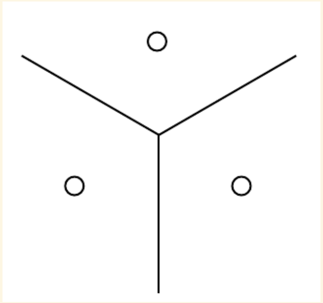
Lecture 2: Deep Learning: Motivation and Basics

Dr. Raghu Krishnapuram
Robert Bosch Centre for Cyber-Physical Systems
Indian Institute of Science, Bangalore
kraghu@iisc.ac.in

Challenges in traditional machine learning that miotivated deep learning

- Traditional methods unable to tackle problems such as image classification when no. of classes is in the thousands.
- Generalizing to new examples becomes exponentially difficult in high-dimensional spaces (curse of dimensionality).
- In very high dimensions, the input data space becomes sparse, and hence traditional machine learning fails (e.g. k-nn).
- Traditionally prior beliefs are introduced to address this problem (e.g., form of function, smoothness, local constancy).

Is smoothness alone sufficient?



Contiguous regions in k-nn

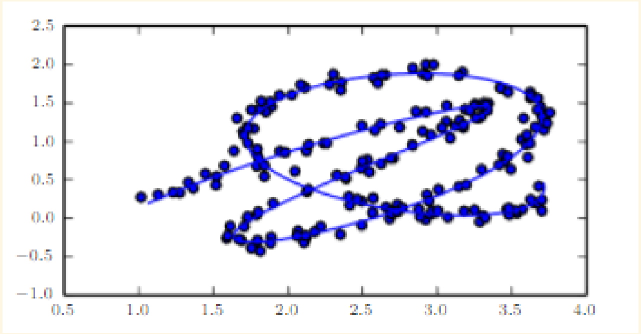
- Smoothness fails in higher dimensions, because the data is sparse.
  - In k-nn, the contiguous regions cannot grow faster than the number of training examples.
  - A decision tree cannot have more leaves than the number of training examples, and this can leave large regions undefined.

How to overcome sparse data issues?

- Roughly, we need $\mathcal{O}(k)$ examples to distinguish regions. For example, if we have a checkerboard pattern of regions, we need at least one sample per square to identify the regions, unless our prior belief allows us to make assumptions.
- In general, we need to introduce dependencies between regions based on assumptions about the probability distribution that generates the data.
- For example, assume that the data is generated by the composition of factors or features.

Manifolds

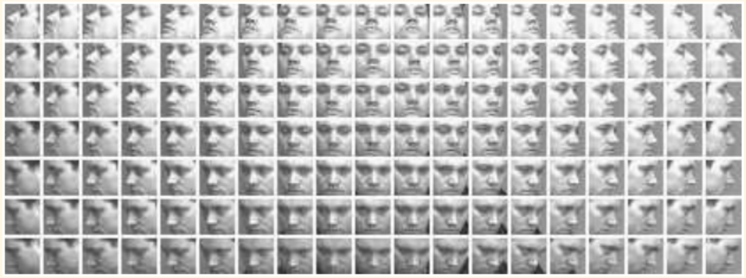
- Roughly speaking, a manifold is a lower-dimensional space embedded in a higher dimensional space.
  - E.g., a ribbon, the surface of a ball, a rod, etc.
- Sometimes the manifold may intersect itself, in which case the dimensionality is different at the intersection.



Example of a self-intersecting manifold

Manifold learning to the rescue?

- In many AI application such as recognition of images, text, and sounds, the assumption that data lies on a manifold seems valid.
  - Probability distributions are concentrated (e.g. text is not random).
  - Concentrated probability regions are connected through transormations (e.g., face rotation, light variations).
- Use of manifolds can bring down the number of parameters significantly. E.g. to fit a polynomial of order M, the number of coefficients tends to D^M.
- Manifold learning is an important topic in deep learning.



What is Deep Learning?

- Deep Learning - looking for solutions to "intuitive" problems that machines are not good at, e.g., recognizing faces, speech, etc.
- Approach is to model the world in terms of a hierarchy of concepts
  - There are limitations, since reasoning will need interactions among concepts in a network that is not hierarchical
- The hierarchy is typically deep - hence "deep learning"

What is Deep Learning? (Contd...)

- Challenge is to learn the “intuitive” knowledge from raw data.
- Ability to acquire knowledge by extracting patterns is “machine learning,” e.g., classification
- “Representation” is a critical task in machine learning and often done through features that are defined by humans
    - This is not always easy

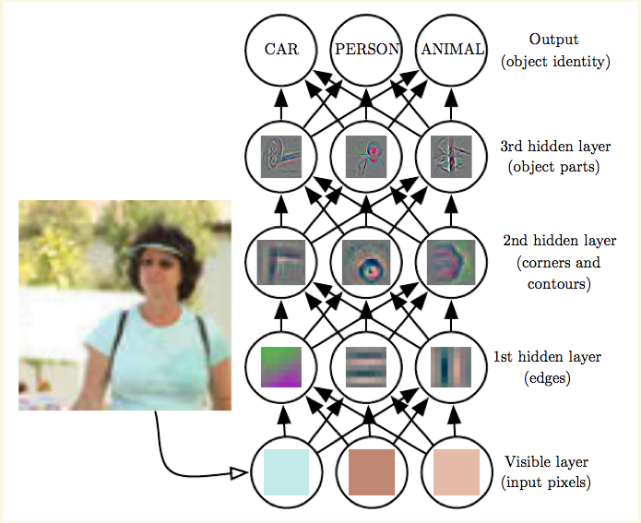
Representation Learning

- We need to disentangle the factors of variation and discard the ones we do not care about.
- An example (at a simple level) is auto-encoders or PCA.
- This task is difficult in general and may be difficult to achieve just using raw data.

Representation learning in Deep Learning

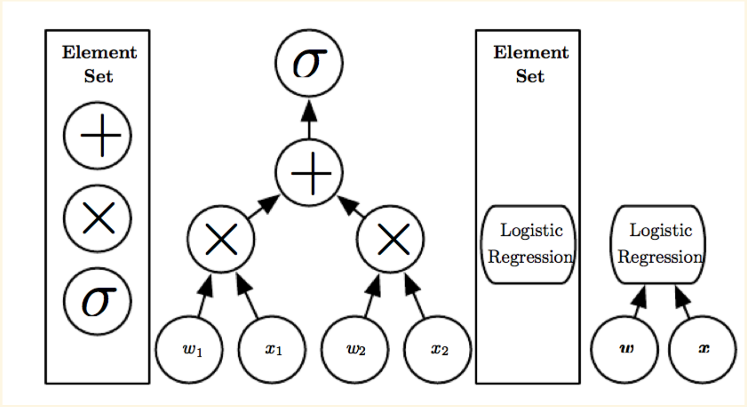
- Deep Learning solves the central problem in representation learning by learning the layers of intermediate representations in terms of simpler representations
- Quintessential example of deep learning: multilayer perceptron (MLP)
  - A complex mapping of input values to output values, composed of many simpler functions
- Each layer of the representation in a deep net can be thought of as a computing step in an algorithm, and the output of the layer is the state of the computer's memory
- Hence, not all information in the activations encode factors of variation
  - Some represent the state

Example of a Deep Learning Model



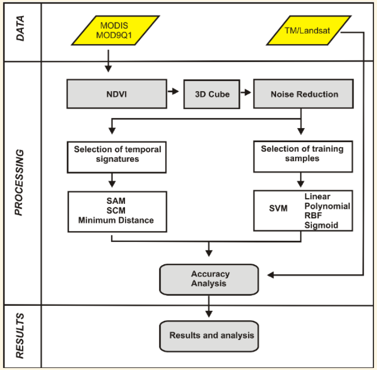
Depth or complexity of a model

- (1) Length of the longest path through the flowchart
  - This again depends on the basic elements used in the flowchart



☛ Depth or complexity of a model (contd...)

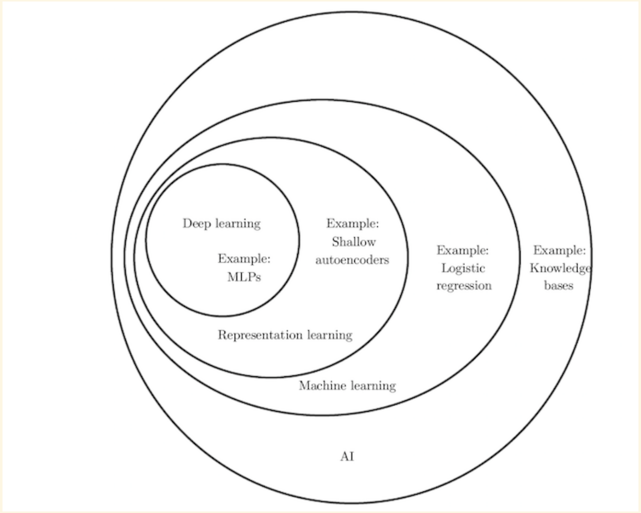
- ☛ (2) Depth of the graph describing how concepts are related to each other
  - ☛ The corresponding flowchart representation might be much deeper
- ☛ There is no general agreement on how to measure complexity



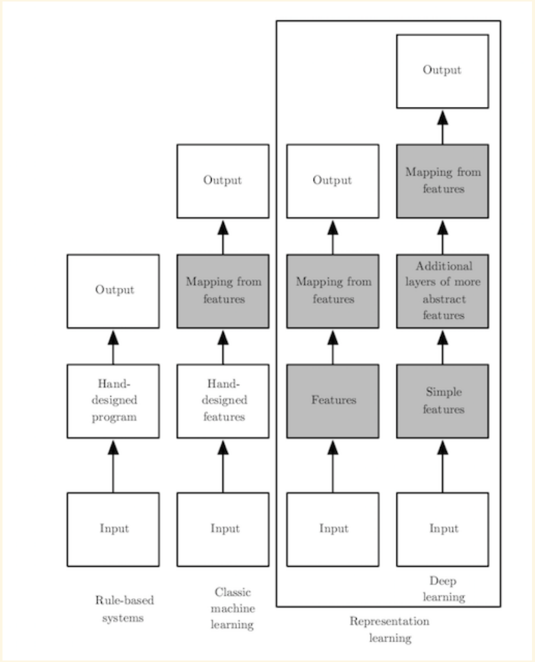
Flow chart of image processing

- **The General Context of Deep Learning**

- Deep Learning is a specific approach to AI and covers certain aspects.
- ML is a general approach to building AI systems that can operate in complex environments.
- AI is a discipline that tries to mimic the way humans perform (complex) tasks.
- Certain important problems, such as complex reasoning, are still unsolved in AI.



Relationships between AI topics


Output
Mapping from features
Output
Output
Mapping from features
Mapping from features
Additional layers of more abstract features
Output
Hand-designed program
Hand-designed features
Features
Simple features
Input
Input
Input
Input
Deep learning
Rule-based systems
Classic machine learning
Representation learning

- **Historical Perspective**

- Deep learning has made significant progress in the recent past primarily due to availability of data and computing power.
- Earlier known as cybernetics (1940-60) and connectionism (1980-90). Later the term artificial neural networks (ANN) has been in parlance.
- The current term 'deep learning' is not based on neuroscientific principles. Just indicates level of composition.

- **Linear Models**

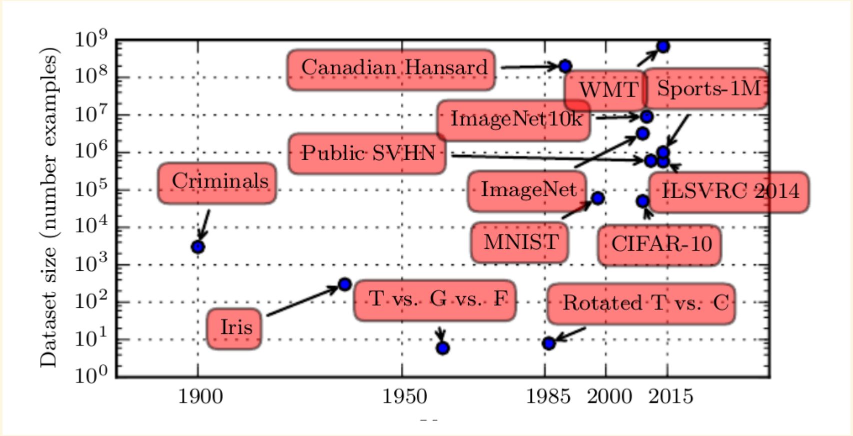
- Linear models are based on:
  - $y = f(\boldsymbol{x}, \boldsymbol{w}) = x_1 w_1 + \cdots + x_n w_n$
  - McCulloch-Pitts used thresholding.
  - Rosenblatt's perceptron was the first to learn the weights.
  - Adaptive linear element (ADALINE) used the actual value.
- Linear models cannot learn some simple functions such as XOR, and hence fell out of favour until the 80s when the backpropagation algorithm for MLPs became popular.
- Fukushima's Neocognitron(1980) is the basis for the modern convolutional network, and uses a more complex versoin of the "rectified linear unit."

- **Connectionist Models:**

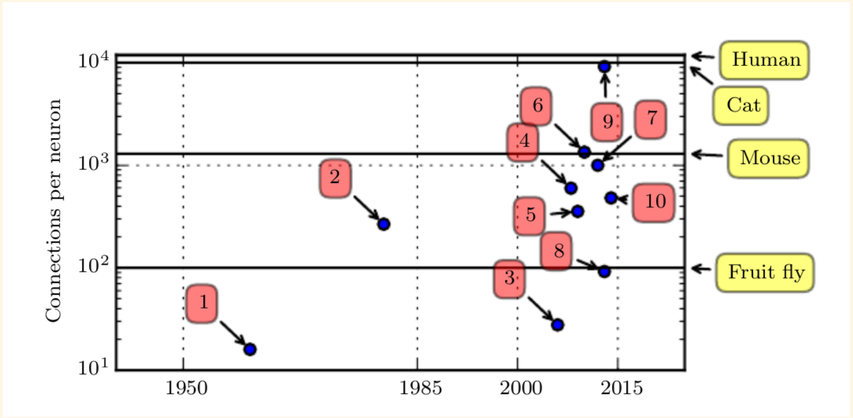
- A large number of simple computational units can achieve intelligent behaviour when networked together.
  - Distributed representation: Each input to a system is represented by many features, and each feature is involved in the representation of many inputs.
  - Other advances: backpropagation algorithm (still used), long short-term memory
  - Connectionist models declined when conventional ML made progress via kernel machines and graphical models.

- **The Third Wave**

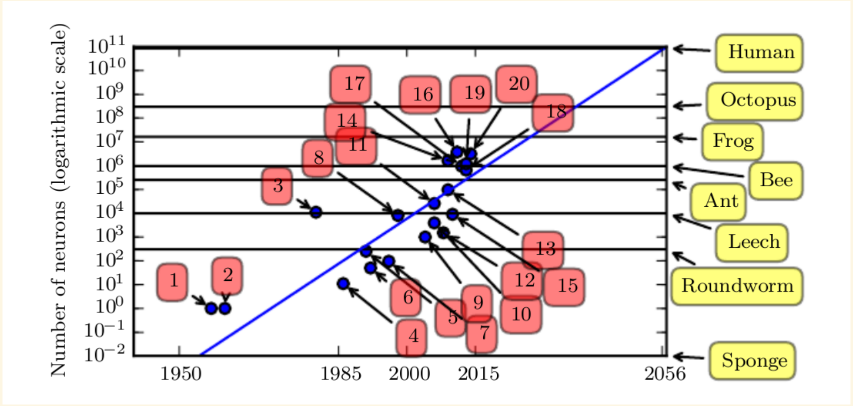
- Third wave began with a breakthrough called greedy layer-wise pre-training (Hinton et al 2006).
- CIFAR (Canadian Institute for Advanced Research) supported much of this work.
- Model sizes have also increased, as now simplified training methods were discovered.
- Also larger models have been shown to give higher accuracies with complex data sets



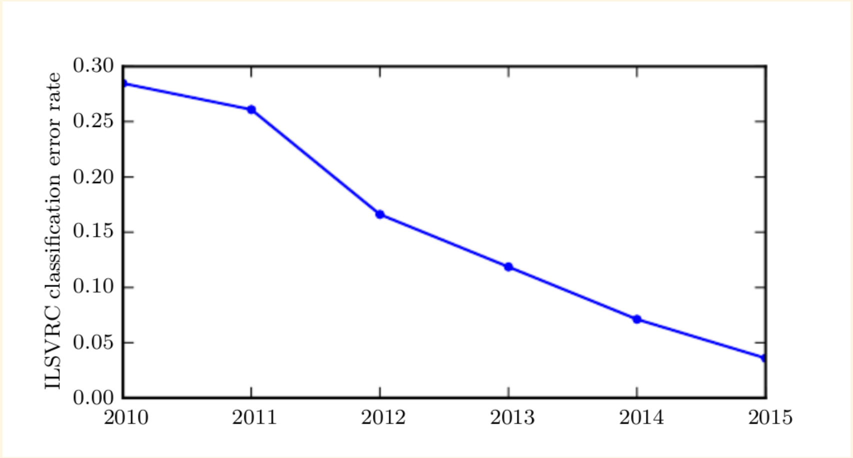
Dataset sizes have changed with time



Number of connections between neurons has been increasing: (1) Adaptive linear element, (2) Neocognitron, (3) GPU-accelerated conv network, (4) Deep Boltzman machine, (5) Unsupervised conv neural network, (6) GPU-accelerated MLP, (7) Distributed autoencoder, (8) Multi-GPU conv network, (9) COTS HPC unsupervised conv net, and (10) GoogLeNet



ANNs have doubled in size every two years:
(1) Perceptron, (2) Adaptive linear element, (3) Neocognitron, (4) Early back-prop network, (5) RNN for speech reco, (6) MLP for speech reco, (7) Mean field sigmoid belief net, (8) LeNet-5, (9) Echo state net, (10) Deep belief net, (11) GPU-acc conv net, (12) Deep Boltzman machine, (13) GPU-acc deep belief net, (14) Unsup conv net, (15) GPU-acc MLP, (16) OMP-1 net, (17) Distributed autoencoder, (18) Multi-GPU conv net, (19) COTS HPC unsup conv net, and (20) GoogLeNet



In 2012, a convolutional network won the ImageNet Large Scale Visual Recognition Challenge by a large margin (Down from 26.1% to 15.3%)

- ☛ Several other impressive applications in speech recognition, other computer vision problems such as pedestrian detection and traffic signs, sequence-to-sequence learning (machine translation), reinforcement learning (DeepMind, other robotic applications)
- ☛ Future: Neural Turing machines (learn to write programs)

- **Neurobiological Models**

- Artificial Neural Networks (ANN) are motivated by the human brain, which is a highly complex, nonlinear, and parallel computer.
- The brain performs perceptual recognition tasks in 100-200 ms, e.g., providing a representation of the surroundings.
- The brain already has considerable structure at birth and learns behavioural rules through "experience," mostly in the first two years.
- ANN is a massively parallel distributed processor made up of simple processing units (neurons), where knowledge is acquired through "experience," and interneuron connections (synaptic weights) store the acquired knowledge.

Some basic properties of neural networks

- Nonlinearity
  - Distributed throughout the network
- Input-output mapping
  - Learning with a teacher or supervised learning
  - Similar to non-parametric statistical inference
- Adaptivity
  - Stability-plasticity dilemma (Grossberg 1988)
- Evidential response
  - Both selection and confidence in the decision
- Contextual information
  - Knowledge is represented by structure and activation state

- **Some basic properties of neural networks (contd...)**

- Fault tolerance
  - Graceful degradation due to its distributed nature
- VLSI implementability
  - Repetitive structure and operations
- Uniformity of analysis and design
  - Neurons are common to all networks
  - Modular structures can be built
- Neurobiological analogy
  - Study of neurological systems has benefited from ANN models (e.g. vestibulo-ocular reflex)
  - Study of the retina has inspired neuro-morphic imaging sensors.

- **The Human Brain**

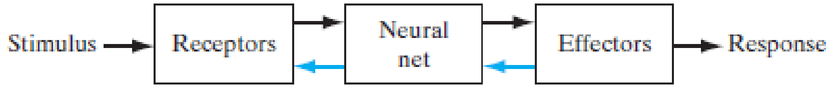


FIGURE 1 Block diagram representation of nervous system.

- Receptors convert input stimuli to electrical impulses
- Both forward and feedback mechanisms exist

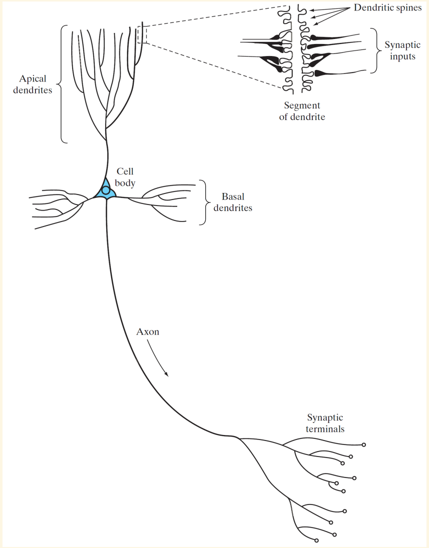
☛ **The Human Brain (contd...)**

- ☛ Ramón Cajál introduced the idea of neurons.
- ☛ They are slow, but massive in number (~ 10 billion in the cortex and about 60 trillion synapses).
- ☛ Energetic efficiency is $\approx 10^{-16}$ joules/op/sec
- ☛ A synapse converts a pre-synaptic electrical signal to a chemical signal and then back to a post-synaptic electrical signal.

☛ The Human Brain (contd...)

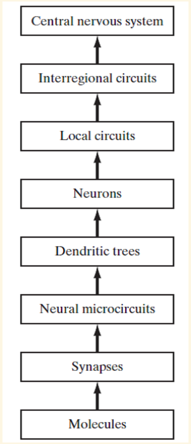
- ☛ Plasticity is achieved by creation of new synaptic connections and the modification of existing ones.
- ☛ Two types of cell filaments: axons (transmission lines) and dendrites (receptive zones).
- ☛ Example: The pyramidal cell can receive 10,000 or mroe synaptic contacts and project onto thousands of target cells.
- ☛ "Action potentials" (spike) originate close to the cell body and propagate across neurons.

☛ The pyramidal cell



☛ The Human Brain (contd...)

- ☛ The brain has both small-scale and large-scale anatomical organization.
- ☛ Topographic maps are organized to respond to incoming sensory information, organized into "sheets."
- ☛ ANN used in practice are primitive in comparison with the brain.



Structural organization of levels in the brain

- **The Human Brain (contd...)**

- The brain has both small-scale and large-scale anatomical organization.
- Topographic maps are organized to respond to incoming sensory information, organized into "sheets."
- ANN used in practice are primitive in comparison with the brain.

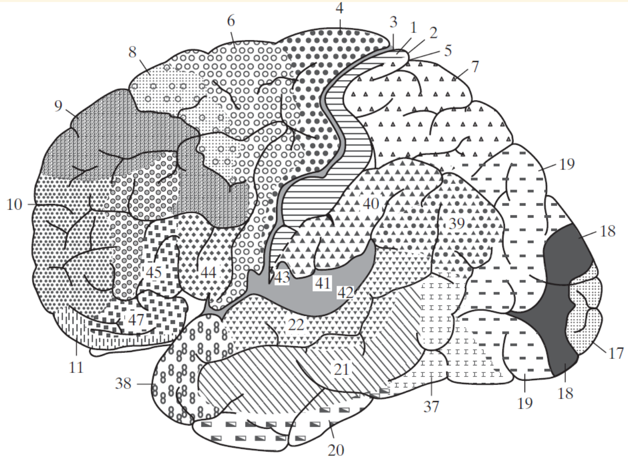
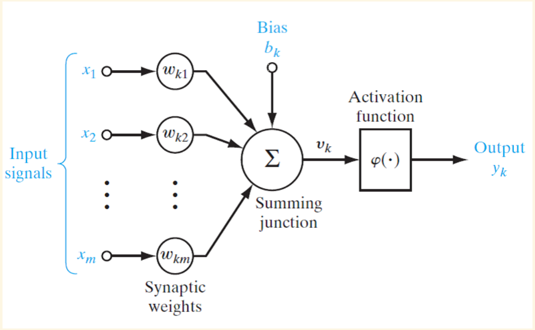


FIGURE 4 Cytoarchitectural map of the cerebral cortex. The different areas are identified by the thickness of their layers and types of cells within them. Some of the key sensory areas are as follows: Motor cortex: motor strip, area 4; premotor area, area 6; frontal eye fields, area 8. Somatosensory cortex: areas 3, 1, and 2. Visual cortex: areas 17, 18, and 19. Auditory cortex: areas 41 and 42. *(From A. Brodal, 1981; with permission of Oxford University Press.)*

Cytoarchitectural map of the brain

☛ Models of a neuron



- ☛ The neuron is the basic information processing unit of a neural net consisting of:
  - ☛ A set of synapses (connecting links) characterized by a weight or strength denoted by w_{kj}, which represents the weight associated with the j-th input to the k-th neuron.
  - ☛ An adder (a linear combiner)
  - ☛ An activation function (sometimes known as the squashing function), which often limits the range of the output to [0,1] or [-1,1].
  - ☛ A bias, denoted by b_k.

☛ **Models of a neuron (contd...)**

☛ The operation can be summarized as:

$$
\begin{aligned}
u_k &= \sum_{j=1}^{m} w_{kj} x_j; \\
v_k &= u_k + b_k; \text{(activation potential or induced local field)} \\
y_k &= \psi(v_k)
\end{aligned}
$$

☛ We can include the bias in the set of weights to simplify the notation, with a dummy input $x_0 = +1$, and $w_{k0} = b_k$, so that $v_k = \sum_{j=0}^{m} w_{kj} x_j$.

☛ Models of a neuron (contd.)

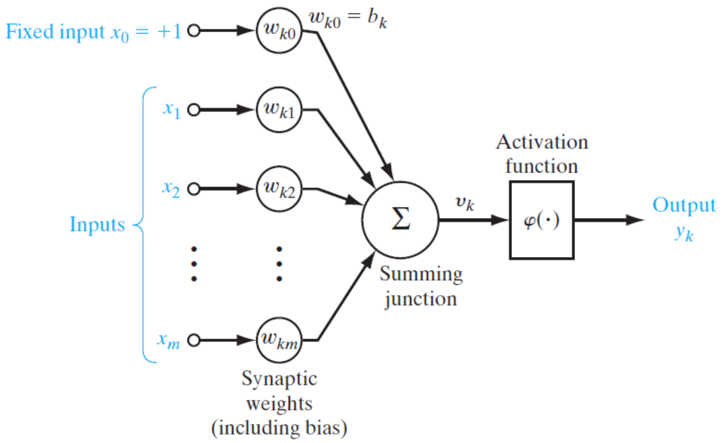
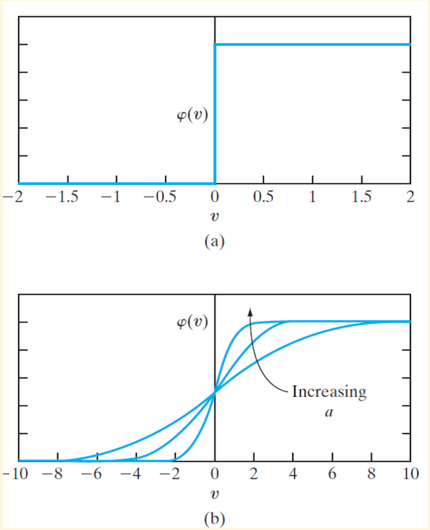


FIGURE 7 Another nonlinear model of a neuron; w_{k0} accounts for the bias b_k.

☛ Activation functions

- ☛ Threshold function (Heavy-side function or McCulloch-Pitts model)
- ☛ $\varphi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases}$
- ☛ Sigmoid function (S-shaped)
- ☛ $\varphi(v) = \frac{1}{1+\exp(-av)}$
  - ☛ The value of 'a' controls the steepness. Slope at origin is $a/4$.
  - ☛ Attractive because it is a differentiable version of the threshold function, but derivatives vanish away from origin.



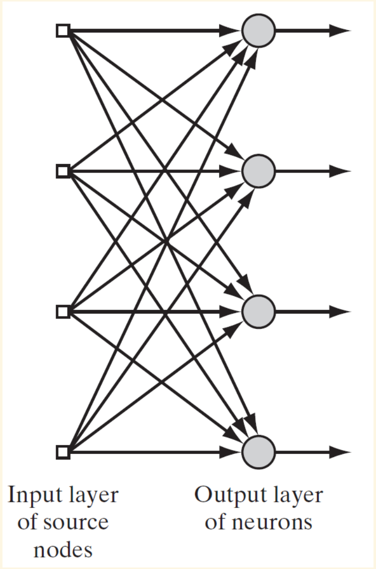
- **Activation functions (contd.)**

- When we need the output to be in the range [-1,1], use a variation called the signum function:
- $\varphi(v) = \begin{cases} 1 & \text{if } v > 0 \\ 0 & \text{if } v = 0 \\ -1 & \text{if } v < 0 \end{cases}$
- The continuous version of the signum function is: $\varphi(v) = \tanh(v)$

- **Stochastic model of a neuron**

- The neuron "fires" with a probability $P(v)$ that is a monotonic function of v:
- $P(v) = \frac{1}{1+\exp(-v/T)}$
- "T" is the "temperature" parameter that controls the shape.
- If x is the state of the neuron, then:
- $x = \begin{cases} +1 & \text{with probability } P(v) \\ -1 & \text{with probability } 1 - P(v) \end{cases}$

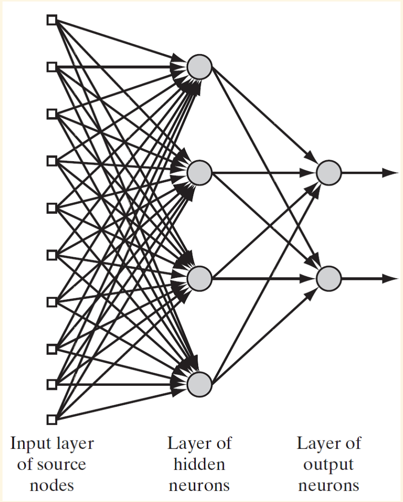
☛ **Network architecture**



☛ Single layer feedforward network

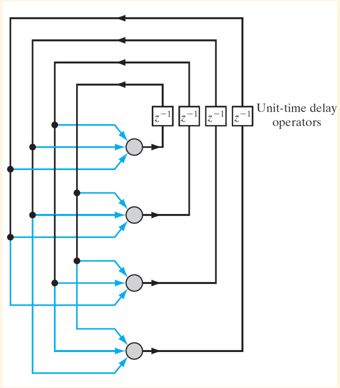
- ☛ Single layer refers to the output layer of computation nodes.

☛ Network architecture - Multilayer feedforward nets



- ☛ Will have hidden neurons and hidden layers.
- ☛ Will enable it to extract higher order statistics and global perspective
- ☛ This is a 10-4-2 network. In general, $m - h_1 - h_2 - \ldots - q$ network
- ☛ Can be fully connected or partially connected

☛ Network architecture - Recurrent nets



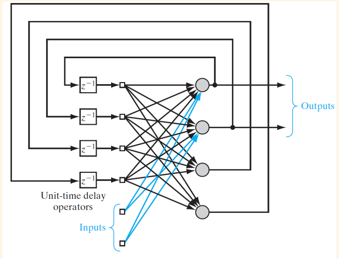


FIGURE 18 Recurrent network with hidden neurons.

- ☛ Will have at least one feedback loop.
- ☛ May or may not have self-feedback loops
- ☛ May or may not have hidden neurons

- **Knowledge Representation**

- Knowledge is stored information or models used to interpret/predict/or appropriately respond to the outside world
- Knowledge representation in a neural network is a design challenge. Need to decide what to represent explicity, and how to encode.
- Knowledge consists of prior information and observations (which are noisy and can be labelled or unlabelled)
- The architecture defines (limits) the type of knowledge that can be learned or stored

Rules of knowledge representation

- Rule 1: Similar input patterns should produce similar representations inside the network (and hence classified as the same class)
- Measure of Similarity:
  - Euclidean distance $d^2(\boldsymbol{x}_i, \boldsymbol{y}_j) = ||\boldsymbol{x}_i - \boldsymbol{x}_j||^2 = (\boldsymbol{x}_i - \boldsymbol{x}_j)^T(\boldsymbol{x}_i - \boldsymbol{x}_j)$
  - Inner product $(\boldsymbol{x}_i \boldsymbol{x}_j) = \boldsymbol{x}_i^T \boldsymbol{x}_j$
  - When $||\boldsymbol{x}_i|| = ||\boldsymbol{x}_j|| = 1$:

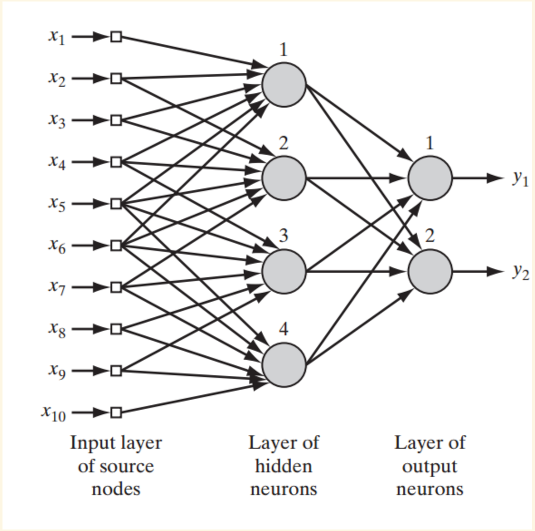
$$d^2(\boldsymbol{x}_i \boldsymbol{x}_j) = 2 - 2\boldsymbol{x}_i^T \boldsymbol{x}_j = 2(1 - \boldsymbol{x}_i^T \boldsymbol{x}_j)$$

  - Hence, they are related. Similarly we can show for show for Mahalanobis distance.

- Rule 2: Objects to be classified differently should represented very differently by the network.
- Rule 3: If a particular feature is important, it should be represented by a large number of neurons (graceful degradation, robustness)
- Rule 4: Prior information should be built into the design of the network, so as to simplify the model and time to learn, as well as speed of response

- **How to build prior information**

- This is still an art, and one of the bottlenecks of deep learning. There are two broad approaches:
  - Restrict the network architecture, e.g., through receptive fields
  - Constrain the synaptic weights, e.g, through weight sharing


x_1
x_2
x_3
x_4
x_5
x_6
x_7
x_8
x_9
x_{10}
1
2
3
4
1
2
y_1
y_2
Input layer of source nodes
Layer of hidden neurons
Layer of output neurons

- Receptive field of a neuron is the region of the input field that can influence the output of the neuron
- A simple way of achieving weight sharing is to use the same value in every receptive field
- e.g., $v_j = \sum_{i=1}^{6} w_i x_{i+j-1} \ , \quad j = 1, 2, 3, 4$
- This architecture has only six independent weight values
- This approach is known as convolutional neural network (LeCun and Bengio, 2003)

☛ How to build invariances into neural network deisgn

- ☛ For example, translation invariance, rotation invariance, scale invariance, affine invariance, more complex distortions
- ☛ Invariance by structure
  - ☛ Structure the network and impose constraints on the weights
  - ☛ As an example, for in-plane rotation invariance, we need to duplicate a synaptic weight at all localtions that are equidistant from the origin
- ☛ Invariance by training
  - ☛ Use examples that correspond to different transformations
  - ☛ Difficult to generate examples. Compulation burden will increase. May not generalize to other classes of objects
- ☛ Invariance can also be achieved by “feature engineering,” if we omit representation learning.

- **Learning processes**

- Two major types:
  - learning with a teacher (supervised learning)
  - learning without a teacher
    - unsupervised learning
    - reinforcement learning

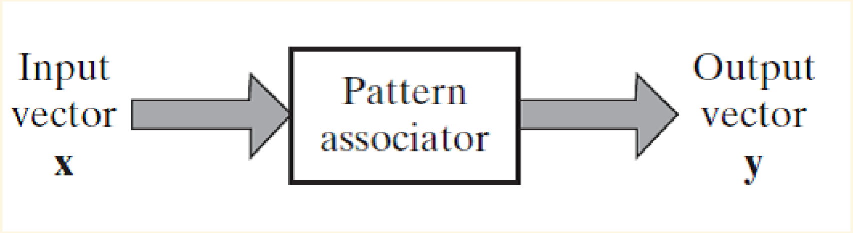
☛ Learning tasks: Pattern Association

- ☛ $\boldsymbol{x}_k \to \boldsymbol{y}_k$, where $\boldsymbol{x}_k$ is the key pattern and $\boldsymbol{y}_k$ is the memorized pattern.
- ☛ Autoassociation
  - ☛ $\boldsymbol{y}_k = \boldsymbol{x}_k$
  - ☛ Retrieve the exemplar when a partial or distorted version is presented.
  - ☛ Uses unsupervised learning
- ☛ Hetero-association
  - ☛ $\boldsymbol{y}_k \neq \boldsymbol{x}_k$
  - ☛ Associates pattern $\boldsymbol{y}_k$ with pattern $\boldsymbol{y}_k$
  - ☛ Uses supervised learning
- ☛ Storage vs. recall phase

☛ **Pattern association (contd...)**

☛ If $\boldsymbol{y} \neq \boldsymbol{y}_j$ for $\boldsymbol{x} = \boldsymbol{x}_j$, the memory has made an error in recall.

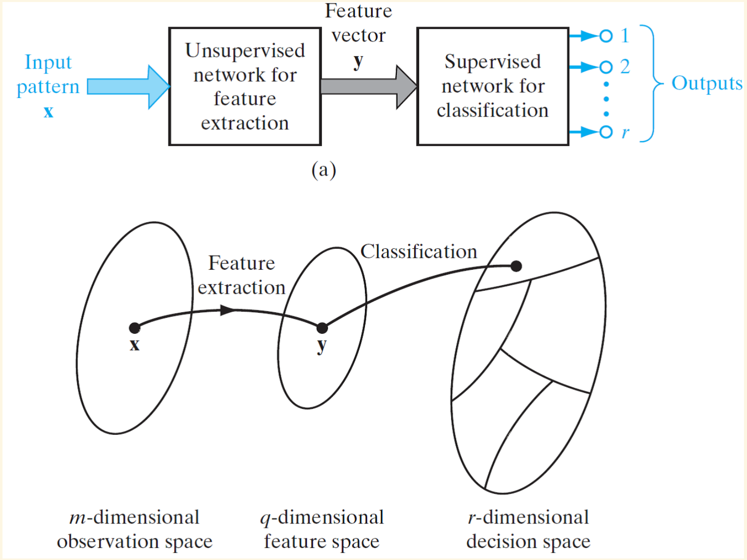
☛ The storage capacity of the network is $q = n/N$, where n is the number of patterns recalled correctly, and N is the total number of neurons used.



- **Pattern recognition**

- Pattern is associated to one of the fixed number of classes.
- Usually a training phase, where the network learns, is followed by a testing phase.
- Statistical in nature where patterns are points in a decision space.
- There is one decision region associated with each class (not necessarily connected).
- Determining the boundaries is the challenge.
- Generally feedforward nets are used, and in modern nets, feature extraction is replaced by representation learning.

☛ Pattern recognition (contd...)

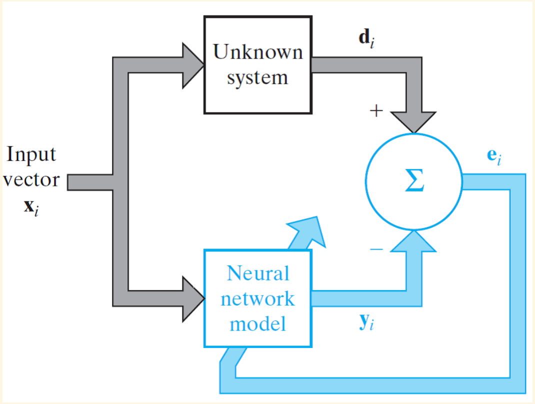

Input
pattern
x
Unsupervised network for feature extraction
Feature vector
y
Supervised network for classification
1
2
r
Outputs
(a)
Feature extraction
Classification
x
y
m-dimensional observation space
q-dimensional feature space
r-dimensional decision space

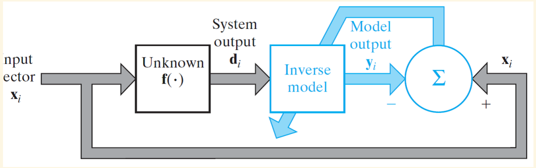
☛ **Function approximation**

☛ How to approximate any memoriless multiple input - multiple output (MIMO) system.

☛ There are two types of problems: system identification, inverse modeling

☛ Inverse modeling is harder because there is no unique solution.





System output $\mathbf{d}_i$
Model output $\mathbf{y}_i$
nput ector $\mathbf{x}_i$
Unknown $\mathbf{f}(\cdot)$
Inverse model
Σ
$\mathbf{x}_i$
−
+

☛ **Control**

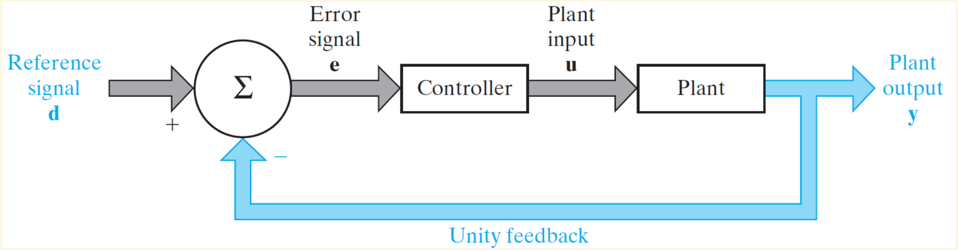
☛ How to control a “plant” using an NN, where the outputs are actions.

☛ There are many applications that need this, e.g., IoT, cyberphysical systems, and autonomous vehicles.

☛ NN solutions are known to scale well, e.g., thousands of actuators.

☛ In a feedback control, the objective is to supply appropriate inputs to make the output track the reference signal.

☛ This is done by inverting the plant’s input-output behaviour.



Control (contd...)

- To adjust the free parameters of the plant, we need to know the Jacobian given by:
- $\boldsymbol{J} = \left\{ \frac{\partial y_k}{\partial u_j} \right\}_{j,k}$
- The partial derivatives vary, depending on the operating point. Therefore, use one of the following approaches:
  - Indirect learning: Learn a neural model to emulate the input-output mapping.
  - Direct mapping: The signs of the partial derivatives are generally known and they do not change in the dynamic range of operation.
  - A distributed version of their absolute values can be learned through the free parameters of the neural controller.