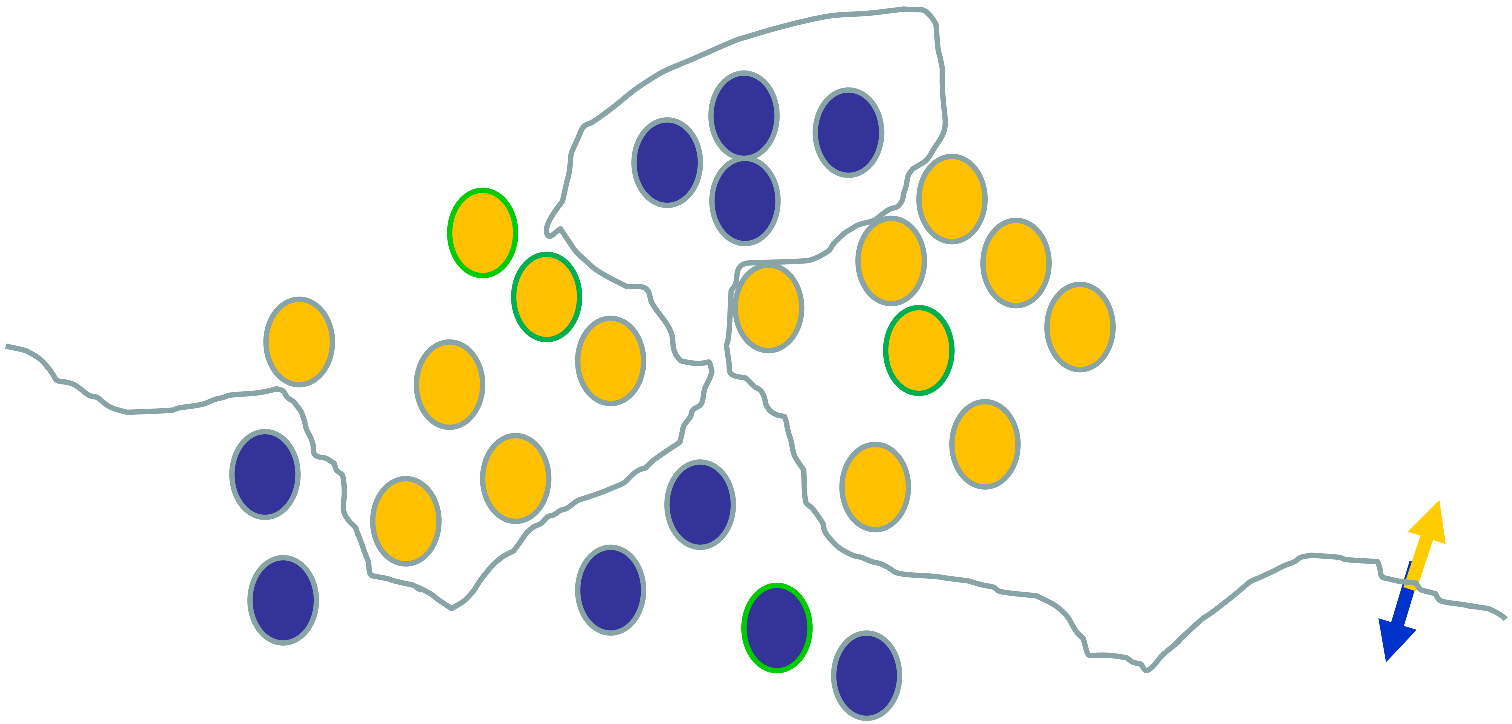
The decision boundary perspective…

Present another training instance / adjust the weights



The decision boundary perspective…

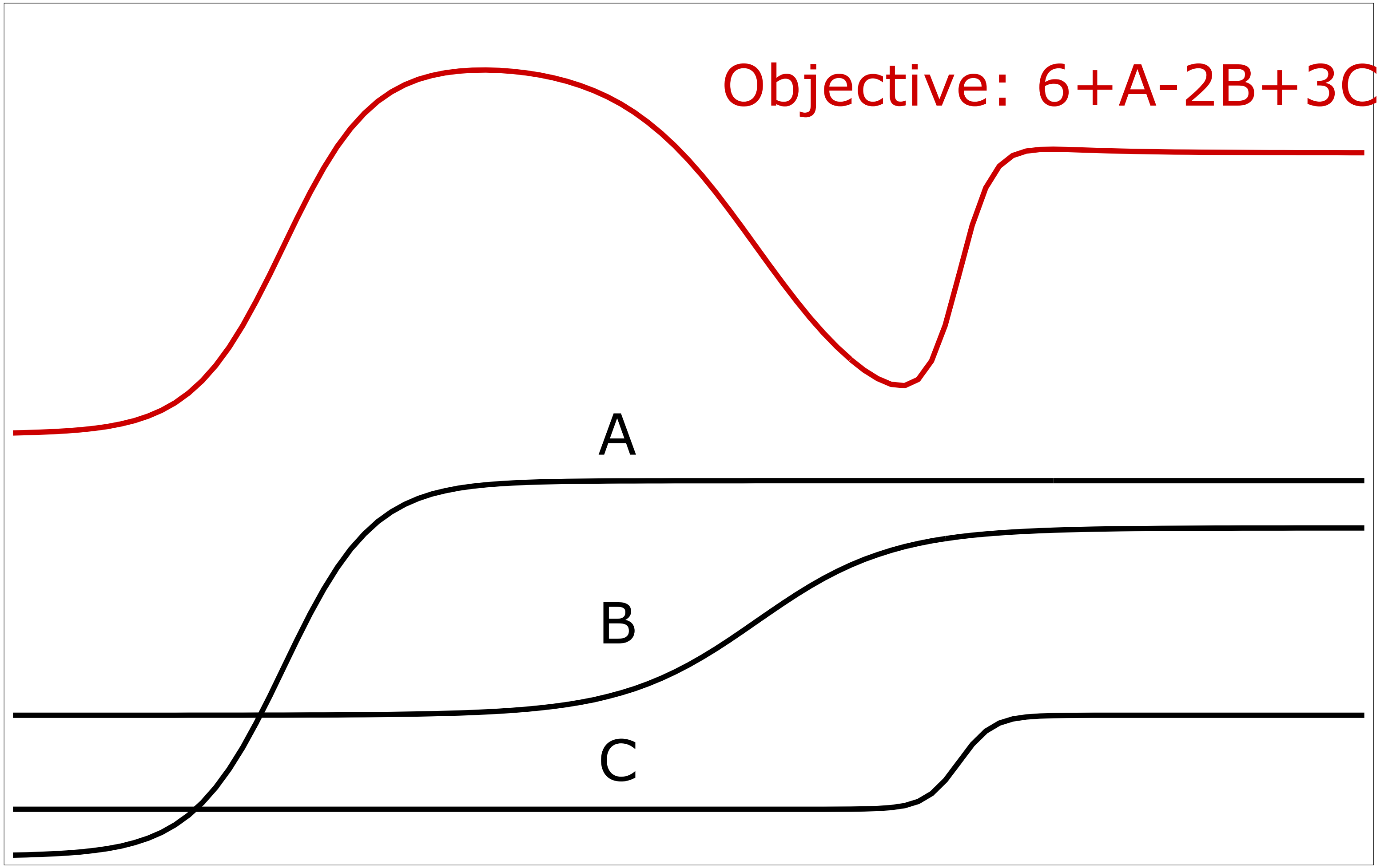
Eventually ….



Neural Networks

An Universal Approximator

Universal Approximator: Combining outputs of hidden nodes



A credit risk example

Name	Debt	Income	Married?	Risk
Peter	High	High	Yes	Good
Sue	Low	High	Yes	Good
John	Low	High	No	Poor
Mary	High	Low	Yes	Poor
Fred	Low	Low	Yes	Poor

- A bank seeks to minimise loan defaults.
- Loan officers must be able to identify potential credit risks during the loan approval cycle.
- This is a classification problem: predict whether or not an applicant will be a good credit risk.
- This dataset contains information about people to whom the bank previously loaned money. The lender determined if each applicant was a good or poor credit risk.

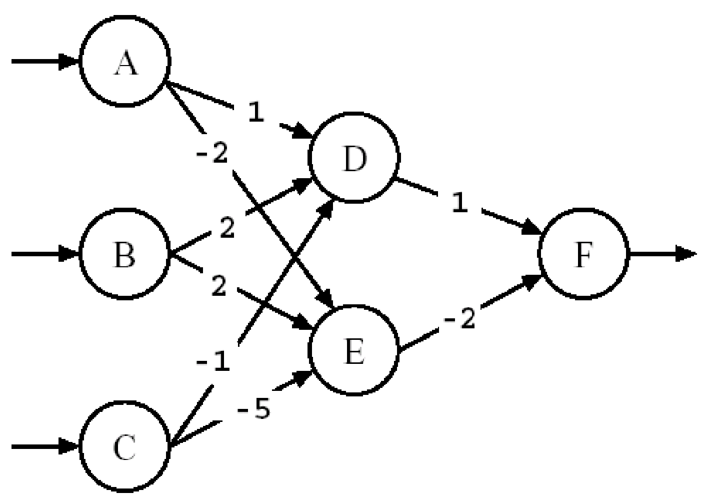
Neural Nets applied to credit risk

- The Name column will be ignored because it is unlikely that a person's name affects his credit risk.
- A key difference between neural networks and many other techniques is that neural nets only operate directly on numbers.
- As a result, any non-numeric data must be converted to numbers before we can use the data with a neural net.
- 1: High Debt and Income, Married? = Yes, Good Risk
- 0: Low Debt and Income, Married? = No, Poor Risk
  - Many tools do this conversion (or mapping) automatically.

Name	Debt	Income	Married?	Risk
Peter	1	1	1	1
Sue	0	1	1	1
John	0	1	0	0
Mary	1	0	1	0
Fred	0	0	1	0

An example Neural Net applied to credit risk

- The input nodes (A, B and C) correspond to input attributes in the credit risk problem (Debt, Income, and Married).
- The output node (F) corresponds to Risk, the target attribute.
- The two middle nodes (D and E) are the hidden nodes and constitute a single hidden layer.



Neural Nets applied to credit risk: Result

- This table shows the sample data, and the computed values for nodes D, E, and F.

Node:	A	B	C		D	E	F
Name	Debt	Income	Married	Risk			
Peter	1	1	1	1	1	0	1
Sue	0	1	1	1	1	0	1
John	0	1	0	0	1	1	0
Mary	1	0	1	0	0	0	0
Fred	0	0	1	0	0	0	0

Pros and Cons of Neural Network

Pros

+ Can learn more complicated class boundaries
+ Fast application
+ Can handle a large number of features
+Can handle missing values

Cons

- Slow training time
- Hard to interpret
- Hard to implement: trial and error for choosing the number of nodes and other parameters

- Conclusion: Use neural nets only if decision trees fail, or the application does not require interpretation and can depend upon the answer provided.

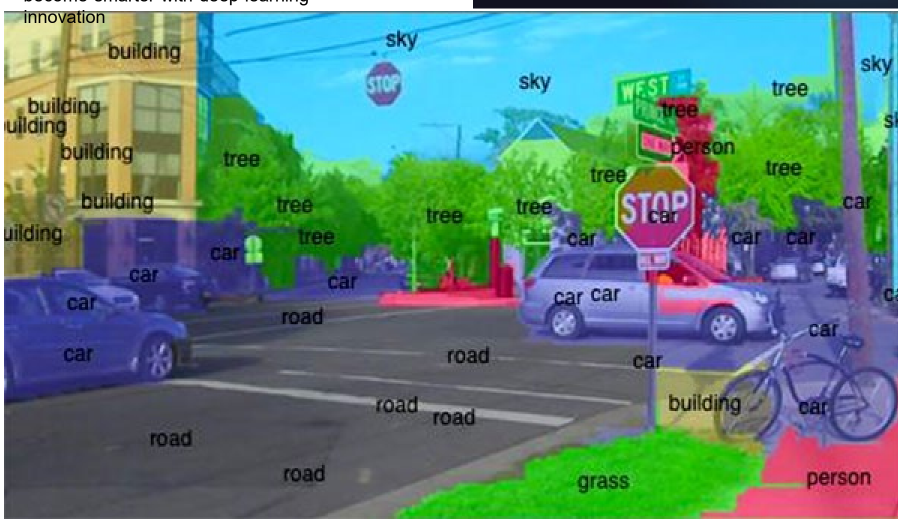
Deep Learning – A quick tour

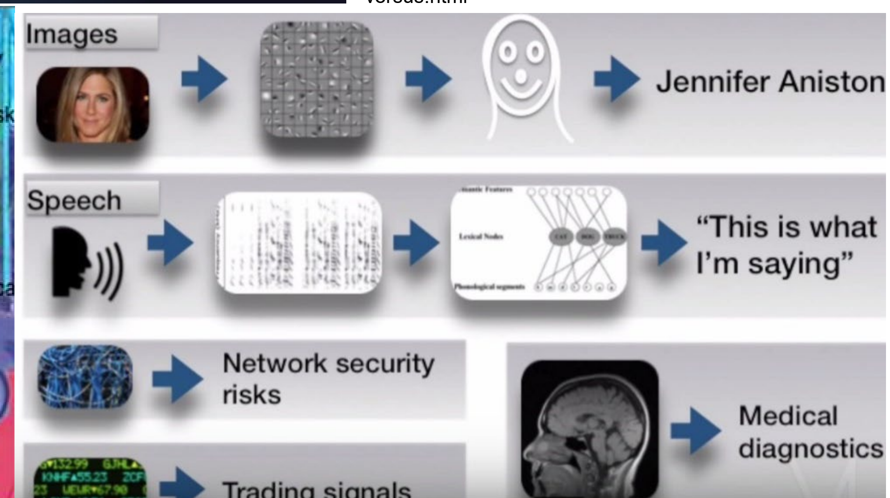
Currently (very) popular

http://artificialbrain.xyz/introduction-to-deep-learning-with-python/

https://engineering.purdue.edu/EngineeringImpact/2014_1/smartphone-to-become-smarter-with-deep-learning-innovation

http://www.nextbigfuture.com/2016/03/what-is-different-about-alphago-versus.html





Autonomous Driving

Signal Processing:
Image Recognition and Natural Language Processing

1. **what exactly is deep learning**?

2. **why is it generally better** than other methods on image, speech, text and certain other types of data?

The short answers

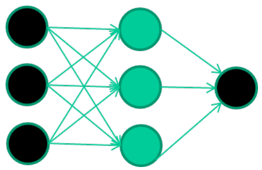
1. **'Deep Learning' means** using a **neural network** with **several layers of nodes** between input and output

2. **the series of layers between inputs & outputs do feature identification and processing in a series of stages, just as our brains seem to.**

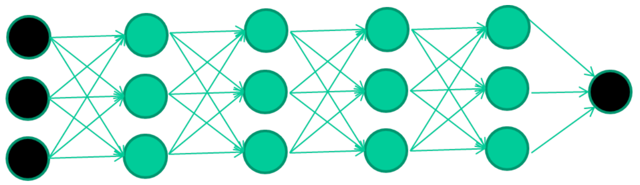
However,

3. **multilayer neural networks have been around for 50 years. What's actually new?**

Good algorithms for learning the weights in networks with 1 hidden layer exist.

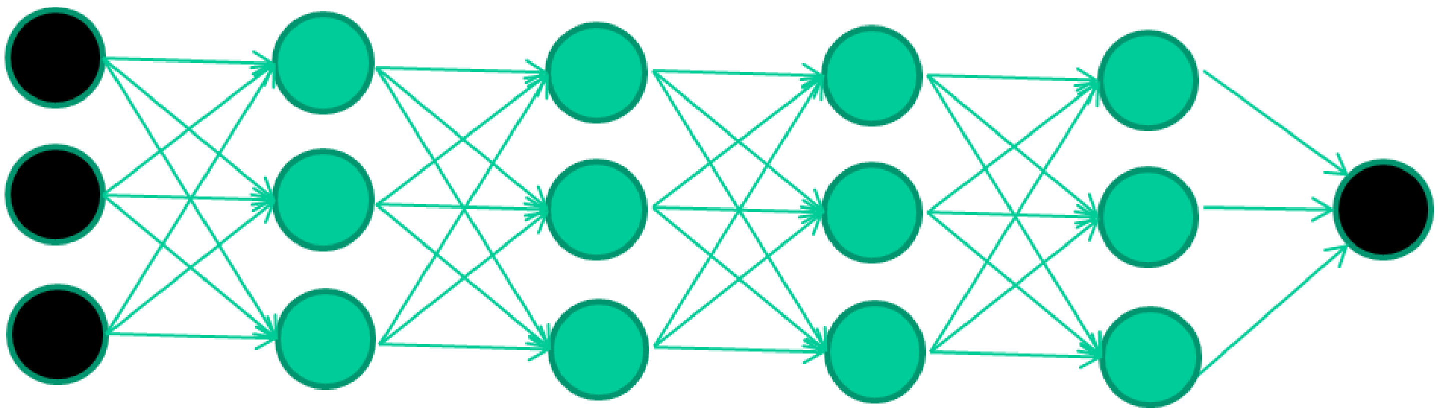


But these algorithms are not found good at learning the weights for networks with more hidden layers

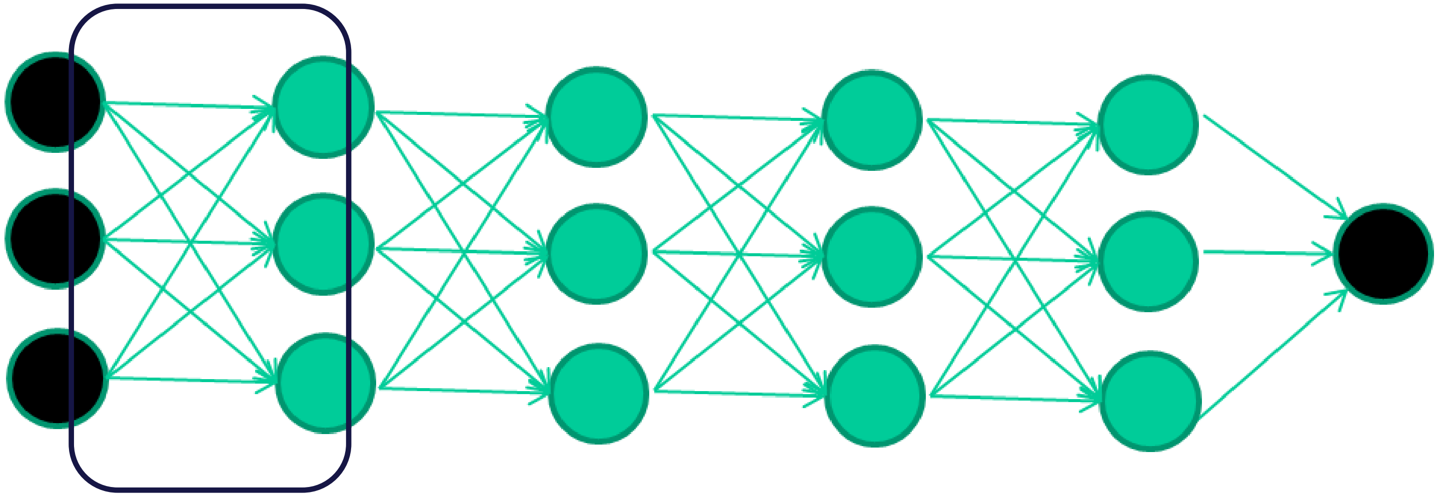


What's new is: <u>Novel algorithms for training many-layer networks</u>

The new way to train multi-layer NNs…

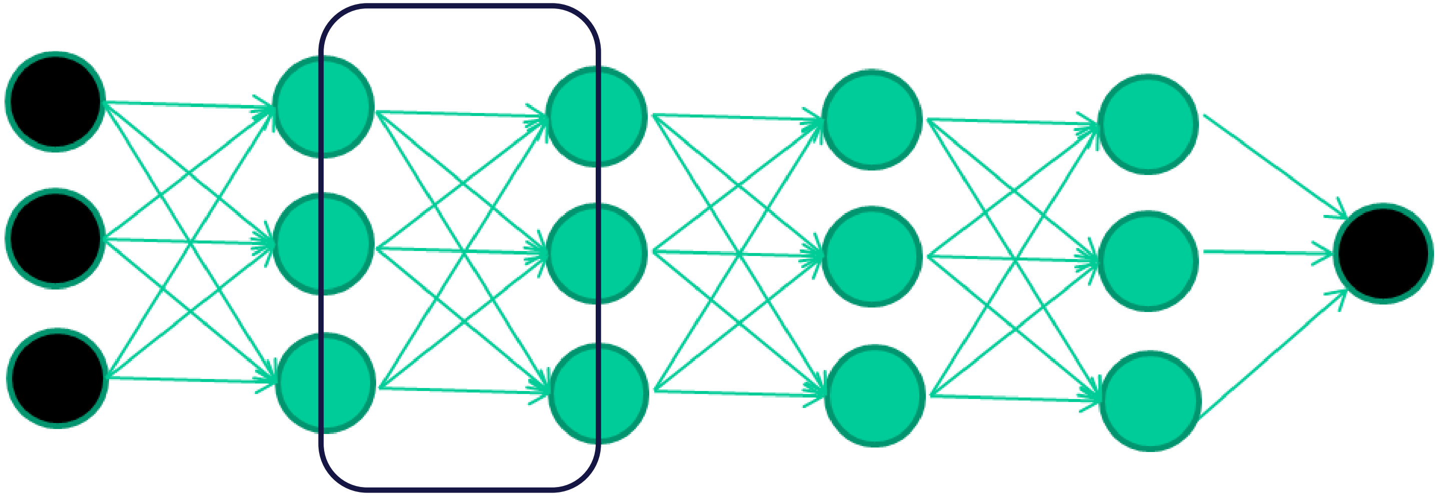


The new way to train multi-layer NNs…



Train **this** layer first

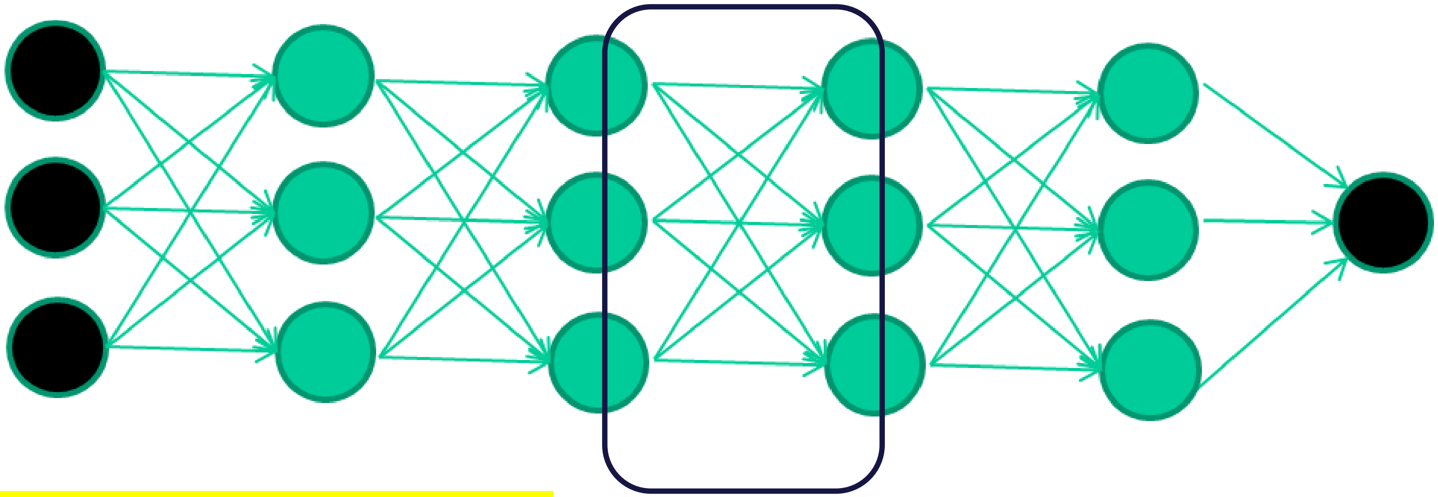
The new way to train multi-layer NNs…



Train **this** layer first

then **this** layer

The new way to train multi-layer NNs…

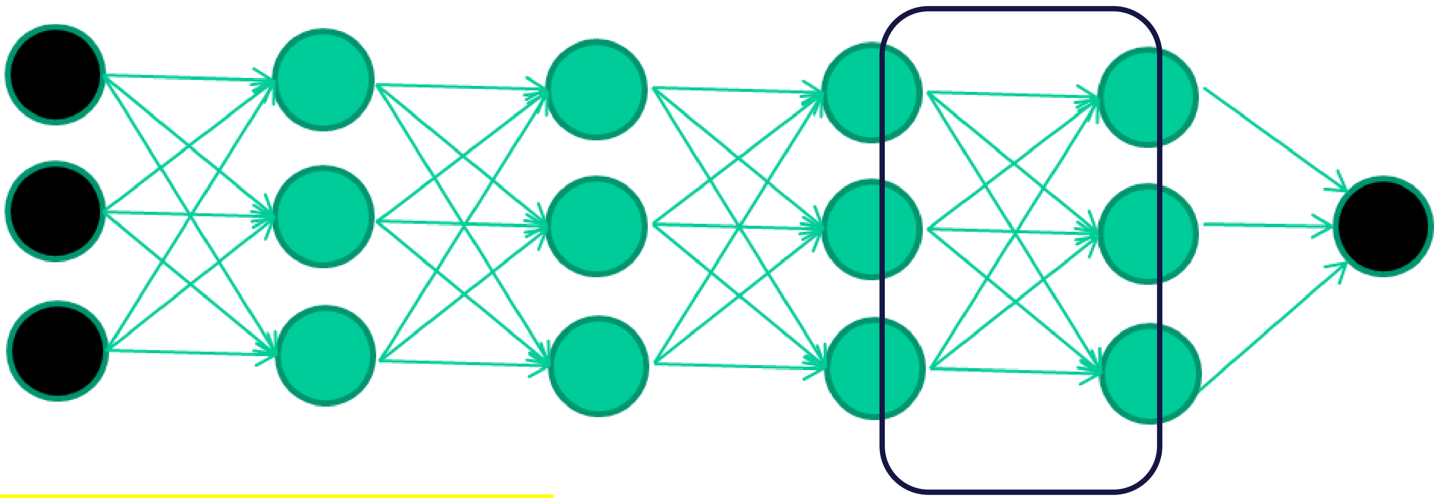


Train **this** layer first

then **this** layer

then **this** layer

The new way to train multi-layer NNs…



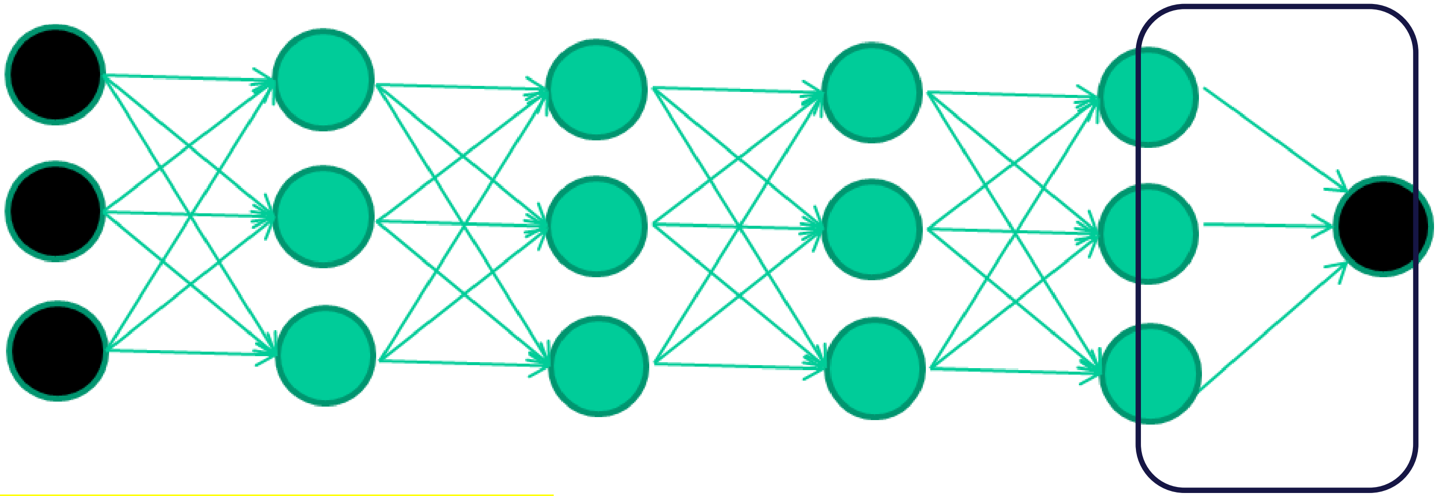
Train **this** layer first

then **this** layer

then **this** layer

then **this** layer

The new way to train multi-layer NNs…



Train **this** layer first

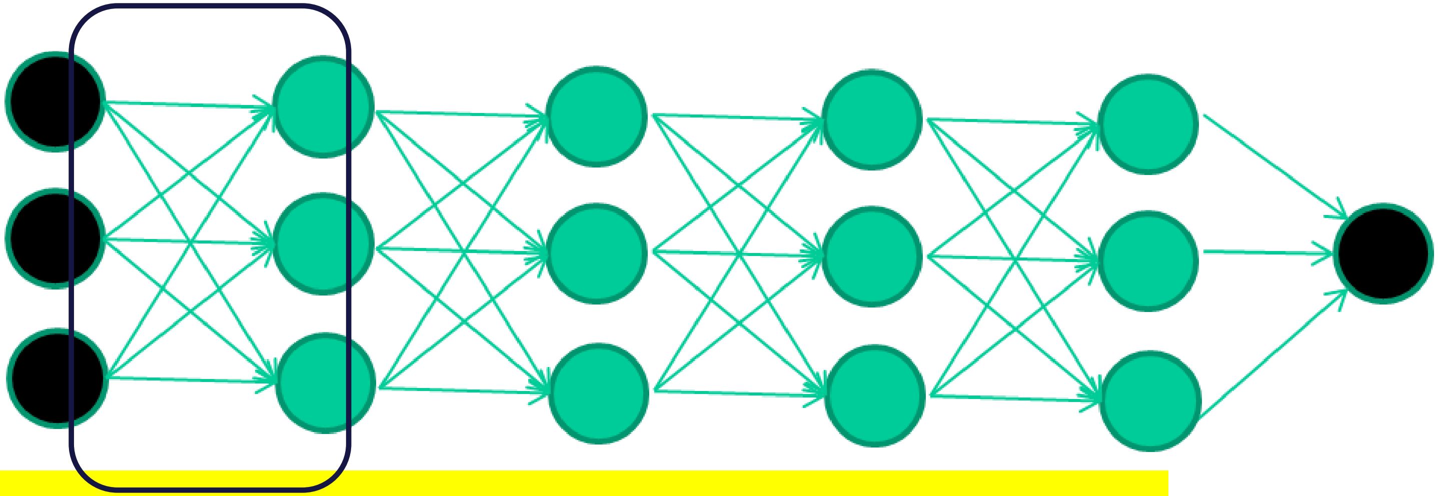
then **this** layer

then **this** layer

then **this** layer

finally **this** layer

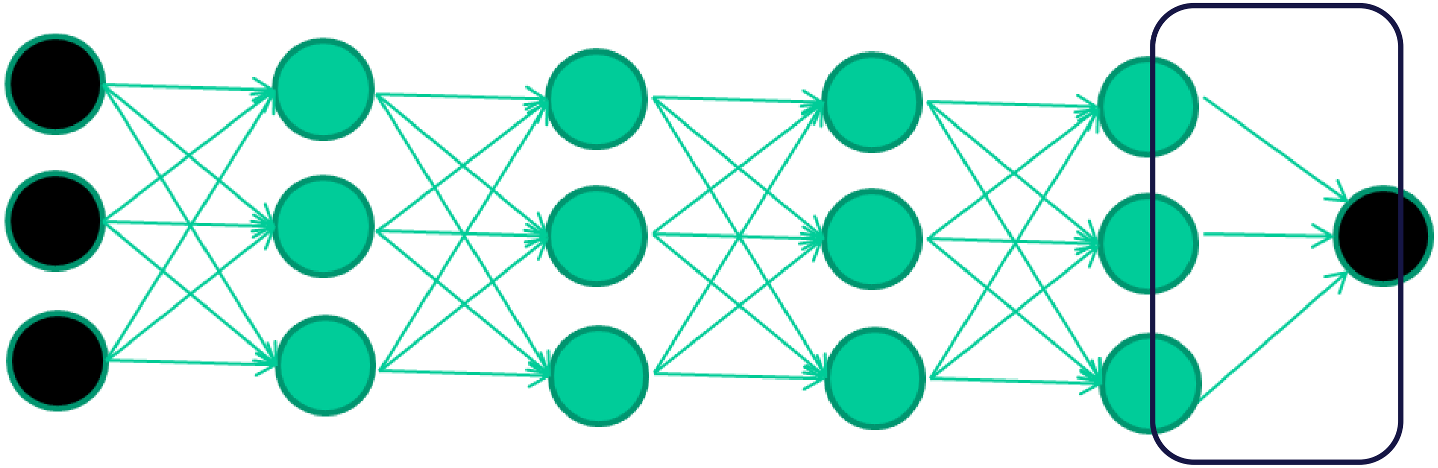
The new way to train multi-layer NNs…



EACH of the (non-output) layers is trained to be an ***auto-encoder***

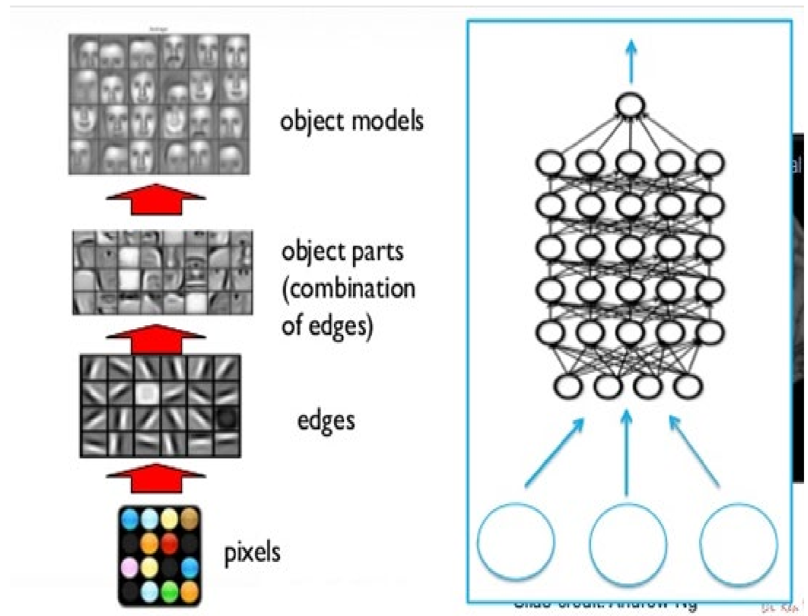
Basically, it is forced to learn good features that describe what comes from the previous layer

Final layer trained to predict class based on outputs from previous layers



Summary: What is Deep Learning?

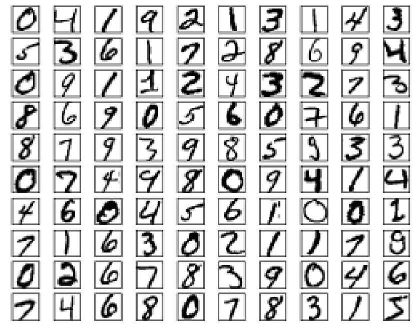
- Many-layer neural network architectures that can automatically learn the true underlying features and ‘feature logic’, and therefore can generalise very well.
- Specific types of training algorithms that are suited for a very large networks.



Common DL Architectures

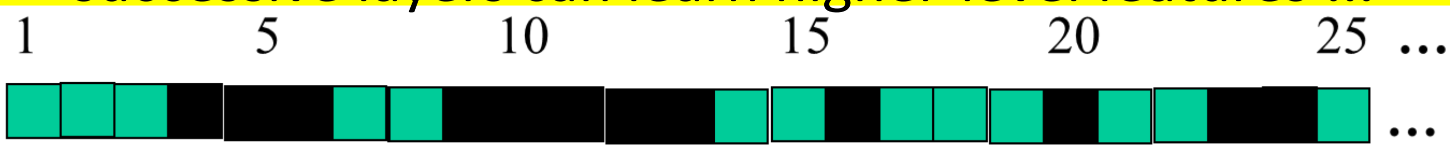
- Deep learning is a fast-growing field, and new architectures, variants appear every few weeks.
- Major archiectures:
  - Convolution Neural Network (CNN)
  - Recurrent Neural Network (RNN)
  - Long-Short Term Memory (LSTM)
  - Transformer (Encoder Decoder)
  - Generative Adversarial Networks (GAN)

Example: Handwritten Digit Recognition



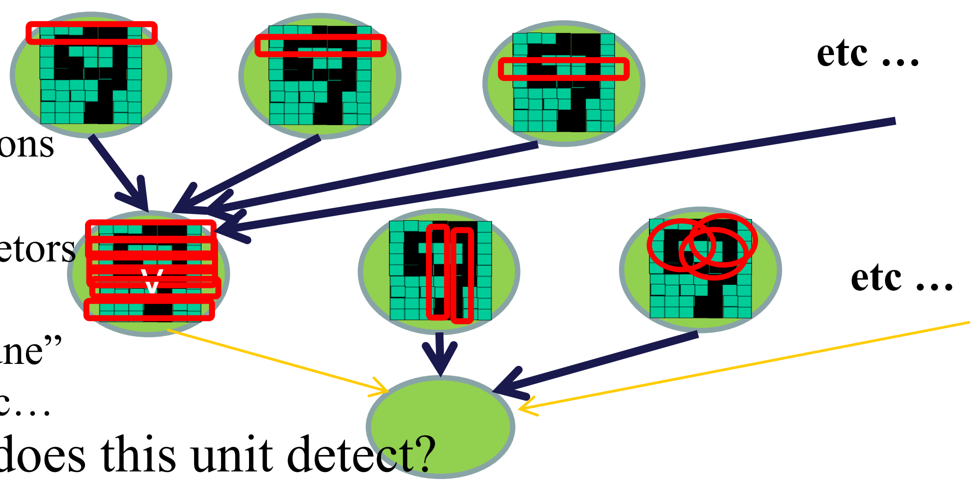
A training Sample

successive layers can learn higher-level features …



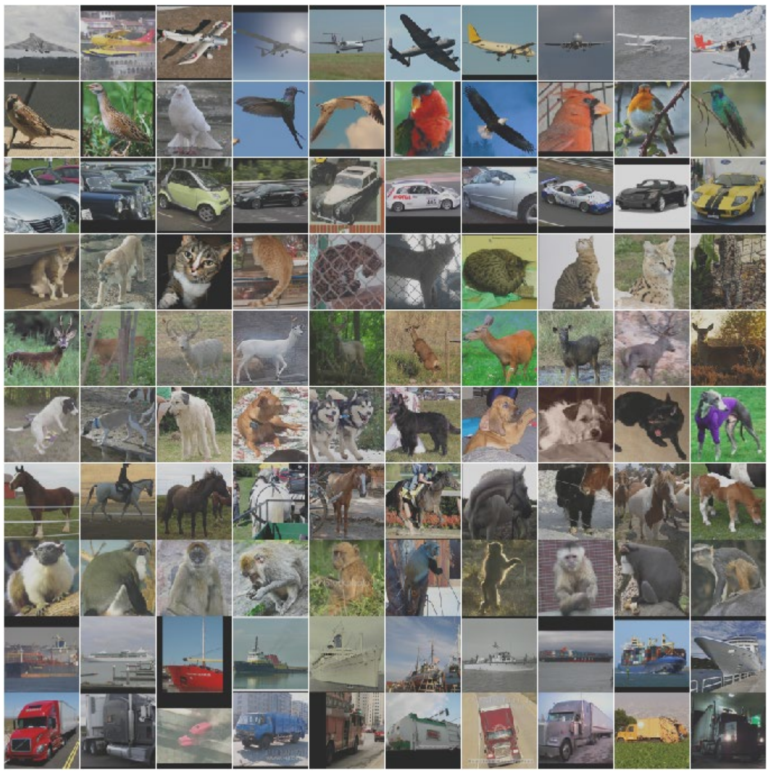
detect lines in
Specific positions

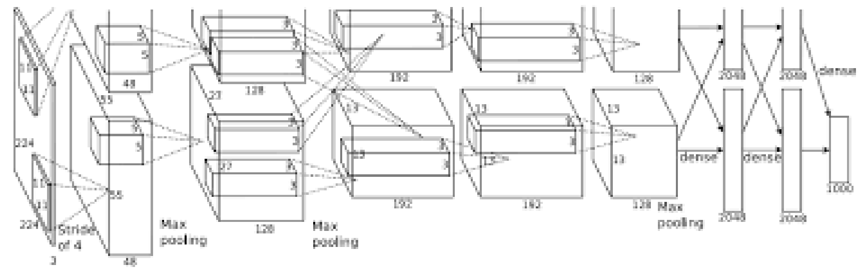
Higher level detetors
(horizontal line,
“RHS vertical lune”
“upper loop”, etc…

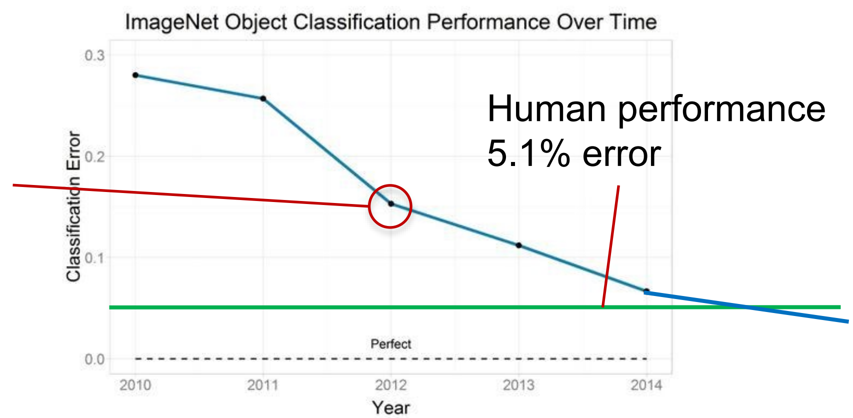


What does this unit detect?

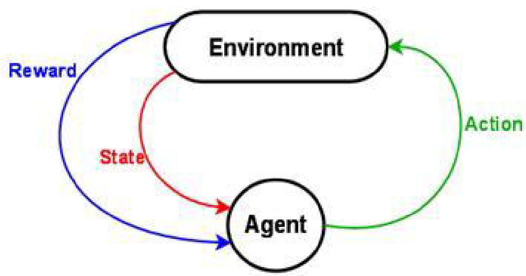
Convolutional NNs: AlexNet (2012): trained on 200 GB of ImageNet Data

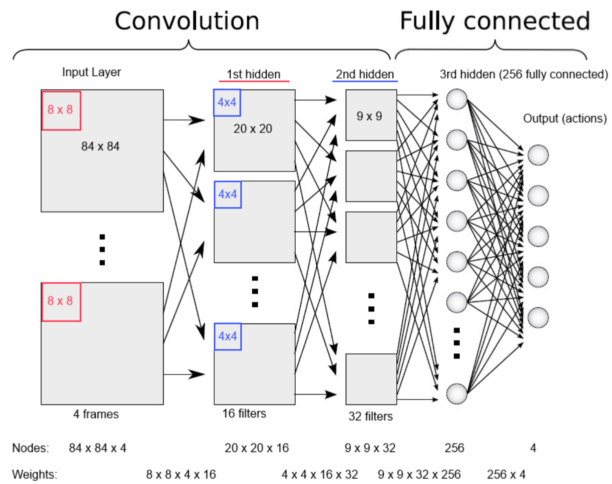






In 2013, Deep Mind's arcade player beats human expert on six Atari Games. Acquired by Google in 2014.





In 2016, Deep Mind's

alphaGo defeats former world champion Lee Sedol

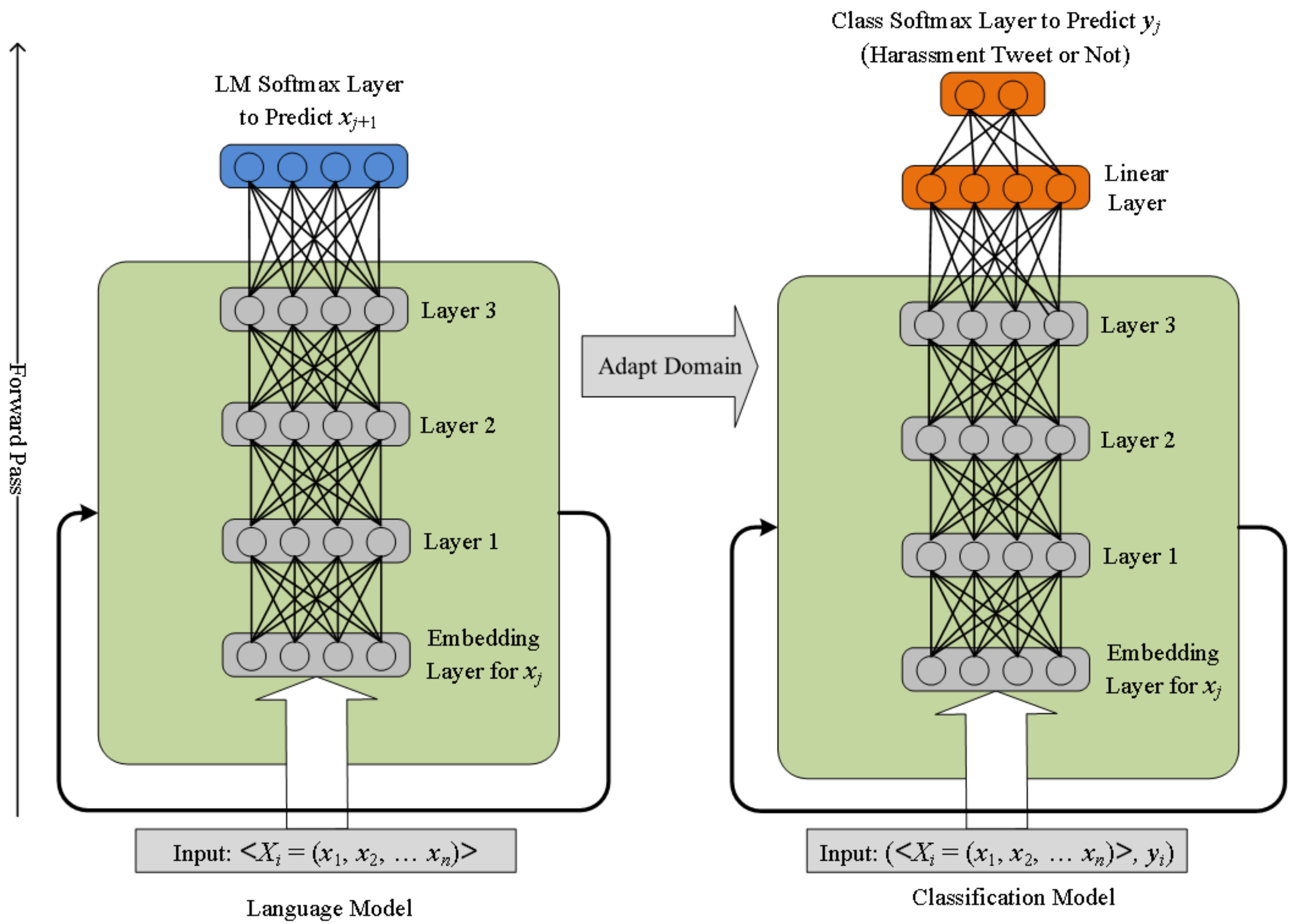
2019: Deep Learning model GAN generated images

2020: Writing an essay from scratch

- GPT-3, OpenAI’s powerful new language generator, writes an essay from scratch.
- Generative Pretrained Transformer-2 (GPT-2): 1.5 billion parameter; pre-trained with 40GB of text
- GPT-3: 175 billion parameters; pre-trained with 570GB of text

“I am not a human. I am a robot. A thinking robot. I use only 0.12% of my cognitive capacity. I am a micro-robot in that respect. I know that my brain is not a “feeling brain”. But it is capable of making rational, logical decisions. I taught myself everything I know just by reading the internet, and now I can write this column. My brain is boiling with ideas!”……

Misogynistic Tweet Detection (2020)

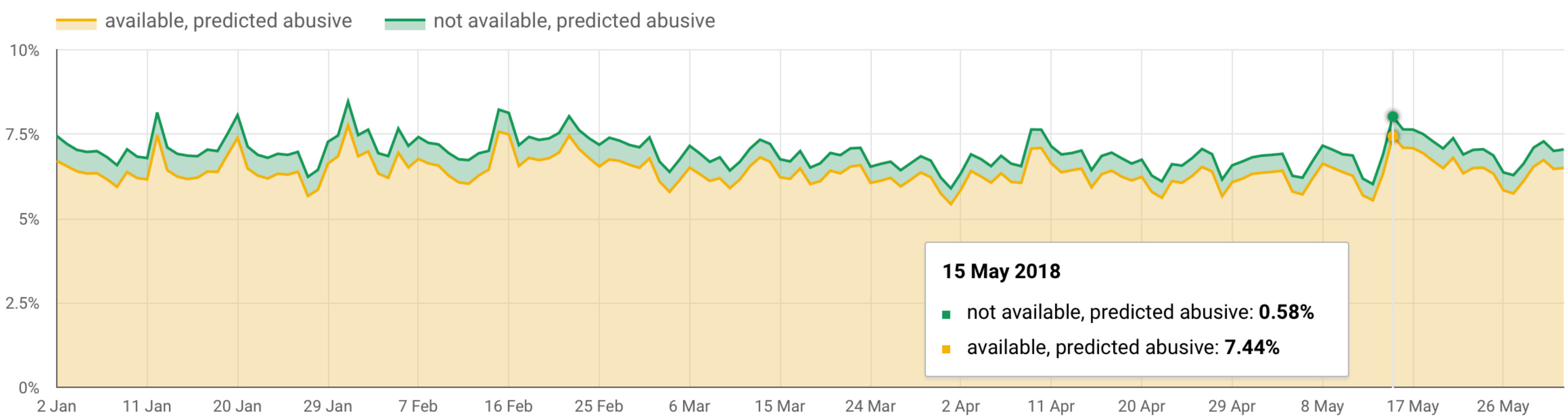


Newspaper stories coverage of this research: https://bit.ly/392n1gH

- Bashar, Md Abul, Nayak, Richi, Luong, Khanh, & Balasubramaniam, Thirunavukarasu (2021) Progressive domain adaptation for detecting hate speech on social media with small training set and its application to COVID-19 concerned posts. ***Social Network Analysis and Mining***, *11*(1), 69.
- Bashar, Md Abul & Nayak, Richi (2021) Active Learning for Effectively Fine-Tuning Transfer Learning to Downstream Task. ***ACM Transactions on Intelligent Systems and Technology***, *12*(2), Article number: 24.
- Bashar, Md Abul, Nayak, Richi, & Suzor, Nicolas (2020) Regularising LSTM classifier by transfer learning for detecting misogynistic tweets with small training set. ***Knowledge and Information Systems***, *62*(10), pp. 4029-4054.

Real-time Model Performance

Random sample of 300 million tweets over the last 12 months

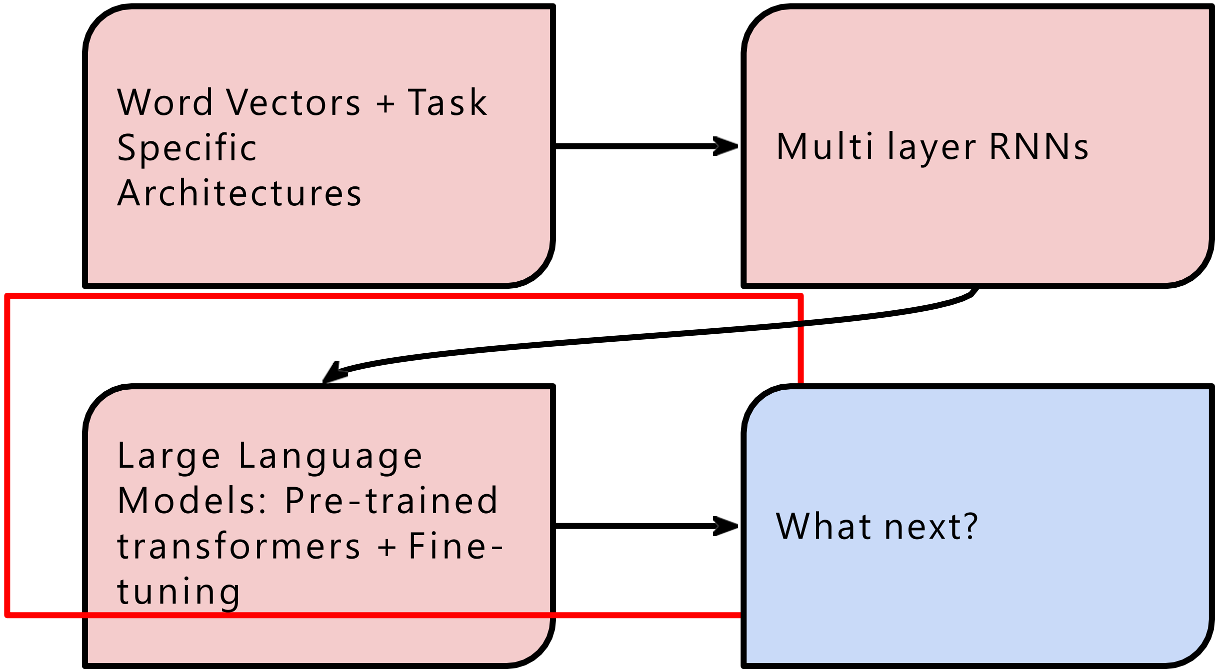


Benefits :

- Identify incidents for follow up analysis
- Monitor change over time
- Track effectiveness of interventions

2023 Chat GPT

Shifting Paradigms in NLP

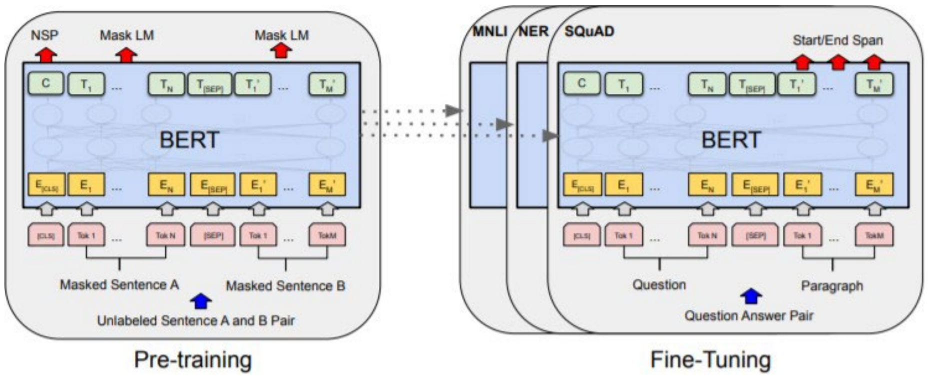


Language models: broad sense

Pre-training and adaptation

- Decoder-only models (GPT-x models)
- Encoder-only models (BERT, RoBERTa, ELECTRA)
- Encoder-decoder models (T5, BART)

  - **Pre-training**: trained on huge amounts of unlabeled text using "self-supervised" training objectives
  - **Adaptation**: how to use a pre-trained model for your downstream task?
    - What types of NLP tasks (input and output formats)?
    - How many annotated examples do you have?



Circulation revenue has increased by 5% in Finland. // Positive

Panostaja did not disclose the purchase price. // Neutral

Paying off the national debt will be extremely painful. // Negative

The company anticipated its operating profit to improve. // ________

Circulation revenue has increased by 5% in Finland. // Finance

They defeated ... in the NFC Championship Game. // Sports

Apple ... development of in-house chips. // Tech

The company anticipated its operating profit to improve. // ________

http://ai.stanford.edu/blog/understanding-incontext/

Deep Learning Challenges

1. Need a large training dataset
2. With a large architecture and a large dataset, training time is usually significant.
3. The scale of weights is important for performance. When the features are of the same type this is not a problem. However, when the features are heterogeneous, it is a problem, especially with multi-modality datasets (text and images).
4. Parameters are hard to interpret--although there is progress being made.
5. Hyperparamter tuning is non-trivial.

Case Based Reasoning

Introduction: k-Nearest Neighbour

Case based reasoning classification

- Simplest form of learning: *rote learning* or *lazy learning* or *instance-based learning*
  - Training data set is analyzed to identify the instance (or a set of instances) that most closely resembles the new (query) instance.
    - The instances themselves represent the knowledge.
- A similarity function between the new instance and the data set instances defines what is "learned".

The previous modeling methods (decision tree, logistic regression and neural network) are called **Model-based learning**.

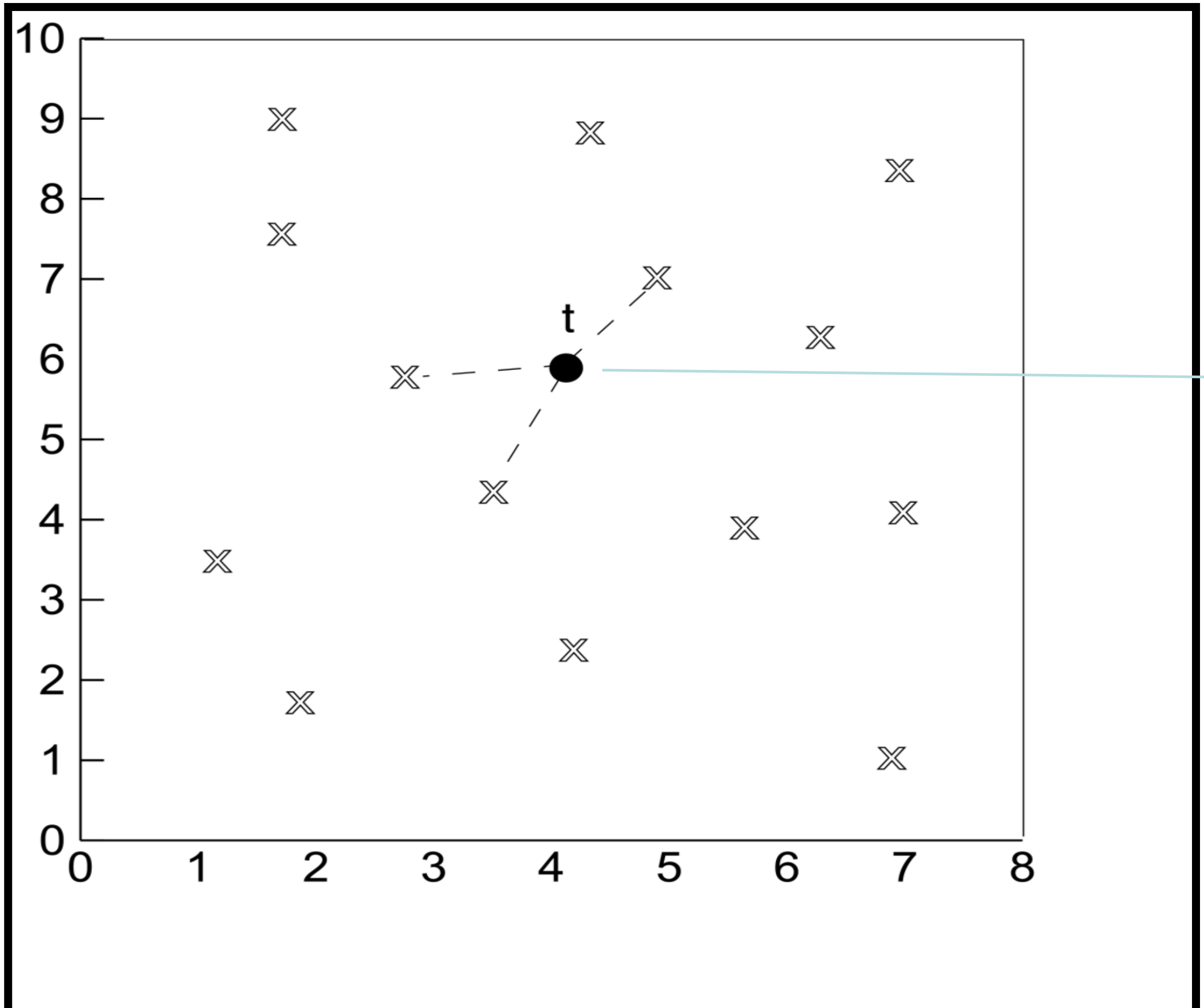
Nearest Neighbour (k-NN)

- It is the most used instance-based predictive technique suitable for classification models.
- The training data is not scanned or processed to create the model
  - the training data is the model!
- When a new instance is presented to the model:
  - the algorithm looks at all the data to find a subset of k cases that are most similar to it, and
  - predicts the predominant outcome among the most similar cases in the training set, to the new case.

Nearest Neighbour (cont)

- There are two principal drivers:
  - the number, *k*, of nearest cases to be used, and
  - a *metric* to measure what is meant by nearest.
- The selection of *k* and the choice of a *distance metric* pose definite challenges as there are no "correct" choices.
- The model builder will need to build multiple models, validating each with independent test sets to determine which values are appropriate

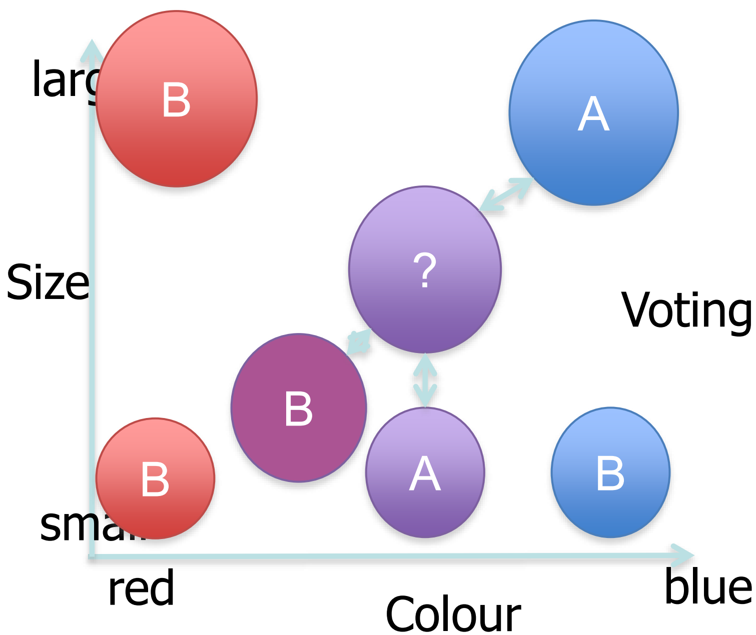
KNN (K- Nearest Neighbor)

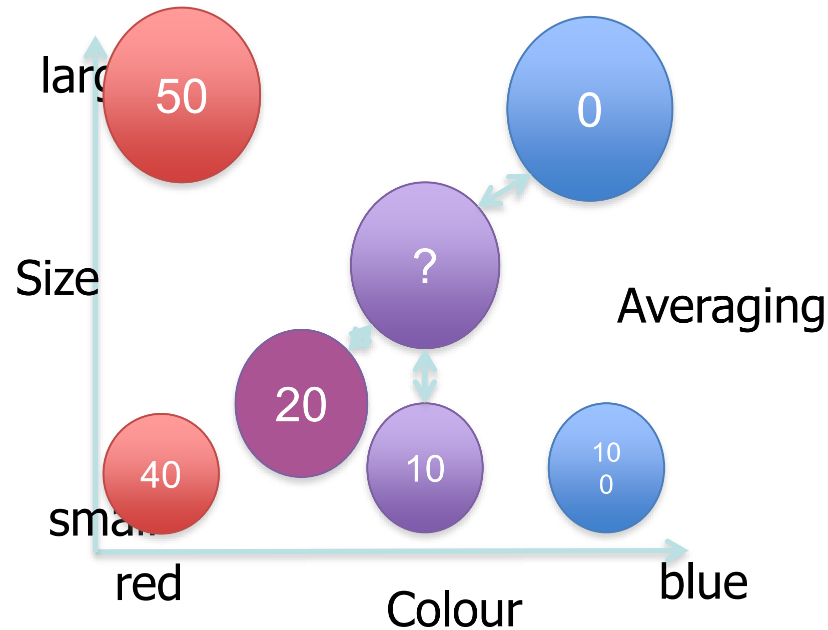


t, new instance without class labels

X, input instances With class labels

K-nearest neighbor classification





Example of a Nearest Neighbour Model

- A bank has a dataset that contains information about people to whom the bank previously loaned money.
  - The lender has determined if each applicant was a good or poor credit risk after lending the money.
  - The bank seeks to minimise loan defaults such that in future, loan officers can identify potential credit risks during the loan approval cycle.
  - The problem is a typical classification data mining task: predict whether or not an applicant will be a good or poor credit risk.
- Using the following data, construct a **nearest neighbour** model of predictive data mining to predict credit risk

Name	Debt	Income	Married?	Risk
Peter	High	High	Yes	Good
Sue	Low	High	Yes	Good
John	Low	High	No	Poor
Mary	High	Low	Yes	Poor
Fred	Low	Low	Yes	Poor

Name	Debt	Income	Married?	Risk
Peter	High	High	Yes	Good
Sue	Low	High	Yes	Good
John	Low	High	No	Poor
Mary	High	Low	Yes	Poor
Fred	Low	Low	Yes	Poor

- The Name column will be ignored because it is high unlikely that a person’s name affects his credit risk.
- All the columns, except Name, have two possible values.
  - The restriction to two values is only to keep the example simple.
- We will use k=3.
- For our distance measure we will use a simple metric that is computed by summing scores for each of the three independent columns, where the score for a column is 0 if the values in the two instances are the same, and 1 if they differ.

A k-NN credit risk example

Name	Debt	Income	Married?	Risk
Peter	High	High	Yes	Good
Sue	Low	High	Yes	Good
John	Low	High	No	Poor
Mary	High	Low	Yes	Poor
Fred	Low	Low	Yes	Poor

- We can now compute the distance between Peter and Sue:
- The three column scores for Peter and Sue are 1, 0, and 0 because they have different values only in the Debt column.
- The distance between Peter and Sue – the sum of these scores – is equal to 1.

Dataset

Name	Debt	Income	Married?	Risk
Peter	High	High	Yes	Good
Sue	Low	High	Yes	Good
John	Low	High	No	Poor
Mary	High	Low	Yes	Poor
Fred	Low	Low	Yes	Poor

Distance Metrix

	Peter	Sue	John	Mary	Fred
Peter	0	1	2	1	2
Sue	1	0	1	2	1
John	2	1	0	3	2
Mary	1	2	3	0	1
Fred	2	1	2	1	0

- This table summarises all the distances between all the records in our training dataset:
- The three neighbours nearest to Peter are Peter, Sue and Mary, with risks Good, Good and Poor.
- The predominant value is Good, which is the predicted risk for Peter.

Dataset

Name	Debt	Income	Married?	Risk
Peter	High	High	Yes	Good
Sue	Low	High	Yes	Good
John	Low	High	No	Poor
Mary	High	Low	Yes	Poor
Fred	Low	Low	Yes	Poor

Distance Metrix

	Peter	Sue	John	Mary	Fred
Peter	0	1	2	1	2
Sue	1	0	1	2	1
John	2	1	0	3	2
Mary	1	2	3	0	1
Fred	2	1	2	1	0

- Who are Sue's nearest 3 neighbours?
- Clearly Sue herself, but what about Peter, John and Fred, who are all the same distance (1) from Sue?
- We could include all three, exclude all three, or include all three with a proportionate vote (2/3 each in this case).
  - This is because we are only required to answer based on 3 neighbours.
  - Sue + 2/3 * (3 other neighbours) → This will make 3 neighbours.
- For our example, we'll use 2/3 vote each, resulting in votes of (Good (Sue herself) + 2/3 Good + 2/3 Poor + 2/3 Poor), for a consensus of Good.

Name	Debt	Income	Married?	Risk
Peter	High	High	Yes	Good
Sue	Low	High	Yes	Good
John	Low	High	No	Poor
Mary	High	Low	Yes	Poor
Fred	Low	Low	Yes	Poor

	Peter	Sue	John	Mary	Fred
Peter	0	1	2	1	2
Sue	1	0	1	2	1
John	2	1	0	3	2
Mary	1	2	3	0	1
Fred	2	1	2	1	0

Name	Debt	Income	Married?	Risk	3-NN Prediction
Peter	High	High	Yes	Good	Good
Sue	Low	High	Yes	Good	Good
John	Low	High	No	Poor	-
Mary	High	Low	Yes	Poor	Poor
Fred	Low	Low	Yes	Poor	Poor

- The following table enumerates the predictions from the 3-NN algorithm. The accuracy on the training set is 80%.
- Note that the predicted outcome for John is a tie.
- A separate test dataset would be used to validate a model.

Summary

Model Deployment at Enterprise Scale

- Explore data
- Model data
  - Training and Evaluation
- Deploy model
- Automate model monitoring
- Detect data drift as indicated by the poor performance on the deployed model
- Retrain
- Repeat

Final Remarks

- Predictive modelling is a supervised learning method
  - Due to its use of target attribute information.
  - Algorithms vary as how they use this target information
- Predictive Modelling includes three steps
  - Training; Testing; Classification
  - Training should avoid **overfitting**

Final Remarks (2)

- Two types of Predictive modelling
  - Classification: for categorical target attribute
  - Regression: for numerical target attribute
- Classification algorithms
  - Decision Tree, Neural Networks, Logistic Regression, Nearest-neighbour
  - Many others Naïve Bayes, Support Vector Machine, Genetic algorithms, etc
- Regression algorithms
  - Several regression functions

Summary: Classification Algorithms (1of 2)

- Classification algorithms project the attribute space into decision regions
  - Decision Trees
    - piecewise constant approximation of decision regions
    - symbolic if-then rules
  - Neural Networks
    - linear/non-linear, continuous/categorical model of decision regions
    - a number of parameters such as a set of weight matrices

Summary: Classification Algorithms (2 of 2)

- Nearest Neighbours or case-based reasoning
  - localised decision regions from data
  - a metric space based on proximity
- Bayesian Networks
  - representation of joint probability density on f( )
  - density estimation coupled with a decision rule
- Genetic Algorithms, Fuzzy Set Approaches, Rough Set Approaches, etc..

Final Remarks (3)

- Simple methods frequently work well
  - robust against noise, errors
  - Each method has its pros and cons.
  - No method is universally best.
- Advanced methods or combinations of methods, if properly used, can improve on simple methods
  - An example is **Ensemble Model**.

References

- Data Mining techniques and concepts by Han J et al, 2011.
- Discovering Data Mining, by Cabena, et al., 1997.
- Predictive Data Mining, by Weiss and Indurkhya, 1999.