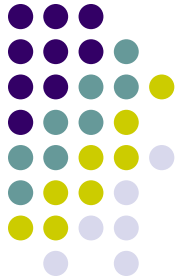
Birkbeck
UNIVERSITY OF LONDON



(Concepts of) Machine Learning

Lecture 1:

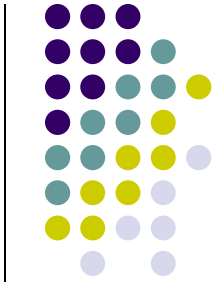
Module overview, knowledge representation and learning from data fundamentals

Prof George Magoulas
Email: gmagoulas@dcs.bbk.ac.uk

Outline

- The module
- Machine intelligence
  - two radically different paradigms: knowledge-based and data-driven
- Knowledge-based systems
- Machine Learning
- Learning from data: patterns, features, similarity, uncertainty, feature selection and generalisation
- Neural and genetic computing
- Swarm intelligence

The module



Covers bio-inspired methods (neural networks, fuzzy systems, supervised and unsupervised learning, genetic algorithms, evolutionary algorithms, swarm intellligence), and basic concepts of feature selection and generalisation.

Other staff

Cosmin Stamate (MSc Intelligent Technologies from Birkbeck) is working towards a PhD on machine learning and deep networks for psycho-physiological signals processing, modelling, and classification.

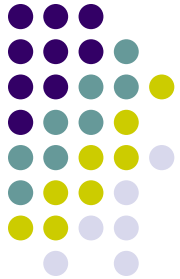
Michal Grochmal (MSc Intelligent Technologies from Birkbeck) is working towards a PhD on bio-inspired machine learning, combining deep learning and statistical mechanics.

The module

Assessment:

- *UG:* 2-hour exam (try 4 out of 5 questions), counts 100%
- *PG:* 2-hour exam (includes compulsory part plus 2 out of 3 questions), counts 80%, and practical assignment, counts 20%.
- *Module URL:* The module will use Moodle.
- *Coursework* (PG): deadline-> April 27, 2020; cut-off: May 7th, 2020

PG coursework



Mini project that will involve implementation of algorithm and testing with real datasets

You can write your own code in any programming language you wish, or use any tool/library you wish.

You report on your methodology to solve the problem, the tools you used and how you used them, your implementation, your results.

Publication date: February 20th, 2020

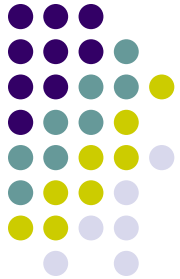
The module

No prerequisite module but essential prior knowledge of

- calculus and linear algebra (vector, matrices and their operations, functions and graphs, gradient, derivative)
- trigonometry concepts
- statistical concepts and the notion of probability
- data structures and algorithms
- first-order and second-order optimisation methods and general algorithmic concepts.

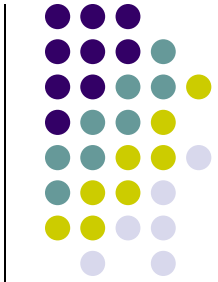
Schedule

(indicative, as they might be room changes- always consult Moodle and your personal timetable on MyBirkbeck)



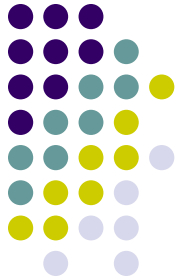
Week 1: 16 Jan	Knowledge representation, learning from data, key concepts
Week 2: 23 Jan	Modelling data uncertainty and Fuzzy Logic
Week 3: 30 Jan	Feature selection and learning paradigms
Week 4: 06 Feb	Neural computing and deep learning
Week 5: 13 Feb	Genetic and Evolutionary Computing
Week 6: 20 Feb	Advanced learning and evolution (ensembles, hybrid systems, swarms)
Week 7: 27 Feb	Lab 1 – Fuzzy
Week 8: 05 March	Lab 2 – Neural Networks and Deep learning-I
Week 9: 12 March	Lab 3 – Neural Networks and Deep learning-II
Week 10: 19 March	Lab 4 – Neurofuzzy and Unsupervised learning and clustering
Week 11: 26 March	Lab 5 – Genetic algorithms, Neuroevolution

Labs



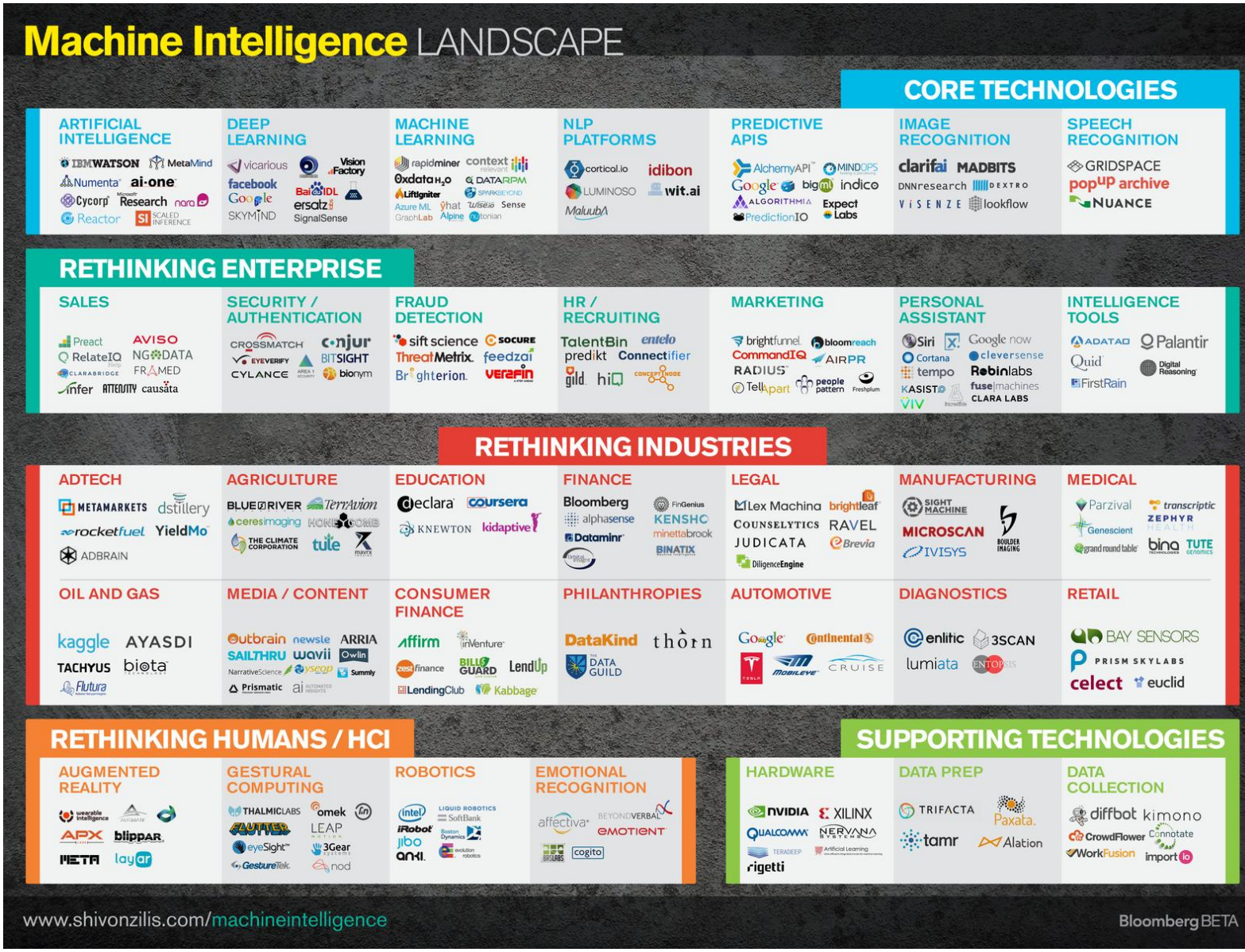
- Lab sessions focus on illustrating fundamental concepts and problem-solving tasks.
- We do not teach programming in the module but we use MATLAB and Python in the labs.
- Material to familiarise yourself with Matlab will be posted on Moodle. I suggest you try these activities before the start of the lab sessions in week 7.

Outline

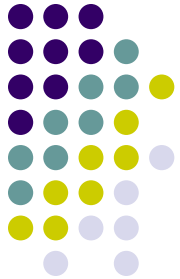


- The module
- Machine intelligence
  - two radically different paradigms: knowledge-based and data-driven
- Knowledge-based systems
- Learning from data: patterns, features, similarity, uncertainty, feature selection and generalisation
- Neural and genetic computing
- Swarm intelligence

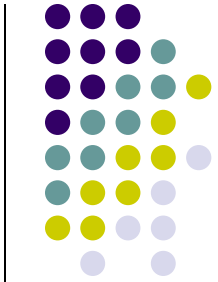
Machine Intelligence



Machine intelligence

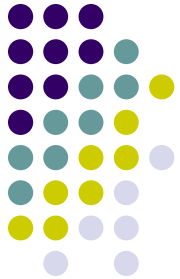


- Philosophers have been trying for over 2000 years to understand and resolve two *Big Questions* of the Universe: **How does a human mind work, and Can non-humans have minds?** These questions are still unanswered.
- ***(human) Intelligence*** is the ability to understand and learn things. ***Intelligence*** is the ability to *think* and understand instead of doing things by instinct or automatically. (*Essential English Dictionary*, Collins, London, 1990)



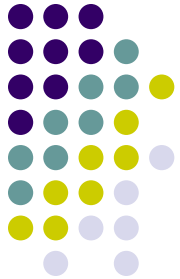
- In order to think, some*one* or some*thing* has to have a brain, or an organ that enables some*one* or some*thing* to *learn and understand* things, to solve problems and to make decisions. So we can define intelligence as ***the ability to learn and understand, to solve problems and to make decisions***.
- The goal of ***artificial intelligence*** (AI) as a science is to make machines do things that would require intelligence if done by humans.

We need knowledge- what is knowledge?



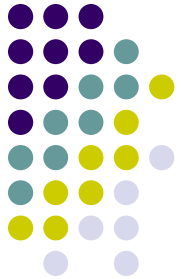
- **Knowledge** is a theoretical or practical understanding of a subject or a domain.
- Those who possess knowledge in a domain are called experts.
- Anyone can be considered a **domain expert** if he or she has deep knowledge (of both facts and rules) and strong practical experience in a particular domain. The area of the domain may be limited. In general, an expert is a skilful person who can do things other people cannot.

How knowledge can be captured?



- Observations and interviews of experts; express knowledge in the form of rules
- Learning from data, usually supervised because we look for reactions to given states
- "Trial and error" guided by an error signal that tells us how well we are performing.

Knowledge-based systems



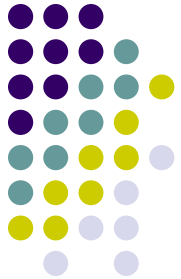
knowledge-based systems: software packages designed to assist humans in situations in which an expert in a specific area is required.

Major tasks:

- obtain the required knowledge.

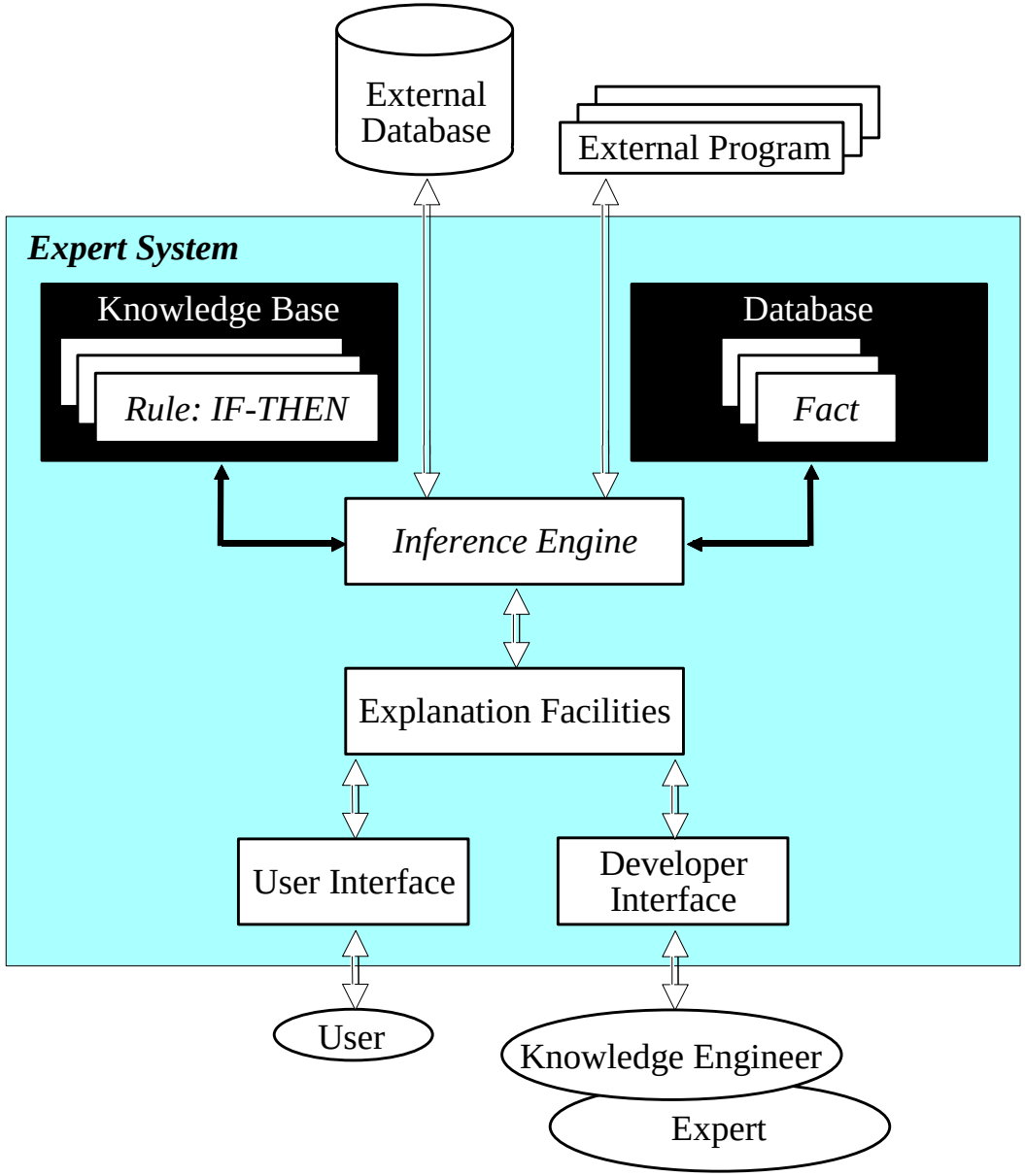
- express knowledge as a collection of rules in the form of logical implications (knowledge base)

Characteristics of a KBS



- Simulates human reasoning about a problem domain
  - Emulates an expert's problem-solving abilities
  - Deals with subject matter of realistic complexity that normally requires a considerable amount of human expertise
  - Performs reasoning (inference engine) over representations of human knowledge (knowledge base), in addition to doing numerical calculations or data retrieval
- Solves problems by heuristic or approximate methods which, unlike algorithmic solutions, are not guaranteed to succeed.
- Explains and justifies solutions or recommendations
  - to convince the user that its reasoning is in fact correct.

Structure of a rule-based KBS

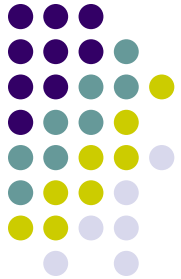


Example types of KBS

Category	Problem Addressed
Control	Interpreting, predicting, repairing, and monitoring system behaviours
Design	Configuring objects under constraints
Diagnosis	Inferring system malfunctions from observables
Instruction	Diagnosing, assessing, and repairing student behaviour
Monitoring	Comparing observations to plan vulnerabilities
Planning	Designing actions
Prediction	Inferring likely consequences of given situations
Repair	Executing a plan to administer a prescribed remedy

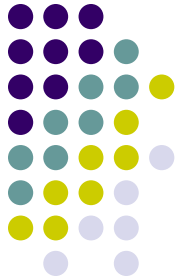
Exsys.com

Knowledge Acquisition



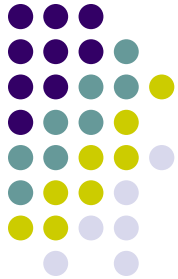
- Acquisition of knowledge from human experts, books, documents, etc.
  - Domain specific knowledge
  - Problem-solving procedures
  - General knowledge
  - Meta-knowledge – knowledge about knowledge
    - Information about how experts use their knowledge to solve problems and about problem-solving procedures in general

Knowledge Representation



- Knowledge acquired from experts or induced from a set of data must be represented in a format that is both understandable by humans and executable on computers

Knowledge representation-Rules

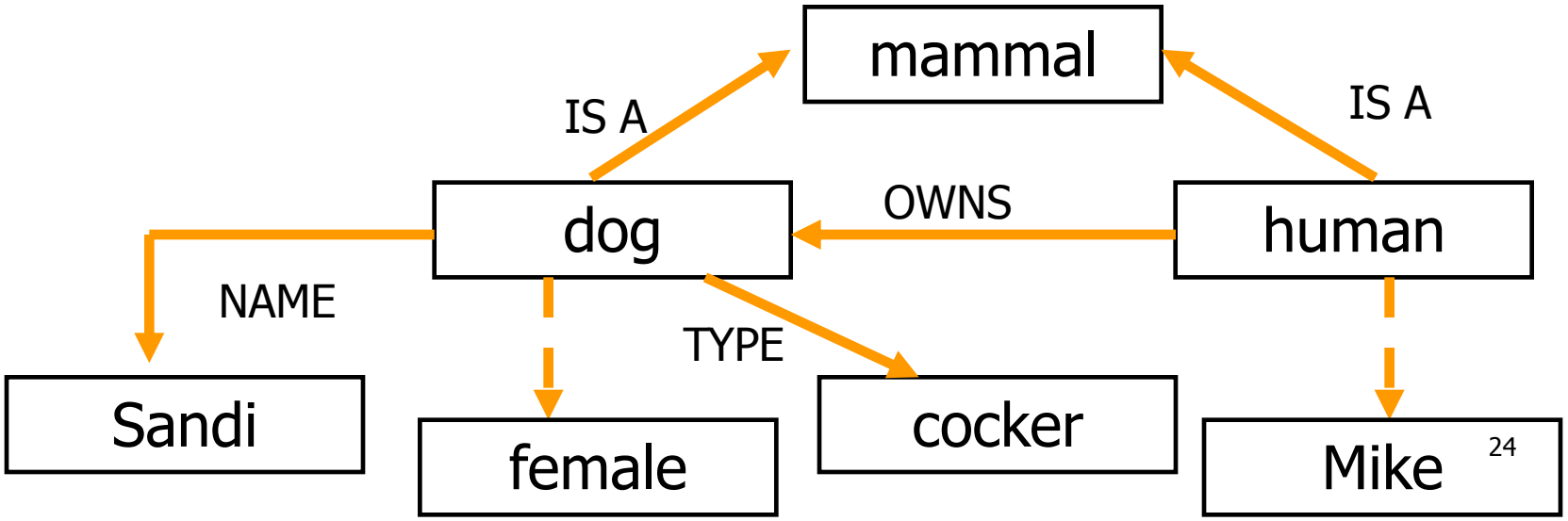


- Knowledge is represented in the form of condition/action pairs:
  - IF this condition (or premise or antecedent) occurs
  - THEN some action (or result or conclusion) will occur
- Example: MYCIN Rule
  - IF
    1. The infection is meningitis,
    2. Only circumstantial evidence is available,
    3. The type of infection is bacterial,
    4. The patient is receiving corticosteroids,
  - THEN
    - There is evidence that the organisms involved are:
      E.coli (0.4), Klebsiella pneumoniae (0.2) or Pseudomonas aeruginosa (0.1)

Knowledge representation –Semantic networks

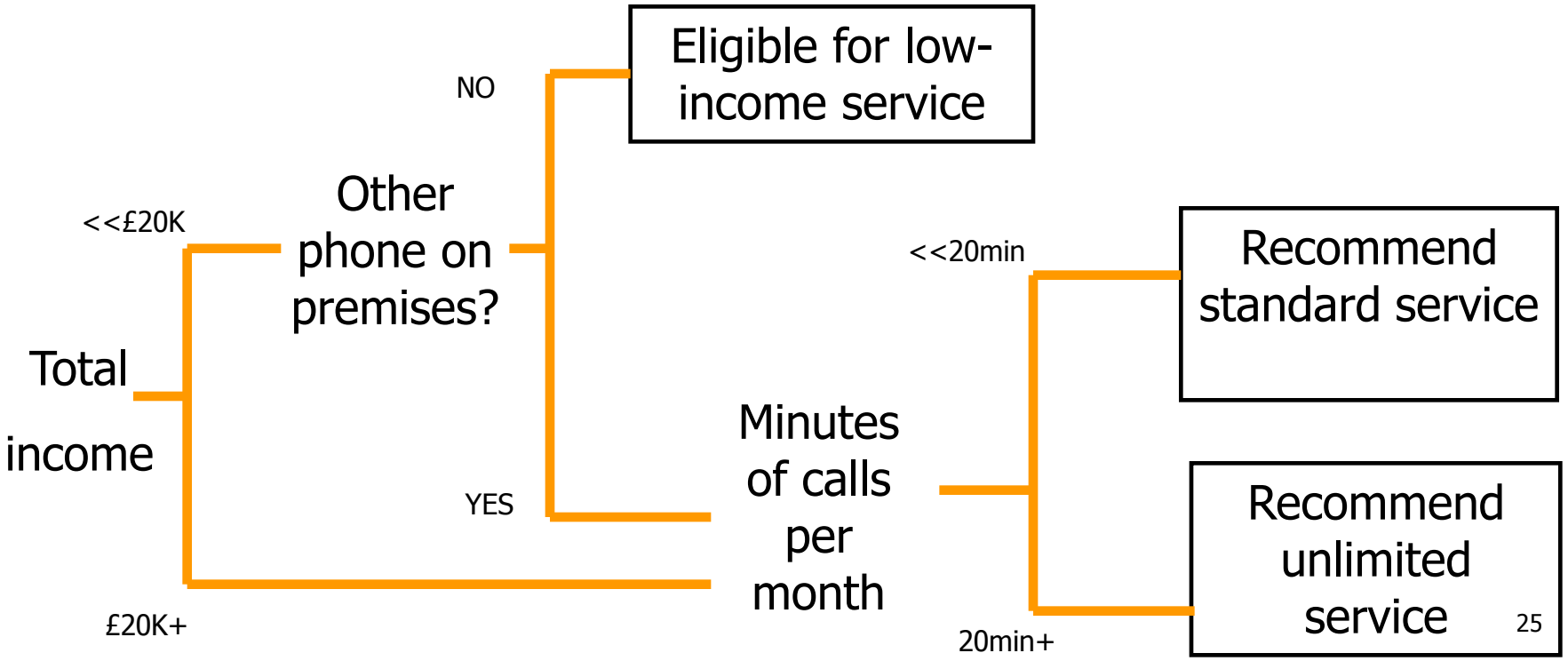
SEMANTIC NETWORKS are contextual maps of relationships utilising nodes and links. The nodes represent objects, events, or concepts. The links show the relationships between nodes.

Advantages: easily represent inheritance, flexible



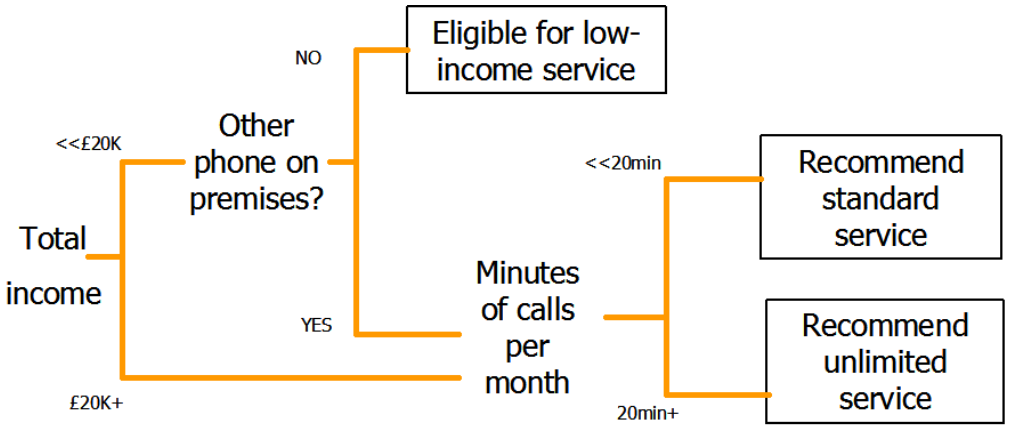
Knowledge representation- decision trees

DECISION TREES may be thought as hierarchical semantic networks, where nodes represent goals and links represent decisions. Always examine the tree from left to right.



Knowledge representation-Rules

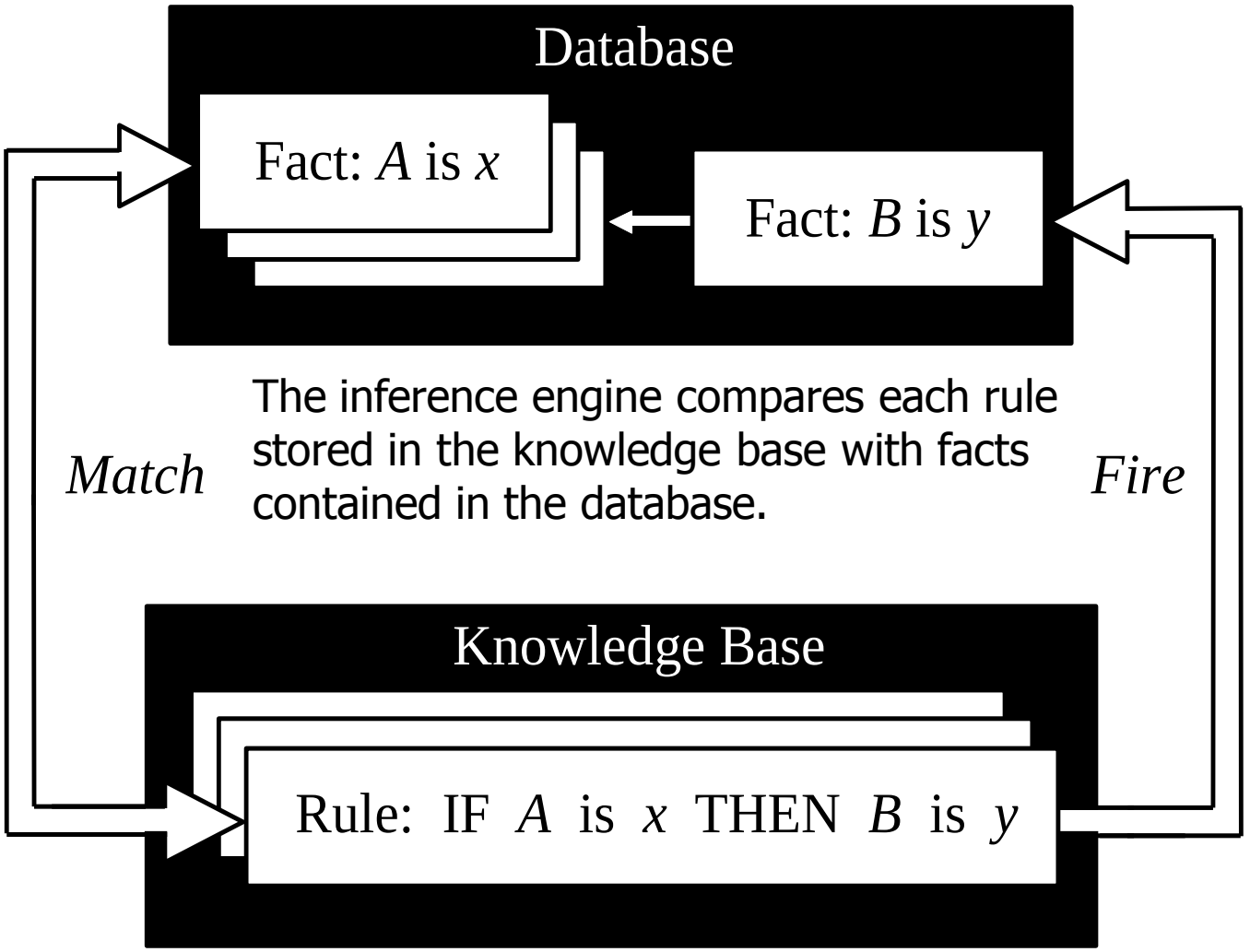
RULES are intuitive for most people and heuristic knowledge is easily represented. Easily modified.



Strategy for selecting a type of service.

```
IF total income is less than £20K and there is no other phone on the premises then select low-income service
```

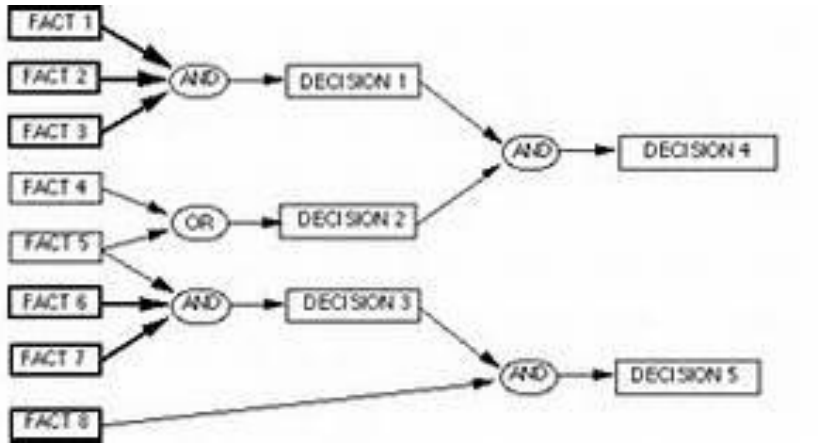
Inference engine cycles



the rule is **fired** and its THEN (action) part is executed

Forward Chaining

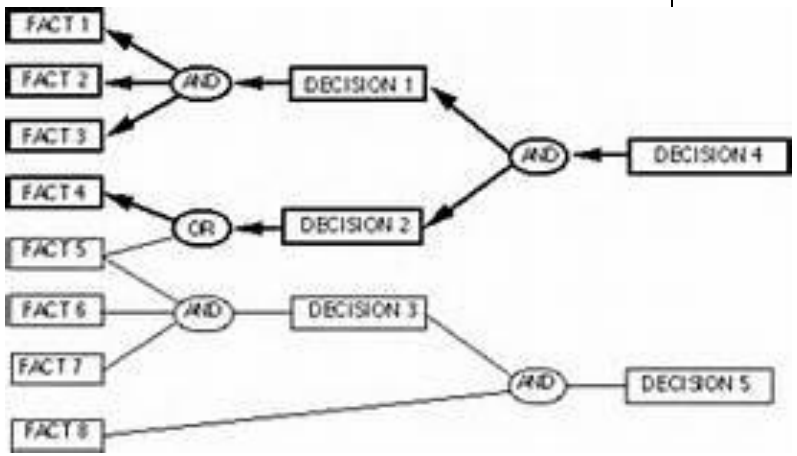
- Forward chaining is the **data-driven reasoning**.
  - The reasoning starts from the known data and proceeds forward with that data.
  - Each time only the topmost rule is executed.
  - When fired, the rule adds a new fact in the database.
  - Any rule can be executed only once.
  - The match-fire cycle stops when no further rules can be fired.



- Forward chaining is a technique for gathering information and then inferring from it whatever can be inferred.
- However, in forward chaining, many rules may be executed that have nothing to do with the established goal.
- Therefore, if our goal is to infer only one particular fact, the forward chaining inference technique would not be efficient.

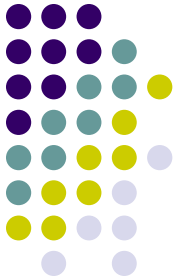
Backward chaining

- Backward chaining is the **goal-driven reasoning**.
  - A system has the goal (a *hypothetical solution*) and the inference engine attempts to find the evidence to prove it.
  - First, the knowledge base is searched to find rules that might have the desired solution. Such rules must have the goal in their THEN (action) parts.
  - If such a rule is found and its IF (condition) part matches data in the database, then the rule is fired and the goal is proved.



- Thus the inference engine puts aside the rule it is working with (the rule is said to ***stack***) and sets up a new goal, a subgoal, to prove the IF part of this rule.
- Then the knowledge base is searched again for rules that can prove the subgoal.
- The inference engine repeats the process of stacking the rules until no rules are found in the knowledge base to prove the current subgoal.

Outline



- The module
- Machine intelligence
  - two radically different paradigms: knowledge-based and data-driven
- Knowledge-based systems
- Machine learning
- Learning from data: patterns, features, similarity, uncertainty, feature selection and generalisation
- Neural and genetic computing
- Swarm intelligence

Machine learning

Machine learning is the part of computer science that attempts to make computers act like human beings, learning from their experiences and from data/information.

Machine learning methods automate model building. Using algorithms that iteratively learn from data, machine learning allows computers to find hidden insights without being explicitly programmed where to look.

Machine learning as a discipline explores the study and construction of algorithms that can learn from and make predictions on data– such algorithms overcome following strictly static program instructions by making data-driven predictions or decisions, through building a model from sample inputs.

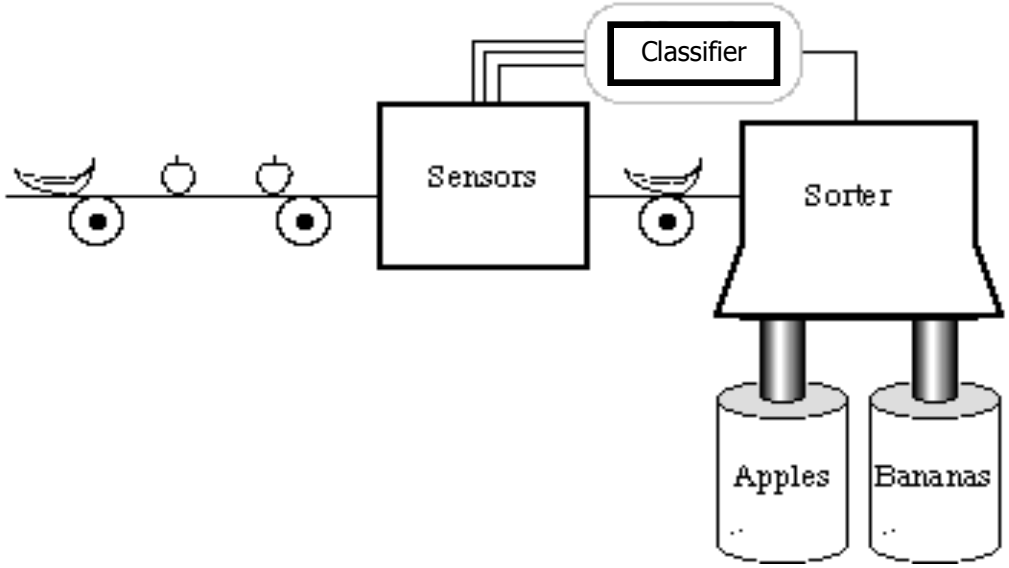
In Machine Learning, model building is considered as approximation of some kind of "true" underlying function or distribution:

- Formulate a space of functions (implemented by a particular type of machine learning algorithm) in which you can search for a good function approximation easily; e.g. linear classifiers (a linear combination of some basis functions), or the space of functions induced by a neural network.
- Formulate a loss function which indicates how close are the predictions of the current function approximation. For neural networks, this is mean-squared or sum-squared error. For deep learning, it is often cross-entropy.
- Formulate a regulariser which allows you to vary the complexity of the function approximation so that overfitting/underfitting is avoided.
- Perform optimisation to minimize the sum of the loss and regularisation.

Data-driven approaches-1

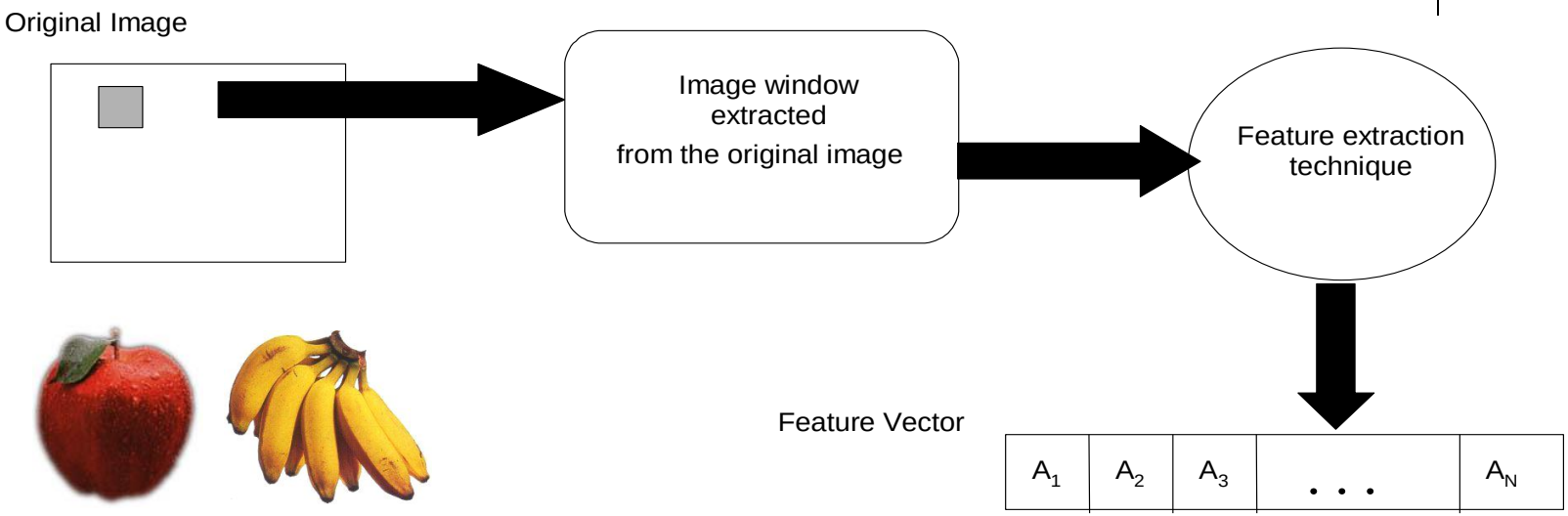
Patterns are "physical" representations of objects. Also called cases or samples.

Similarity among objects is evaluated through the use of *features* or *attributes*.



Example of *features*: COLOUR, SHAPE
The *prototypes*: "RED APPLE" and "BANANA".
Possible representations of the prototypes: "GREEN APPLE"

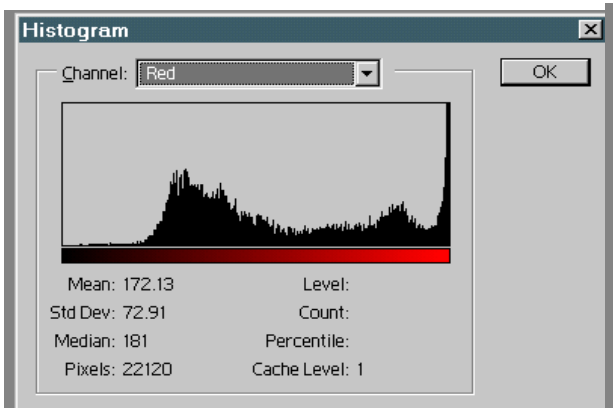
Data-driven approaches-2



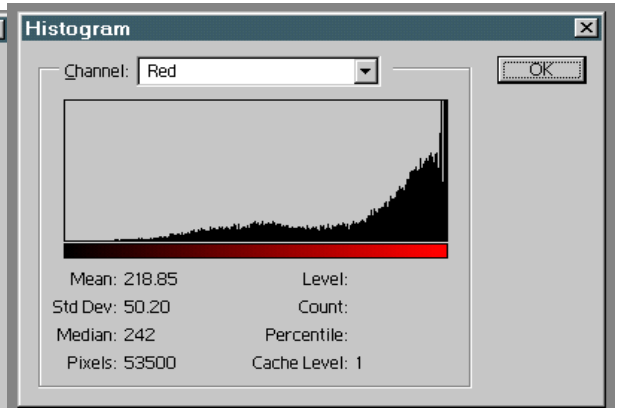
Feature 1: COLOUR→ IMAGE WINDOW → RGB components → Intensity histogram

Feature 2: SHAPE → WIDTH+HEIGHT → NORMALISE DISTANCE (W/H)

Data-driven approaches-3



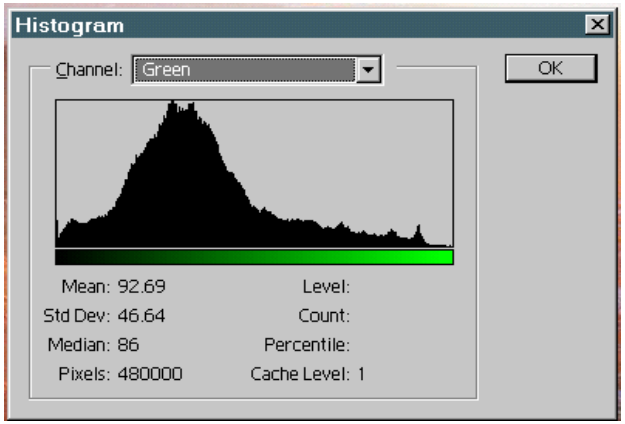
RED APPLE



BANANA

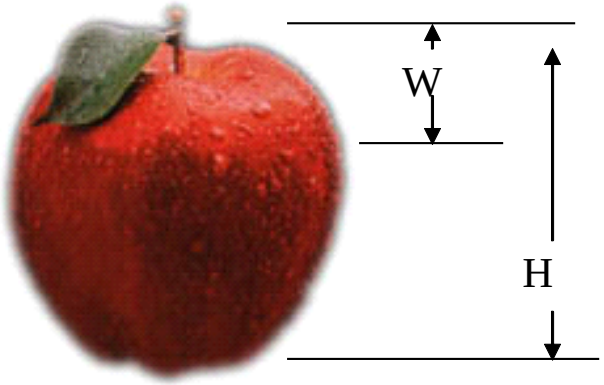
Feature-1:
COLOUR→
IMAGE WINDOW →
RGB components →
Intensity histogram

GREEN APPLE



Data-driven approaches-4

Feature-2:
SHAPE →
WIDTH+HEIGHT →
NORMALISE DISTANCE
(W/H)



Pattern or feature vector:

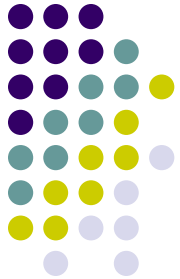
$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} COLOUR \\ SHAPE \end{bmatrix}$$

Data-driven approaches-5

The RED APPLE prototype: $\mathbf{x} = \begin{bmatrix} 1.85 \\ 0.37 \end{bmatrix}$

General case: $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \mathrm{M} \\ x_d \end{bmatrix}$ is in the *d*-dimensional domain of the feature vectors

Data-driven approaches-6

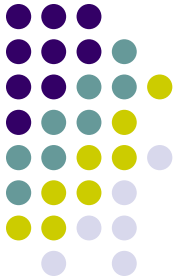


• **Similarity between two objects**. We recognise two objects as being similar because they have *similarly-valued common attributes.*

• **Similarity between an object and a target concept**. For example, we recognise an object as being a car because *it corresponds, in its features, to an idealised image, concept or prototype*, we may have of a car, i.e. the object is similar to that concept and dissimilar from the others, e.g. from a tree.

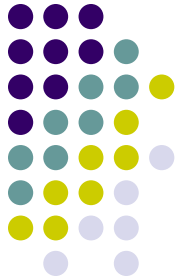
How do we select attributes/features?

Feature selection



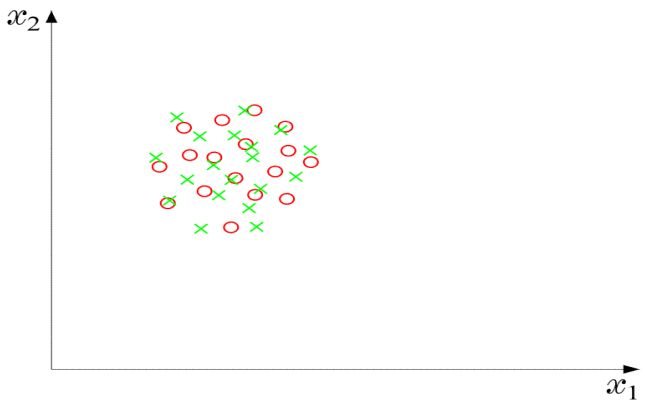
- The goals:
  - Select the "optimum" number, l , of features
  - Select the "best" l features

- Large l has a three-fold disadvantage:
  - High computational demands
  - Low generalisation performance
  - Poor error estimates

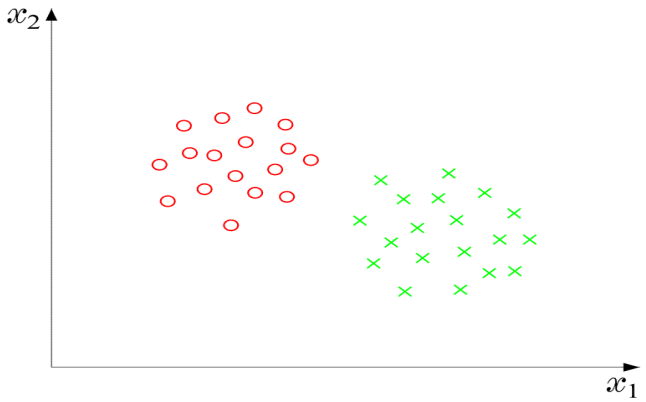


- l must be large enough to learn:
  - what makes classes different
  - what makes patterns in the same class similar
- l must be small enough not to learn what makes patterns of the same class different

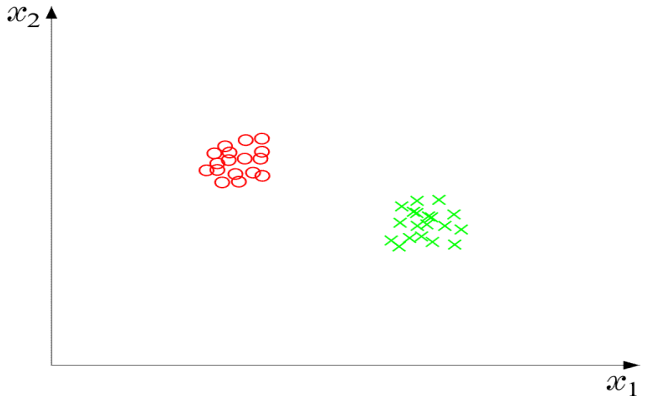
- Once l has been decided, choose the l most informative features
  - Best: **Large** between class distance, **Small** within class variance



Bad choice

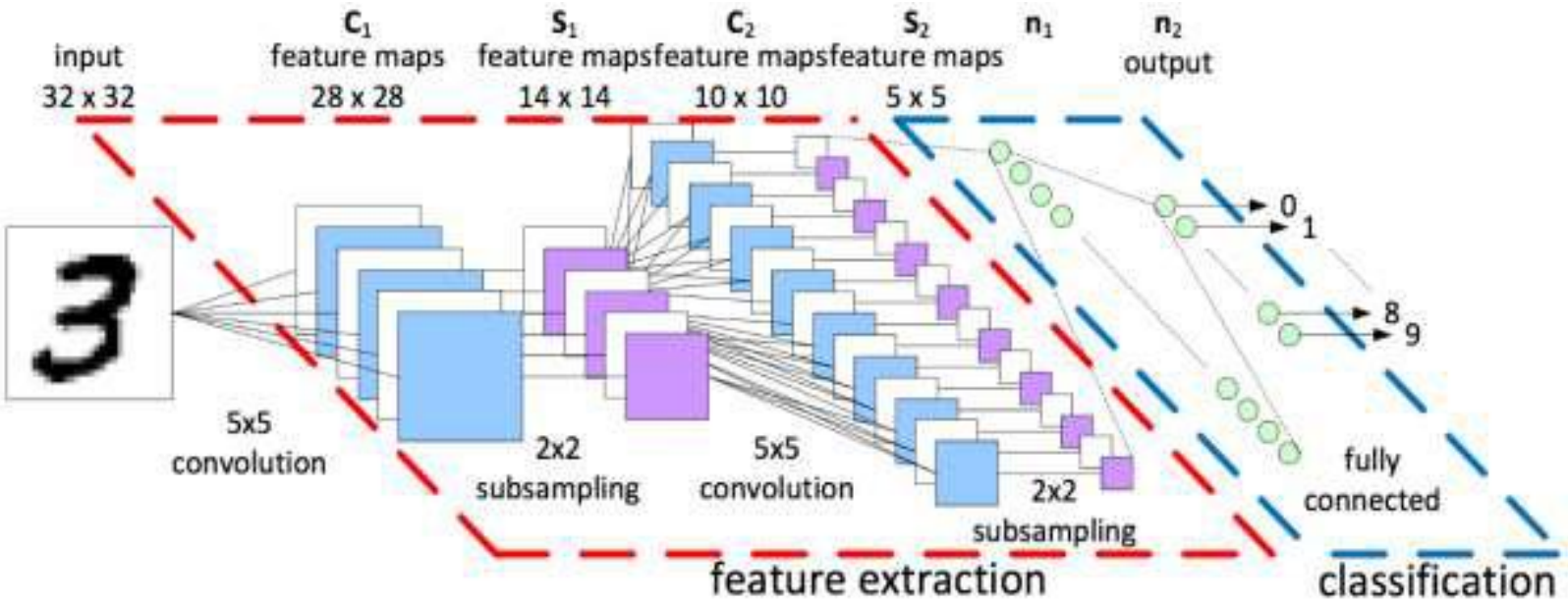


Not bad choice

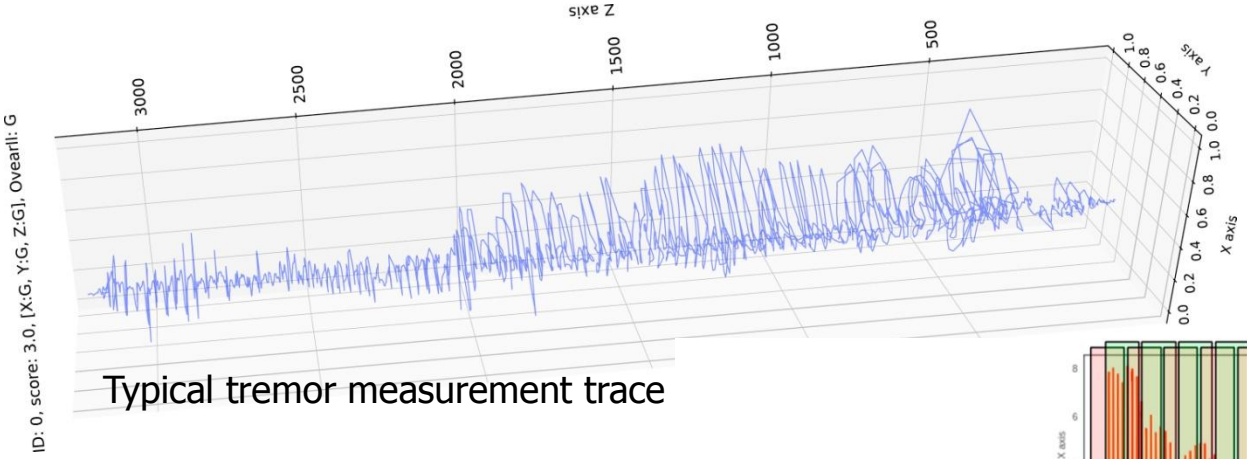


Good choice

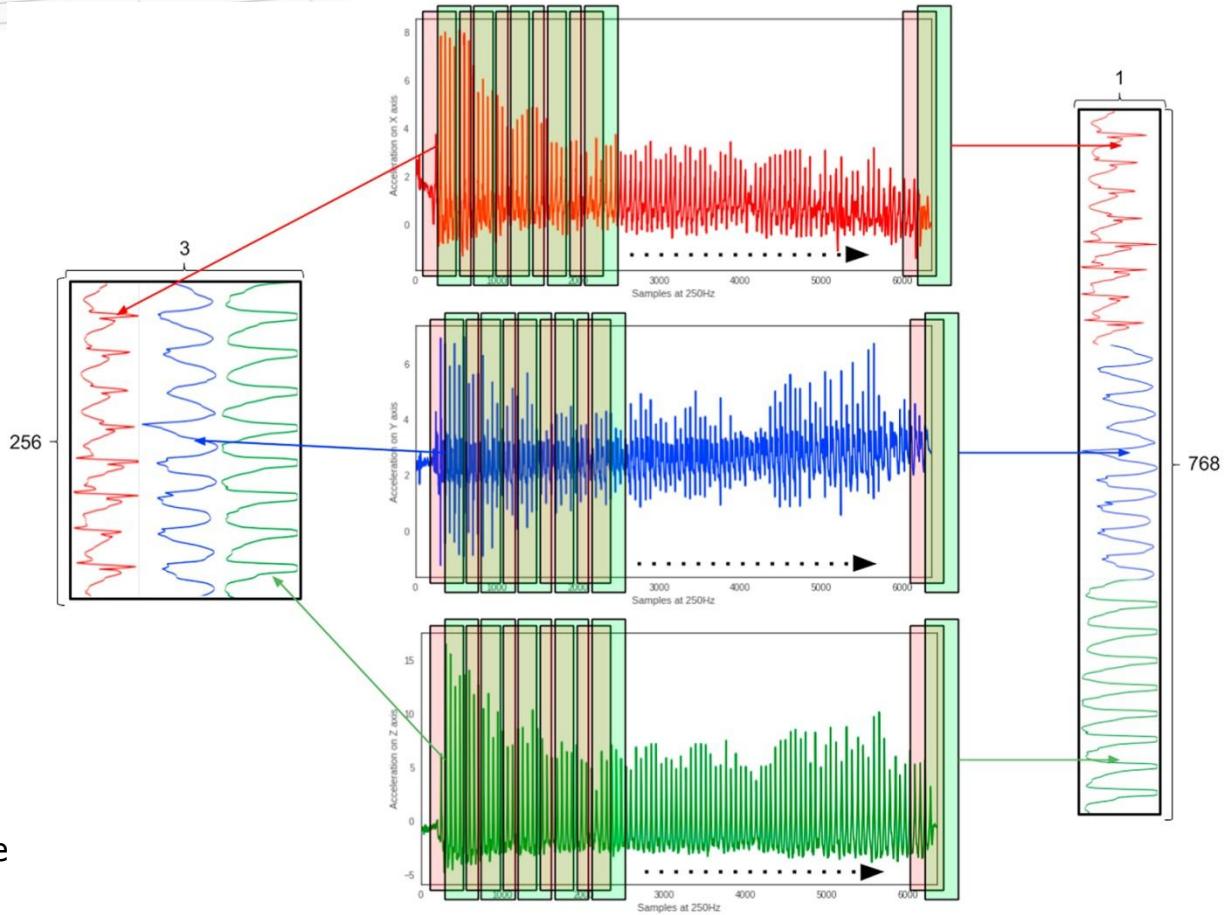
Another approach- features extracted as part of training process in deep learning methods



Parkinson's tremor measurements

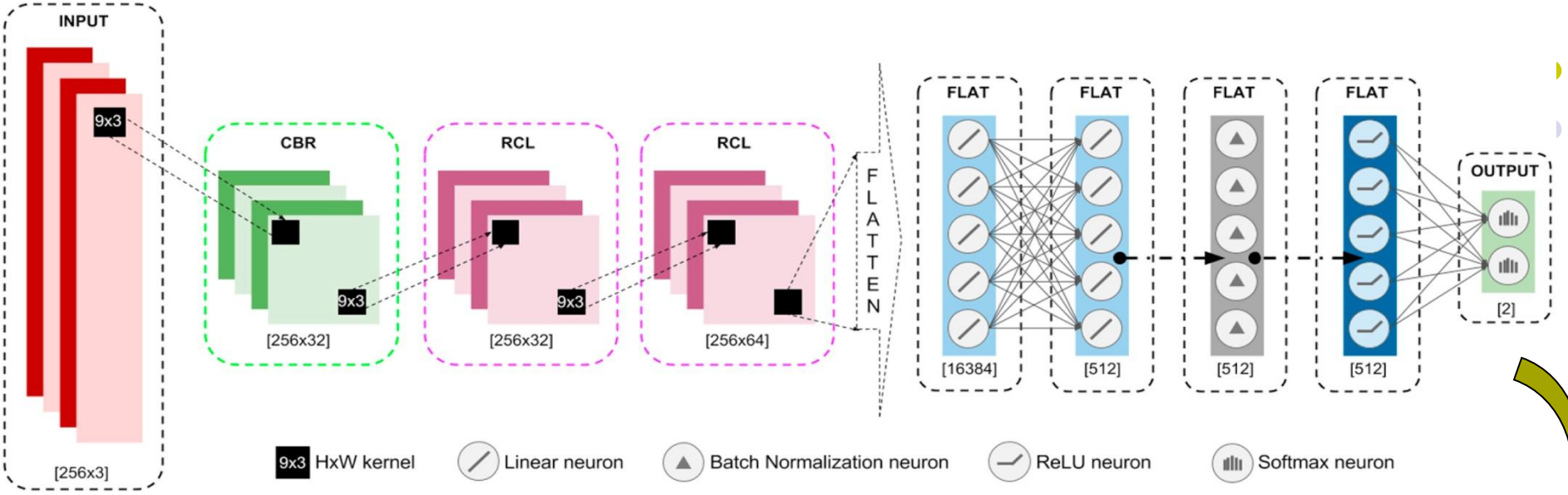


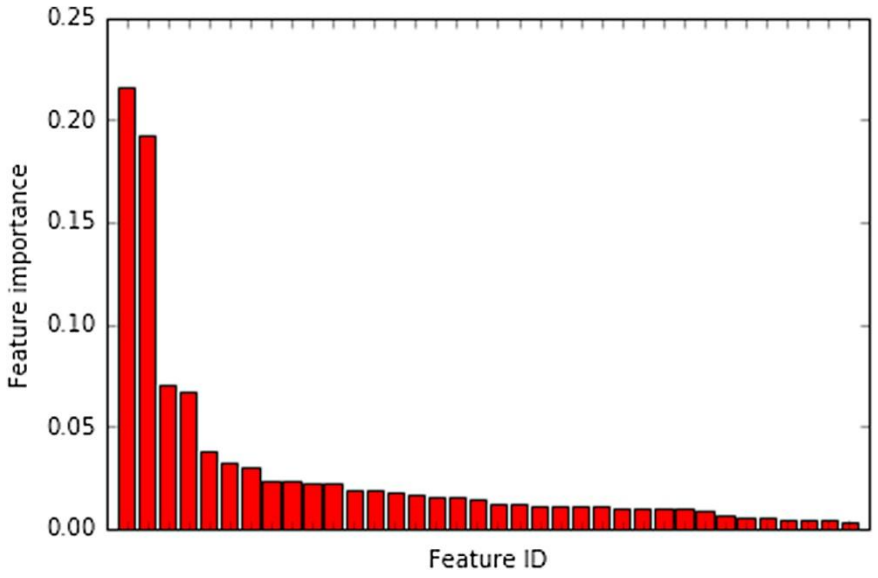
Typical tremor measurement trace



signal segments along the x, y, z acceleration axes

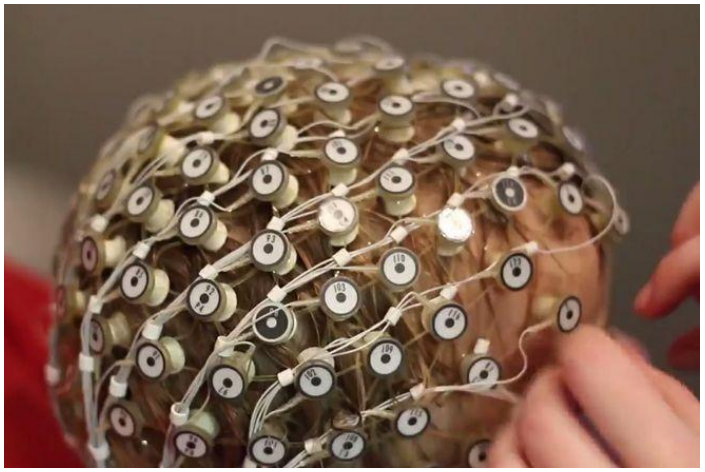
The cloudUPDRS app: A medical device for the clinical assessment of Parkinson's Disease
https://doi.org/10.1016/j.pmcj.2017.12.005



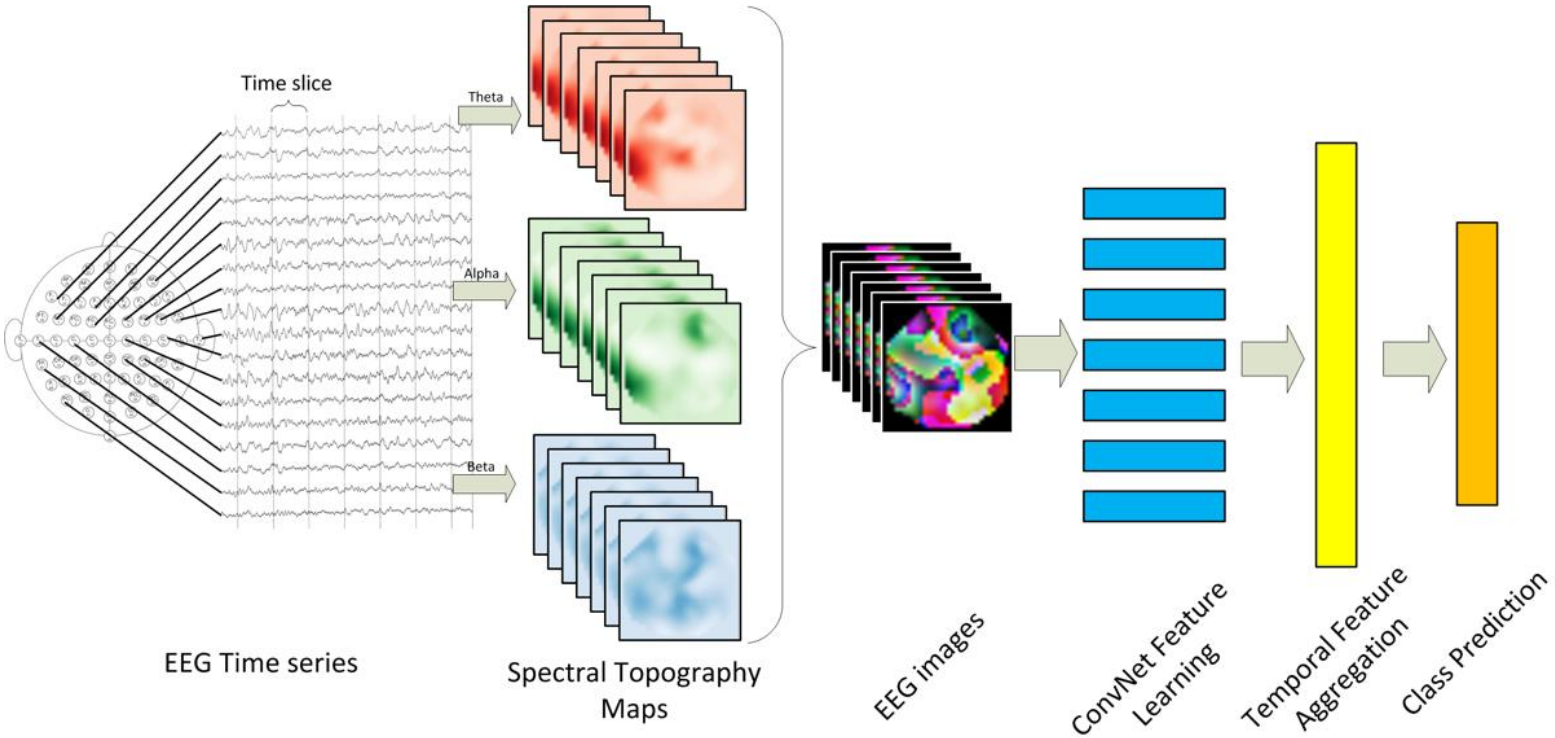


Select features that offer the highest predictive power of the patient's overall motor performance

Two and three features per tremor and bradykinesia test respectively. Select the subgroup of top performing features which account for at least 80% of the variance in the overall UPDRS score.



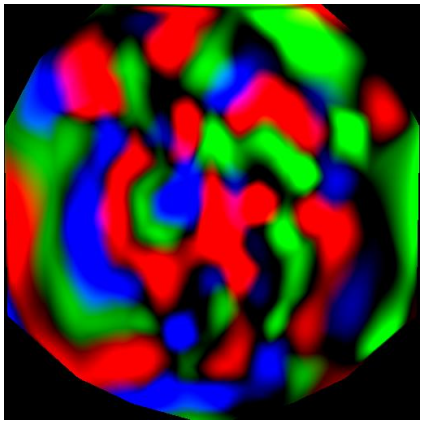
transform EEG activities into a sequence of topology-preserving multi-spectral images



Cognitive load classification

Learning Representations from EEG with Deep Recurrent-Convolutional Neural Networks- https://arxiv.org/abs/1511.06448

EEG hyper-connectivity for early detection of autism

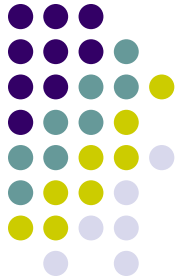


- 65% (imbalanced classes)
- 80% (balanced)

Wavelet decomposition of the signal:
Use wavelet coefficients as features

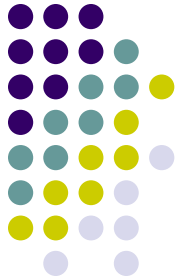
- 94% (balanced)

Generalisation



- How does the classifier perform on a problem with respect to both seen and unseen data?
- How many training instances are required for good generalisation?
- What size classifier gives the best generalisation?
- What kind of neural architecture is best for modelling the underlying problem?
- What learning algorithm can achieve the best generalisation?

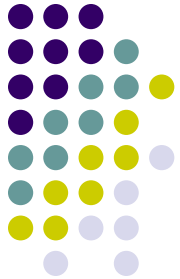
Generalisability



- **Generalisation theory** aims at providing general bounds that relate the error performance of a classifier with the number of training points, N, on one hand, and some classifier dependent parameters, on the other. So far, the classifier dependent parameters that we considered were the number of free parameters of the classifier and the dimensionality of the subspace, in which the classifier operates.

- **Generalisability** is the ability of a model to generalise on unseen data points (not used during training). A model is not effective if it is too specific to the training data, i.e., "too specific" means it is over-fitting and its performance on unseen data cannot be guaranteed.

Evaluating generalisability



- One systematic method to check for this is cross-validation. Cross-validation is a strategy where the model-building process is applied on a subset of the data such that the model has not "seen" all the data available. This learning process is then followed by an evaluation step on the "unseen" portion of the data. This is an approach to ensure that the model is not over-fitted to the data and an acceptable degree of generalisability is ensured.

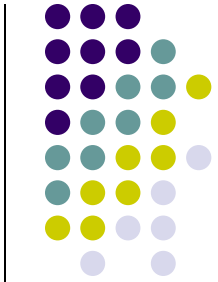
- One approach is **the holdout method** where the data set is separated into two random sets, called the training set and the testing set. And the model is trained on the former and evaluated on the latter.
- Another version is the **k-fold cross-validation** approach, where the data set is divided into k subsets, and the holdout method is repeated k times. Each time, one of the k subsets is used as the test set and the other k-1 subsets are put together to form a training set. Then the average error across all k trials is computed.

Boosting

Technique first introduced in [Schapire (1990), JML; Schapire, Freund (1996), Experiments with a new boosting algorithm, 13th ICML]

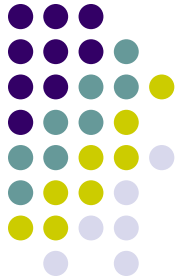
- *Learners (models) are trained sequentially, using a sample from the original dataset, with the prediction error from the previous round affecting the sampling weight for the next round.*
- *After each round of boosting, the decision can be made to terminate and use a set of calculated weights to apply as a linear combination of the newly created set of learners.*

Uncertainty



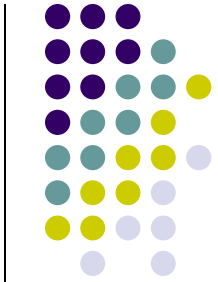
Uncertainty *can be defined as the lack of exact knowledge that would enable us to reach a perfectly reliable solution.*

What are the sources of uncertainty ?



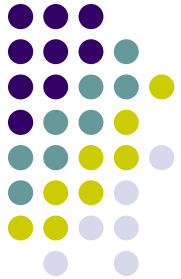
- **Weak implications.** In KBS, domain experts and knowledge engineers have the painful task of establishing concrete correlations between IF (condition) and THEN (action) parts of the rules. Therefore, KBS need to have the ability to handle vague associations, for example by accepting the degree of correlations as numerical certainty factors.

What are the sources of uncertainty ?



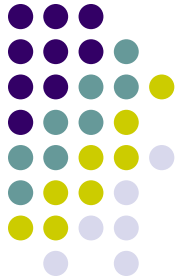
- **Imprecise language.** Our natural language is ambiguous and imprecise. We describe facts with such terms as *often* and *sometimes*, *frequently* and *hardly ever*. As a result, it can be difficult to express knowledge in the precise IF-THEN form of production rules. However, if the meaning of the facts is quantified, it can be used in KBS.

What are the sources of uncertainty ?



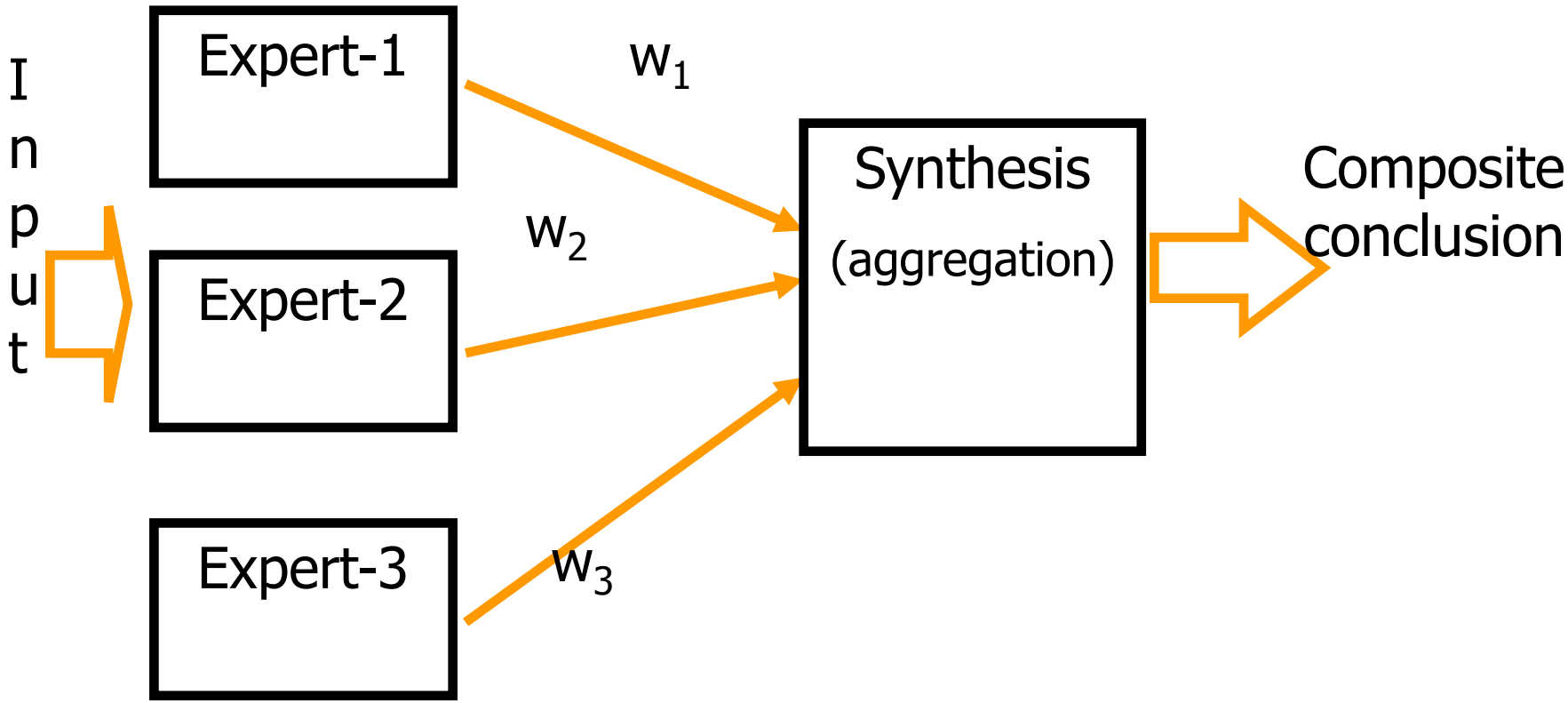
- **Unknown data.** When the data is incomplete or missing, the only solution is to accept the value "unknown" and proceed to an approximate reasoning with this value.

What are the sources of uncertainty ?

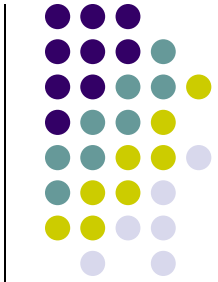


- **Combining the views of different experts/models.** AI systems usually combine the knowledge and expertise of a number of experts or models. Unfortunately, experts often have contradictory opinions and produce conflicting rules; sometime models do the same. To resolve the conflict, we can attach a weight to each "expert" and then calculate the composite conclusion/consensus.

Combining the views of different experts



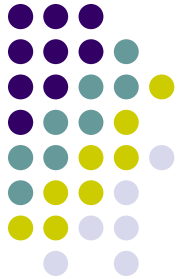
Probabilistic approaches



Bayesian approaches- advantages, drawbacks

- Very popular for modelling
- Delivers fast results and performs well in many domains
- Assumes events or objects to be conditionally independent
- Needs at least a few examples to induce a rough hypothesis; where do prior probabilities come from?

Reasoning under uncertainty and the role of probability

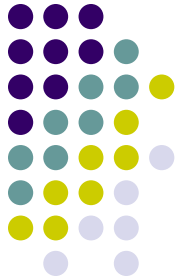


The probability of success and failure:
P(success)= the number of successes / the number of possible outcomes

P(failure)= the number of failures / the number of possible outcomes

The probability of getting a 6 from a single throw of a dice. If we assume 6 as the only success then the number of successes is 1 and the number of possible outcomes is 1+5 (there are five ways of not getting a 6 in a single throw)
P(success)=1/(1+5)=0.1666
P(failure)=5/(1+5)=0.8333

Reasoning under uncertainty and the role of probability



Let A be an event and B be another event. Suppose that both can happen simultaneously but occur conditionally on the occurrence of the other.

Conditional probability: the probability that A will occur if B occurs

$$p(\mathrm{A}\,|\,\mathrm{B}) = \frac{\text{the number of times A and B can occur}}{\text{the number of times B can occur}} = \frac{p(\mathrm{A}\cap\mathrm{B})}{p(\mathrm{B})}$$

Joint probability: the probability that both A and B will occur

$$p(\mathrm{A}\cap\mathrm{B}) = p(\mathrm{A}\,|\,\mathrm{B})\times p(\mathrm{B}) = p(\mathrm{B}\,|\,\mathrm{A})\times p(\mathrm{A}) = p(\mathrm{B}\cap\mathrm{A})$$

Bayesian rule

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

where:

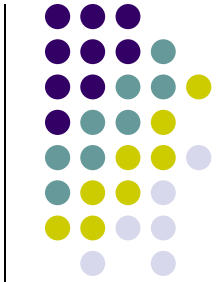
$p(A|B)$ is the conditional probability that event A occurs given that event B has occurred;

$p(B|A)$ is the conditional probability of event B occurring given that event A has occurred;

$p(A)$ is the probability of event A occurring;

$p(B)$ is the probability of event B occurring.

Bayesian reasoning



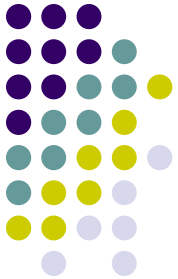
Suppose all rules in the knowledge base are represented in the following form:

IF *E* is true
THEN *H* is true {with probability *p*}

This rule implies that if event *E* occurs, then the probability that event *H* will occur is *p*.

In knowledge-based systems, *H* usually represents a hypothesis
and *E* denotes evidence to support this hypothesis.

The Bayesian rule expressed in terms of hypotheses and evidence looks like this:



$$p(H|E) = \frac{p(E|H) \times p(H)}{p(E|H) \times p(H) + p(E|\neg H) \times p(\neg H)}$$

where:

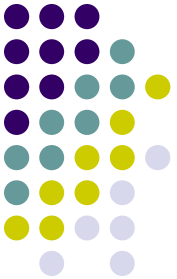
$p(H)$ is the prior probability of hypothesis H being true;

$p(E|H)$ is the probability that hypothesis H being true will result in evidence E;

$p(\neg H)$ is the prior probability of hypothesis H being false;

$p(E|\neg H)$ is the probability of finding evidence E even when hypothesis H is false.

how are all these used in KBS?



- The probabilities required to solve a problem are provided by experts. An expert determines the **prior probabilities** for possible hypotheses $p(H)$ and $p(\neg H)$, and also the **conditional probabilities** for observing evidence E if hypothesis H is true, $p(E|H)$, and if hypothesis H is false, $p(E|\neg H)$.
- Users provide information about the evidence observed and the expert system computes $p(H|E)$ for hypothesis H in light of the user-supplied evidence E. Probability $p(H|E)$ is called the **posterior probability** of hypothesis H upon observing evidence E.
- *What happens when there are many hypotheses and evidence observations?*

- We can take into account both multiple hypotheses H_1, H_2,..., H_m and multiple evidences E_1, E_2,..., E_n. The hypotheses as well as the evidences must be mutually exclusive and exhaustive.
- Single evidence E and multiple hypotheses follow:

$$p(H_i|E) = \frac{p(E|H_i) \times p(H_i)}{\sum_{k=1}^{m} p(E|H_k) \times p(H_k)}$$

- Multiple evidences and multiple hypotheses follow:

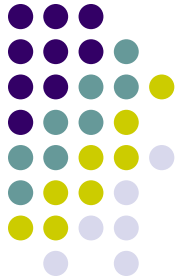
$$p(H_i|E_1\,E_2 \ldots E_n) = \frac{p(E_1\,E_2 \ldots E_n|H_i) \times p(H_i)}{\sum_{k=1}^{m} p(E_1\,E_2 \ldots E_n|H_k) \times p(H_k)}$$

Is all that realistic?

- This requires to obtain the conditional probabilities of all possible combinations of evidences for all hypotheses, and thus places an enormous burden on the expert.
- Therefore, in practice (e.g. in KBS), conditional independence among different evidences is assumed. Thus, instead of the *unworkable equation*, we attain:

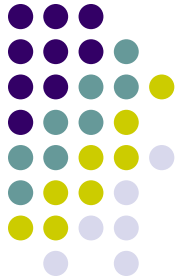
$$p(H_i|E_1\,E_2\,\ldots E_n)=\frac{p(E_1|H_i)\times p(E_2|H_i)\times\ldots\times p(E_n|H_i)\times p(H_i)}{\sum_{k=1}^{m} p(E_1|H_k)\times p(E_2|H_k)\times\ldots\times p(E_n|H_k)\times p(H_k)}$$

Bias of the Bayesian method



- The framework for Bayesian reasoning requires probability values as primary inputs. The assessment of these values usually involves human judgement. However, psychological research shows that humans cannot elicit probability values consistent with the Bayesian rules.
- This suggests that the conditional probabilities may be inconsistent with the prior probabilities given by the expert.
- Use a lot of data to estimate probability.

From probabilities to fuzzy logic



- In 1965 Lotfi Zadeh, published his famous paper "Fuzzy sets". Zadeh proposed a formal system of mathematical logic, and introduced a new concept for applying natural language terms. This new logic for representing and manipulating fuzzy terms was called **fuzzy logic.**

Fuzzy logic is a set of mathematical principles for knowledge representation based on degrees of membership.

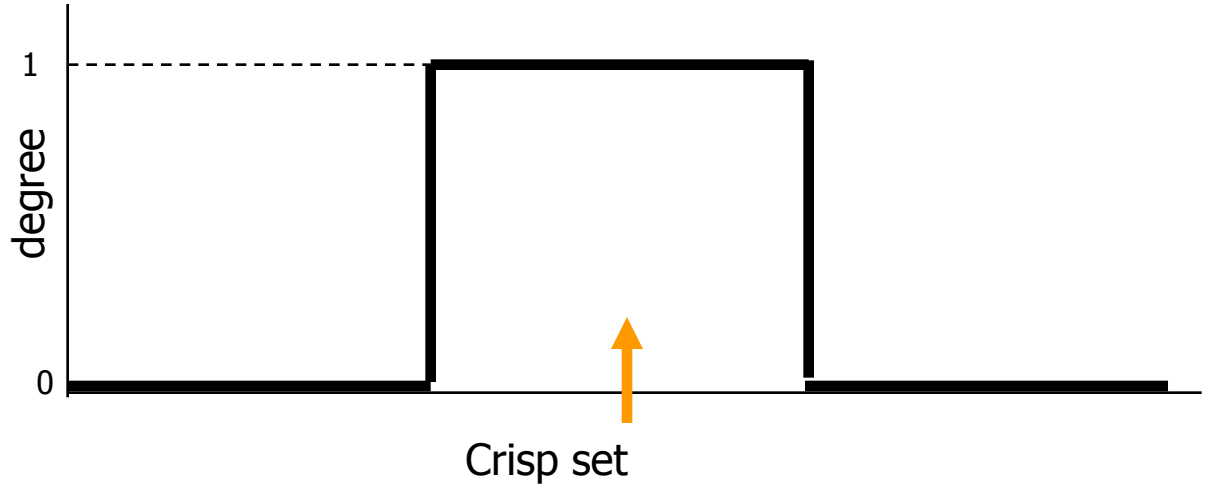
Unlike two-valued Boolean logic, fuzzy logic is **multi-valued**. It deals with **degrees of membership** and **degrees of truth**. Fuzzy logic uses the continuum of logical values between 0 (completely false) and 1 (completely true). Instead of just black and white, it employs the spectrum of colours, accepting that things can be partly true and partly false at the same time.

Crisp sets and fuzzy sets

Crisp set X= $\{x_1,x_2,x_3,x_4,x_5\}$

Fuzzy set X= $\{(x_1,0), (x_2,1), (x_3,1), (x_4,0), (x_5,0)\}$

It is a set of pairs $\{(x_i, \mu_X(x_i)\}$, where $\mu_X(x_i)$ is the membership function of element x_i in the set X.

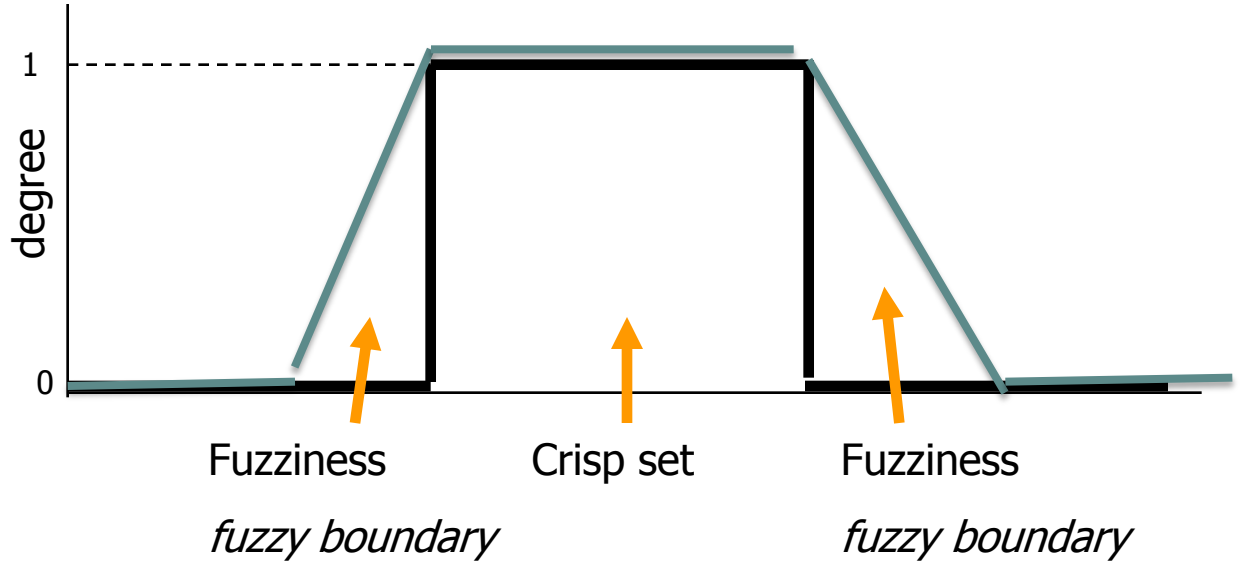


Crisp sets and fuzzy sets

Crisp set X= $\{x_1,x_2,x_3,x_4,x_5\}$

Fuzzy set X= $\{(x_1,0), (x_2,1), (x_3,1), (x_4,0), (x_5,0)\}$

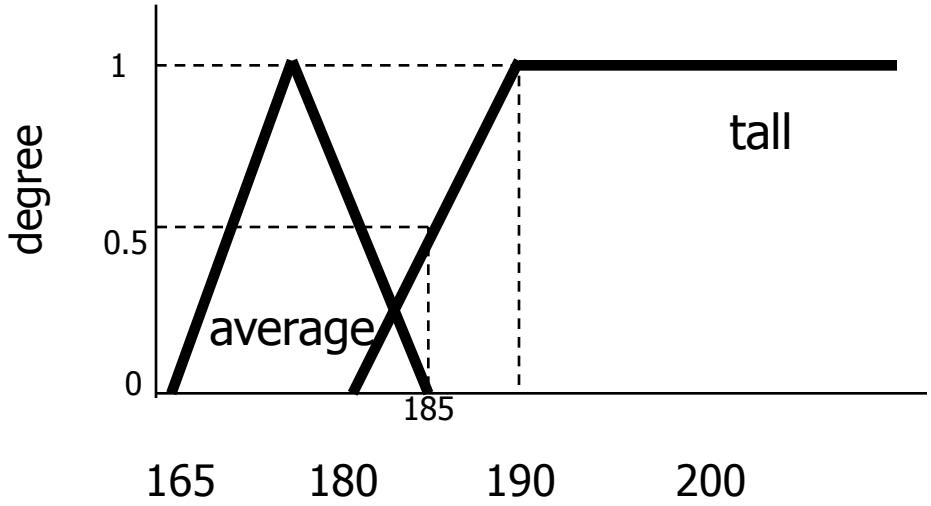
It is a set of pairs $\{(x_i, \mu_X(x_i)\}$, where $\mu_X(x_i)$ is the membership function of element x_i in the set X.



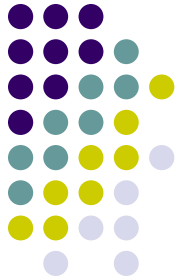
Crisp sets and fuzzy sets

Fuzzy sets: *tall men*= {0/180, 0.5/185, 1/190}

average men= {0/165, 1/175, 0/185}



Linguistic variables: some examples



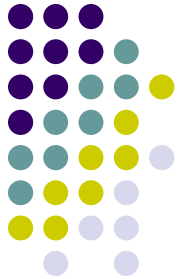
HOW CAN WE DECIDE ON THE RANGE AND THE TERMS USED?

Reading speed, counted from user's self-paced reading, can be determined by computing the average reading time per 100 word.

Use universe of discourse that is from 15 to 65 sec.

It can be associated with linguistic values {Slow, Medium, Fast}. Medium speed is around 35 sec

What is the difference between classical and fuzzy rules?



A classical IF-THEN rule uses binary logic, for example,

Rule: 1
IF speed is > 100
THEN stopping_distance is long

Rule: 2
IF speed is < 40
THEN stopping_distance is short

The variable *speed* can have any numerical value between 0 and 220 km/h, but the linguistic variable *stopping_distance* can take either value *long (300m)* or *short (0m)*. In other words, classical rules are expressed in the black-and-white language of Boolean logic.

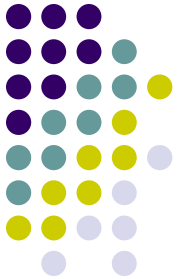
We can also represent the stopping distance rules in a fuzzy form:

Rule: 1
IF speed is fast
THEN stopping_distance is long

Rule: 2
IF speed is slow
THEN stopping_distance is short

In fuzzy rules, the linguistic variable *speed* also has the range (the universe of discourse) between 0 and 220 km/h, but this range includes fuzzy sets, such as *slow*, *medium* and *fast*. The universe of discourse of the linguistic variable *stopping_distance* can be between 0 and 300 m and may include such fuzzy sets as *short*, *medium* and *long*.

Neural computing



- A **neural network** can be defined as a model of reasoning based on the human brain. The brain consists of a densely interconnected set of nerve cells, or basic information-processing units, called **neurons**.

EE Times
Connecting the Global Electronics Community
designlines MCU

News & Analysis

Qualcomm Reveals Neural Network Progress

R. Colin Johnson

10/11/2013 03:40 PM EDT
24 comments

PORTLAND, Ore. — Biologically inspired neural processing units (NPUs) were recently described by Qualcomm Inc. in San Diego at the MIT Technology Review's EmTech conference. Qualcomm chief technology officer (CTO) Matt Grob described a new generation of NPUs and design tools that Qualcomm hopes to make available to developers next year.

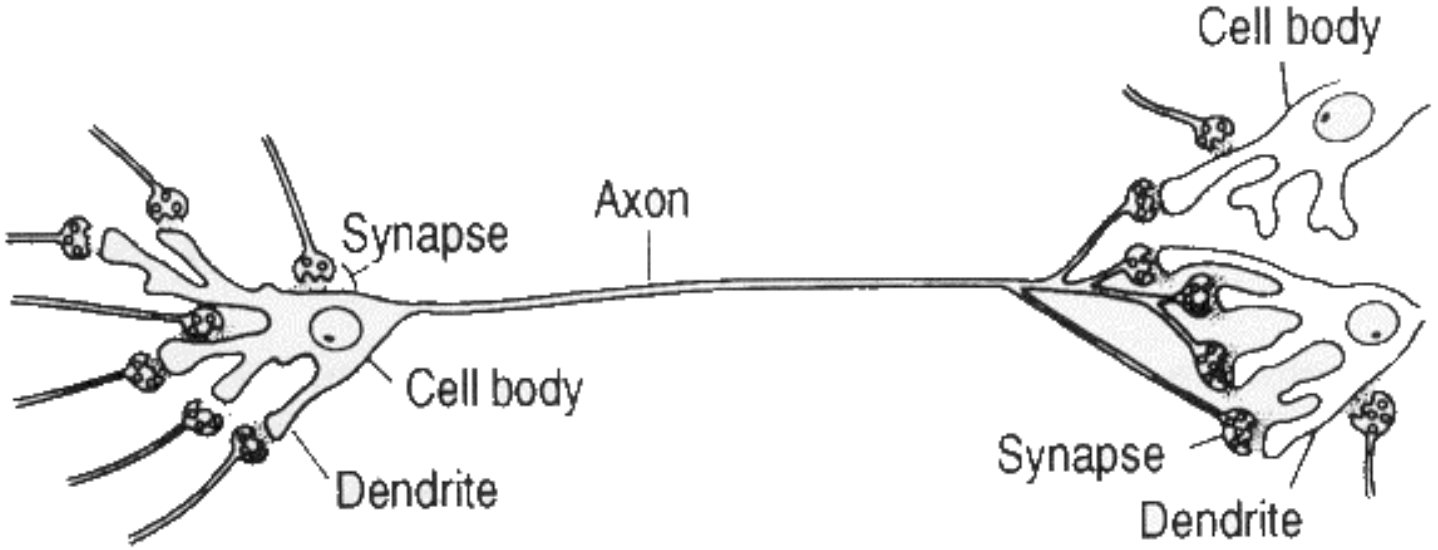
Purdue University researchers also use neural networks to create an image-processing application that can categorize objects from a moving car in real-time.
(Source: Eugenio Culurciello, Purdue University/Qualcomm at MIT's EmTech)

At the conference, Grob showed videos of what it calls its Zeroth Robot prototype -- named after Isaac Asimov's Zeroth Law of Robotics (that no robots shall harm humanity). These robots were not powered by a conventional computer but instead by biologically inspired NPUs modeled on the human brain and created in cooperation with Brain Corp, which receives funding from Qualcomm Ventures and operates its labs inside Qualcomm's facility.

According to Grob, even though these early prototypes are general-purpose image

Neural computing fundamentals

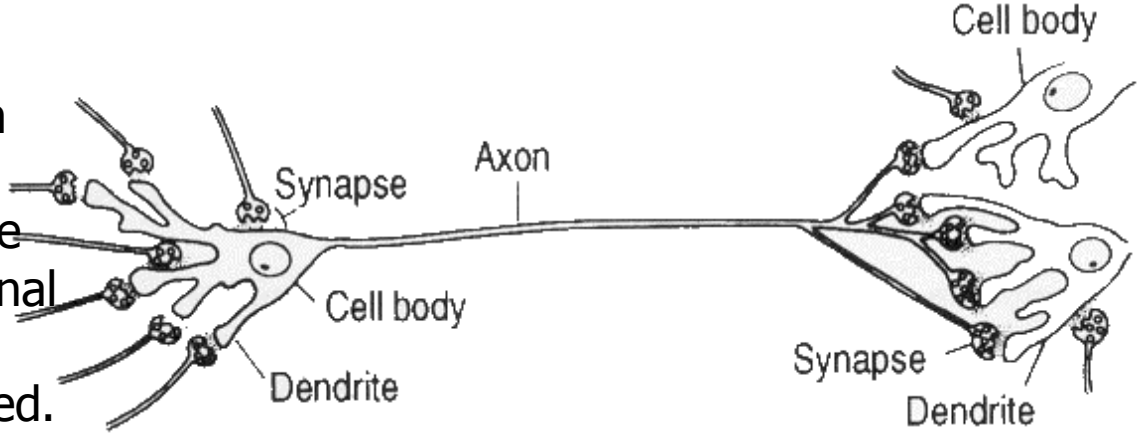
Biological neurons: Input is received by the dendrites from other neurons. If the sum of these inputs is sufficiently large the neuron fires, passing a signal down its axon to the other neurons to which it is connected.



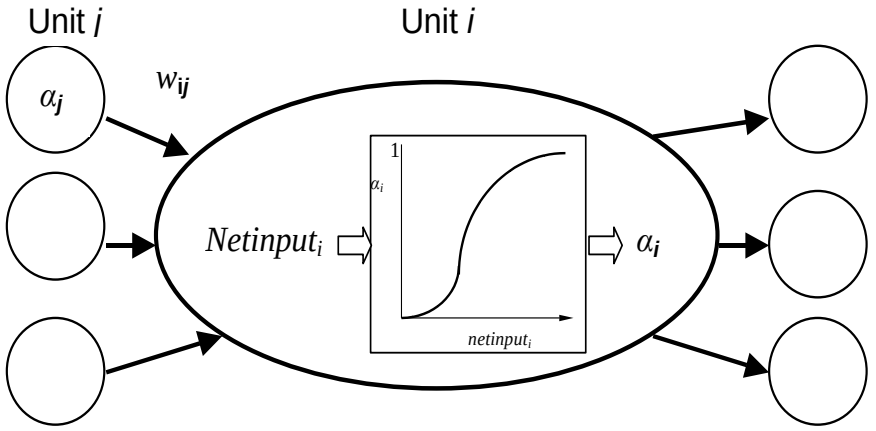
- Our brain can be considered as a highly complex, non-linear and parallel information-processing system.
- Information is stored and processed in a neural network simultaneously throughout the whole network, rather than at specific locations. In other words, in neural networks, both data and its processing are **global** rather than local.
- Learning is a fundamental and essential characteristic of biological neural networks. The ease with which they can learn led to attempts to emulate a biological neural network in a computer.

Neural computing fundamentals

Biological neurons: Input is received by the dendrites from other neurons. If the sum of these inputs is sufficiently large the neuron fires, passing a signal down its axon to the other neurons to which it is connected.



Artificial neuron: The computational operation performed by a unit in an artificial neural network. The operation can be broken into three steps: (1) integrate the inputs from the previous layer to create a *netinput*; (2) use an activation function to convert the *netinput* to an activation level; (3) output the activity level as input to the units in the next layer.

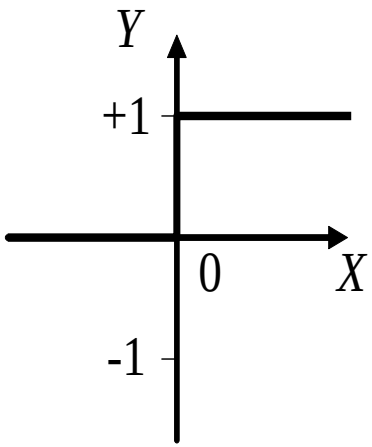
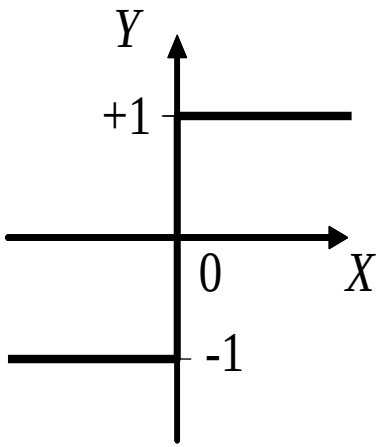
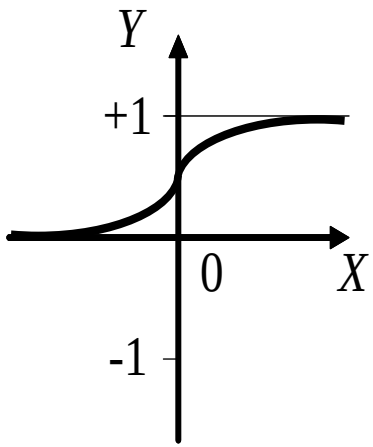
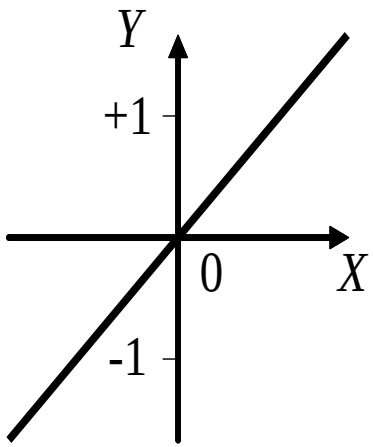


Integrate input from previous layer

Transform *netinput* to activity level (α_i)

Transmit activity level to units in next layer

Activation functions of a neuron

Step function	*Sign function*	*Sigmoid function*	*Linear function*
			
$Y^{step} = \begin{cases} 1, \text{ if } X \geq 0 \\ 0, \text{if } X < 0 \end{cases}$	$Y^{sign} = \begin{cases} +1, \text{if } X \geq 0 \\ -1, \text{if } X < 0 \end{cases}$	$Y^{sigmoid} = \dfrac{1}{1+e^{-X}}$	$Y^{linear} = X$

Fundamental of neural networks

Neural computing assumptions about computation in the brain

1. **Neurons integrate information**. The neuron receives signals, either excitatory or inhibitory, from other neurons via synaptic connections onto the dendrites. If the sum of these signals exceeds a threshold, the neuron fires. This is communicated to other neurons by a signal passing down its axon. This signal acts as part of the input to the dendrites of the other neurons.

2. **Neurons pass information about the level of their input**. Each unit has an activity level, which is related to the input level- the higher the input, the higher the activity. The activity level is transmitted as a single value to all the units to which it is connected.

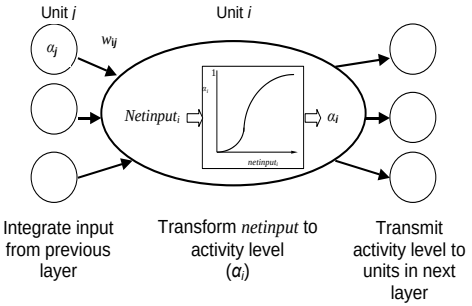
3. **Brain structure is layered**. Information is processed in the brain by a flow of activity passing through a sequence of physically independent structures. Example: the visual processing system.

4. **The influence of one neuron on another depends on the strength of the connection between them**. The effect of one neuron on another (whether it makes it much more or less likely to fire or whether it only slightly changes the probability) is determined by the strength of the synaptic connection between them, which is called the weight of the connection.

5. **Learning is achieved by changing the strengths of connections between neurons**. Experience can change the behaviour of an organism in response to a particular stimulus. Learning is implemented by rules, which determine how the weights of the connections between units are changed.

Symbols and elementary equations

Learning by weight change: If the response of an output unit is incorrect then the network can be changed so that it is more likely to produce the correct response the next time that the stimulus is presented. This is achieved by changing the connection weights.



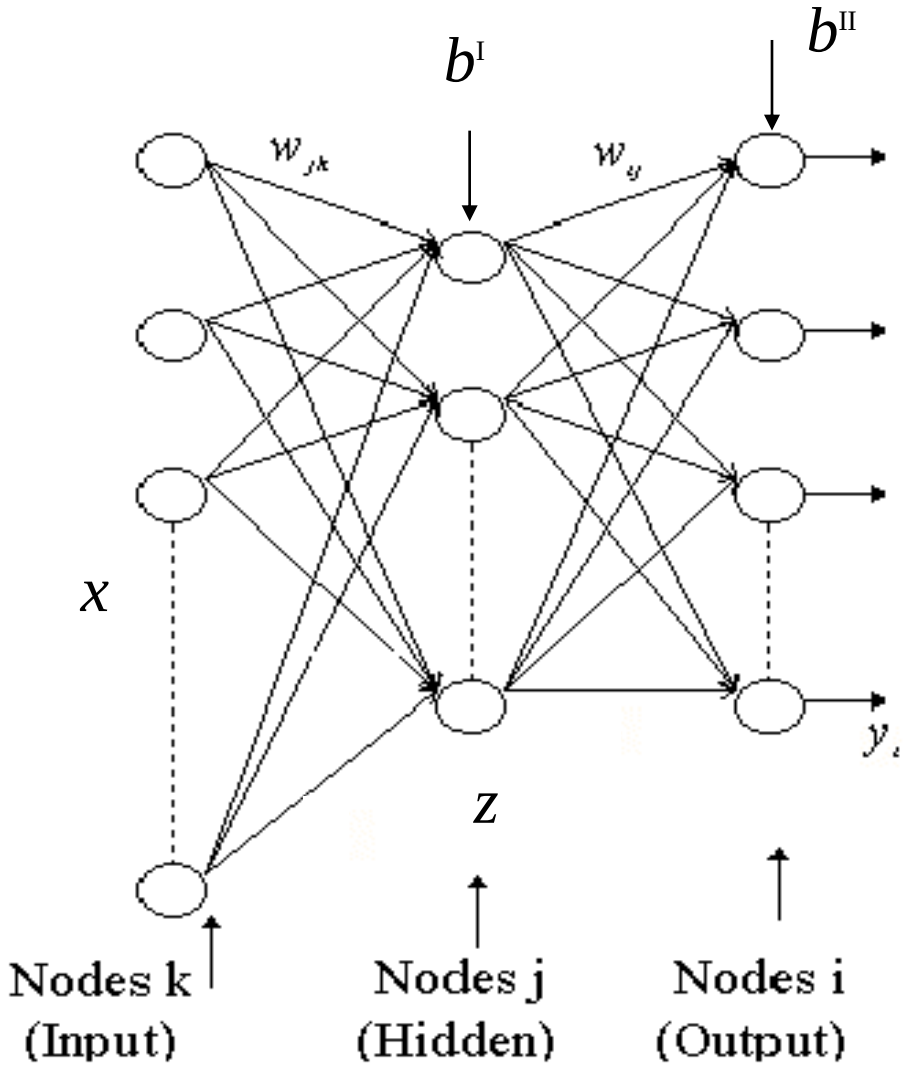
$$w_{ij}^{new} = w_{ij}^{old} + \Delta w_{ij}$$

$$\Delta w_{ij} = -\eta \frac{dE}{dw_{ij}} = -\eta \frac{d\left[a_i^{desired} - a_i^{obtained}\right]^2}{dw_{ij}}$$

Δw_{ij} is the change in the connection weight w_{ij} from unit *j* to unit *i*

Bias: There is one special input unit, which is called bias unit. The bias unit receives no input itself, and its activity is always set at +1. The weight from the bias unit to the unit of interest can be positive or negative and changes just like any other weight during learning.

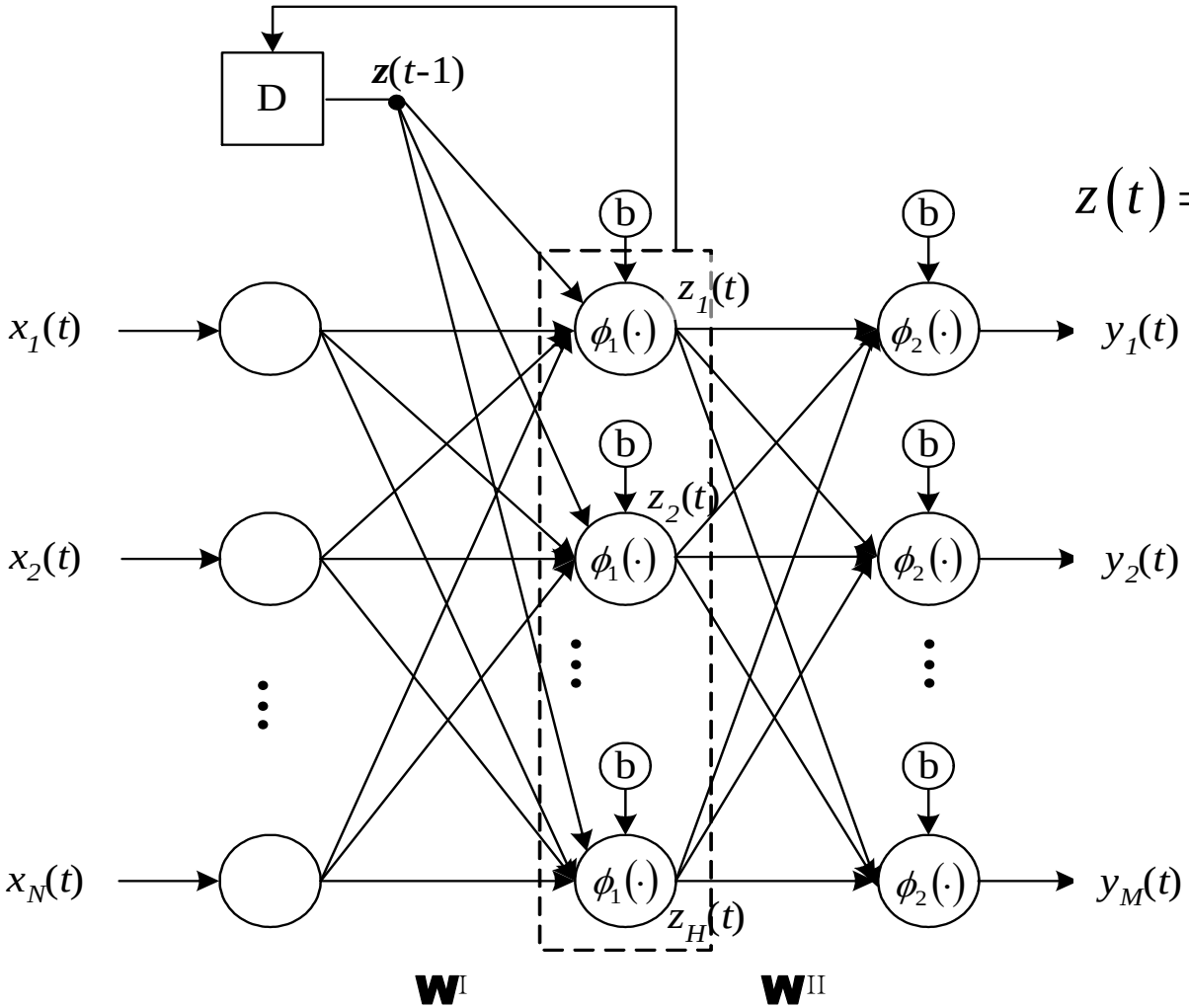
Feed-forward network



$$z = \phi\left(\mathbf{W}^{\mathrm{I}} x + b^{\mathrm{I}}\right),$$

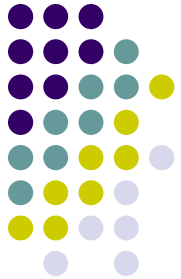
$$y = \phi\left(\mathbf{W}^{\mathrm{II}} z + b^{\mathrm{II}}\right),$$

Layered recurrent network



$$z(t) = \phi\left(\Lambda z(t-1) + \mathbf{W}^{\mathrm{I}} x(t) + b^{\mathrm{I}}\right),$$

Λ is a diagonal matrix

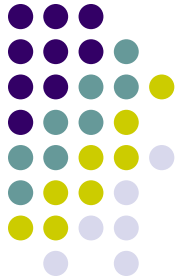


Error correction (perhaps the most popular)

$$E(w) = \sum_{p=1}^{P} \left| y_p - t_p \right|^2$$

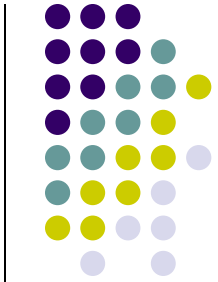
One weakness of the sum-of-squared-error function is that it is influenced by the relative magnitude of the input attributes. Therefore, a process of normalization, the exact way of which depends on the nature of the problem and on domain knowledge, is often necessary. Nevertheless, the choice $r = 2$ has become very popular because it can be shown that under some assumptions it minimises both the sum-of-squared-error and the probability of prediction error

Evolution and learning



- Evolution is a type of adaptation that captures relatively slow environmental changes that involves several generations, i.e. evolution operates at the *phylogenetic level* (evolutionary relationships among biological entities - often species, individuals or genes).
- Learning includes various set of mechanisms that lead to adaptive changes in an individual during its lifetime, i.e. learning operates on the *ontogenetic level* (developmental history of an organism within its own lifetime).

Evolution and learning



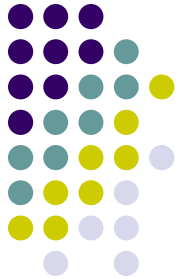
- The key concept is that **what a species must initially learn during each individual's lifetime, can overtime become part of the genetic makeup of that species**, i.e. what is initially learned eventually becomes innate
- The structure of all cognitive abilities that we possess like language acquisition, reasoning arise from the interactions between learning and evolution.

Genetic computing

- GAs consist of a finite repetition of the three steps:
  - Selection of the parent strings
  - Recombination
  - Mutation

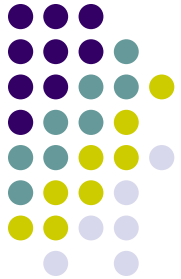
```
begin
    t ← 0
    initialise P(t)
    evaluate P(t)
    while (not termination condition) do
    begin
        t ← t +1
        select P(t) from P(t -1)
        alter P(t)
        evaluate P(t)
    end
end
```

Swarm Intelligence- 1



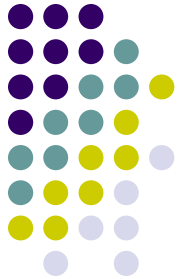
- A **swarm** can be defined as a structured collection of interacting organisms or agents.
- Within the computational study of swarm intelligence, individual organisms have included ants, bees, wasps, termites, fish (in schools) and birds (in flocks).
- Individuals in these swarms are relatively simple in structure, but their collective behaviour can become quite complex.
- The global behaviour of a swarm of social organisms emerges in a nonlinear manner from the behaviour of individuals in that swarm.

Swarm Intelligence - 2



- The behaviour of the swarm determines the conditions under which an individual performs actions. These actions may change the environment, and thus the behaviours of the that individual and its peers may also change.
- Interaction between individuals plays a vital role in shaping the swarm's behaviour.

Summary



- Introduction to basic concepts that will be used in the module.
- Overview of the topics covered in this module.

Useful Reading

....on the last slide of each lecture.

Negnevitsky, Artificial Intelligence: a Guide to Intelligent Systems, Chapters 1-2 (available at the BBK Library)- covers key concepts introduced and KBS.

Davis, Shrobe, Szolovits, What is Knowledge representation? *AI Magazine*, 14(1):17-33, 1993. Also online: http://medg.lcs.mit.edu/ftp/psz/k-rep.html

R. Rojas (1996), *Neural Networks-A Systematic Introduction.* Available online: http://page.mi.fu-berlin.de/rojas/neural/

Eiben A.E., Smith J.E. (2007), Introduction to Evolutionary Computing, Springer. Available online: https://docs.google.com/file/d/0By995HEqDrWQakRxNFc4OVVPQWc/edit?usp=sharing

Poli R., Kennedy J., Blackwell T. (2007), Particle Swarm Optimisation: an overview, Swarm Intelligence Journal, vol. 1, no. 1, 33-57. Available online: http://dces.essex.ac.uk/staff/rpoli/papers/PoliKennedyBlackwellSI2007.pdf

Next session

Uncertainty modelling and Fuzzy Logic