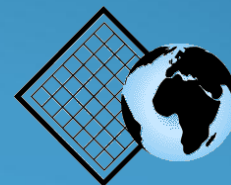


# Klasifikace obrazu III – strojové učení



# Klasifikace obrazu metodami strojového učení

Alternativa k tradičním statistickým metodám per pixel

Výhody strojového učení:

- možnost zohlednění jiných než spektrálních charakteristik např.
  - ✓ objektově orientovanou klasifikaci
  - ✓ segmentaci obrazu
  - ✓ okolí a sousedy pixelů

Rozdělení metod dle disciplín:

- 1. Statistické metody:** regresní metody, diskriminační a shluková analýza, Bayesovské metody
- 2. Symbolické metody UI:** rozhodovací stromy, asociační a rozhodovací pravidla
- 3. Subsymbolické metody UI:** neuronové sítě, Bayesovské sítě, genetické algoritmy

# Neuronové sítě

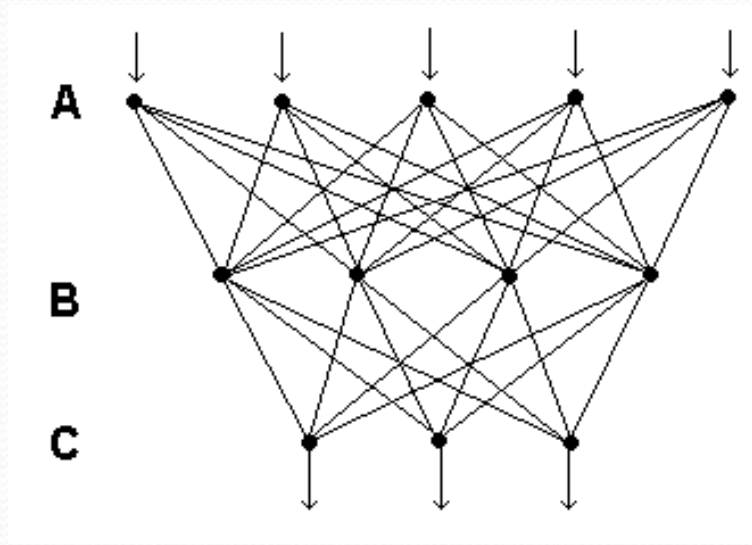
- počítačová architektura, která se snaží napodobit procesy probíhající v nervové soustavě
- je nezávislá na statistickém rozložení dat
- je odolná proti chybám, má schopnost učit se (asociativní učení), dovede abstrahovat i generalizovat
- dokáže odhadnout nelineární vztah mezi vstupními a požadovanými výstupními daty
- umožňuje v procesu klasifikace kombinovat různé typy vstupních dat.

název sítě	úloha
HEBB	logické obvody
MLP	optické rozpoznávání znaků
OLAM	dolování dat
RBF	odstranění šumu
SOM	řeč, výslovnost

*Zdroj: Kukač, 2012*

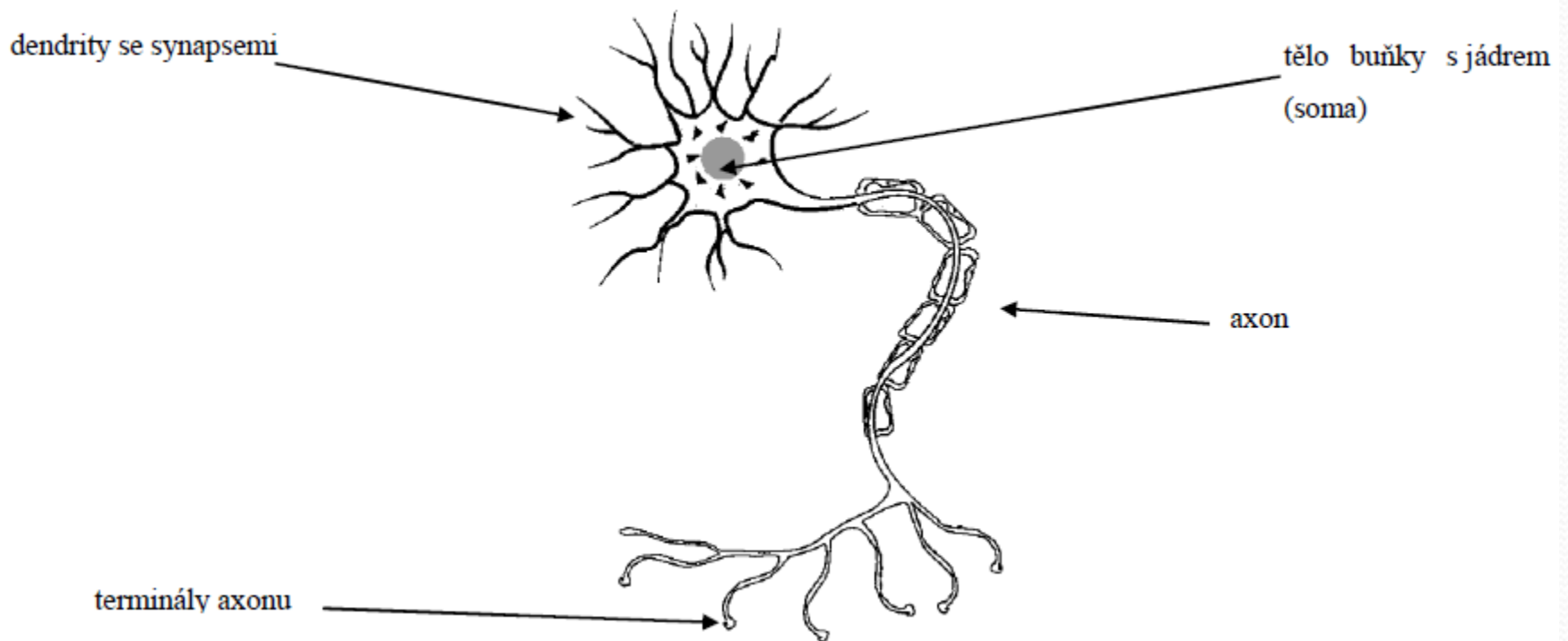
# Základní pojmy

- neuron – výkonný prvek NS
- synapse – spojení neuronů
- váhové koeficienty neuronů
- adaptivní fáze – učící
- aktivní fáze – vybavovací



Schematizované uspořádání neuronové sítě A - vstupní vrstva, B - skrytá vrstva, C - výstupní vrstva

# Stavba lidského neuronu



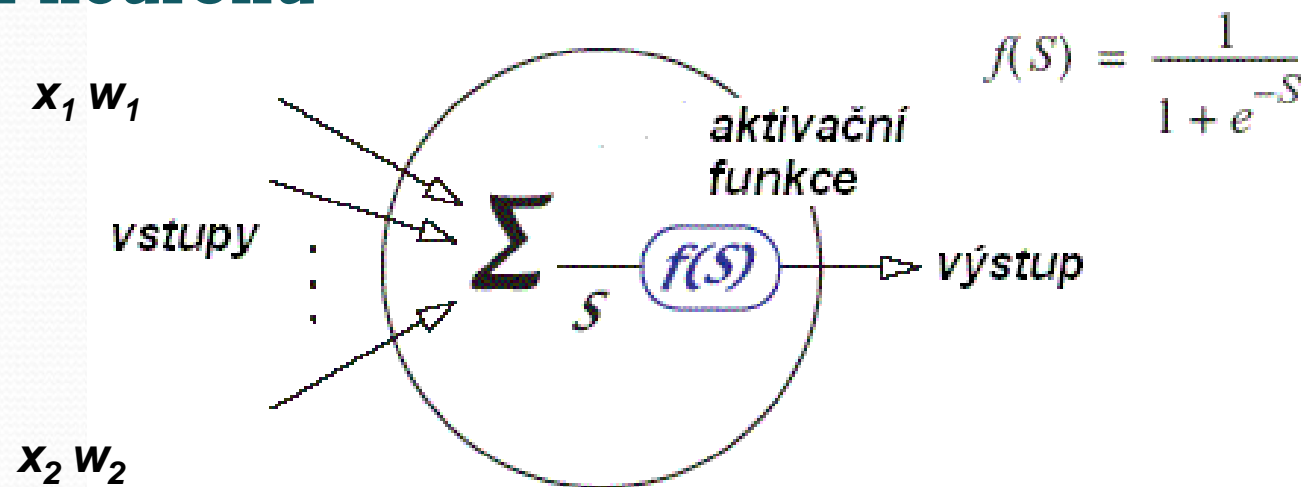
analogie s UNS

dendrity = vstupy (kvantitativní, kvalitativní), sloučením vznikne vstupní vektor

synapse zajišťují spojení s terminály axonů = váhy

funkce axonu = přenosová (aktivační) funkce

# Model neuronu



*kde*  $x_i$  - hodnota  $i$ -tého vstupu  
 $w_i$  - váha  $i$ -tého vstupu

$$S(x) = \sum_i w_i x_i$$

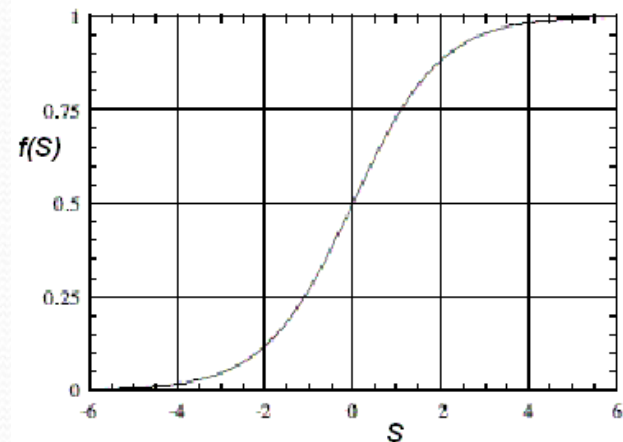
Neuron provádí tři akce:

- sumuje vstupy z jiných neuronů
- provádí prahování
- posílá výstup do jiných neuronů

# Model neuronu

- Neurony v síti jsou propojeny tzv. **váhovými koeficienty**, které zesilují nebo zeslabují signál přicházející z předchozích neuronů.
- Suma těchto vážených signálů určuje aktivaci neuronu, která ovlivňuje další výstup z neuronu.
- Výstup z neuronu je funkcí této aktivace, kdy výstup je vypočten na základě logistické aktivační funkce - **sigmoida**.
- Výstup z uzlu je realizován pouze překračuje-li určitou prahovou hodnotu.
- Váhové koeficienty jsou na počátku náhodnými čísly

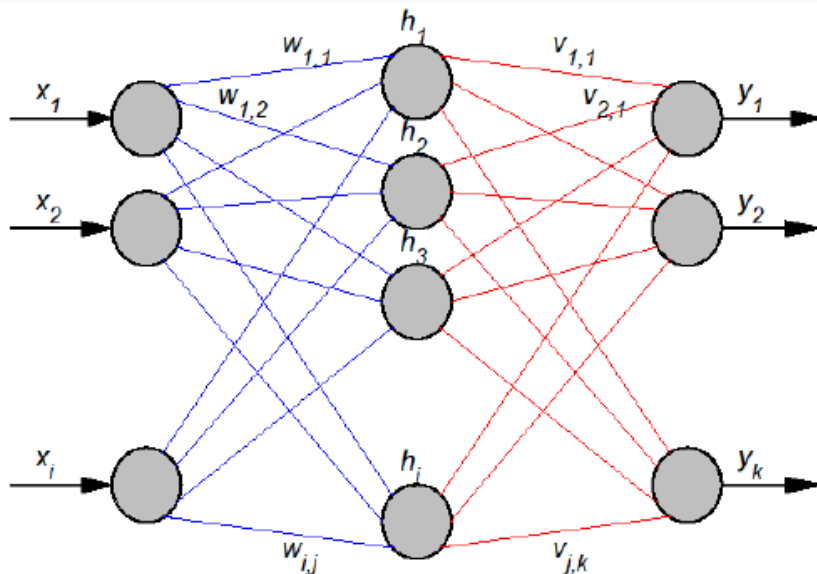
Příklad aktivační funkce neuronu  
(sigmoida)



# MLP – multi layer perceptron

## - vícevrstvá perceptronová síť

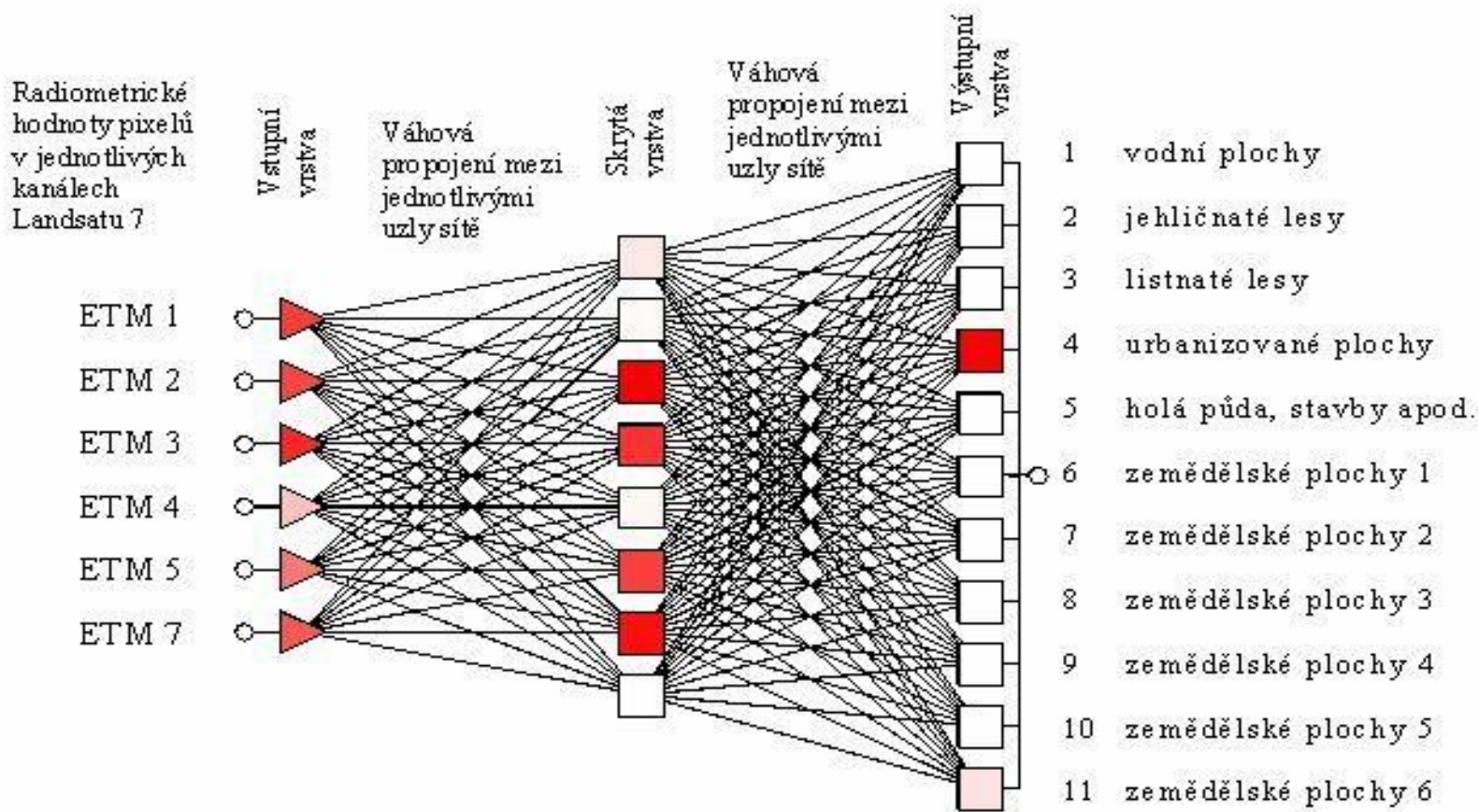
- patří k dopředným umělým neuronovým sítím se skrytou vrstvou neuronů a sigmoidální přenosovou funkcí
- neurony jsou do vrstev rozděleny tak, že výstupy každého neuronu jedné vrstvy jsou napojeny na vstupy všech neuronů vrstvy následující
- neexistují žádné vazby mezi neurony nesousedních vrstev, stejně tak ani mezi neurony nacházející se ve stejné vrstvě
- každý neuron má tak právě tolik vstupů, kolik neuronů se nachází v nižší vrstvě



$x_i$  jsou vstupní neurony (složky vstupního neuronu),  
 $w_{i,j}$  představují váhy skryté vrstvy,  
 $h_j$  symbolizují skryté neurony  
 $v_{j,k}$  popisují váhy výstupní vrstvy  
 $y_k$  značí výstupní neurony



# NN pro klasifikaci multispektrálního snímku



Příklad třívrstvé perceptronové sítě se 6 vstupními, 8 skrytými a 11 výstupními uzly (MLP 6-8-11) s příkladem vstupu a výstupu při klasifikaci družicových snímků s vyznačenými aktivacemi zastavěné plochy

# Učící - adaptivní fáze

- **Učení** – váhy na spojích mezi jednotlivými výkonnými prvky sítě se mění podle určitého tzv. učícího algoritmu.
- Použije se trénovací soubor, ve kterém známe správné zařazení pixelů do jednotlivých klasifikačních tříd.
- **Učící algoritmus** – předpis, podle kterého se předkládají síti vzory k učení a podle kterého se mění váhy jednotlivých spojení mezi neurony.
- **Učení se s učitelem** – analogie řízené klasifikace Algoritmus zpětného šíření (**Back propagation**) – supervised learning, známé požadované výsledky, modifikují se váhy a prahy, autoasociativní (stejně vstupy a výstupy)x heteroasociativní, offline = počítají se synaptické váhy a prahy (bias) pro všechny vzorky naráz (dávkový trénink batch training), on-line = inkrementální, parametry sítě jsou modifikovány pro průchodu každého vzorku (patternu) sítí, konverguje rychleji než offline
- **Učení se bez učitele** – analogie shlukové analýzy (neřízené klasifikace) – **samoorganizující** se sítě (Self-organizing) – **Kohonenova síť**, schopnosti rozeznat na vstupních vzorech stejné nebo blízké vlastnosti a třídit přicházející vektory dle nich, podobné vektory sdružuje do shluků (clusterů)
- **Přeučení (overfitting)** - stav, kdy je systém příliš přizpůsoben množině trénovacích dat, ale nemá schopnost generalizace a selhává na testovací (validační) množině dat. To se může stát např. při malém rozsahu trénovací množiny nebo pokud je systém příliš komplexní (např. příliš mnoho skrytých neuronů v neuronové síti.)
- **Ustálení** – dosažení stabilního stavu sítě

# Algoritmus zpětného šíření (Back propagation)

- signály se nejprve vyšlou směrem dopředu
- u výstupních neuronů se porovnají výstupy s požadovanými
- zjištěné chyby se použijí ke změně nastavení vah v síti

**Výstup ze sítě** má formu vektoru:

Očekávaný výstup (z učící množiny):  $o = (0,0,0,1,0,0)$

Aktuální výstup:  $a = (0,1,0,0,0,0)$

**Hodnocení úspěšnosti učení** – chyba  $e$ : 
$$e = \frac{1}{2} \sum_{j=1}^k (o_j - a_j)$$

- Učení probíhá iteračním způsobem – cílem je dosáhnout nulové či minimální akceptovatelné chyby.
- Adaptace sítě – úprava vah synapsí - probíhá po krocích.
- Délka kroku se nazývá **learning rate**.
- Velká délka kroku značí rychlejší, ale méně přesné učení

## **Samotný cyklus algoritmu se skládá ze tří částí:**

- dopředného šíření vstupního signálu tréninkového vzoru
  - inicializace váhových hodnot malými náhodnými čísly
  - přiřazení hodnoty
  - aktivace vstupních neuronů
  - výpočet vstupní hodnoty skrytých neuronů
  - stanovení výstupních hodnot skrytých neuronů
  - určení skutečných výstupních hodnot
- zpětného šíření chyby
  - přiřazení hodnoty požadovaného výstupu ke každému neuronu ve výstupní vrstvě
  - výpočet chybové funkce
- aktualizace váhových hodnot na spojeních
  - aktualizace váhových hodnot na spojení každého neuronu ve výstupní vrstvě
  - aktualizace váhových hodnot na spojení každého neuronu ve skryté vrstvě
  - pokud již nenastávají žádné změny váhových hodnot nebo byl vykonán maximální počet váhových změn → stop; jinak pokračovat.

## **Přednosti neuronových sítí**

- nezávislost na statistickém rozdělení
- schopnost generalizace
- síť je tolerantní k šumu v učicích datech
- možnost kombinace různých vstupních dat

## **Nedostatky neuronových sítí**

- problém návrhu architektury sítě (počet skrytých vrstev a počet neuronů v nich)
- dlouhá doba učení
- problém lokálního minima (oscilace)
- nastavení úvodních (náhodných) vah synapsí

# NN v programu GEOMATICA

Návrh architektury sítě

- počet vstupních neuronů
- počet skrytých vrstev
- počet skrytých neuronů
- počet výstupních neuronů (tříd)

Vytvoření požadované sítě

Nastavení parametrů trénování:

- rychlost učení
- moment
- maximální počet iterací

Adaptace (trénování) sítě

Zpráva o průběhu a stavu trénování

Klasifikace natrénovanou sítí

Vytvoření výstupů

- mapy klasifikace
- pravděpodobnostní mapy

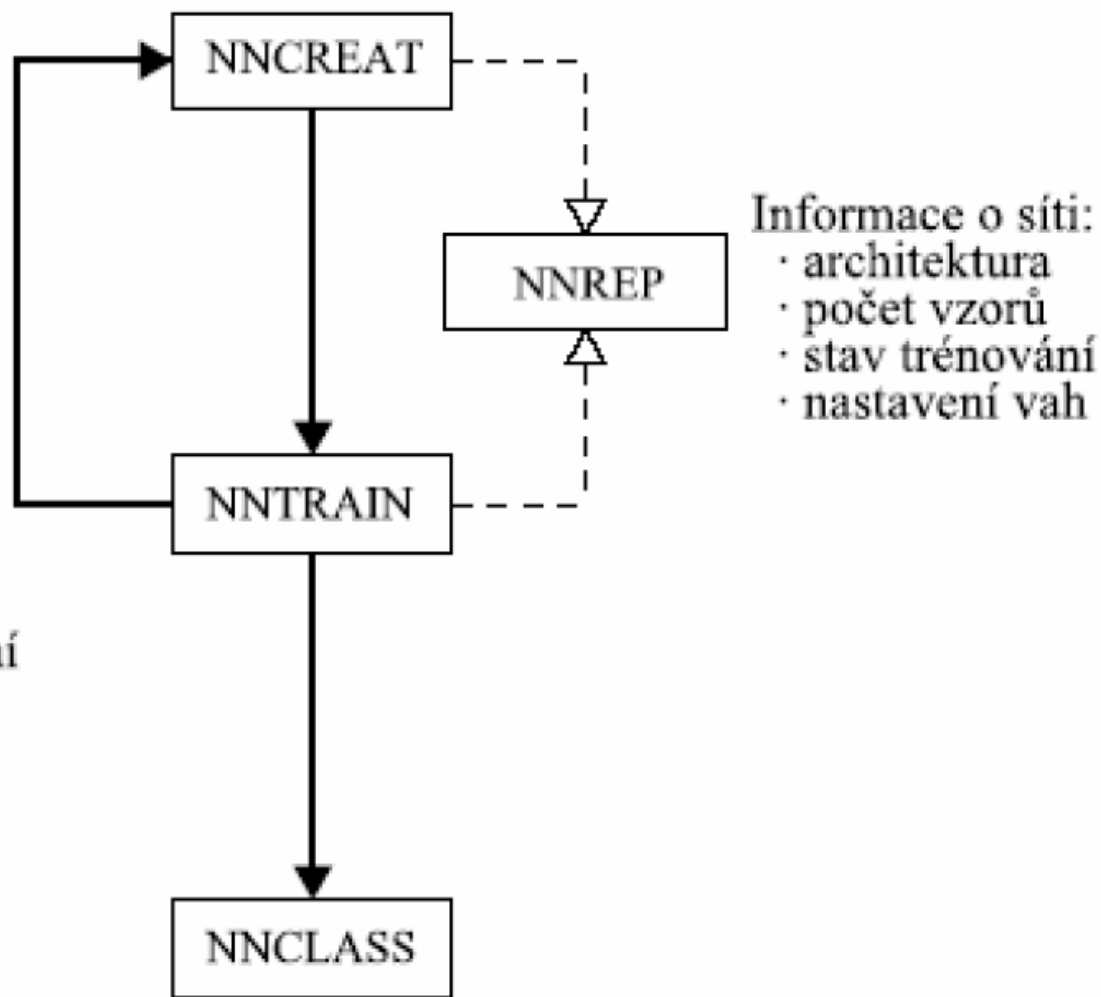


Schéma práce s neuronovými sítěmi v programu GEOMATICA.

# Klasifikace rozhodovacími stromy (Decision Trees)

## Základní pojmy:

- Pravidlo rozděl a panuj (divide and conquer)
- Název atributu - uzel
- Aritmetický či logický výraz - větev stromu
- Název třídy - list stromu
- Dělení do tříd může být spojitě i výčtové (výčet = počet větví)

**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)**

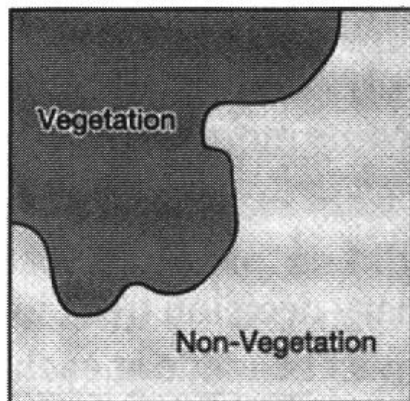
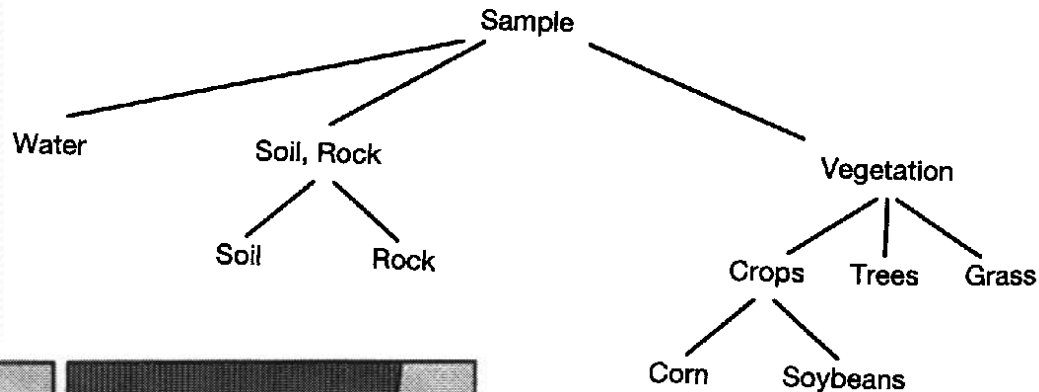
| **TM6 > 23 : poleb (26.0)**

- Mohou ale i nemusí být založeny na binárním třídění, jsou neparametrické, symbolické klasifikátory (nemusí být numerické)
- Umožňují testování sousedů - grafy sousednosti a topologických vazeb (meet, contain, overlap, ...)
- Umožňují testování atributů různé povahy
- Možnost klasifikace po vrstvách – hierarchické třídění („layered classification“), prořezávání = omezení větvení (košatosti stromu)
- Problém objektivního hodnocení („ground truth“)
- Rozhodovací strom se převádí na soubor pravidel = srozumitelnost, kódovatelnost

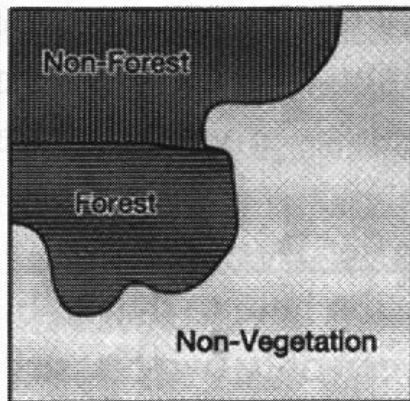
# Klasifikace rozhodovacími stromy

## Hierarchická (layered) klasifikace

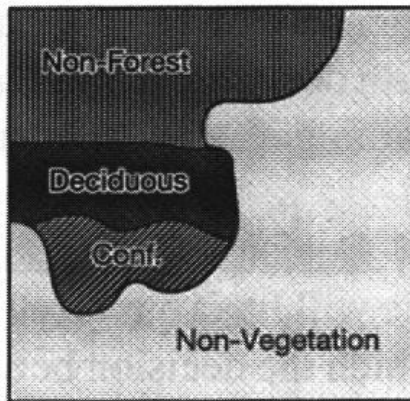
LAYERED CLASSIFIER



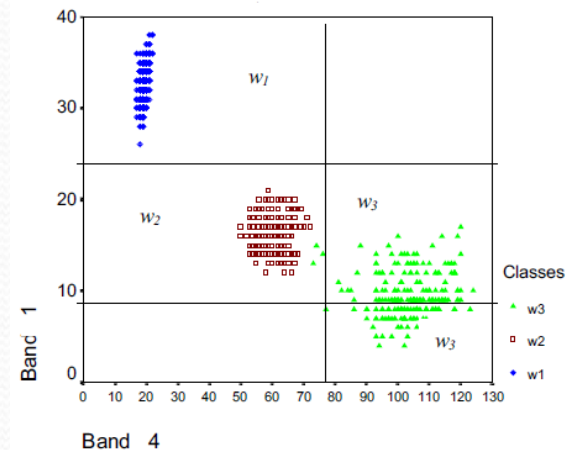
First Layer Partitioning



Second Layer Partitioning



Third Layer Partitioning





## Některé typy rozhodovacích stromů v digitálním zpracování obrazu

- **Algoritmus C4.5** – pevně daná množina atributů, (diskrétní nebo spojitá), spíše vhodný pro menší počet vstupních atributů, učení s učitelem, předdefinované kategorie - trénovací případy zařazené do jedné z kategorií, nepřekrývající se kategorie, dostatek trénovacích dat, konstrukce stromu na principu „rozděl a panuj“, v každém uzlu vybrán atribut s nejvyšším informačním ziskem
- **Random Forest** – klasifikátor tvořený více rozhodovacími stromy vybranými náhodně, a naučené na náhodně vybraných příkladech, výsledná třída pro každý vstup je volena jako modus výsledků všech stromů, nepoužívá se prořezávání, náhodnost zamezuje přeučení.

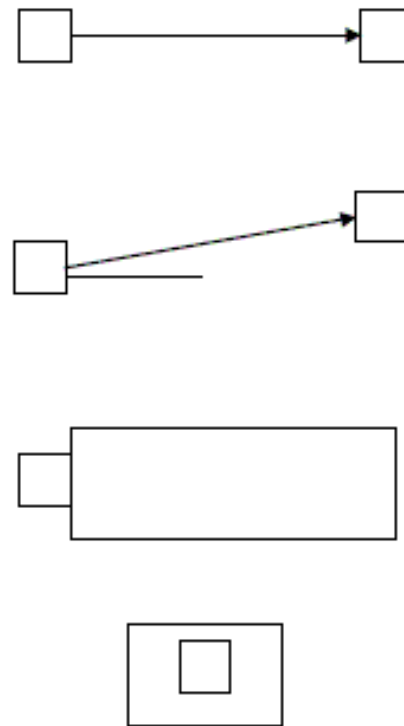
# Kontextuální klasifikátory

## (blíže viz. objektová klasifikace)

- Zahrnují postupy hodnotící polohu či prostorové uspořádání (strukturu) objektů v obraze.
- K rozpoznávání a klasifikaci objektů používají **prostorových příznaků** – kontextu.
- Kontext (pattern) je popisován v obraze jako celku.
- Mohou pracovat i s jednotlivými obrazovými daty, většinou jsou však aplikovány na obraz rozdělený na objekty (např. segmentací) či na obrazová data již klasifikovaná (např. shlukovou analýzou).
- Prostorové příznaky jsou hodnoceny **topologickými** vztahy

## Základní topologické vazby

- **vzdálenost** (určitý objekt se může nacházet pouze v určité vzdálenosti od jiného objektu)
- **směr** (orientace - určitý objekt se může nacházet pouze v určitém směru od jiného objektu)
- **konektivita** (spojení - dva objekty mohou anebo nemusí být spojeny jiným objektem)
- **dotyk** (sousedství - dva objekty se mohou a nemusí dotýkat)
- **vnoření** (obsaženost - jeden objekt se musí nebo naopak nesmí vyskytovat uvnitř jiného).



# Základní topologické vazby

- **Mapová algebra** jako nástroj k sestavování logických pravidel a topologických vztahů
- Hlavní problém – kvantifikace vztahů, regionálně omezená platnost (příklad. lužního lesa)
- Výhoda – možnosti uvažování i „ne rastrových“ dat
- Lze je považovat také za metodu úpravy výsledků klasifikace

## **Tvrdé a měkké klasifikátory, princip neurčitosti (fuzzy logic)**

- Tvrdé klasifikátory jednoznačně rozhodují o přiřazení pixelu do některé z tříd (řízená klasifikace per pixel...)
- Měkké klasifikátory umožňují hodnotit smíšené pixely. Pixel obsahuje na 51 % listnatý les a na 49 % jehličnatý les
- Umožňují hodnotit sílu důkazu pro zařazení pixelu do dané třídy (maximální pravděpodobnost může být malá s ohledem na absolutní hodnoty). Pixel je z 26 % jehličnatý les, z 19 % listnatý, ale z 55 % nějaký jiný, neznámý (nenatrénovaný) povrch.
- Umožňují použití GIS databáze pro vyhodnocování výsledků. Př.: Zastavěné plochy vs. vektorová databáze komunikací (do každého sídla vede cesta).

## Měkké klasifikace – „neostrá vyhodnocení“

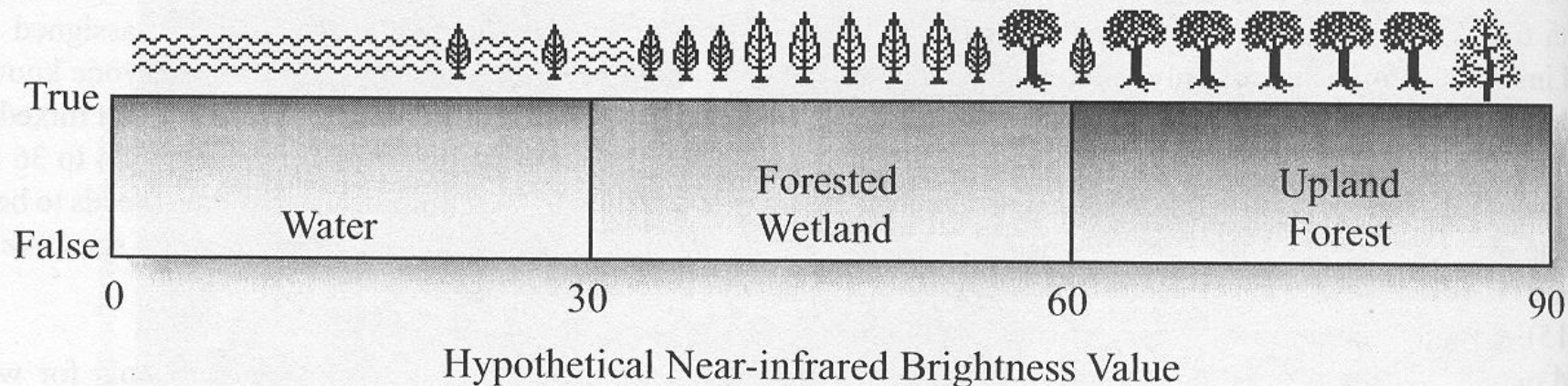
„**fuzzy**“ - neurčitý, nepevný, nedeterminovaný

**Fuzzy logic** - skupina algoritmů a rozhodovacích pravidel, které nejsou pevné, v každém kroku jednoznačně definované, ale umožňují pracovat s jistou mírou nejistoty.

- Nepřiřazují jednoznačně pixel jedné třídě.
- Pro každou třídu zjišťují pravděpodobnost s jakou pixel této třídě náleží.
- Výstupem z těchto klasifikátorů není jedna tématická mapa, ale skupina rastrů, ve kterých hodnoty představují pravděpodobnost s jakou pixel spadá do dané třídy.

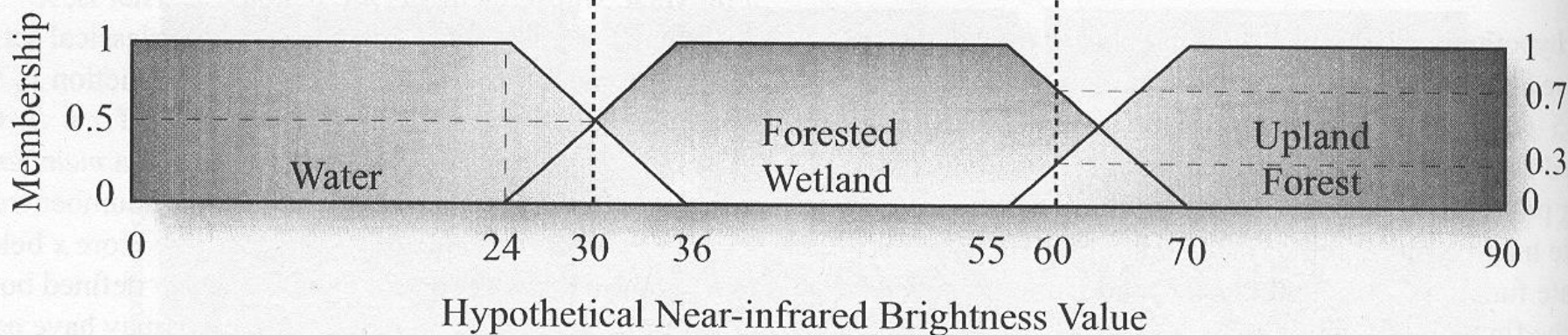
# Měkké klasifikace – „neostrá vyhodnocení“

## Conventional Classification



a.

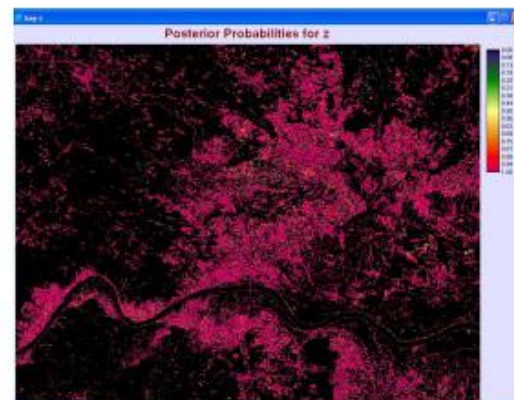
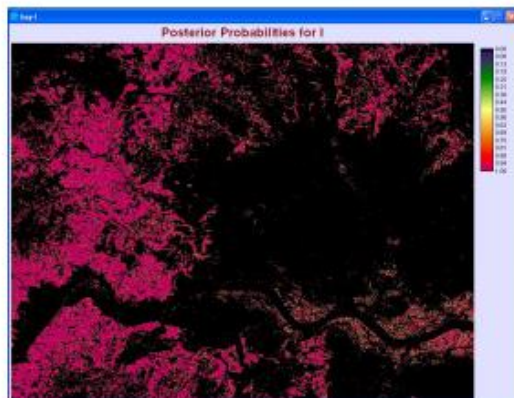
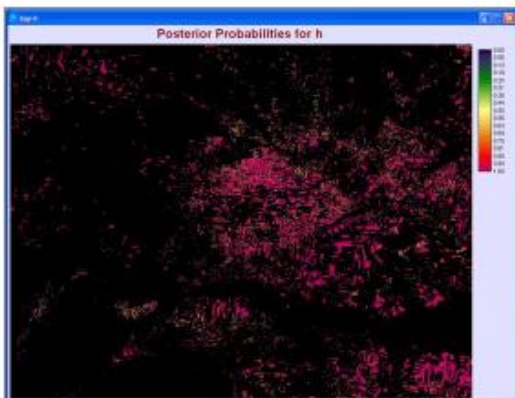
## Fuzzy Classification



# Výsledek tvrdé klasifikace



## Výsledek měkké klasifikace

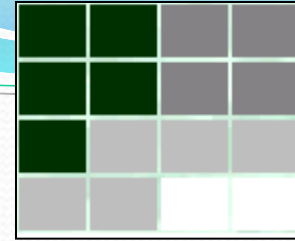




# Texturální klasifikátory

- Určitou třídu povrchu nemusí charakterizovat homogenní množina pixelů ale naopak množina pixelů s **typickou** heterogenitou hodnot
- Textura jako interpretační znak – plošná tónová proměnlivost skládající se ze shodných opakujících se elementů
- Statistické míry textury vycházejí z **GLCM (Grey Level Co-occurrence Matrix)**
- Je to čtvercová matice, která vyjadřuje, jak často se určité kombinace DN hodnot pixelů v obraze vyskytují
- Z GLCM lze vypočítat popisné charakteristiky textury. Těch potom můžeme využít např. jako vstupu do klasifikace v případech, kde selhává použití spektrálních příznaků.

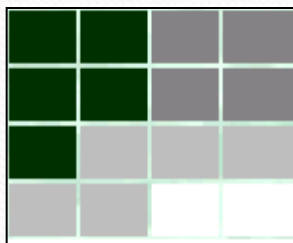
# Výpočet GLCM



- Textura je počítána ze **vztahu** několika pixelů – pro jednoduchost dvou. Definujeme tzv. Referenční pixel (REFERENCE) a sousední pixel (NEIGHBOUR).
- N pixel je k R definován v určitém směru. Např. N je vždy o jeden pixel vpravo ( na východ) do R. Tedy pracujeme s texturou určenou v předem dohodnutém směru.
- Těchto směrů může být 8. Prostorový vztah R a N pixelu lze vyjádřit zápisem  $(1,0)$  – tedy jeden pixel v ose x a žádný v ose y. Vztah R a N ve směru západním by byl zapsán  $(-1,0)$ . Každý sousední pixel se stává referenčním atd.
- Matice GLCM je počítána pro okno předem definované velikosti – stejně jako v případě filtrace obrazu.
- Vztah mezi R a N je také definován vzdáleností mezi R a N (tzv. offset). V našem případě je 1, ale může být i více – pak je nutné zvětšit okno.

# Výpočet GLCM

GLCM je čtvercová matice. Počet řádků (sloupců) je dán počtem hodnot, kterého mohou pixely analyzovaného obrazu nabývat. V našem případě bude rozměr matice 4, pixely v testovacím obrázku nabývají hodnot 0,1,2,3.



0	0	1	1
0	0	1	1
0	2	2	2
2	2	3	3

	0	1	2	3
0	2	2	1	0
1	0	2	0	0
2	0	0	3	1
3	0	0	0	1

Hodnoty R pixelů jsou čteny v řádcích, hodnoty N pixelů ve sloupcích. Každý prvek GLCM matice nese informaci, kolikrát se daná kombinace hodnot v okně vyskytuje.

## Výpočet GLCM

GLCM musí být **symetrická**. Pro definování textury je jedno, zda jde o směr  $(1,0)$  či  $(-1,0)$

Relaci  $(-1,0)$  dostaneme záměnou R a N pixelů (zaměníme řádky a sloupce – transponujeme matici) a obě GLCM matice sečteme.

GLCM				GLCM transponovaná				GLCM symetrická			
2	2	1	0	2	0	0	0	4	2	1	0
0	2	0	0	2	2	0	0	2	4	0	0
0	0	3	1	1	0	3	0	1	0	6	1
0	0	0	1	0	0	1	1	0	0	1	2

# Výpočet GLCM

Prvky GLCM se vyjadřují v hodnotách pravděpodobnosti výskytu

Je pravděpodobnost najít v originálním snímku kombinaci 2,2 větší jak kombinaci 2,3?

Kombinace 2,2 se vyskytuje 6 krát z 24 možností – tedy pravděpodobnost výskytu je  $6/24$  tj.  $1/4$  tj. 0,25. Pravděpodobnost členu 2,3 je 0,042.

Výpočtem pravděpodobnosti je matice **normalizována**

<b>0,166</b>	<b>0,083</b>	<b>0,042</b>	<b>0</b>
<b>0,083</b>	<b>0,166</b>	<b>0</b>	<b>0</b>
<b>0,042</b>	<b>0</b>	<b>0,250</b>	<b>0,042</b>
<b>0</b>	<b>0</b>	<b>0,042</b>	<b>0,083</b>

## Vlastnosti GLCM

- Na diagonále GLCM jsou prvky, které reprezentují dvojice pixelů, jejichž DN hodnoty jsou stejné (0-0, 1-1, 2-2,...).
- Mají-li buňky matice na diagonále vysoké hodnoty, znamená to, že velké procento pixelů má stejnou hodnotu jako jejich sousedé – to svědčí o malém kontrastu analyzovaného snímku.
- Paralelně s diagonálou se v GLCM nacházejí buňky, které nesou informaci o pravděpodobnosti výskytu kombinací pixelů, které se liší v horizontálním směru o hodnotu 1 (0-1, 1-2, 2-3), dále potom kombinace pixelů, které se liší o hodnotu 2 (0-2, 1-3,...).
- Tedy čím dále od diagonály GLCM, tím je větší rozdíl v hodnotách R a N pixelu.

## Výpočet měr textury

- Většina měr textury je **váženým průměrem** buněk GLCM.
- Princip výpočtu míry textury je stejný jako v případě filtrace obrazu. Okno předem definované velikosti – čtvercové s lichým počtem řádků a sloupců se pohybuje po originálním obraze pixel po pixelu.
- Pro každou pozici okna je vypočtena GLCM a určitá míra textury jako vážený průměr. Tato hodnota je přiřazena střednímu pixelu v rámci každé pozice okna.
- Výsledkem je tedy **TEXTURE IMAGE**, ve kterém každý pixel nese hodnotu určité míry textury
- Míry textury se třídí podle účelu použitých vah.

## Míry textury založené na kontrastu

Používá se váh, které se vztahují ke vzdálenosti buněk od diagonály GLCM. Prvky na hlavní diagonále vypovídají o kombinacích stejných pixelů – tedy s nulovým kontrastem. S rostoucí vzdáleností od diagonály roste kontrast – tedy zvětšují se váhy.

$$\textit{Kontrast} = \sum_{i,j=0}^{N-1} P_{i,j} \cdot (i - j)^2$$

**Vysvětlení:** Je-li  $i$  a  $j$  stejné (na hlavní diagonále) váha je 0. Liší-li se  $i$  a  $j$  o 1 váha je 1, liší-li se o 2 váha je 4 atd. Váhy **exponenciálně** rostou.



# Míry textury založené na pravidelnosti

Vyjadřují, jak pravidelně jsou pixely v rámci okna uspořádány. Na obrázku je část snímku s pravidelným uspořádáním pixelů a nepravidelným - v obou případech se každý pixel liší o 1 oproti svému pravému sousedu – tedy kontrast snímků určený z GLCM je stejný.

V případě měr pravidelnosti jsou váhy sestavovány na základě **četnosti výskytu** jednotlivých kombinací pixelů.

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>
<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>

V případě měr pravidelnosti tvoří váhy přímo hodnoty prvků GLCM.

Váhy, které budou zvyšovat svoji hodnotu s čtenějším výskytem dané kombinace pixelů budou produkovat charakteristiku textury, která bude mít vyšší hodnoty se zvyšující se pravidelností uspořádání pixelů ve snímku.

Tyto míry textury dávají vysoké hodnoty pro pravidelně uspořádané části obrazu

ASM (Angular Second Moment) a Energy

$$ASM = \sum_{i,j=0}^{N-1} P_{i,j}^2 \quad Energy = \sqrt{ASM}$$

# Míry textury založené na popisných statistikách GLCM

Míry se počítají z hodnot GLCM

GLCM **průměr** vypovídá o úrovni souboru hodnot  $P_{i,j}$  v GLCM, **rozptyl** je mírou disperze hodnoty kolem průměru – mírou proměnlivosti. **Korelace** je potom mírou lineární závislosti sousedních pixelů.

GLCM mean (průměr)

$$\mu_i = \sum_{i,j=0}^{N-1} i \cdot (P_{i,j})$$

GLCM rozptyl

$$\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} P_{i,j} \cdot (i - \text{Průměr}_i)^2$$

GLCM korelace

$$\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \frac{P_{i,j} \cdot (i - \text{Průměr}_i) \cdot (j - \text{Průměr}_j)}{\sqrt{\text{Rozptyl}_i \cdot \text{Rozptyl}_j}}$$

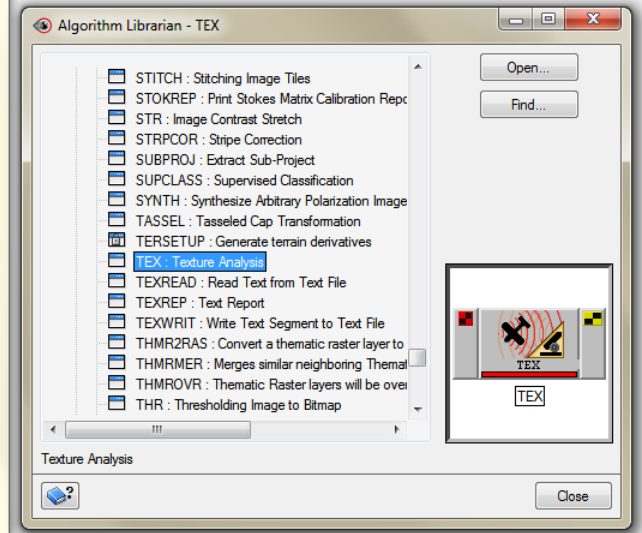
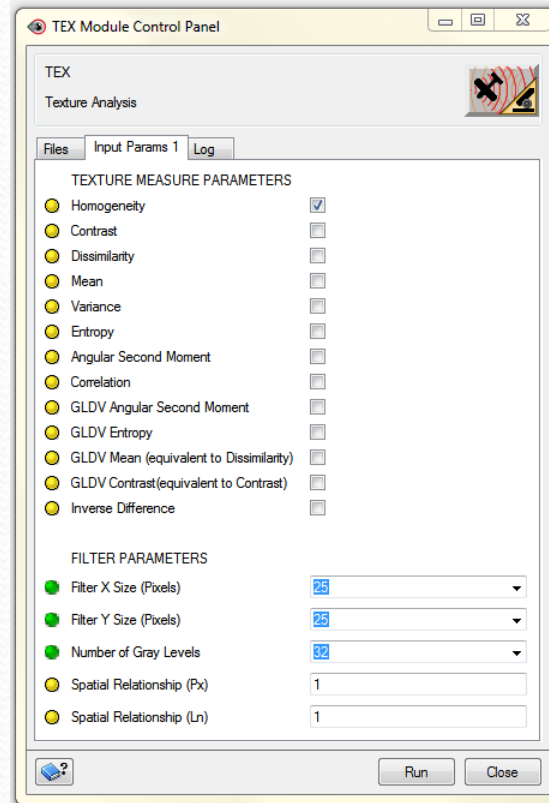
## Použití měr textury, výhody a nevýhody

- GLCM a míry textury jsou počítány pro určitou část snímku – většinou čtvercové okno o lichém počtu řádků (sloupců). Okno se volí větší než při klasické filtraci, aby postihlo typickou texturu heterogenní plochy.
- Předpokládáme-li, že různé kategorie povrchů budou mít různé hodnoty měr textury, potom by velikost okna měla být dostatečná, aby zachytila charakteristickou tónovou proměnlivost objektu, ale na druhé straně také dostatečně malá – menší jak samotný objekt.
- Např. texturu lesního komplexu nelze zachytit v okně, které je menší jak jeden strom, na druhé straně okno větší jak lesní komplex nelze také použít.
- Problémem je vysoká korelace mezi jednotlivými charakteristikami textury.

# Použití měr textury, výhody a nevýhody

- Míry textury mohou být počítány z jednopásmových dat – nevyžadují multispektrální data.
- Dvě či tři míry textury mohou být zobrazeny ve formě RGB barevné syntézy.
- Měr textury existuje celá řada a často jsou si značně podobné. (entropie, korelace)
- Pro extrahování podstatné informace z měr textury a odstranění vysoké korelace je možné využít například analýzu hlavních komponent.

Míry textury –  
knihovna  
algoritmů  
Geomática



# Vlastní klasifikace texturálních příznaků

Klasifikace založená na mírách textury je analogií filtrace a následného shlukování.

Je určena prahová hodnota proměnlivosti. Je-li rozptyl hodnot v okně menší než prahová hodnota, je okno považováno za "homogenní", je-li naopak větší, je okno považováno za "heterogenní".

Střed prvního homogenního okna je středem prvního shluku, střed druhého homogenního okna je středem druhého shluku atd.

Jakmile je dosaženo maximálního počtu shluků, definovaného zpracovatelem, klasifikátor hodnotí vzdálenosti mezi nalezenými středy shluků.

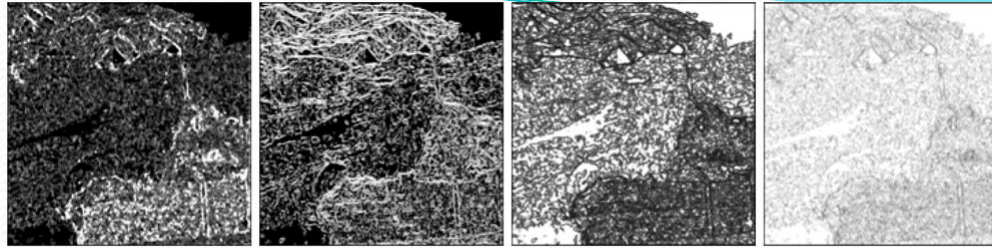
Dva nejbližší spojí do jednoho shluku a kombinuje jejich statistiku. Tak klasifikátor pokračuje s dalšími dvěma nejbližšími shluky, až zhodnotí celou scénu.

Výsledné středy shluků jsou statisticky popsány z hlediska své separability na základě zpracovatelem definovaných statistických vzdáleností (např. rozptyl, směrodatná odchylka, atd.).

Shluky, mezi nimiž je tato vzdálenost menší než prahová hodnota, jsou opět spojovány.

Výsledné shluky jsou použity ke klasifikaci celé scény klasifikátorem minimální vzdálenosti nebo maximální pravděpodobnosti.

# Příklad TEXTURE IMAGE



(a) Contrast    (b) Correlation    (c) Energy    (d) Homogeneity

