Remote land su

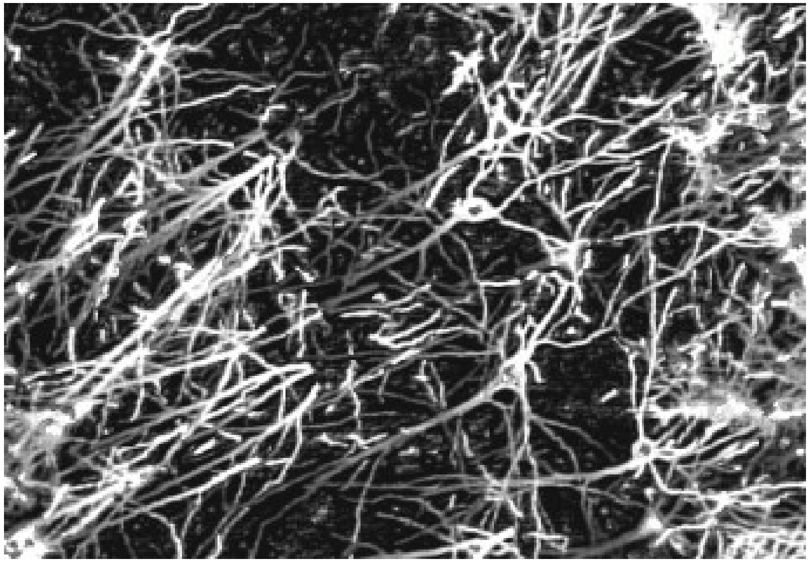
Institute of Geoinformation Technologies,
Faculty of Forestry and Timber MENDELU

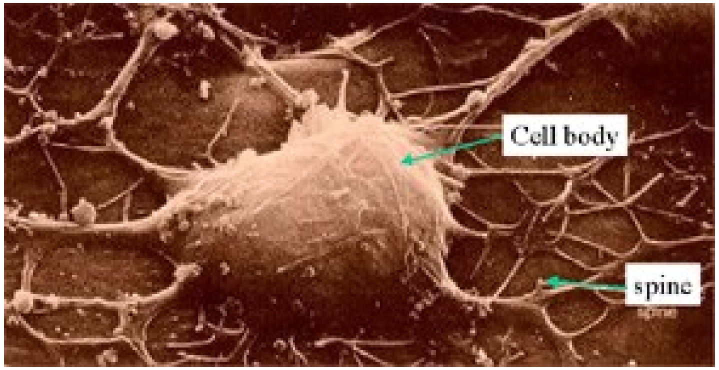
Classification based on machine learning

- Machine learning is a subfield of artificial intelligence, dealing with algorithms and techniques that enable a computer system to 'learn'.
- By learning in a given context, we mean such a change in the internal state of the system that increases the ability to adapt to changes in the surrounding environment.
- For the classification of remote sensing data, the following are most often used:
  - **neural networks**(Artificial Neural Networks (ANN)
  - **decision trees**(Decision Trees)

NEURON NETWORKS

- Neural networks are the result of natural evolution. They have proven themselves excellently in complex biological systems. Living organisms equipped with a neural network can behave adaptively. This is because the neural network is able to learn. Learning means drawing conclusions from experience .
- If some of the neurons supplying the information drop out, the resulting behavior of the neural network will not change.





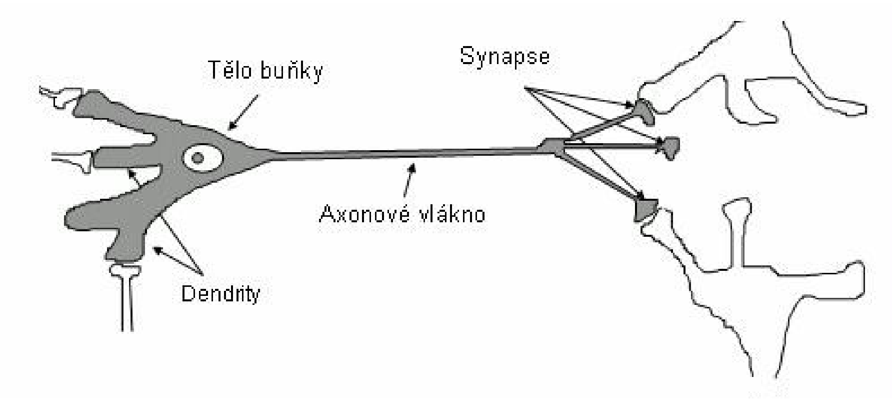
Electron microscope image of a neuron

• Neural networks are black boxes into which data enters and processed data depending on the input data is output.
These networks are composed of individual**neurons**.

• Each neuron has**many inputs and one output**. The inputs are called **dendrites** and are connected to the outputs (**axons**) of other neurons.

• Information is transmitted at connections - synapses.

• In the human brain, one neuron has connections with about 10,000 on average. -100 thousand neurons.



Biological neuron

Hebb's Law of Learning

- Canadian psychologist
- Apprenticeship Act 1949

- Two connected neurons:

  - both are active: the bond is strengthened
  - both are inactive: the bond weakens
  - one active, the other inactive: no bond modification

- 1943 – the first mathematical model of a neuron (McCulloch and Pitts)

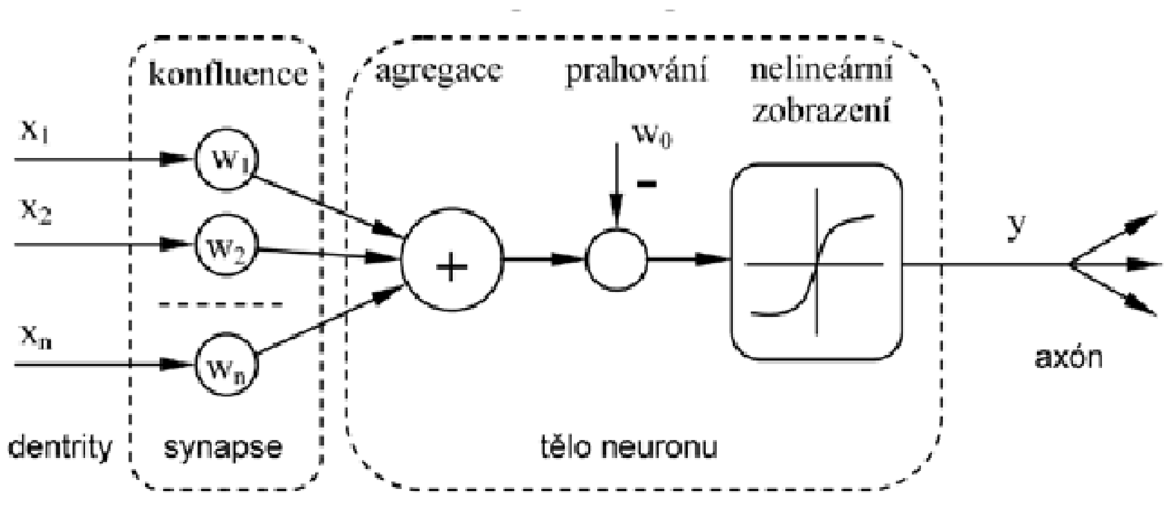
ARTIFICIAL NEURAL NETWORKS

- The biological network of neurons inspired scientists to create an artificial neural network
- Neurophysiological findings made it possible to create simplified mathematical models that can be used for neurocomputations in solving practical tasks in the field of artificial intelligence
- It is not a copy of the human brain, but an imitation of its basic function
- The brain learns by strengthening connections between neurons or making new connections
- A nerve cell receives an impulse from several neighboring neurons, adds up the energy and, if a certain threshold value is exceeded, sends the signal on

Formal neuron - a model of an artificial neuron

- *Dendrites*-they represent the point of entry of the signal into the body of the neuron.
- *Neuron body*-sums up the signals given by surrounding neurons. The internal potential determined in this way leads to the excitation of the neuron.
- *Axon fiber*-brings the signal given by the degree of excitation to the synapses.
- *Synapse*-they form the output devices of neurons that amplify or weaken the signal and transmit it to other neurons.

• An artificially created neuron functionally corresponds to its biological model and forms the basic "computational unit" of a more complex complex - neural networks



Comparison computer *x* human brain (Hlaváč 2006)

	Computer	**Human brain**
Computing unit	1 CPU	10^{11} cells
Memory	10^{9} bits of RAM, 10^{11} bits on disk	10^{11} neurons, 10^{14} synapses
Cycle length	10^{-8} seconds	10^{-3} seconds
Bandwidth	10^{9} bits per second	10^{14} bits per second
Recovery speed	10^{9} computing elements	10^{14} neurons per second

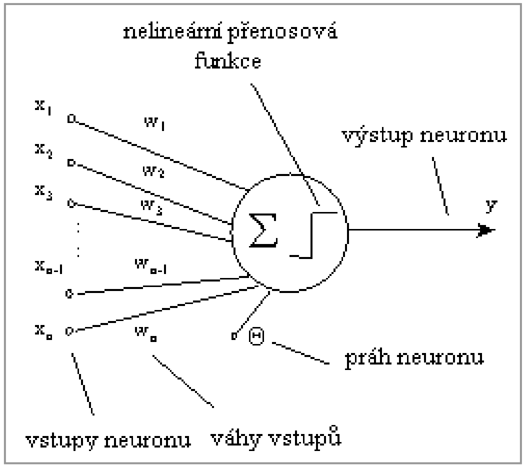
In general, we can formally describe a neuron according to the relation:

$$y = f\left(\sum_{i=1}^{N} w_i x_i + \Theta\right)$$

where y is the output of the neuron
x_{and} neuron inputs (total N)
w_{and} synaptic weights
F nonlinear transfer function of a neuron Θ
threshold

The expression in the parentheses of the relation is sometimes referred to as **the internal potential of the neuron** ***at***. Scales w_{and} for each neuron they represent its **local memory**; by connecting all neurons
we get **total memory** networks.

The learning of the network is done by changing these weights, but also by shaping the transfer function, changing the number of neurons in the network, or even the topological arrangement of the network.



Neuron model (McCulloch-Pitts)

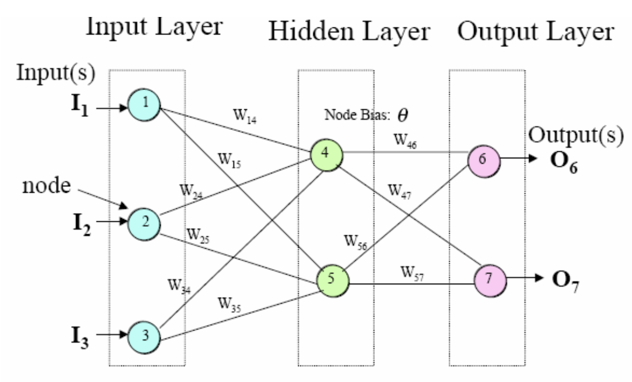
Properties of artificial neural network

- An artificial neural network is a distributive, adaptive, generally non-linear learning machine that is composed of many different information processing elements.
- When performing calculations, neural networks use distributed, parallel processing of information - storage, processing and transmission of information takes place through the entire neural network.
- Each element is connected to other elements or to itself through feedback.

- The basic element, i.e. the "computing unit" of a neural network is an artificial neuron. The strength of signals can be represented by real numbers and thereforea neural network can be described mathematically.
- The values of the signals transmitted between individual elements change depending on the adjustable parameters, which are called**Scales**, w_{ij}. The element sums all incoming weighted connection values and produces a result value that is a non-linear (static) function of its sum.

- An element's output can be a system output or can be sent to the same or a different element. The output of a neural network depends on the cooperation between individual neurons within the network.
- A specific feature of neural networks is that they are able to work even if some of the neurons are not working, because the processing relies on the collective result of the function of individual neurons. It means that neural networks are quite resistant to disturbances and errors.
- An important feature of neural networks is the ability to learn. Knowledge is stored primarily through the strength of connections between individual neurons.
- The number of neurons and their interactions **(topology)**neural networks.
- A neural network evolves over time, m

- **Network dynamics**:
  - – Organizational – topologist change
  - – Active – status change
  - – Adaptive – changing configurations

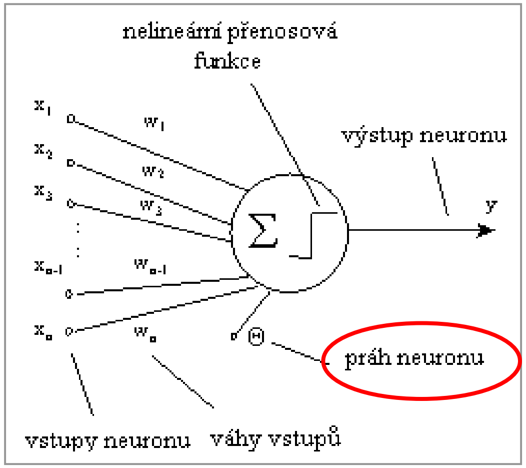


- Organizational dynamics
  - Specifies the architecture of the neural network and its possible change
  - Network architecture: network interconnection, number of layers
  - Cyclic (**recurrent**) network, acyclic (forward,**feed-forward**) Sew
- Active dynamics
  - Specifies the initial state of the network and how it changes over time with a fixed topology and configuration
  - Activation function – the procedure by which its output is calculated from the inputs of the nodes (**sigmoid fce**: step, standard sigmoid, hyperbolic tangent,...)
- Adaptive dynamics
  - Specifies the initial configuration of the network and how the weights on the connections between individual neurons change over time
  - The adaptive mode is used to learn the activation function - usually specified by the so-called training set, an iterative process
  - Two types of learning:**with the teacher**(error learning) a**without a teacher**(self-organization)

- Network modification can take place by changing the synaptic weights, adapting the transfer function or changing the typology

Threshold of a neuron Θ

- Threshold or the threshold value means the barrier that the neuron's input signal must overcome in order to propagate further through the neural network.
- The threshold value therefore determines when the neuron is active or inactive.
- If the value of the input signal of the neuron is lower than the threshold value, the output of the neuron is a signal corresponding to the passive state of the neuron. Once a threshold value is exceeded (this threshold value can also change during the process), the neuron becomes active and the output signal from the neuron increases up to a certain maximum value. This is given by the range of values of the relevant activation function.

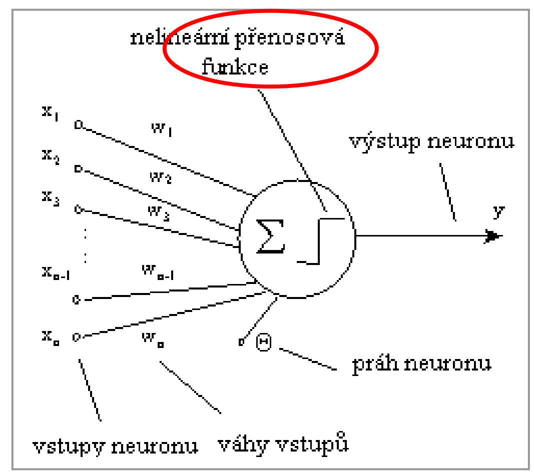


The transfer (activation) function of a neuron *F*

- The task of the transfer (activation) function of the neuron is to convert the value of the input potential to the output value from the neuron.
- The specific shapes of transfer functions tend to be very diverse.
  In principle, these functions can be divided into linear and non-linear, or continuous and discrete.
- The selection of a suitable transfer function depends on the specific type of problem being solved, or on the specific position of the neuron in the neural network.

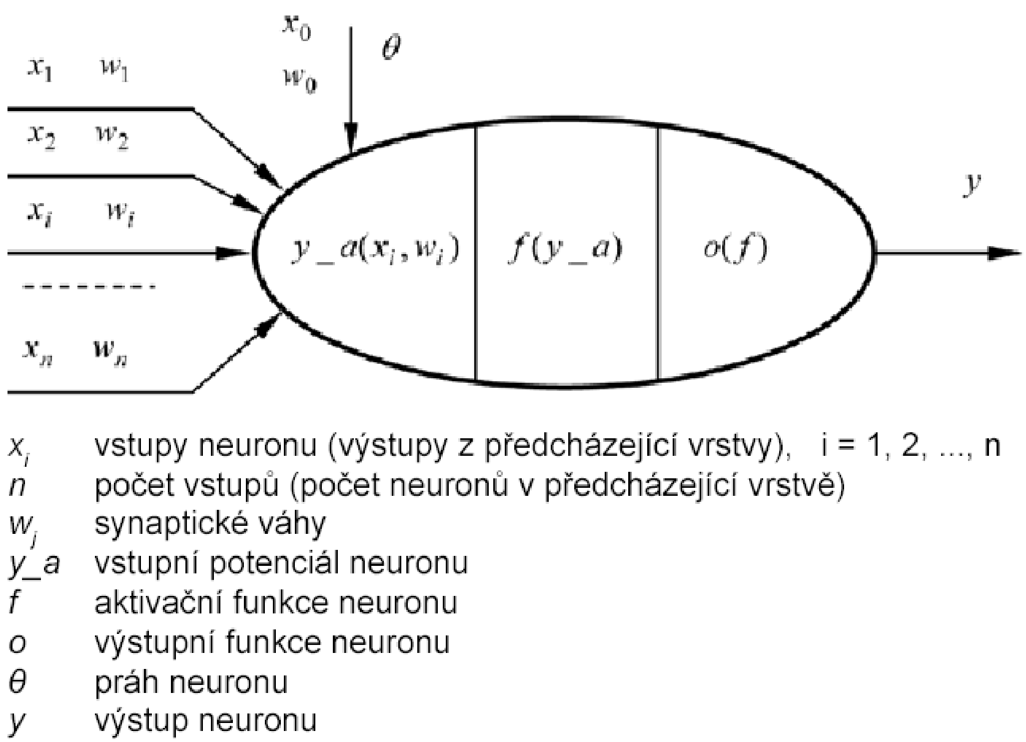
Sigmoidální funkce	$f_s(u) = \frac{1}{1+e^{-u}}$
Hyperbolický tangens	$f_h(u) = tgh\,(u)$
Znaménková funkce	$f_z(u) = sgn\,(u)$
Heavisideova funkce	$f_H(u) = \begin{cases} 1 & pro\ u > 0 \\ 0 & pro\ u \leq 0 \end{cases}$

*at*the internal potential of the neuron



Output function *about*

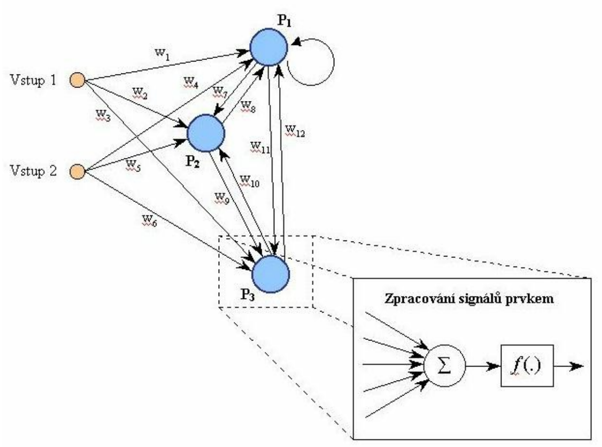
simulates signal transmission through an axon. This function "finishes" the final value of the output from the neuron.



Schematic of an artificial neuron with an output function

The structure of an artificial neural network

- The structure of a neural network consists of many nodes (neurons) connected by directional connections. Connections correspond to signals entering or leaving a given node, which transforms the input signals into a single output signal according to the specified relationship.
- Each neuron can simultaneously receive any finite number of different input signals. It can pass any finite number of identical information about the status of its output to other executive elements.
- At least part of the nodes is adaptive, i.e. that the outputs from them depend on modifiable parameters.
- The goal is to set these parameters so that the behavior of the network minimizes a certain criterion. Nodes are grouped into individual layers.

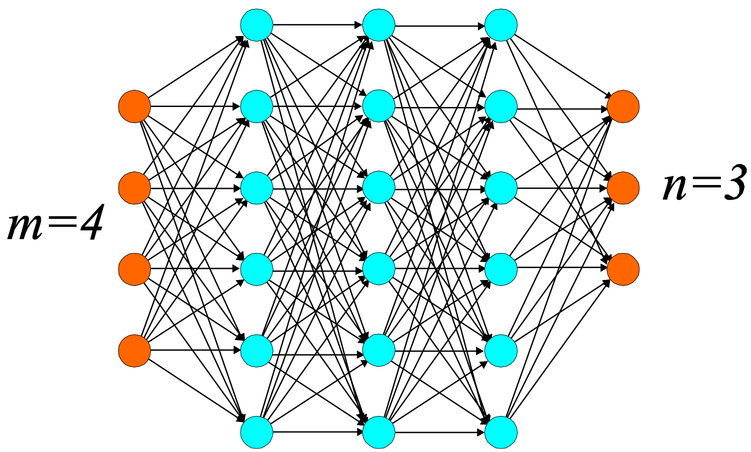


An artificial neural network with three P neurons and

Layered neural networkit is made up of at least three layers of neurons: input, output and at least one inner (hidden) layer.

There are always so-called complete interconnections of neurons between two adjacent layers. Such a network is able to approximate any mathematical function, while the accuracy of this approximation is greater the more neurons the network contains.

By adding more neurons to the network, the network can learn.



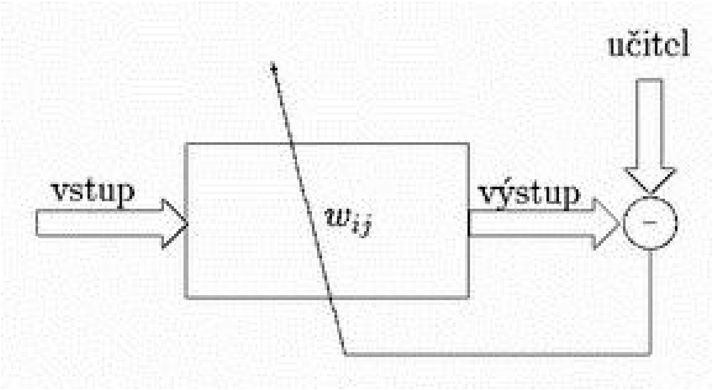
A layered network with an input layer of dimension *m,*with the dimension output layer *on* with *year*hidden layers (*grid 4-6-6-6-3).*

Learning an artificial neural network

The process of setting network parameters is called learning. According to the learning method, we divide neural networks into two basic types: learning with a teacher and learning without a teacher.

Neural network learningwith the teacher

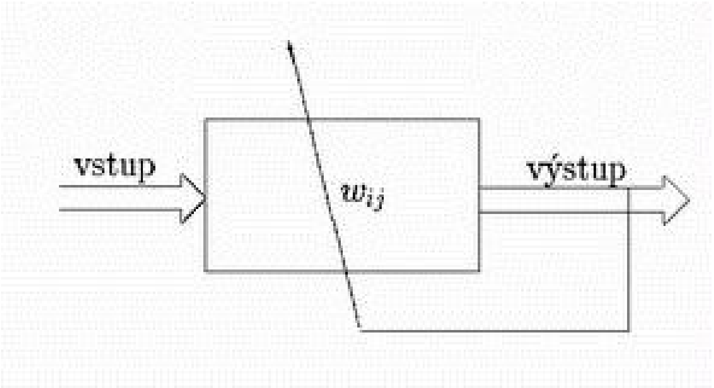
When learning with a teacher, the artificial neural network learns by comparing its output with the desired (teacher) output by adjusting the synapse weights (values in the matrix) to reduce the difference between the actual and desired output. (The desired output corresponds to the given training set.)



If the network is properly trained and the training data set sufficiently characterizes the entire data set, then it is likely that the neural network will behave "reasonably" even for data on which it was not trained.

Two types of tutored learning algorithms are distinguished. At**learning offline**(after batches) the values of the network parameters change only after passing through the entire training set of data. At**learning online** (sample-by-sample learning) the parameters of the network are modified after passing each data sample from the training set. In practice, both methods are often combined.

Network learning without a teacher



The network has no comparison signal to determine correctness. The connection weights are adjusted so that the network provides a response for the same or similar input vectors. The algorithm looks for samples with certain properties in the input data according to dependence, correlation.

For example, it is possible to analyze the effect of the season on the stock market, the number of mice on the harvest,
etc.

Learning without a teacher is referred to as **self-organization** (see e.g. Kohonen network, ART network). The working method is **clustering**. Similar inputs activate neurons that are close to each other.

The most famous neural networks used to process DPZ data

Neural networks

Fixed weights:

- Linear associative memory (LAM)
- Hopfield network

Learning with a teacher:

- Simple perceptron
- Multi-layered perceptron (MLP)

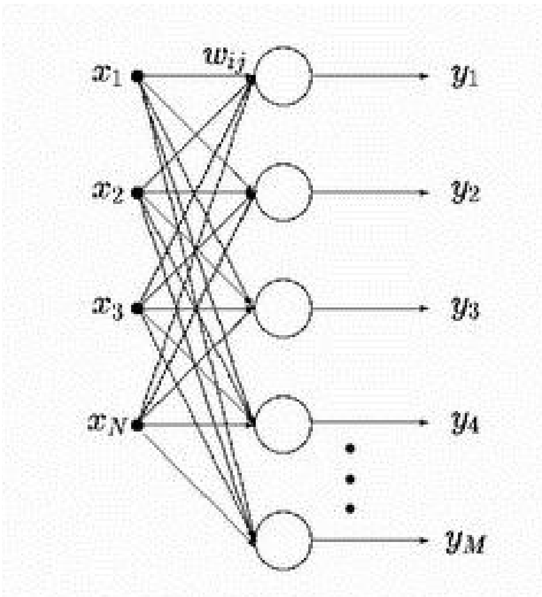
Learning without a teacher:

- Kohonen network
- ART network (Adaptive Resonance Theory)

Linear associative memory (LAM)

- single-layer feedforward network
- Input vectors can be binary (0
- It is used, for example, to reconstruct new buildings

Z



Výstupní hodnoty neuronů:

- pro reálné vstupy

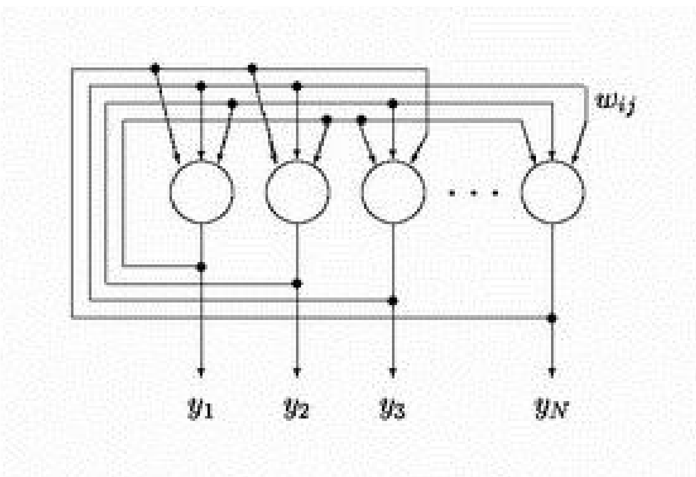
$$y_i = \sum_{j=1}^{N} w_{ij} x_j, \qquad pro \quad i = 1, \ldots, M$$

- pro binární vstupy

$$y_i = f_H\left(\sum_{j=1}^{N} w_{ij} x_j - \Theta_i\right), \qquad pro \quad i = 1, \ldots, M$$

Hopfield network

- A single-layer recurrent network with fixed
- Input vectors must be binary (0/1)
- It is used as an associative memory, for damaged images and for optimization



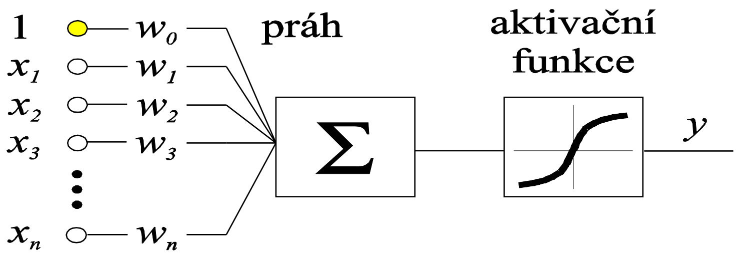
Neuronová aktivační funkce:

– funkce jednotkového skoku $f_H(u) = \begin{cases} 1 & u > 0 \\ 0 & u \leq 0 \end{cases}$

– znaménková funkce (pro bipolární vstupní vektory)

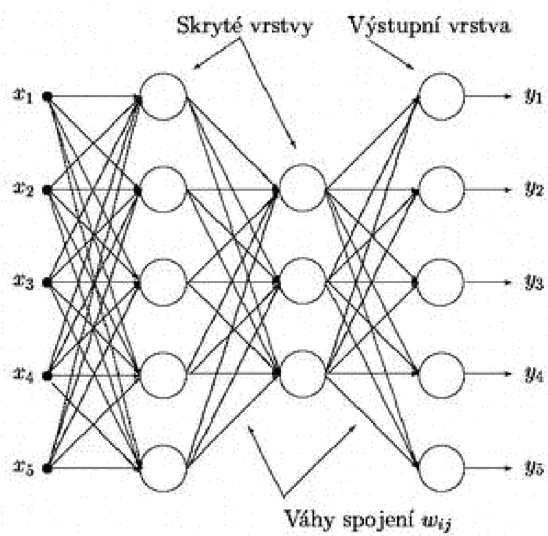
A simple perceptron

- Simple forward propagation neuron
- The network has *n* inputs and one working neuron connected to all its inputs
- Each connection is assigned a certain weight
- The transmitted signal is either binary (0,1) or bipolar (-1,0,1)
- Learning with a teacher
- Neuron activation function: familiar function, unit jump
- Classifier for linearly separable images



A multilayer perceptron

- Multi-layer network with forward propagation
- Learning with a teacher
- Neuron activation function: sigmoidal function, hyperbolic tangent (or other non-linear continuous function)
- Use: image classification, function approximation, time series prediction



Whom

- Yippee
- Mon

competitive strategy

teaching

- Vs

$$X = [x_1, \dots, x_N]^T, \quad x_i \in \mathbf{R}$$

- Ex

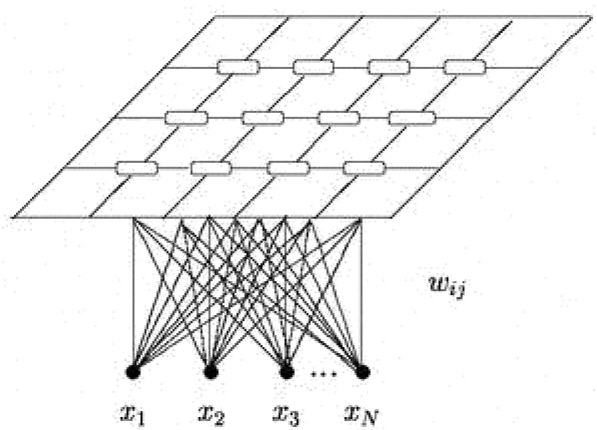
ost between the input and

Mrs

$$y_i(t) = \sum_{j=1}^{N} (x_j(t) - w_{ij}(t))^2$$

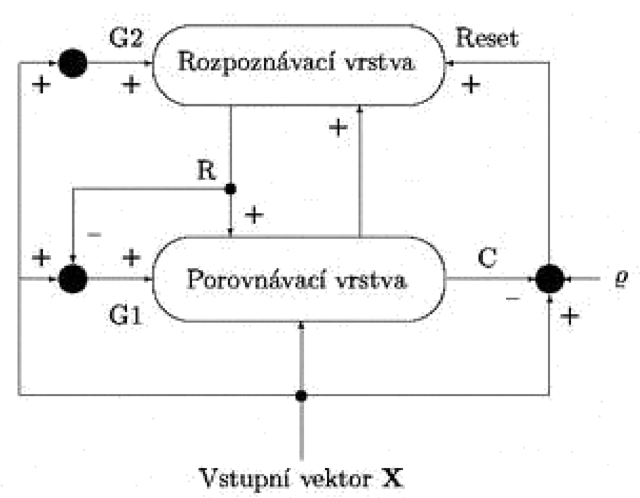
- Mon

ch maps



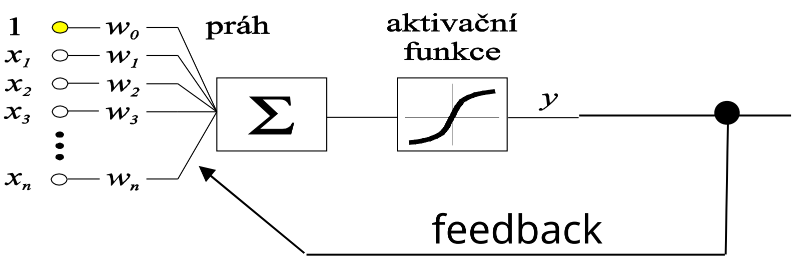
Adaptive Resonance Theory (ART) network

- Two-layer recurrent network
- Learning without a teacher
- Input vector X: binary for ART-1 model, real for ART-2 model
- NS usually cannot learn new information without damaging information already learned (variable stability of the network), however:
- Associative NS switch between malleable (learning) and stable state without damage to already learned information
- Application: clustering, recognition of characters, speech segments, etc.

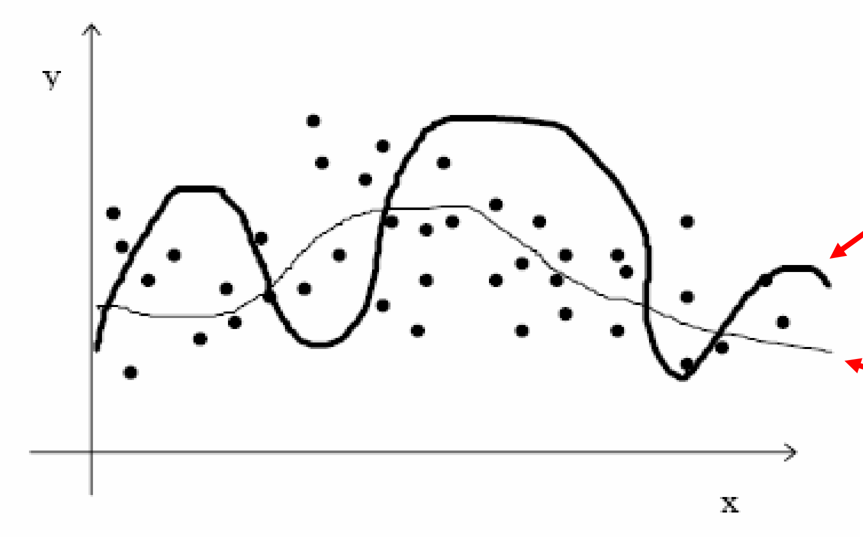


Error backpropagation(error back-propagation)

- Probably the most widespread method of connecting neurons with a sigmoid function is multilayer networks, it is used in 80% of NS
- 3 stages:
  - forward (feedforward) propagation of the input signal of the training pattern
  - error back propagation
  - update of weight values on connections
- During the adaptation of the neural network, the calculated activations are compared with the defined input values for each neuron in the output layer and for each training pattern. Based of this comparison is the defined error of the neural network for which it is calculated factor that propagates back to neurons from the previous layer.



- The backpropagation algorithm takes into account not only changes in the weights in the direction of the gradient of the error function, but also the previous change, the so-called moment (momentum).
- The moment determines the degree of influence of the previous change, usually 0.9 (0-1) is chosen
- Using the moment, the gradient method better describes the shape of the error function because it takes into account the previous gradient
- The architecture of a multilayer NS – i.e. determining the appropriate number of internal neurons and their connections – should correspond to the complexity of the problem being solved, i.e. the number of training patterns, their inputs and outputs and the structure of the relationships they describe
- A small network cannot solve a complicated problem
- Rich architecture – with a larger number of scales, the computational complexity of adaptation increases



Retrained network

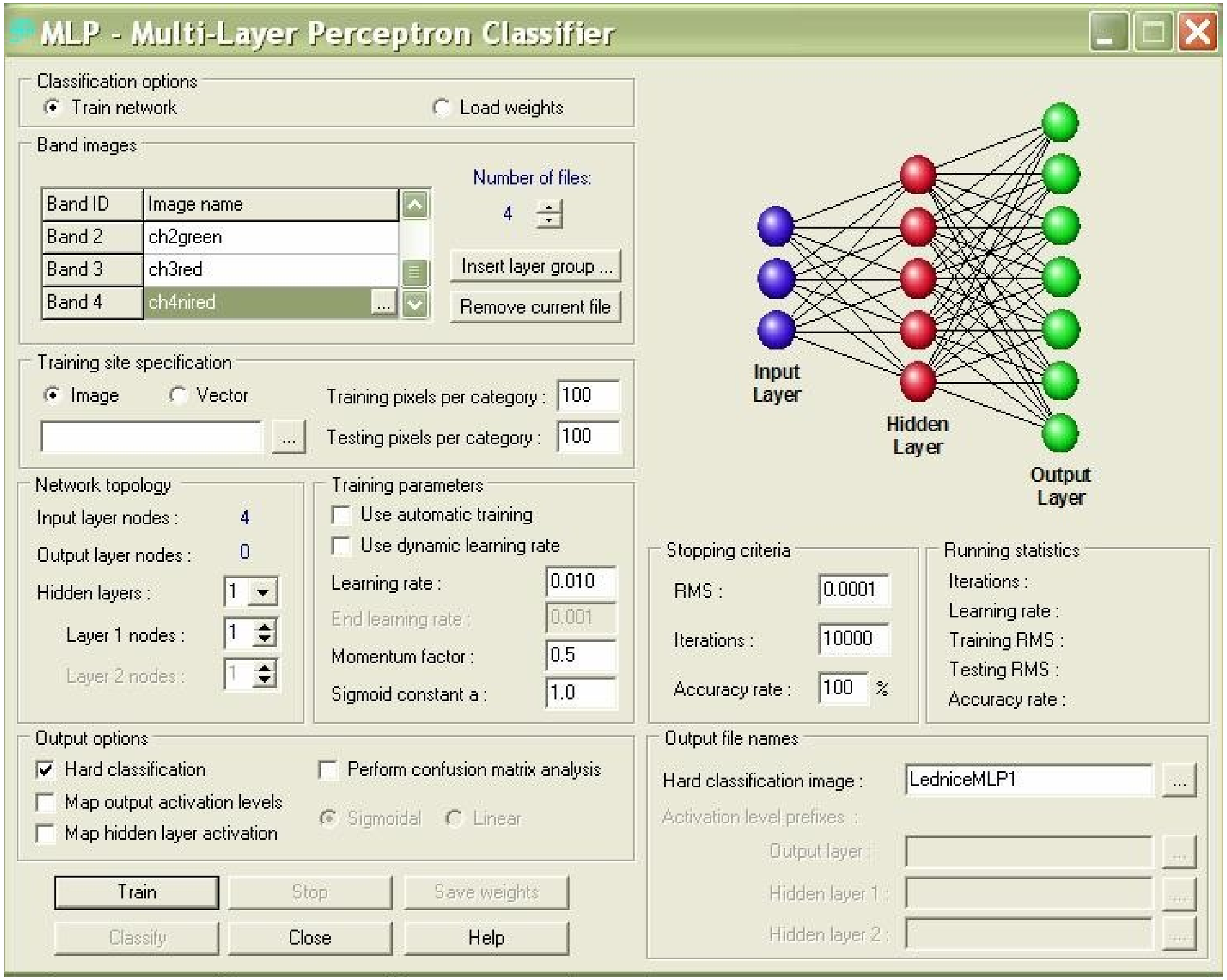
A "correct" generalization

Neural classifiers in Idrisi 15 Andes and 16 Taiga

- Idrisi 15 Andes and 16 Taiga software packages contain three classifiers, based on automated machine learning and neural networks.

  They are:

  - Multi-Layer Perceptron ► Multi-Layer Perceptron –**MLP**
  - Self-organizing Kohonen network ► Self-Organizing Map –**I AM**
  - Fuzzy map according to Adaptive Resonance Theory (ART)
    ► **Fuzzy ARTMAP**



Workflow

First it is necessary choose a method classification.

Two are offered options:

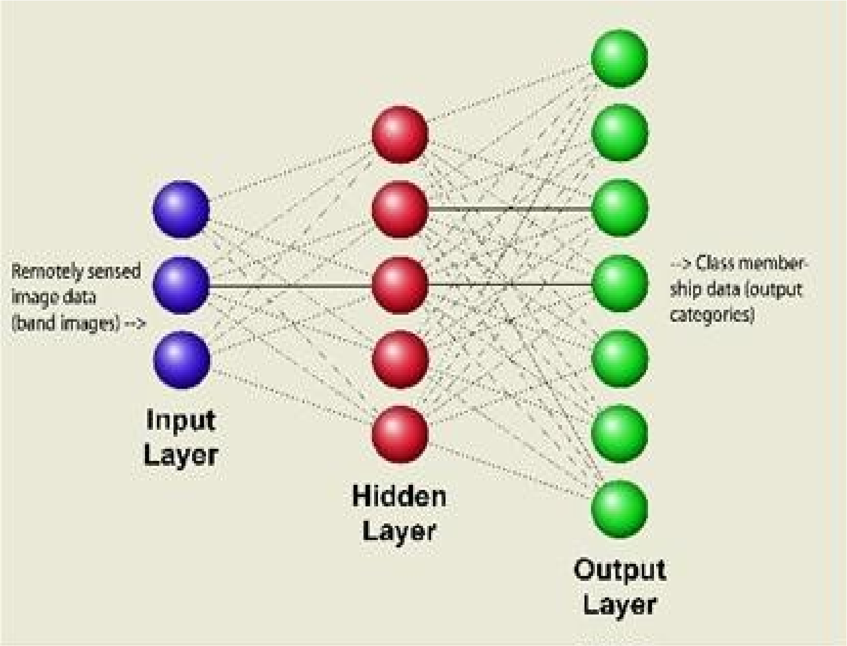
- Network training (Train network)
- Implementation existing set of scales (Load weights)

Working methods of the MLP module

The MLP module enables controlled classification. It uses an algorithm to train the network**error back-propagation**.

A layered neural network in this case contains one input layer, one output layer and one or more hidden layers. The function of hidden layers is analogous to lines that allow points in the multispectral space to be assigned to appropriate classification classes.

Backpropagation involves two main steps that allow the state of the neural network to be modified – **forward and back propagation**. The calculation depends on the information from the training sets, on the basis of which the synapse weights are corrected.



During the training phase, each sample (e.g. a feature vector corresponding to one pixel) is fed to the input layer and the receiving node (neuron) sums the weighted signals from all the nodes with which it is connected in the previous layer.

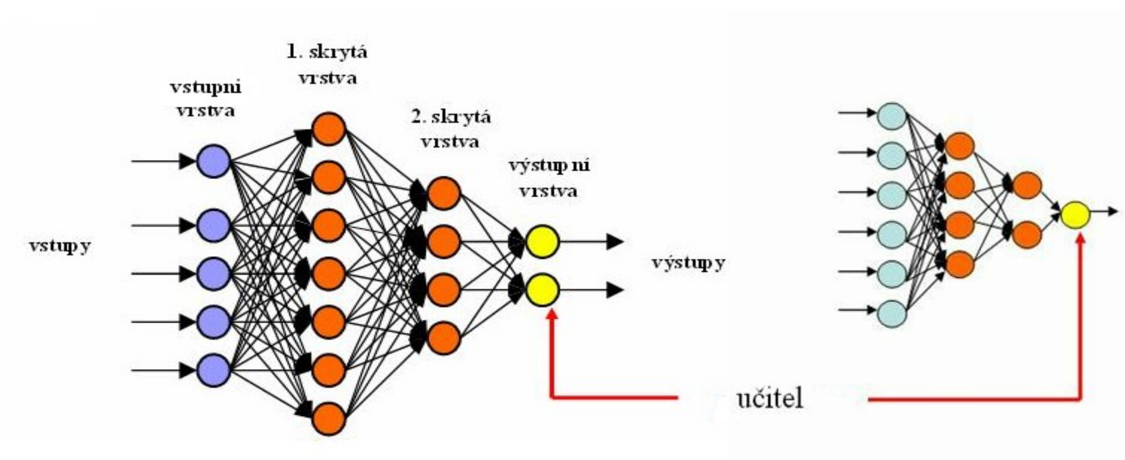
When the forward step is finished, the results at the output nodes are compared with the expected results.

Each output node is associated with a class. When a pattern is fed into the network, each output node generates a value that indicates the similarity between the input pattern and the corresponding class.

The actual output usually differs from the expected (ie the teacher's) output; the difference corresponds to the network error. This**the error then propagates backwards**with weights adjusted for individual links based on a relationship known as the delta rule:

$$\Delta w_{ji(t+1)} = \eta \partial_{ji} o_i + \alpha \Delta w_{ji(t)}$$

where η is the learning rate, α is the momentum factor and δ is the calculated error.

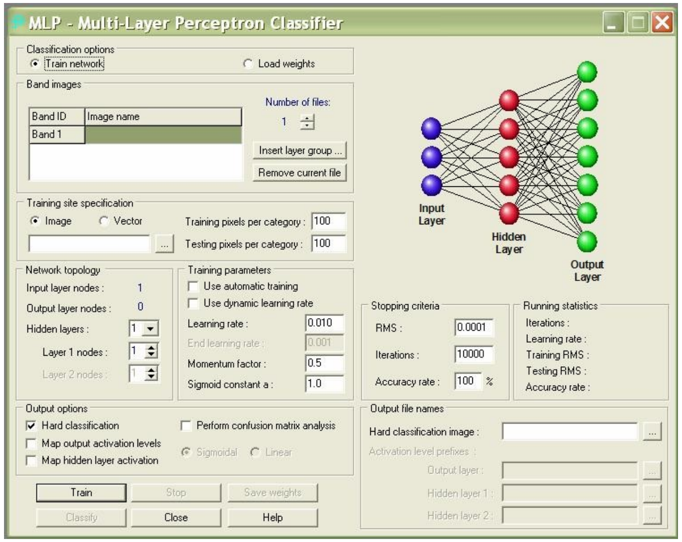


Forward and backward steps continue until the network "learns" the characteristics of all classes.

The goal of learning the network is to obtain the correct weights for the connections between the input layer and the hidden layers, and between the hidden layers and the output layer.

This then allows the classification of unknown pixels.

Classification



It is possible to set the classification output as: a)

hard classification map

b) output activation level maps

c) activation level maps of hidden layers.

Confusion matrix analysis can be chosen, which allows determining the accuracy of the classification.

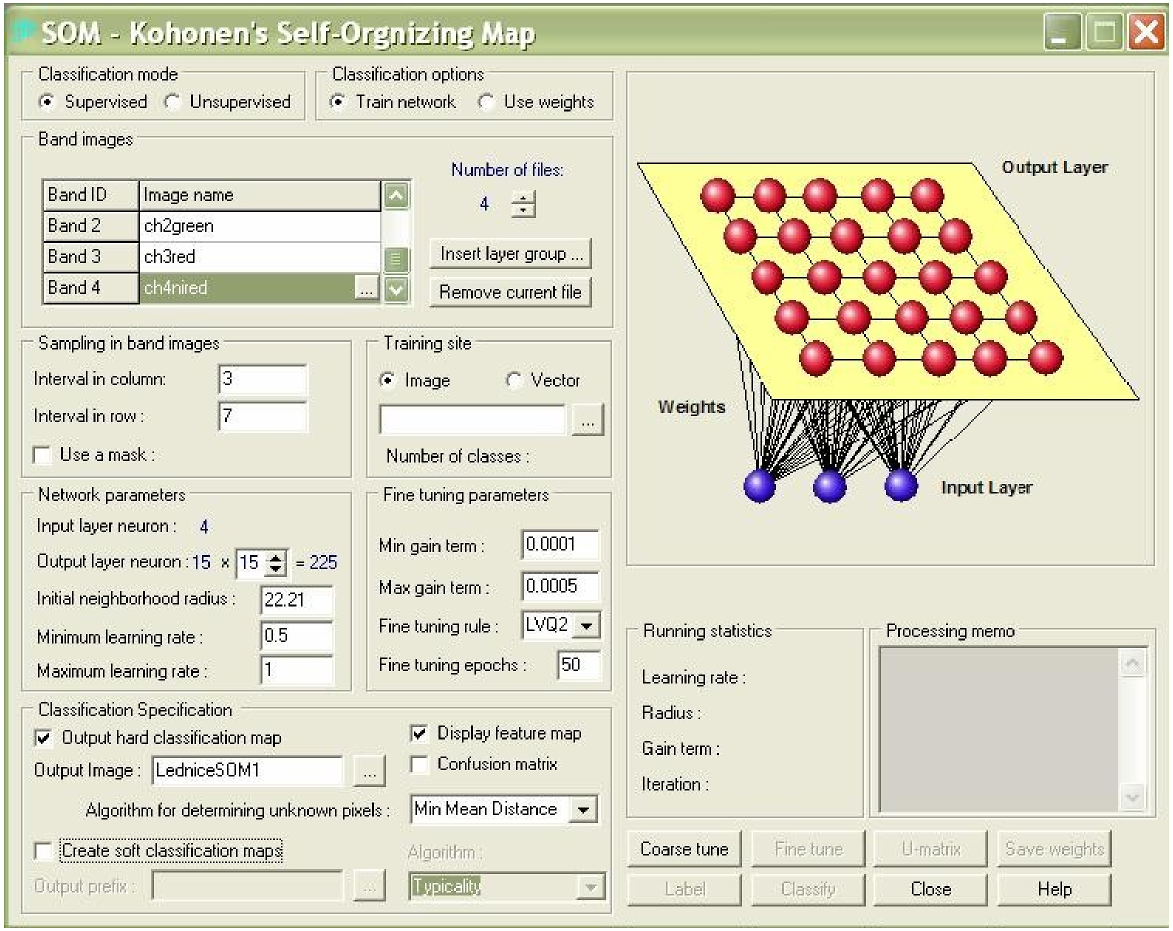
MLP can be considered both a hard and a soft classifier.

The hard classification result provides a map in which each pixel belongs to a certain class.

Soft classification result – activation level maps provide a group of images recording the degree to which each pixel belongs to each possible class. The output is therefore not a simple land cover classification map, but a set of images (one for each class).

Unlike probability maps, the sum of the values for each pixel position is not necessarily 1. This is because the outputs of the neural network are obtained by fuzzing the signals to values in the interval 0-1 through an activation function. Higher values represent a higher degree of belonging to the corresponding class.

Self-organizing Kohonen network ► Self-Organizing Map - SOM



Workflow

1. First, it is necessary to select a classification variant. There are two options:

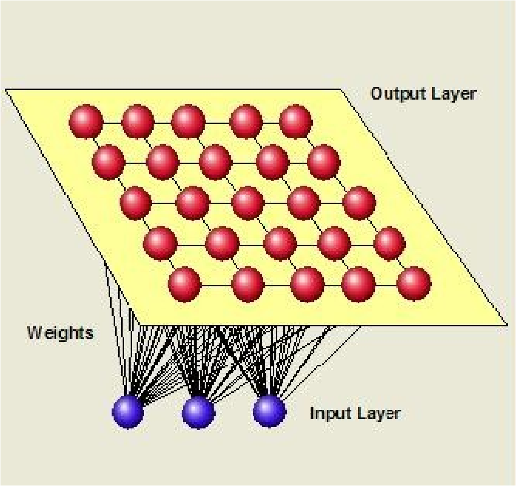
- Controlled classification (Supervised)
- Unsupervised classification (Unsupervised)

2. The method of classification is then determined:

- By training the network (Train network)
- Using an existing set of weights (use weights)

The SOM module implements both unsupervised and supervised classification of remote sensing image data based on **Kohonen self-organizing networks.**

It is a type of network that does not need a teacher to learn. It is based on a cluster analysis algorithm. The network algorithm has the ability to find certain properties based on translated training data, without the presence of any external information.



A Kohonen network has only two layers.

The number of inputs is equal to the dimension of the input space, in our case it has a value of three (the first layer contains only three neurons).

The second layer is typically arranged as a two-dimensional (usually square) array of neurons. Each output neuron is connected to all input layer neurons by synaptic weights.

Neurons do not have their own transmission function. The only operation that each neuron in the network performs is to calculate the distance (deviation) of the presented pattern from the pattern encoded in the weights of that neuron according to the relation

$$d = \sum_{i=0}^{N-1} \left[x_i(t) - w_i(t) \right]^2 .$$

Training and classification

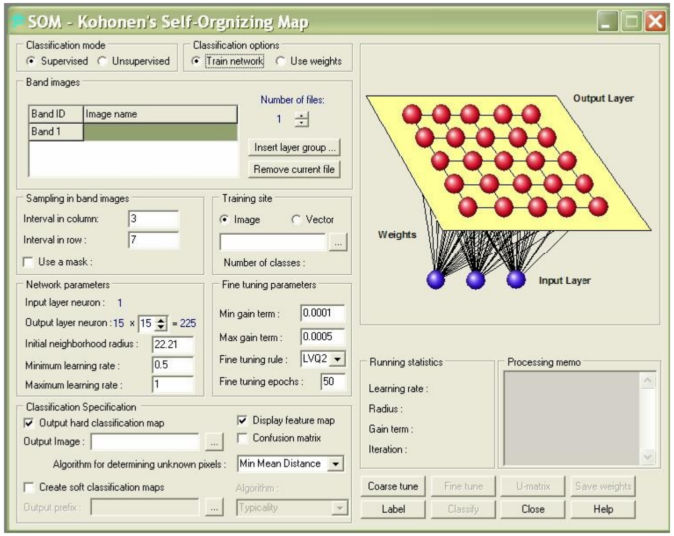
1. Before classifying image data, it is necessary firsttrain the network.

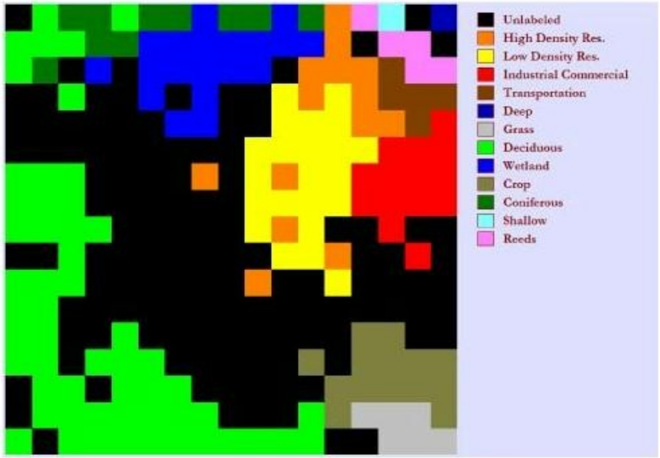
- After all the parameters have been introduced, the training process starts using the Coarse tune option.
- You can continue by selecting Fine tune.
- The U-matrix button allows you to display the U-matrix, which represents a map of the average distance of a neuron to its neighbors. This matrix is useful for detecting existing patterns in analyzed images.

At the end of the training process, the weights can be saved to a file (.som)

2. Own**classification**can be implemented:

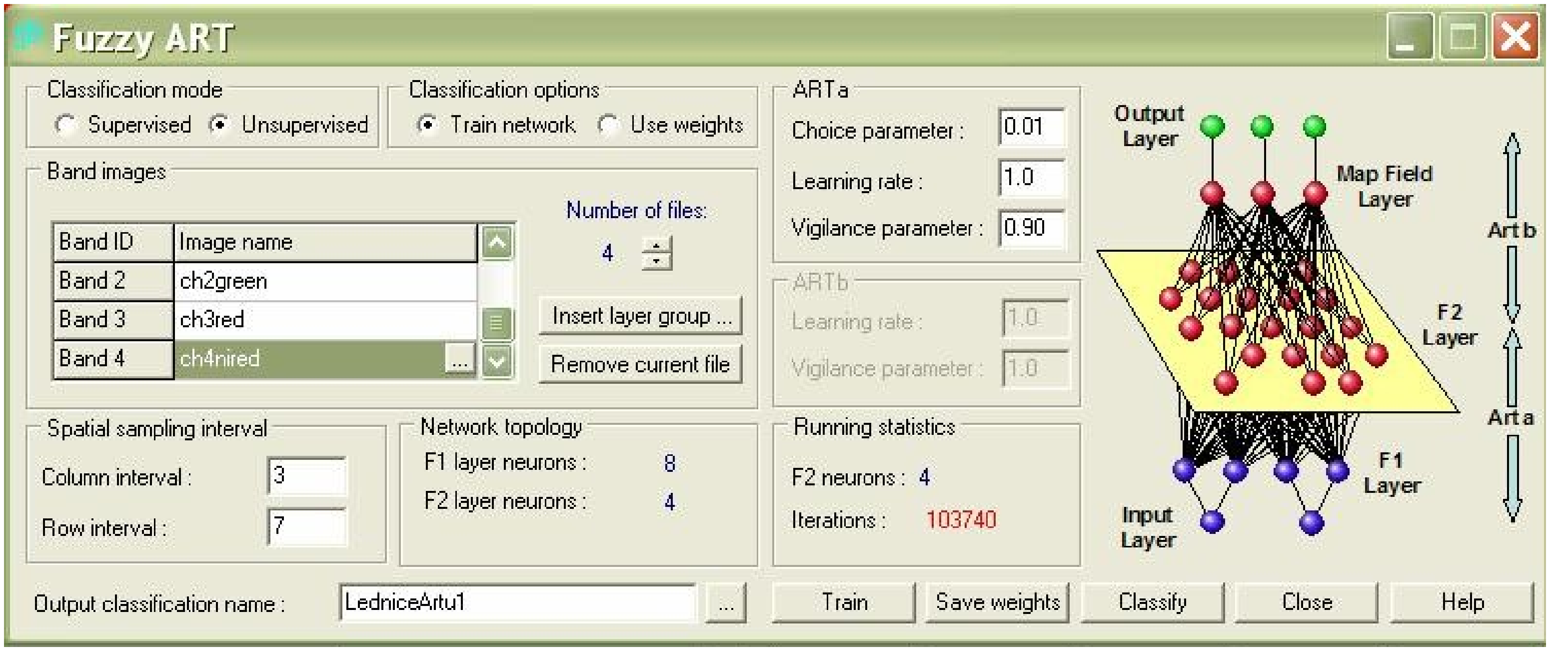
- based on the parameters of the weights achieved by training the network, or
- using an already existing set of weights.





Sample feature map for training data containing 12 classes

Fuzzy map according to Adaptive Resonance Theory (ART) ► (Fuzzy ARTMAP)



Workflow

1. First, the classification option must be selected. There are two options:

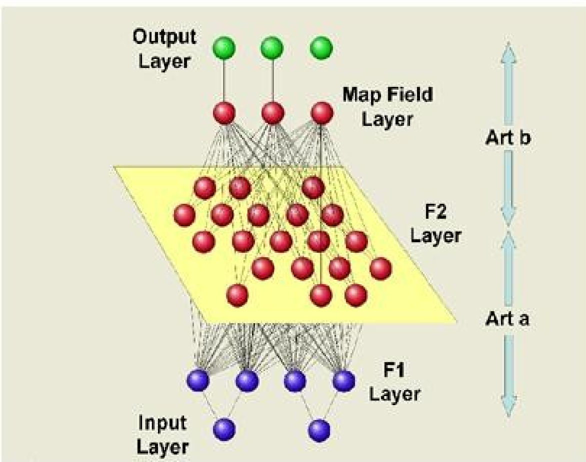
- Supervised classification
- Unsupervised classification

2. The method of classification is then determined:

- By training the network (Train network)
- Using an existing set of weights (Use weights)

The Fuzzy ARTMAP module implements both unsupervised and supervised classification of remote sensing image data. The Fuzzy ART algorithm is a clustering algorithm, working with vectors with fuzzy analog input patterns (real numbers in the interval 0.0 and 1.0) and including an incremental learning method. This enables the network to learn without forgetting previously achieved learning states.

A neural network based on Adaptive Resonance Theory (ART) was created by Grossberg and Carpenter (1991). It was developed from the biological theory of cognitive information processing.



Fuzzy ARTMAP for unsupervised classification has two layers, F1 (input layer) and F2 (category layer). These two layers make up the ARTa model. The F1 layer represents the input feature vector and contains neurons for each respective dimension. The n-dimensional input vector is processed by the relation

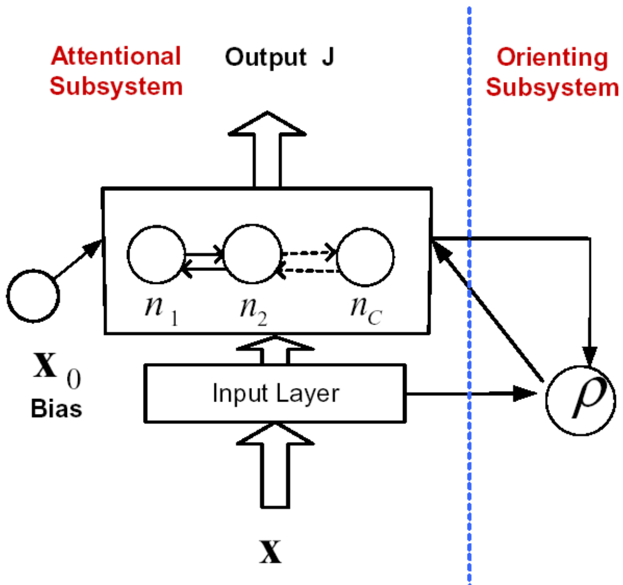
$$I = (a, a^c) = (a_1, a_2, \ldots a_N, 1 - a_1, 1 - a_2, \ldots 1 - a_N) \quad .$$

The number of F2 layer neurons is determined automatically; it starts with a single neuron and grows dynamically during learning.

For controlled classification, Fuzzy ARTMAP has two additional layers, a map field layer and an output layer. These two additional layers form the ARTb model. Both the output layer and the map field layer contain *m* neurons where *m* is the number of output classes. There is a one-to-one connection between the two layers. The vigilance parameter controls the "tightness" of clusters.

Before classifying the image data, the network must first be trained (option**Train network**), or use a file with existing synaptic weights (Option**Use weights**).

The input information oscillates in the form of output values between the two layers of neurons until resonance is reached. At this point, learning, i.e. adaptation of the scales, begins.



Architecture of the simplified ART network

Resonance can occur in two cases: (1) If the network has already processed the same or very similar sample in the past, resonance occurs immediately.

(2) In case the input sample is different from all the previous ones, the process of searching the learned codes is started and their similarity with the presented sample is compared; at the same time, a certain threshold value is defined, which determines the minimum permissible similarity of the winning class. When none of the known classes meet this threshold, the system creates a new class identical to the submitted sample.

This achieves both stability - the network resonates in the case of known input (see 1) and plasticity - the network has the ability to learn new unknown samples (see 2).

Neural classifiers in Idrisi 15 Andes and 16 Taiga - summary

Multi-Layer Perceptron (MLP)

This neural network enables supervised classification. It works with the error backpropagation algorithm. It includes work in automatic (meta-intelligent) mode, progressive learning rate modification, two hidden layers and the ability to map all activation layers, including hidden layers.

Self-Organizing Kohonen Network (Self-Organizing Map - SOM)

This neural classifier can be used in both supervised and unsupervised classification variants. The network takes into account the lateral interactions of neurons. Performs a projection from a multidimensional space to a lower dimensional space. Learning Vector Quantization (LVQ) is available for fine-tuning the network.

Fuzzy map according to adaptive resonance theory ART (Fuzzy ARTMAP)

This neural network provides both supervised and unsupervised classification. It works based on the theory of adaptive resonance. This efficient operation requires a minimum of human interaction. The network exhibits both stability (resonates to known input) and plasticity (ability to learn new unknown patterns).

DECISION TREES

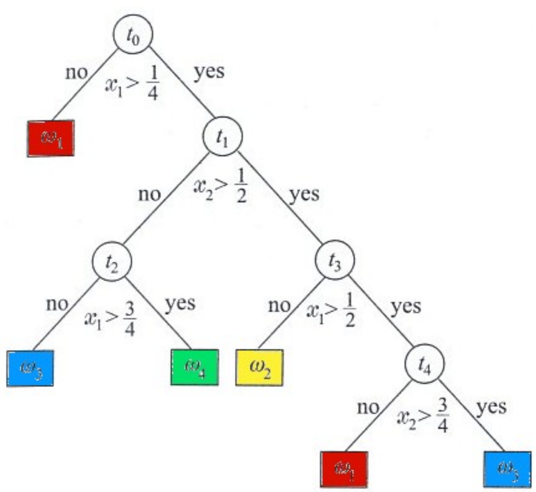
- Representation of knowledge in the form of decision trees is well known from a number of areas. Let's recall the various "keys to determine" different animals, plants or minerals.
- Induction (proceeding from specific cases to a general conclusion) of decision trees is one of the most well-known algorithms in the field of symbolic machine learning methods.
- From the root of the tree, based on the answers to the questions (placed in non-terminal, i.e., non-leaf nodes), the relevant branch progresses deeper and deeper, until the terminal, so-called leaf node, which corresponds to the inclusion of the example in the class.

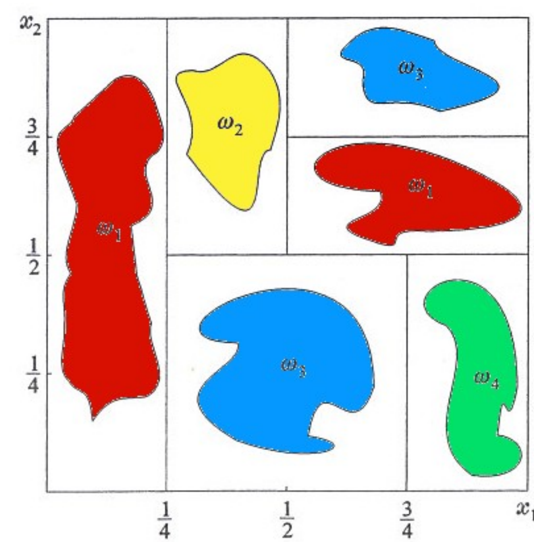
- The goal of decision trees is to identify objects, described by various attributes, into classes. We can imagine them as rows in a table, where individual columns are attributes (e.g. DN values). Since it is a tree, the algorithm is very fast.
- The decision tree must first be created from a set of given objects, which someone (a teacher, another algorithm) must classify into groups=classes (the group is usually referred to as a dependent attribute and written in the table in the last column). So it is learning with a teacher.
- Each node of the tree represents one (selected) property of objects, a finite number of edges - branches lead from this node. Therefore, it is necessary to first discretize the properties (e.g. from real numbers to a finite number of intervals).
- The main problem is creating such a tree. He must differentiate the objects from each other as best as possible. Therefore, an attribute is chosen for the root node that enables the best possible resolution of objects. Entropy (a measure of the attribute's information value) is used as a parameter.

Binary decision trees

- The most famous are ordinary binary decision trees
  (OBCT = ordinary binary classification trees)
- divide space into polygons that have sides parallel to the axes
- the decision sequence is applied to individual symptoms
- decision-making takes place according to questions in the form "is the symptom x**and**≤ α?'

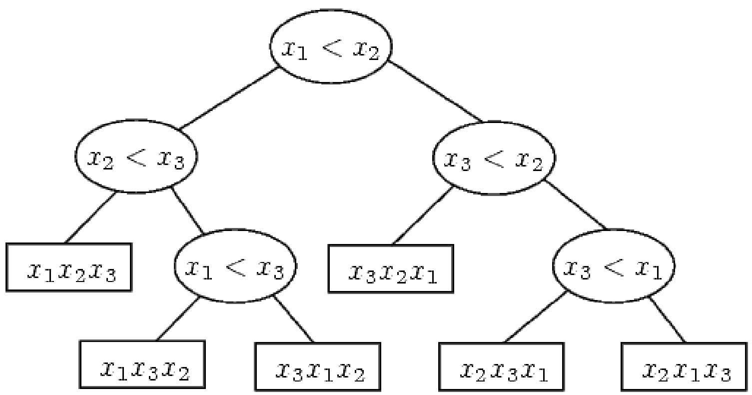
- by gradually dividing the space, we create areas that correspond to individual classes:
- corresponding binary tree:





The tree is composed of:

- **roots**(root), which represents the beginning,
- **nodes**(internode), which provide connections between the root, other nodes and leaves,
- **leaves**, which mean decisions; in the case of image data classifications, they are formed by groups of pixels assigned to the same class.



Source: http://ksp.mff.cuni.cz/tasks/18/k181.png

- Classification using decision trees generates an output in the form of a binary tree, which is usually easy to interpret.
- The decision tree model contains rules that allow arriving at the target variables.
- The classification algorithm provides an easy-to-understand description of the distribution of the processed data.
- The full binary tree algorithm allows for sequential binary splitting of the data set and the creation of homogeneous subsets.
- **The goal is to find a tree consistent with the training data.** Preference is given to smaller trees.

Growth and pruning of a decision tree

- The decision tree classifier works in two phases:
  - **growth phase**–creation of a decision tree
  - **pruning phase**–tree reduction so that instances of one class predominate in the leaf node.
- When creating a decision tree, the "divide and conquer" method is used. The training data is gradually divided into smaller and smaller subsets such that examples of one class dominate those subsets. The goal is to assess the relevant attribute in each node and find an expression that best splits the training data in a given branch. The value of this expression depends on how well it separates the considered classes.
- After the growth phase is completed, a new subtree is created (pruned), based on removing the most frequent errors. In this phase, small nodes are removed, which are the result of noise in the training data
- This results in a more accurate classification of unknown data.

Growth Algorithm

1. Take one attribute as the root of the subtree
2. Divide the data into subsets based on the values of this attribute
3. If all the data in the subset do not belong to the same class, repeat the procedure from point 1 for this subset.

Pruning algorithm

1. Convert the tree to rules
2. Generalize the rule's removal of a condition from an assumption if the estimated accuracy improves
3. Arrange the pruned rules by estimated accuracy; in this order, the rules will be used for classification

Example of a decision tree application (according to Dobrovolný, 2001)

We have at our disposal multispectral data from the Landsat TM satellite, where for each object (pixel) we know its value (attribute) in 6 different parts of the spectrum (TM1 to TM6).

For the training data, we also know the class to which each object belongs:

- Water - bodies of water
- Forest - areas with a predominance of deciduous forest
- Forest - areas with a predominance of coniferous forest
- Topping - areas of fields with vegetation
- Poleb - areas of fields without vegetation
- Zast - built-up areas

The resulting tree may look like this, for example (the object's target class is listed in the tree leaf (after the colon):

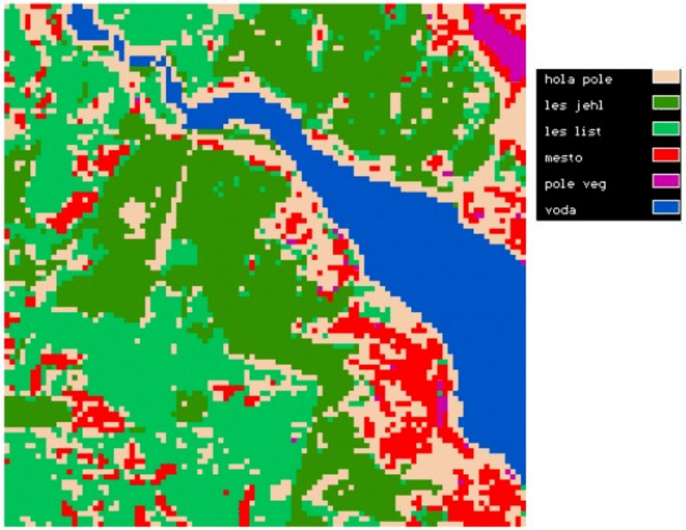
```
TM3 <= 35 :
|   TM4 > 99 : polev (12.0)
|   TM4 <= 99 :
|   |   TM5 > 58 : lesl (30.0/1.0)
|   |   TM5 <= 58 :
|   |   |   TM6 <= 12 : lesl (2.0)
|   |   |   TM6 > 12 : lesj (8.0)
TM3 > 35 :
|   TM6 <= 23 : voda (17.0)
|   TM6 > 23 : poleb (26.0)
```

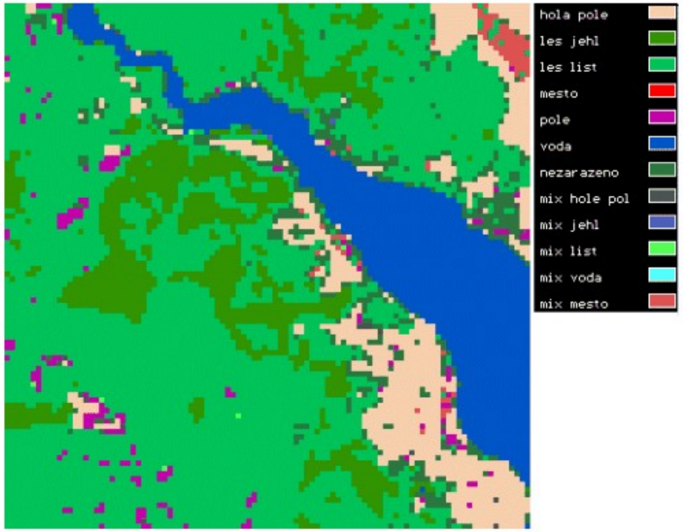
E.g. for an object whose attribute values - DN values of pixels in individual bands - are (20,1,32,97,70,21), the conditions TM3 <= 35, TM4 <=99 and TM5 > 58 are met.

Therefore, this pixel will be classified into the lesl class.

The total number of objects from the training set that fall into this sheet and the number of objects that were misclassified are given in parentheses after the sheet.

- Classified area on color synthesis
- Classification result using the maximum likelihood method (bottom left)
- Result of decision tree classification (bottom right)





General decision trees

- Not only continuous attributes, but also enumeration attributes can be used to create a decision tree.
- A node in the tree that contains an enum attribute will have as many branches as the attribute can take on. Then it is no longer a binary tree.

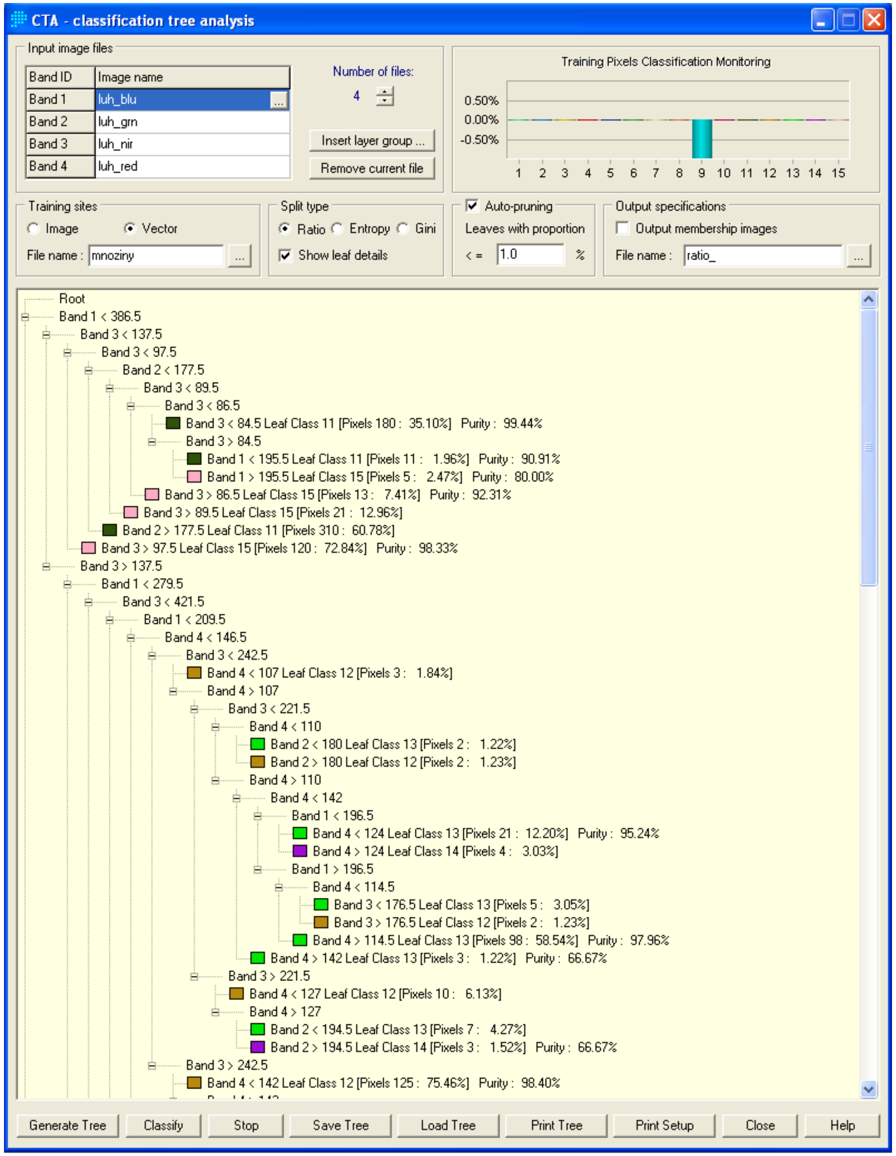
Classification Tree Analysis (CTA) in Idrisi 16 Taiga

- The Classification Tree Analysis (CTA) module enables the classification of remote sensing data as well as auxiliary data. It is suitable for land cover mapping.

- CTA represents a one-dimensional **non-parametric** technique. This means that data associated with a certain class based on a certain attribute may not follow any specific (eg normal) distribution.

- Thanks to this, it can process classes with unusual characteristics, such as impervious surfaces containing elements with high reflectivity (asphalt) and low reflectivity (concrete). *Who knows an example of such a surface?*

- CTA gradually splits the data by creating homogeneous subclasses, resulting in a hierarchical tree of decision rules.


CTA - classification tree analysis
Input image files
Band ID | Image name
Band 1 luh_blu
Band 2 luh_grn
Band 3 luh_nir
Band 4 luh_red
Number of files:
4
Insert layer group ...
Remove current file
Training Pixels Classification Monitoring
0.50%
0.00%
-0.50%
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Training sites
Image
Vector
File name : mnoziny
Split type
Ratio
Entropy
Gini
Show leaf details
Auto-pruning
Leaves with proportion
< = 1.0 %
Output specifications
Output membership images
File name : ratio_
Root
Band 1 < 386.5
Band 3 < 137.5
Band 3 < 97.5
Band 2 < 177.5
Band 3 < 89.5
Band 3 < 86.5
Band 3 < 84.5 Leaf Class 11 [Pixels 180 : 35.10%] Purity : 99.44%
Band 3 > 84.5
Band 1 < 195.5 Leaf Class 11 [Pixels 11 : 1.96%] Purity : 90.91%
Band 1 > 195.5 Leaf Class 15 [Pixels 5 : 2.47%] Purity : 80.00%
Band 3 > 86.5 Leaf Class 15 [Pixels 13 : 7.41%] Purity : 92.31%
Band 3 > 89.5 Leaf Class 15 [Pixels 21 : 12.96%]
Band 2 > 177.5 Leaf Class 11 [Pixels 310 : 60.78%]
Band 3 > 97.5 Leaf Class 15 [Pixels 120 : 72.84%] Purity : 98.33%
Band 3 > 137.5
Band 1 < 279.5
Band 3 < 421.5
Band 1 < 209.5
Band 4 < 146.5
Band 3 < 242.5
Band 4 < 107 Leaf Class 12 [Pixels 3 : 1.84%]
Band 4 > 107
Band 3 < 221.5
Band 4 < 110
Band 2 < 180 Leaf Class 13 [Pixels 2 : 1.22%]
Band 2 > 180 Leaf Class 12 [Pixels 2 : 1.23%]
Band 4 > 110
Band 4 < 142
Band 1 < 196.5
Band 4 < 124 Leaf Class 13 [Pixels 21 : 12.20%] Purity : 95.24%
Band 4 > 124 Leaf Class 14 [Pixels 4 : 3.03%]
Band 1 > 196.5
Band 4 < 114.5
Band 3 < 176.5 Leaf Class 13 [Pixels 5 : 3.05%]
Band 3 > 176.5 Leaf Class 12 [Pixels 2 : 1.23%]
Band 4 > 114.5 Leaf Class 13 [Pixels 98 : 58.54%] Purity : 97.96%
Band 4 > 142 Leaf Class 13 [Pixels 3 : 1.22%] Purity : 66.67%
Band 3 > 221.5
Band 4 < 127 Leaf Class 12 [Pixels 10 : 6.13%]
Band 4 > 127
Band 2 < 194.5 Leaf Class 13 [Pixels 7 : 4.27%]
Band 2 > 194.5 Leaf Class 14 [Pixels 3 : 1.52%] Purity : 66.67%
Band 3 > 242.5
Band 4 < 142 Leaf Class 12 [Pixels 125 : 75.46%] Purity : 98.40%
Generate Tree
Classify
Stop
Save Tree
Load Tree
Print Tree
Print Setup
Close
Help

Decision Tree Classification of Remote Sensing Data Problem

- The existence of correlation between independent variables (which is common in remote sensing data) leads to very complex trees.
- This can be avoided if the principal components (in Idrisi PCA) or canonical components (in Idrisi CCA) transformation is applied first.

- This simplifies the classification tree, but usually makes it more difficult to interpret.

Conclusion on classification by machine learning methods

- The ability of machine learning models to learn from examples is important (from the point of view of knowledge acquisition).
- With decision trees, the knowledge found is comprehensible to the user (so-called white box).
- In a neural network, knowledge is "spread" in the form of weights of individual connections between neurons. A neural network behaves like a black box; it's not very obvious what's going on inside.
- It is up to the user to determine which method he will prefer for his work.

RESOURCES

- Eastman, JR 2009: Idrisi TAIGA Guide to GIS and Image Processing, Clark Labs, Worcester, MA
- Volná, E. 2002: Neural networks 1. University of Ostrava, Faculty of Science. Educational text.
- Vomlelová, M. 2006: Decision trees. http://kti.mff.cuni.cz/~marta/slistromy.pdf
- Vyorálková, I. 2002: DPZ image processing by neural networks. Thesis. Charles University, Prague.
- Matoušek V., 2006: Artificial intelligence and recognition, part of a neural network.
- Veselovský M.: Neural networks. http://avari.cz/uir/index.php
- http://bio.felk.cvut.cz/~novakd/mws/doc/neuron/neursite.html, see NS5
- http://gis.vsb.cz/GIS_Ostrava/GIS_Ova_2001/Sbornik/Referaty/Dobrovolny.htm