Neural Networks: An Introduction

Prof. Sven Lončarić

sven.loncaric@fer.hr

http://www.fer.hr/ipg

Overview of topics

- What are neural networks?
- Organization of the brain
- Neuron models
- Network architectures
- Knowledge representations
- Visualization of learning in neural networks
- Artificial intelligence and neural networks
- History

Motivation

- Brain computation compared to digital computers
  - Neurons are 5-6 orders of magnitude slower than digital logic (ms vs. ns)
  - Brain has a huge number of neurons (10 billion neurons and about 60 000 billions of interconnections)
  - Brain is enormously energy efficient (10^{-16} J per operation per second vs 10^{-6} J per operation per second)
  - Brain is a very complex, nonlinear, parallel computer

Examples of efficacy

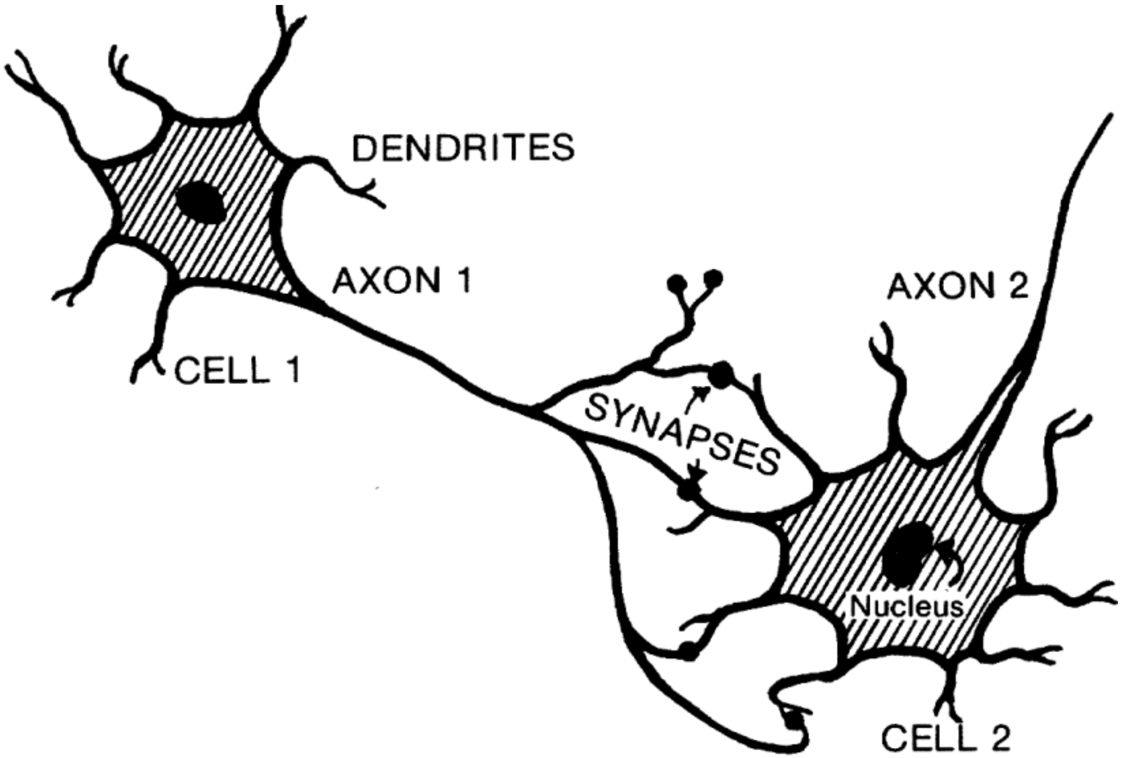
- Human vision as information processing problem
  - Human recognizes faces in 100-200 ms, computes need more time
- Sonar of a bat
  - Detects distance to object (e.g. Insect), velocity, size, azimouth and elevation
  - All operations are done in a plum-sized brain
  - Bat can accurately locate and catch insects

How is this possible?

- After birth brain develops very fast through learning
- Knowledge is gained through years of learning
  - Fastest brain development through first two years of life (1 million synapses formed per second)
  - Brain development continues after the initial phase

Pyramidal neurons

- Dendrites accept inputs from other neurons
- Axon transmist impulses to other neurons
- Synapses are structures where impulses are transfered from one neuron to another



Synapses

- Synapses connect neurons to pass electrical signals from one neuron to another
- Pyramidal neuron can have:
  - 10000 and more of input synapses
  - Its output can be connected to 1000s of other neurons

Neural networks

- Biological neural networks
  - Biological organisms
  - Human and animal brains
  - High complexity and parallelism
- Artificial neural networks
  - Motivated by biological neural networks
  - Much simpler and primitive compared to biological networks
  - Implementation on general purpose digital computers or using specialized hardware

ANN definition

- ANN is a massive parallel distributed processor that is good for memorizing of knowledge
- ANN is similar to biological NN in the following aspects:
  - Knowledge is gained through learning process
  - Knowledge is encoded in mutual connections between neurons

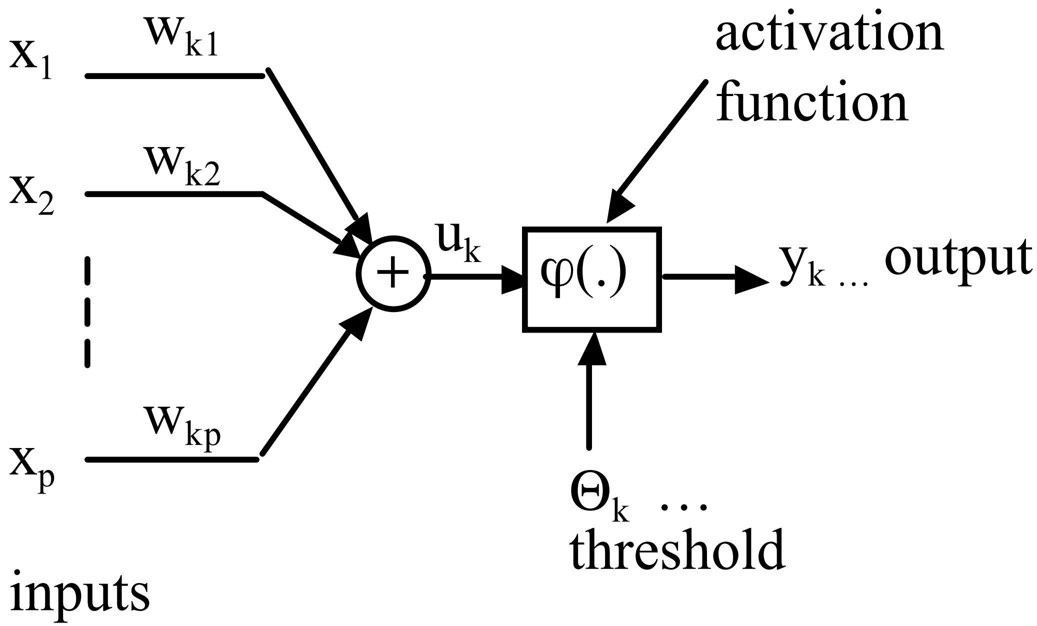
ANN properties

- Nonlinearity
- Input to output mapping (supervised learning)
- Adaptivity
- Fault tolerance
- Possible VLSI implementation
- Neurobiological analogy

Neuron models

- Neuron model elements:
  - Set of synapses, i.e. Inputs with respective weights. (Notation: Signal x_j at input j of neuron k has weight w_{kj})
  - Adder for summation of weighted inputs. These two operations calculate the weighed sum of inputs.
  - Non-linear activation function that limits output of neuron to interval [0,1]

Neuron models

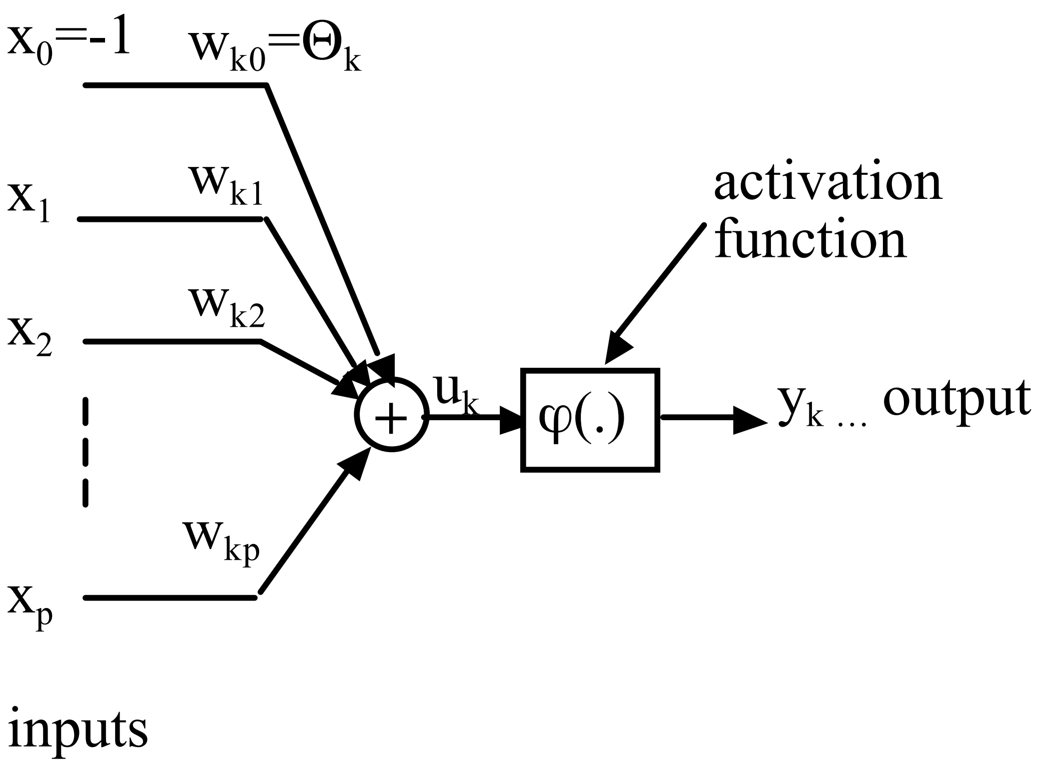


$$u_k = \sum_{j=1}^{p} w_{kj} x_j$$

$$y_k = \varphi\left(u_k - \Theta_k\right)$$

Neuron models

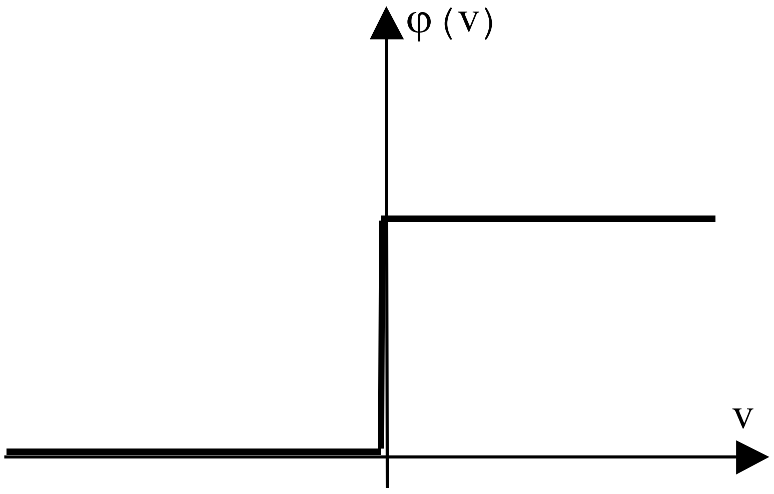
- Threshold Θ_k can be represented as an additional input of value -1 and weight Θ_k



Activation functions

- Type threshold:

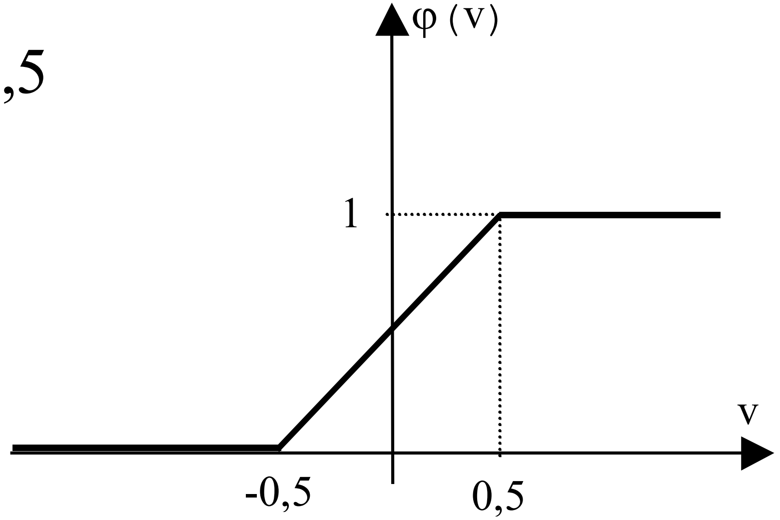
$$\varphi(v) = \begin{cases} 1, & v \geq 0 \\ 0, & v < 0 \end{cases}$$



Activation functions

- Linear :

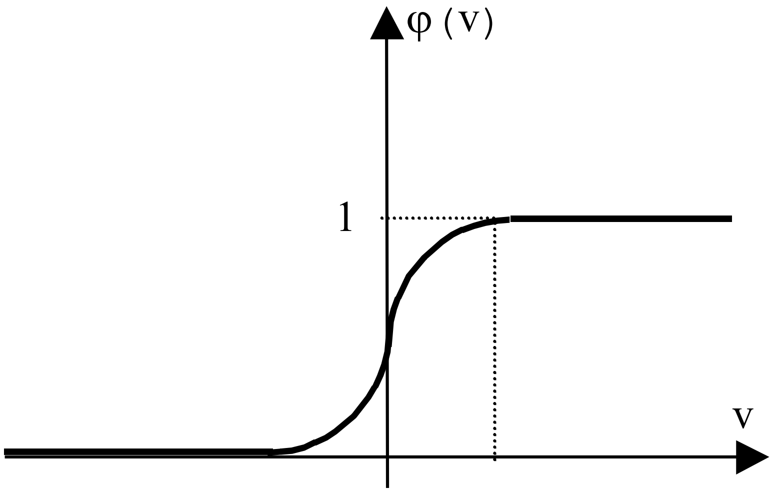
$$\varphi(v)=\begin{cases} 1, & v \geq 0{,}5 \\ v+0{,}5, & -0{,}5 < v < 0{,}5 \\ 0, & v \leq -0{,}5 \end{cases}$$



Activation functions

- Sigmoid activation function

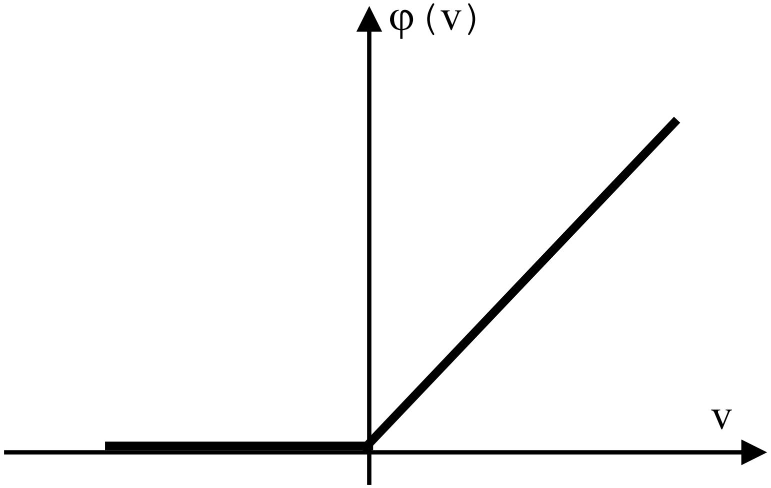
$$\varphi(v) = \frac{1}{1+\exp(-av)}$$



Activation functions

- Rectifier linear unit (ReLU)

$$\varphi(v) = \begin{cases} v, v \geq 0 \\ 0, v < 0 \end{cases}$$



Graphs

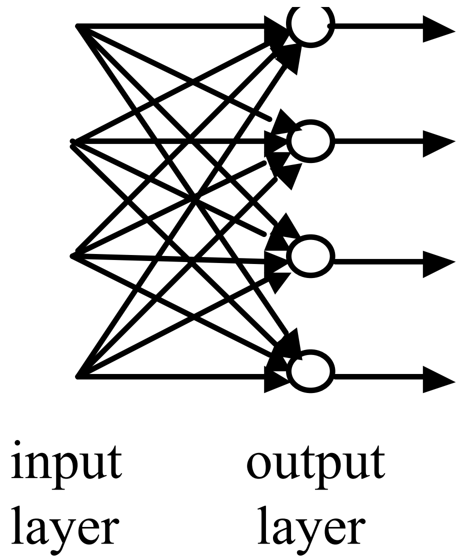
- ANNs can be represented using oriented graphs
- A graph has two kinds of edges:
  - Synaptic edge representing linear input-output relation (multiplication by weight)
  - Activation edge representing a non linear input-output relation of activation function

Network architectures

- Network architectures (or topologies) define ways how neurons are mutually connected
- Four main architectures:
  - Single-layer feed-forward networks
  - Multi-layer feed-forward networks
  - Recurrent networks
  - Lattice structures

Single-layer networks

- Has a single neuron layer (output layer)
- Input layer does not count due to lack of processing
- Network inputs are connected to neuron inputs
- Neuron outputs are also network outputs
- No feedbacks from outputs to inputs

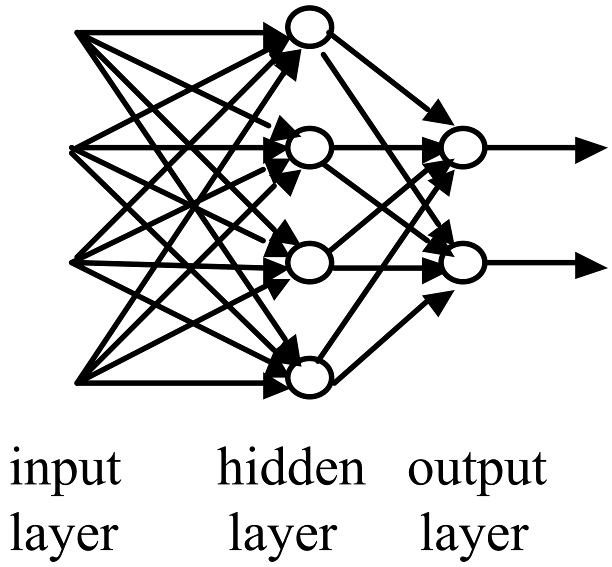


Multilayer networks

- Multilayer networks have one or more hidden layers, in addition to input and output layers
- Outputs from n-th layer are inputs to n+1-th layer
- Connectedness:
  - A network is fully connected when each neuron in a layer is connected to all neurons in the next layer
  - If some connections are missing the network is partially connected

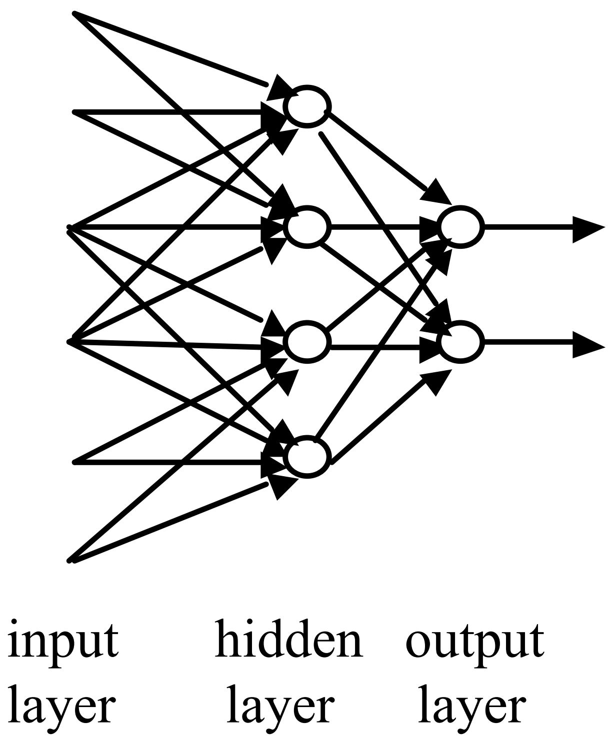
Multilayer networks

- An example of network with one hidden layer with four neurons
- The network has four input neurons
- There are two output neurons



Multilayer networks

- An example of partially connected networks are locally connected networks
- Example: Each hidden neuron is connected only to neighboring input neurons
- Such a set of localized inputs is called receptive field

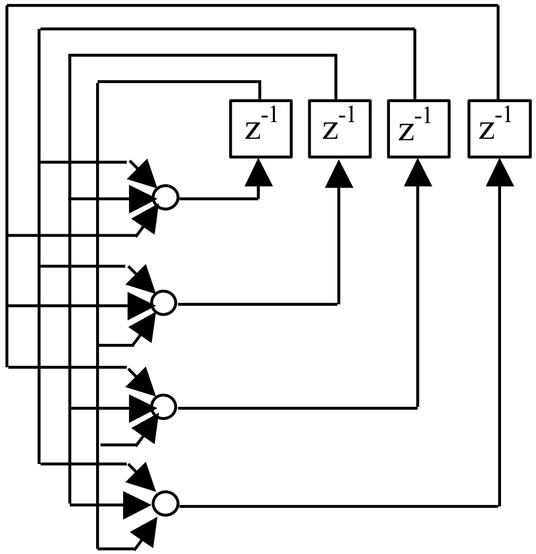


Recurrent networks

- Have at least one feedback
- Can have hidden neurons
- Feedbacks give additional quality to recurrent networks
- Higher complexity for network analysis
- In combination with delay elements we obtain nonlinear dynamic systems which is crucial for ability to memorize patterns

Recurrent networks

- Example of recurrent network without hidden neurons
- Such network is a nonlinear time-discrete system

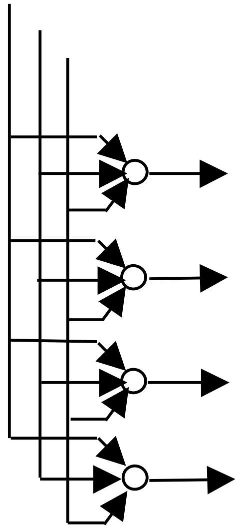


Lattice networks

- Lattice networks consist of 1-D, 2-D or n-D array of neurons with a set of input nodes
- Each input is connected to all neurons in the array
- Such networks are feedforward networks where neurons are arranged in an array

Lattice networks

- An example of 1-D lattice network



Knowledge representation

- A definition of knowledge says: “Knowledge is stored information or models that is used by a person or a machine to interpret, predict or react to the outside world”
- The main characteristics of knowledge are:
  - Stored information
  - How is information encoded for later use
- Good intelligent systems depend on well chosen knowledge representation

ANN learning

- The main task of ANN is to learn the model of envirnoment in which it will operate and to maintain the model accurate in order to achieve the goals of the system
- The knowledge about the world consists of:
  - A priori information about everything that is known
  - Opservations of the workd that are used as examples for ANN learning
- Every learning example consists of an input-output value pair

ANNs for pattern recognition

- Neural networks are often used for pattern recognition
- For pattern recognition a classifier determines a class for each input vector
- ANN is a non-linear classifier that divides the input space into classes that have non-linear borders
- Applications: image analysis, speech analysis, signal analysis, time series analysis, etc.

An Example: OCR

- For character recognition, input can be a vector of pixel values (e.g. from a 5x7 pixel matrix), and output could be an identity of a digit (0-9)
- Input layer can have 5x7=35 inputs
- Output layer could have 10 neurons (one for each digit)
- ANN would be trained with pairs of known input and output vectors (aka learning phase)
- After learning is completed, ANN could recognize previously unseen digits

Comparison with classical approaches

- This example illustrates difference between classical and ANN approach to character recognition
  - Classical approaches require development of mathematical models of measured data, which is used to develop a PR system based on measured data
  - ANN approach works directly with data (ANN learns from examples) and it is not necessary to know the model of measured data
  - ANN learns the model implicitly and computes the required output

Rules for knowledge representation

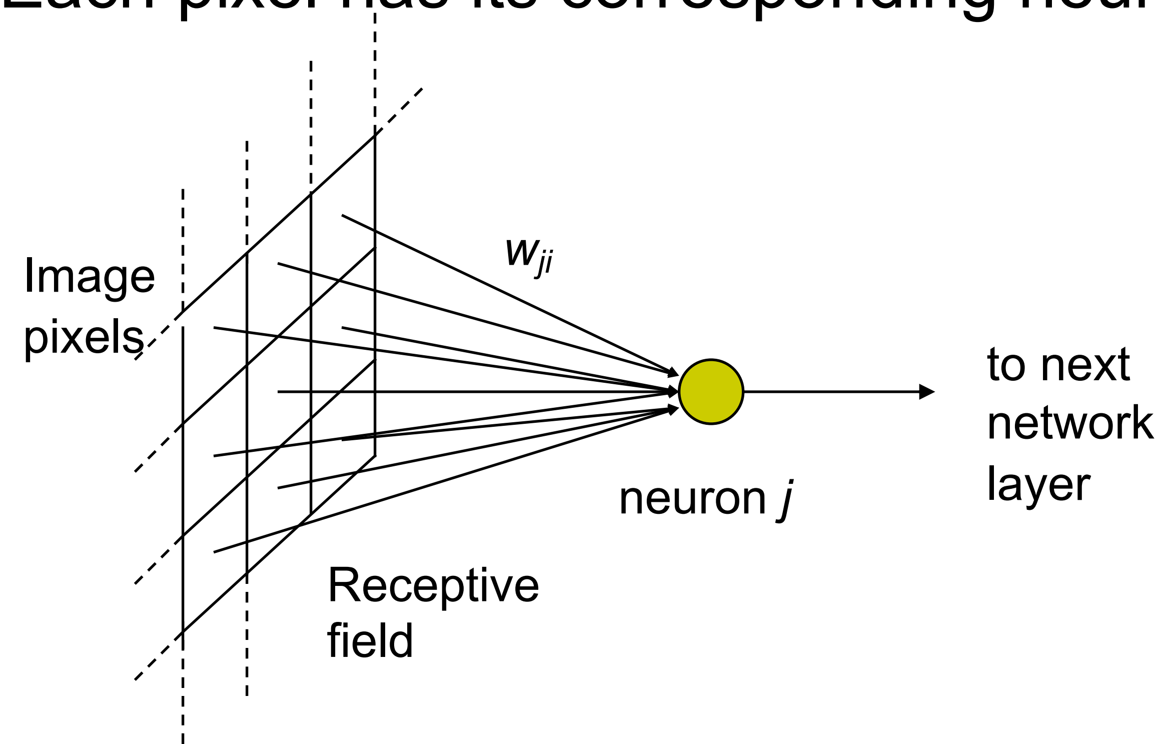
1. Similar inputs from similar classes should have similar representations and should be classified to the same class
2. Objects belonging to different classes should have sufficiently different representations in the network
3. Important object features must be represented using a larger number of neurons
4. A priori information and invariance should be built into ANN architecture so that ANN does not have to learn them

Use of a priori information

- In character recognition example a priori information is
  - Input data (image) is two-dimensional
  - Input data has local structure (characters are localized)
- For this reason we should form ANN so that synaptic connections in input layer are localized (locally connected network)
  - Such local input region for a neuron is called receptive field

Receptive field

- In this example neuron *j* has receptive field of dimensions 3x3 that is centered around the pixel at this position
- Each pixel has its corresponding neuron



Receptive field

- Receptive field of each neuron is the same (weights repeat for each field)
- Let $\{w_{ji} \mid i=0,1, ..., p-1\}$ be weights for neuron j
- Output of neuron j iz given by expression (convolution):

$$y_j(n) = \sum_{i=0}^{p-1} w_{ji} x(n-i)$$

where $x(n)$ are pixel values within receptive field

Receptive field

- Because of convolution calculation such networks are called convolutional networks
- Processing is reduced to convolution with 2-D kernel of small dimensions followed by limiting using activation function
- Because of neuron performing the same function at diferent image regions the technique is called weight sharing
- Advantage of weight sharing is in reduction of the number of unknown network parameters

A priori information - Summary

- A priori knowledge about the problem can be built into ANN in two ways:
  - By restricting network topology through use of local connectedness (local receptive fields)
  - By limiting the choice of synaptic weights - by use of weight sharing
- Methods are very application-dependent

Invariance property in pattern recognition

- If an object image is rotated, translated or scaled we can still recognize the object
- If a speaker talks louder or silent, in low or high voice pitch, or if he has a cold we can still recognize the words spoken
- In general, a pattern recognition system must be invariant to certain transformation of input information (e.g. signal, image)
- Classification result must not depend on such transformations of input information

Realization of invariance

- Some techniques to construct ANNs that will be invariant to certain informations of input information are:
  - Invariance based on ANN architecture (topology)
  - Invariance through learning
  - Invariance based on use of invariant features at ANN input

Invariance by architecture

- ANN architecture can be chosen to be invariant to certain transformations
- Example: Let us assume that we want to achive rotational invariance around the image center:
  - Let w_{ji} be the weight of the neuron *j* linked to the pixel *i* in input image
  - If $w_{ji}=w_{jk}$ for each two pixels *i* and *k* that are located at an equal distance from the image center than the network will be invariant to rotation around the image center

Invariance by learning

- Invariance can be achieved by learning so that ANN is trained with examples corresponding to transformed versions of the same object
- E.g. ANN can be trained with rotated versions of the same object in order to achieve rotational invariance
- Drawbacks of this approach are:
  - Invariance may not be achieved for rotated versions of some other object which was not used in training
  - Increased learning and computational demands due to increased number of training examples

Invariance by features

- Invariance can be achieved by selecting features that are invariant to desired transformations
- If such features are input into ANN than the network will not have to solve the problem of invariance
- Drawback: In this case features have to be selected manually or using some other approach – ANN is not involved in feature extraction/selection

Invariance by features

- Advantages of achieving invariance by features:
  - Input vector dimension can be reduced because only selected features are used for input, not the original data (e.g. image)
  - Demands on ANN are reduced
  - Invariance is achieved for all objects (not only for ones used in learning)

Visualization of learning process in ANNs

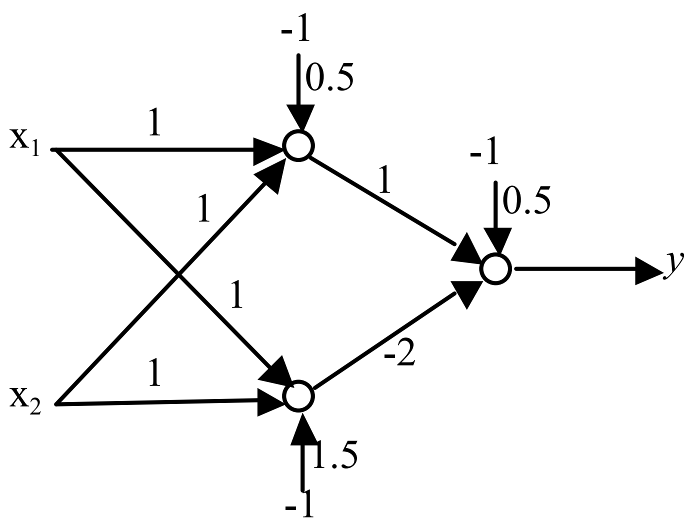
- Learning process is complex and involves determining a large number of unknown network parameters
- Visualization of the learning process can be useful for better understanding of learning
- Still an open research issue
- Examples of two visualization approaches:
  - Hinton diagram, Rumelhart i McClelland, 1986
  - Bond diagram, Wejchert i Tesaro, 1991

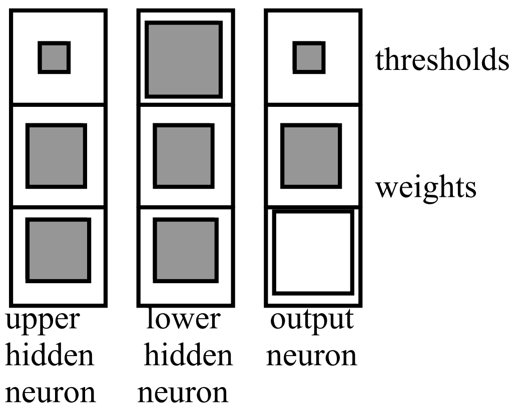
Hinton diagram

- Hinton diagram consists of columns composed of squares
- Each column has one square for threshold value and other squares for weight values
- Square size is proportional to weight value
- Tone of the square (black or white) represents the sign of the weight value (1 or -1)

Example: Hinton diagram

- ANN has two input neurons, two hidden neurons and one output neuron





Drawbacks of Hinton diagram

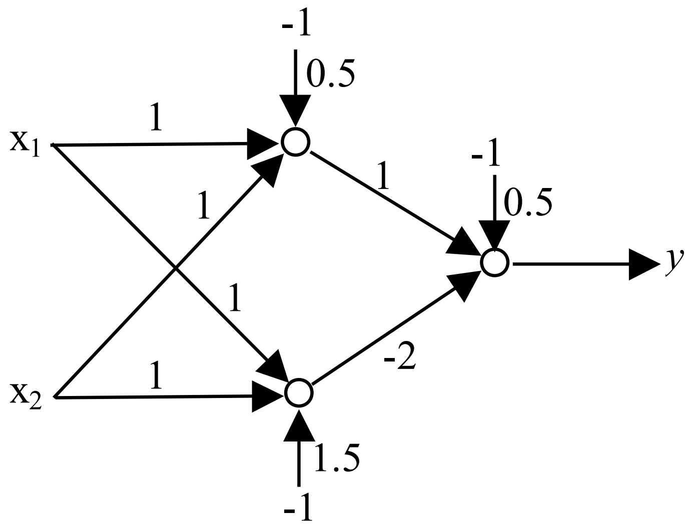
- Hinton diagram only shows weight and threshold values, but does not show their relationship to the ANN architecture
- It is desirable to have a visualization of weight and threshold values integrated into graphical representation of ANN topology
- Bond diagram overcomes this shortcoming of the Hinton diagram

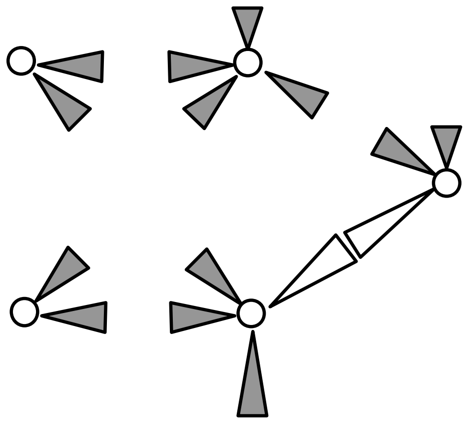
Bond diagram

- Weight values are shown using bonds between neurons
- Weight value is proportional to bond length
- Weight polarity is represented using color (black or white)
- Advantages compared to Hinton diagram are:
  - Diagram shows connections between neurons
  - Set of weights represents a pattern for the observer who can draw global observations from bond diagram

Example: Bond diagram

- In this example we have ANN with two input, two hidden, and two output neurons





Intelligent systems and ANNs

- Goal of artificial intelligence (AI) is to develop paradigms and algorithms that enable machines to perform tasks that require intelligence when people perform them
- Tasks solved by AI include perception, language, and problem solving
- Intelligent system must be able to perfrom three tasks:
  - Knowledge representation
  - Use of knowledge to solve a given task (reasoning)
  - Knowledge acquisition (learning)

Knowledge representation

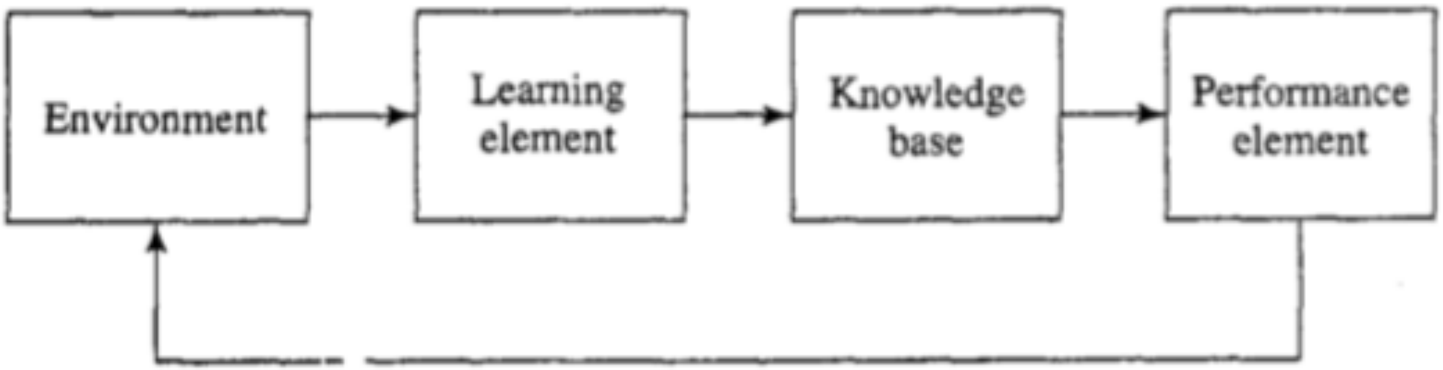
- Knowledge in intelligent systems can be represented as a database
- Knowledge can be declarative or procedural
- Declarative knowledge representation is in the form of a set of static facts with a small number of procedures that manipulate the facts
- In procedural representation knowledge is in the form of program code that performs reasoning when executed

Reasoning

- Reasoning is ability to solve problems
- Every reasoning system must have these properties:
  - The system must be able to solve a wide set of problems
  - The system must be able to explicitely express any known implicit information
  - The system must have a control mechanism that determines operations that must be applied to solve a specific problem

Learning

- A simple learning model is shown below:



Learning

- During learning the environment sends information to the learning element
- Learning element uses this information to modify (augment) the knowledge base
- New knowledge (hypothesis) is tested providing feedback to the learning element
- A machine learns in this manner and gets a grade from a teacher (element for testing of new knowledge)

Comparison of classical AI and ANNs

- Comparison of classical AI and ANNs can be done at three levels:
  - Operation level
  - Type of processing
  - Knowledge representation structures

Operation level

- AI systems:
  - Use symbolic representations
  - Representations are discrete and abstract
  - Model reasoning as sequential processing of symbolic representations
- ANNs:
  - Use parallel distributed information processing (interaction of a large number of neurons)
  - Emphasize on neurobiological background of the reasoning phenomenon

Type of processing

- AI systems:
  - Work sequentially like conventional computers
  - Even if order of operations is not strictly defined, operation is sequential
- Neural networks:
  - Parallel operation (massive parallelism with a large number of processing elements – neurons)
  - Robustness, reduced sensitivity to noise, due to massive parallelism

Knowledge representation

- AI systems:
  - Use symbolic representations that have quasi-linguistic structure (elements composed into sentences)
  - Discussion of the nature of reasoning (debate between AI and ANN communities about nature of mental processes)
- ANNs:
  - Have distributed representations
  - ANN architectures are ad-hoc designs without a theoretical background

History

- Modern era of ANNs begins with the paper of McCullocha and Pitts, 1943, who described a neuron model and ANNs
- Wiener book “Cybernetics” was published in 1948 and describes concepts of control, communications and statistical signal processing
- Hebb's book “The Organization of Behavior” was published in 1949 – first view of learning as a process of synaptic modifications

History

- Taylor, 1956, associative memory
- von Neumann, 1956, idea of redundant systems (ANNs) composed from many unreliable components (neurons)
- Rosenblatt, 1958, perceptron
- Widrow i Hoff, 1960, LMS (least mean squares) algorithm and Adaline (adaptive linear element)
- Fast development until 1969 when Minski and Papert published the book “Perceptrons” where they analyzed limitations of neurons and ANNs

History

- Ten year slowdown (1970-1980) due to:
  - Technological reasons (no computers)
  - Policy/financial reasons (Minski-Papert book)
- Grossberg, 1980, adaptive resonance theory
- Hopfield, 1982, energy function, relation to statistical physics
- Kohonen, 1982, self-organizing maps
- Kirkpatrick, Galatt, Vecchi, 1983, simmulated annealing for combinatorial problems

History

- Rumelhart, Hinton, Williams, 1986, back-propagation learning
- Rumelhart, McClelland, 1986, two influential books, “Parallel Distributed Processing: Exploration in Microstructures of Cognition”
- Broomhead, Lowe, 1988, radial-basis function (RBF) networks

History

- Most recently – development of deep neural networks
- GPU computing for higher parallelism and increased computational power
  - Enables faster simulation of more complex network architectures and learning algorithms (more layers, more neurons, more learning examples)

Homework

- Problem 1.15.
  - Multi layer feed-forward network has neurons working in linear mode. Show that such a network is equivalent to a single layer network.
- Problem 1.19.
  - A recurrent ANN has 3 input neurons, 2 hidden neurons and 4 output neurons. Draw a graph representing the network.