



FAKULTA  
APLIKOVANÝCH VĚD  
ZÁPADOČESKÉ  
UNIVERZITY  
V PLZNI

KATEDRA  
KYBERNETIKY



# Metody Počítačového Vidění

## 9. přednáška: Strojové učení

Ing. Lukáš Bureš  
Ing. Petr Zimmermann  
Katedra kybernetiky

Plzeň  
6. listopadu 2014

# Obsah

<b>1</b>	<b>Metody Počítačového Vidění 9. přednáška</b>	<b>1</b>
1.1	Bayesovská klasifikace . . . . .	1
1.2	Rozhodovací strom . . . . .	3
1.3	Rozhodovací les . . . . .	7
1.4	Náhodný les . . . . .	7
1.5	Neuronové sítě . . . . .	8
1.6	Konvoluční neuronové sítě . . . . .	12
1.7	Zdroje . . . . .	14

# Kapitola 1

## Metody Počítačového Vidění

### 9. přednáška

#### 1.1 Bayesovská klasifikace

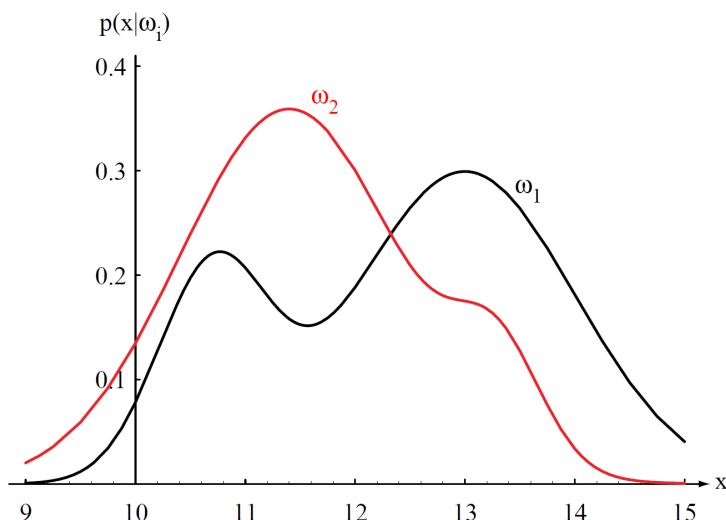
Bayesovská rozhodovací teorie je základní statistický přístup ke klasifikaci. Tento přístup je založen na vyčíslení kompromisu mezi různými klasifikačními rozhodnutími za využití pravděpodobnosti a nákladů, které doprovázejí rozhodnutí. Za předpokladu, že problém rozhodování je představen pomocí pravděpodobnostních vztahů, kde jsou všechny hodnoty pravděpodobností známe.

Označme  $\omega$  (*přírozený stav*) jednotlivé třídy, do kterých budeme klasifikovat. Při klasifikaci do dvou tříd mějme  $\omega_1$  a  $\omega_2$ , tedy  $\omega = \omega_1$  a  $\omega = \omega_2$ . Protože *přírozený stav* je nepredikovatelný, považujeme  $\omega$  za proměnou, která musí být popsána pravděpodobnostně.

Předpokládejme, že tu je nějaká *apriorní* pravděpodobnost  $P(\omega_1)$ , taková, že vzorek patří do třídy  $\omega_1$  a  $P(\omega_2)$  do třídy  $\omega_2$ . Dále předpokládejme, že neexistují žádné další třídy, proto  $P(\omega_1) + P(\omega_2) = 1$ . Tato *apriorní* znalost reflektuje jaké je pravděpodobnost, že vzorek bude patřit do třídy  $\omega_1$  nebo  $\omega_2$ , ještě dříve než se vzorek zobrazí.

Předpokládejme pro tuto chvíli, že chceme rozhodnout do jaké třídy vzorek patří bez jeho pozorování. Jediná informace, kterou je možné využít je *apriorní* pravděpodobnost. Pokud musí být učiněno rozhodnutí s takto malou informací, zdá se logické použít následující rozhodovací pravidlo: vzorek patří do třídy  $\omega_1$  pokud  $P(\omega_1) > P(\omega_2)$ , jinak patří do třídy  $\omega_2$ .

Ve většině případů nejsme nuceni učinit rozhodnutí pouze na základě *apriorní* pravděpodobnosti. Uvažujme  $x$ , jako spojitou náhodou veličinu, jejíž distribuce závisí na *přírozeném stavu* a je vyjádřena jako  $p(x|\omega_1)$ , což je podmíněná hustota pravděpodobnosti třídy  $\omega_1$ . Přesněji lze podmíněnou hustotu pravděpodobnosti  $p(x|\omega_1)$  zapsat jako  $p_X(x|\omega_1)$ , což naznačuje, že se jedná zejména o náhodnou proměnnou  $X$ . Tedy, pro upřesnění, v případě  $p_X(\cdot)$  a  $p_Y(\cdot)$  se jedná o dvě různé funkce, které můžeme zapsat jako  $p(x)$  a  $p(y)$ . Rozdíl mezi  $p(x|\omega_1)$  a  $p(x|\omega_2)$  popisuje rozdíl mezi jednotlivými třídami, viz Obr. 1.1.



Obrázek 1.1: Hypotetické funkce hustot podmíněných pravděpodobností ukazují hustotu pravděpodobnosti měření hodnoty příznaku  $x$  za předpokladu, že vzorek náleží třídě  $\omega_i$ . Hustotní funkce jsou normalizované a plocha pod každou křivkou má obsah roven 1.

Předpokládejme, že známe apriorní pravděpodobnosti  $P(\omega_j)$  a podmíněné závislosti  $p(x|\omega_j)$ . Spojitá hustota pravděpodobnosti, že vzorek patří do třídy  $\omega_j$  a má hodnotu příznaku  $x$  a může být zapsáno dvěma způsoby:  $p(\omega_j, x) = P(\omega_j|x)p(x) = p(x|\omega_j)P(\omega_j)$ , přeuspořádáním tohoto vztahu dostaneme známý Bayesův vztah

$$P(\omega_j|x) = \frac{p(x|\omega_j)P(\omega_j)}{p(x)}, \quad (1.1)$$

kde v případě dvou tříd

$$p(x) = \sum_{j=1}^2 p(x|\omega_j)P(\omega_j) \quad (1.2)$$

Bayesův vztah může být vyjádřen slovně jako

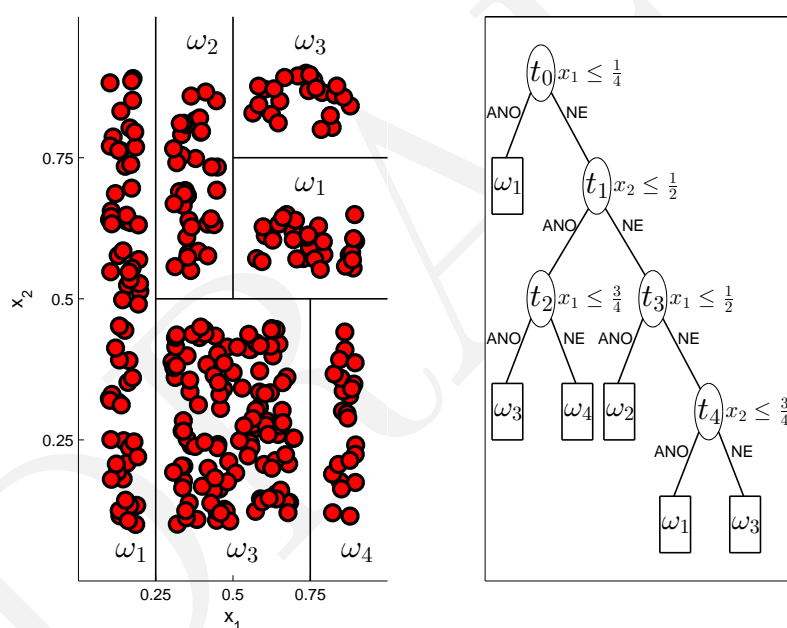
$$\text{posterior} = \frac{\text{likelihood} \times \text{prior}}{\text{evidence}}. \quad (1.3)$$

Bayesův vztah ukazuje, že pozorováním hodnoty  $x$  můžeme konvertovat apriorní pravděpodobnost  $P(\omega_j)$  na aposteriorní pravděpodobnost  $P(\omega_j|x)$  - pravděpodobnost přirozeného stavu  $\omega_j$  je dána pozorovanou hodnotou  $x$ .

## 1.2 Rozhodovací strom

Rozhodovací strom je příkladem nelineární klasifikační metody. Představuje vícepatrový rozhodovací systém, kterým příznakové vektory postupně procházejí až do bodu, kdy jsou zařazeny do odpovídající třídy.

Sestavení rozhodovacího stromu se provádí postupným dělením prostoru trénovací množiny  $X$  na navzájem se neprolínající regiony, přičemž každý region odpovídá právě jedné klasifikační třídě. Rozhodovací strom  $T$  se skládá z uzlů, uvnitř kterých dochází k porovnávání jednotlivých prvků příznakového vektoru  $\mathbf{x} \in X$ . Na základě těchto porovnání vektor postupně propadne až do jednoho z listů stromu, které odpovídají jednotlivým klasifikačním třídám. Obecně nejpoužívanější forma rozhodovacího stromu je ta, která rozděluje prostor vstupní množiny na kvádry, jejichž hrany jsou rovnoběžné se souřadnicovými osami tohoto prostoru. Takový strom se nazývá stromem binárním a prvky příznakového vektoru jsou porovnávány na základě splnění nerovnosti  $x_i \leq \alpha$ , kde  $x_i$  je  $i$ -tý prvek příznakového vektoru  $\mathbf{x}$  a  $\alpha$  značí zvolený práh. Nerovnost  $x_i \leq \alpha$  je označována jako rozhodovací pravidlo.



Obrázek 1.2: Ukázka rozdělení prostoru trénovací množiny (vpravo) pomocí rozhodovacího stromu (vlevo).

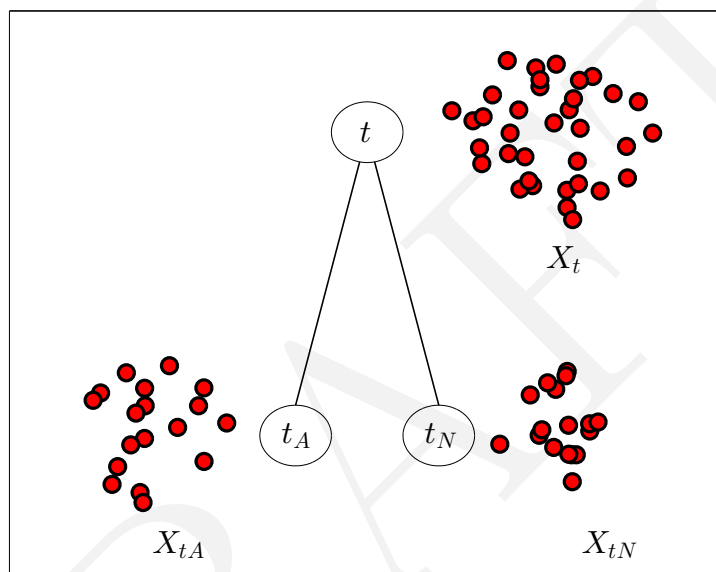
Příklad zobrazený na Obr. 1.2 ukazuje princip konstrukce rozhodovacího stromu. Je zřejmé, že výsledný tvar stromu je ovlivněn volbou rozhodovacího pravidla a pořadím, ve kterém jsou prvky příznakových vektorů porovnávány. Konstrukce každého stromu musí splňovat následující podmínky:

- Každý uzel  $t$  odpovídá podmnožině  $X_t \subset X$ , která vznikla na základě rozhodovacího pravidla předešlého uzlu (výjimku tvoří první uzel, tzv. kořen stromu, pro který  $X_t = X$ ). Rozhodovacím pravidlem uzlu  $t$  je podmnožina  $X_t$  dále rozdělena na  $X_{tA}$  a  $X_{tN}$ .  $X_{tA}$  v sobě slučuje vektory, které splnily rozhodovací pravidlo uzlu  $t$  a  $X_{tN}$  naopak slučuje ty, které pravidlo nesplnily (viz Obr. 1.3). Každé takové rozdělení

musí splňovat:

$$\begin{aligned} X_{tA} \cap X_{tN} &= \emptyset, \\ X_{tA} \cup X_{tN} &= X_t. \end{aligned}$$

- Ze všech možných rozdělení  $X_t$  je zvoleno to, které nejlépe splňuje rozdělovací kritérium.
- Je nutné zvolit zastavující podmínku, která reguluje narůstání stromu. Uzel, který tuto podmínku splňuje je označena za list stromu.
- Každému listu musí být přiřazena odpovídající klasifikační třída.



Obrázek 1.3: Příklad dělení uzlu  $t$  a jemu odpovídající množina vektorů  $X_t$

Pojmy rozdělovací kritérium, zastavující podmínka a princip přiřazení klasifikačních tříd listům stromu jsou vysvětleny v následujících podkapitolách.

## Volba rozhodovacího pravidla

Jak bylo výše uvedeno, v případě binárního stromu má rozhodovací pravidlo tvar  $x_i \leq \alpha_i$  ( $\alpha_i$  je práh rozdělovací vektory na základě porovnání jejich  $i$ -tého prvku). Úkolem je tedy zvolit jak hodnotu prahu  $\alpha_i$ , tak i index prvku  $i$ . Prah však může nabývat nekonečně mnoha hodnot, jelikož  $\alpha_i \in \mathbb{R}$  a otestovat všechny z nich je tak nemyslitelné. Díky existenci trénovací množiny  $X$  lze ovšem vymezit konečnou množinu hodnot, které je pro získání  $\alpha_i$  nutné otestovat. Pokud je porovnáván  $i$ -tý prvek,  $x_i$ , vektoru  $\mathbf{x}$ , jsou veškeré  $i$ -té prvky všech příznakových vektorů uvnitř trénovací množiny vzestupně seřazeny a tato řada pak tvoří množinu variant, které je nutné následně podrobit testování. Obdobně se postupuje pro všechny prvky  $x_i, i = 1, 2, \dots, l$  všech vektorů  $\mathbf{x} \in X$ . Postupně tak vznikne několik možností rozdělení podmnožiny  $X_t$  a rozdělovacím kritériem je vybráno to nejlepší z nich. Tímto způsobem je uzlu  $t$  určeno, který prvek vektoru  $\mathbf{x}$  se bude uvnitř něj porovnávat s nalezeným prahem (dojde ke konkretizaci rozhodovacího pravidla  $x_i \leq \alpha_i$ ).

## Rozdělovací kritérium

Z důvodu, aby bylo možné kvalitativně porovnávat vzniklá rozdělení  $X_t$  z předchozího kroku, je definována tzv. míra rozmanitosti uzlu.

Nechť  $P(\omega_i | t)$  značí pravděpodobnost jevu, že vektory uvnitř podmnožiny  $X_t$ , náležící uzlu  $t$ , patří do třídy  $\omega_i$ ,  $i = 1, 2, \dots, M$ . Míra rozmanitosti uzlu je definována jako:

$$I(t) = - \sum_{i=1}^M P(\omega_i | t) \log_2 P(\omega_i | t). \quad (1.4)$$

Je vhodné poznamenat, že předpis 1.4 představuje entropii z teorie informace spjatou s uzlem  $t$ . Ta nabývá svého maxima pokud se všechny pravděpodobnosti  $P(\omega_i | t) = \frac{1}{M}$ , naopak je rovna nule, pokud všechny vektory uvnitř množiny spadají do jedné třídy, což znamená, že právě jedna pravděpodobnost  $P(\omega_i | t) = 1$  a ostatní jsou nulové. V praxi jsou pravděpodobnosti  $P(\omega_i | t)$  odhadovány pomocí vztahu  $\frac{N_t^i}{N_t}$ , kde  $N_t^i$  je počet vektorů v  $X_t$  patřících do třídy  $\omega_i$  a  $N_t$  je celkový počet vektorů uvnitř podmnožiny  $X_t$ .

Dalším rozdělením  $X_t$  na dvě podmnožiny  $X_{tA}$  a  $X_{tN}$ , kde  $X_{tA}$  má  $N_{tA}$  počet vektorů a  $X_{tN}$  má  $N_{tN}$  počet vektorů, dojde k nárůstu rozmanitosti uzlu definovaného jako

$$\Delta I(t) = I(t) - \frac{N_{tA}}{N_t} I(t_A) - \frac{N_{tN}}{N_t} I(t_N), \quad (1.5)$$

kde  $I(t_A)$ ,  $I(t_N)$  jsou míry rozmanitosti uzlů  $t_A$  a  $t_N$ . Cílem rozdělovacího kritéria tedy je zvolit takové rozdělení uzlu  $t$  (resp. podmnožiny  $X_t$ ), pro které dojde k nejvyššímu nárůstu rozmanitosti uzlu,  $\Delta I(t)$ .

## Zastavující podmínka

Dojde-li k naplnění zastavující podmínky pro některý z uzlů rozhodovacího stromu, je takový uzel prohlášen za list stromu a jeho dělení je ukončeno. Zastavující podmínka může být definována prahem  $Tr$ , který určuje minimální potřebný nárůst rozmanitosti uzlu,  $\Delta I(t)$ . Pokud je po rozdělení uzlu  $t$  maximální hodnota  $\Delta I(t)$  přes všechna možná rozdělení uzlu menší než práh  $Tr$ , je uzel  $t$  prohlášen za list. Další možností volby zastavující podmínky je stanovení minimálního počtu vektorů uvnitř podmnožin vzniklých dělením uzlu  $t$ .

## Princip přiřazení třídy listům

Jakmile je uzel rozhodovacího stromu prohlášen za list, musí mu být přiřazena jedna z klasifikačních tříd. Listu je přiřazena třída  $\omega_j$ , kde index  $j$  odpovídá

$$j = \arg \min_i P(\omega_i | t). \quad (1.6)$$

Jinými slovy, listu  $t$  je přiřazena taková třída, jenž je uvnitř  $X_t$  nejčastěji zastoupena.

Vedle binárních rozhodovacích stromů lze v odborné literatuře dohledat i stromy, jejichž rozhodovací pravidlo má tvar  $\sum_{i=1}^l c_i x_i \leq \alpha$ . Jedná se tedy o nalezení přímk, nikoli

pouze prahů, dělících jednotlivé prvky příznakových vektorů. Tímto rozhodovacím pravidlem lze sice v některých případe vhodněji rozdělit prostor trénovací množiny  $X$ , ovšem konstrukce stromu je mnohem složitější.

Nevýhodou klasifikační metody založené na rozhodovacím stromu zůstává princip sestavování samotného stromu. Pokud dojde byť k malé změně trénovací množiny  $X$ , dojde zároveň ke změně tvaru rozhodovacího stromu  $T$ , což může vést k nárůstu chyby klasifikace. Tento nedostatek lze odstranit zapojením více než jednoho rozhodovacího stromu do procesu klasifikace. Taková klasifikační metoda je označována za rozhodovací les.



## 1.3 Rozhodovací les

Hlavní předností rozhodovací lesa je to, že účinně potlačuje citlivost výsledného tvaru rozhodovacího stromu  $T$  na změny v trénovací množině  $X$ . Toho je docíleno tak, že množina  $X$  obsahující  $N$  prvků je algoritmem *bootstrap aggregating* rozdělena na  $n$  množin,  $X_{(n)}$ , obsahujících  $N_{(n)}$  prvků, přičemž  $N_{(n)} \leq N$ . A pro každou množinu  $X_{(n)}$  je následně zkonstruován rozhodovací strom  $T_{(n)}$  postupem, který je popsán v kapitole 1.2.

Klasifikace neznámého vektoru  $\mathbf{y}$  probíhá tak, že je tento vektor přiveden na vstupy všech rozhodovacích stromů  $T_{(n)}$ . Každý rozhodovací strom poté určí příslušnost vektoru  $\mathbf{y}$  do některé třídy  $\omega_i$ ,  $i = 1, 2, \dots, M$ . Finální index třídy  $i$  je vybrán na základě nejčastějšího výsledku mezi všemi provedenými klasifikacemi (tj. výstupy stromů  $T_{(n)}$ ) neznámého vektoru  $\mathbf{y}$ .

Dalším faktorem, jenž ovlivňuje výsledný tvar rozhodovacího stromu, je pořadí prvků příznakových vektorů, které je během nalézání rozhodovacích pravidel pro jednotlivé uzly uvažováno. Doposud byly v tomto textu představeny rozhodovací stromy využívající ke své konstrukci prvky vektorů popořadě. Následující kapitola popisuje modifikaci rozhodovacích lesů, jež stanoví toto pořadí náhodně.

## 1.4 Náhodný les

Náhodný les je modifikací metody rozhodovacího lesa, která vznikla za účelem ještě většího zpřesnění výsledku klasifikace. Jsou-li během konstrukce stromů porovnávány prvky příznakových vektorů popořadě, zvyšuje se tím riziko toho, že velká část takto vzniklých stromů bude vzájemně korelovaná. Ovšem vysoká korelace stromů, jak bylo prokázáno, uvnitř rozhodovacího lesa zvyšuje chybu klasifikace. Z tohoto důvodu používá náhodný les zcela nový princip konstrukce jednotlivých rozhodovacích stromů.

Prvním krokem této metody zůstává rozdělení trénovací množiny  $X$  na  $n$  množin pomocí algoritmu *bootstrap aggregating*. Rozdíl nastává v bodě konstrukce jednotlivých stromů. Během té je nejprve uživatelem zvolen parametr  $m$  ( $m \ll l$ , kde  $l$  je dimenze vektoru  $\mathbf{x} \in X$ ), který ovlivňuje volbu rozhodovacího pravidla pro jednotlivé uzly konstruovaných stromů. Jako rozhodovací pravidlo je totiž zvoleno to, které vykáže nejlepší výsledek mezi  $m$  náhodně vybranými prvky příznakového vektoru. Tímto způsobem vznikají všechny stromy tvoří finální náhodný les. Klasifikace neznámého vektoru  $\mathbf{y}$  pak probíhá stejně, jako tomu bylo v případě klasického rozhodovacího lesa.

## 1.5 Neuronové sítě

Dříve popsaná klasifikační metoda SVM usiluje o nalezení takové nadroviny  $g(x)$ , která rozděluje trénovací množinu  $X$  do dvou tříd  $\omega_1$ ,  $\omega_2$ . V případě lineárně separovatelné množiny nalezená nadrovina  $g(x)$  klasifikovala trénovací vektory,  $\mathbf{x} \in X$ , bez chyby. Pokud ovšem trénovací množina lineárně separovatelná nebyla, cílem metody SVM bylo zkonstruovat nadrovinu  $g(x)$  minimalizující chybu klasifikace. Klasifikátory pracující na principu neuronových sítí byly primárně navrženy právě k řešení lineárně neseparovatelných problémů. Řešení je nalezeno pomocí transformace těchto problémů do prostoru, ve kterém již existuje adekvátní lineární řešení.

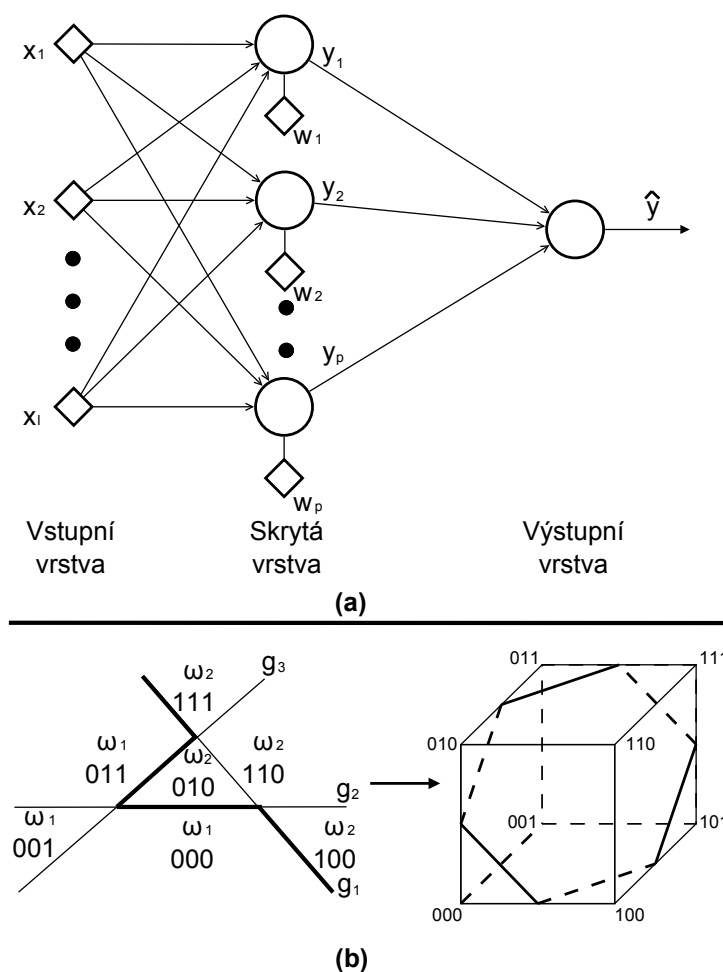
Základním stavebním kamenem neuronové sítě je neuron, na jehož vstup je přiveden vektor  $\mathbf{x}$  a výstupem je pak 1 (je-li  $\mathbf{x} \in \omega_1$ ) či 0 (je-li  $\mathbf{x} \in \omega_2$ ). Neuron s takovýmto binárním výstupem je označován jako perceptron. Neuronová síť, skládající se výhradně z perceptronů, se nazývá perceptronová síť a pokud v ní dochází k řazení perceptronů za sebe, je taková síť označována jako vícevrstvá perceptronová síť (MLP - *Multilayer perceptron*). Princip fungování MLP bude popsán v následující části textu na příkladu dvouvrstvé perceptronové sítě.

### Dvouvrstvá perceptronová síť

Úkolem této neuronové sítě je klasifikovat  $l$ -dimenzionální vektor  $\mathbf{x} \in \mathcal{R}^l$  pomocí  $p$  perceptronů (viz Obr. 1.4a). Tyto perceptrony se nacházejí v tzv. skryté vrstvě a jejich úkolem je namapovat  $l$ -dimenzionální vstupní vektor do  $p$ -dimenzionálního prostoru  $H_p$ , který si lze představit jako vrcholy prostorového mnohoúhelníku definovaného:

$$H_p = \{[y_1, \dots, y_p]^T \in \mathcal{R}^p, y_i \in \{0, 1\}, 1 \leq i \leq p\}.$$

Každý perceptron ve skryté vrstvě sítě vlastně představuje jednu klasifikační nadrovinu a převádí lineárně neseparovatelnou trénovací množinu  $X$  do prostoru, v němž je ji možné bez chyby rozdělit do dvou tříd (viz Obr. 1.4b).

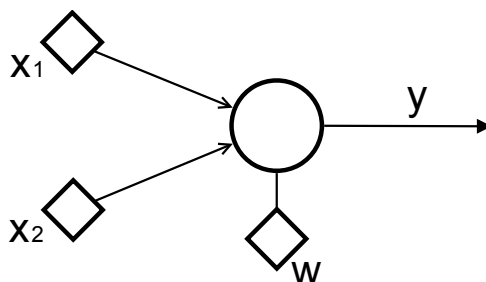


Obrázek 1.4: (a) Dvouvrstvá perceptronová síť, (b) Příklad transformace nelineárně separovatelného problému do prostoru, ve kterém existuje lineární řešení; neuronová síť je tvořena třemi perceptrony ve skryté vrstvě (perceptrony představují klasifikační nadrovinu  $g_1, g_2, g_3$ ).

Vícevrstvé perceptronové sítě se používají v případě komplikovanějších klasifikačních úloh. Jejich konstrukce obsahuje více skrytých vrstev a pracují na principu rozložení složitějšího problému na menší mnohem jednodušší problémy. Kombinací řešení těchto dílčích problémů je následně dosaženo hledaného výsledku

## Aktivační funkce

Uvažujme jednoduchý perceptron s dvěma vstupy  $x_1$ ,  $x_2$  a jedním výstupem  $y$  (viz Obr. 1.5). Výstup takového perceptronu odpovídá předpisu:



Obrázek 1.5: Perceptron.

$$y = f(a), \text{ kde } a = \sum_{i=1}^2 (wx_i).$$

Funkce  $f(a)$  je označována jako aktivační funkce neuronu a  $w$  je jeho synaptická váha. Mezi nejčastěji používané aktivační funkce patří:

- Skoková funkce

$$f(a) = \begin{cases} 1 & \text{pro } a > 0 \\ 0 & \text{pro } a < 0 \end{cases} \quad (1.7)$$

- Sigmoida

$$f(a) = \frac{1}{1 + \exp(-\lambda a)} \quad (1.8)$$

kde  $\lambda$  je parametr sklonu.

- Hyperbolický tangens

$$f(a) = \frac{1 + \exp(-\lambda a)}{1 - \exp(-\lambda a)} \quad (1.9)$$

Proces trénování neuronové sítě spočívá v nastavení synaptických vah jednotlivých neuronů. Jelikož je k dispozici trénovací množina jedná se opět o tzv. učení s učitelem, které je realizováno např. algoritmem zpětného šíření (*Backpropagation Algorithm*).

## Algoritmus zpětného šíření

Algoritmus vychází z trénovací množiny, která obsahuje  $N$  dvojic  $(y(i), \mathbf{x}(i))$ , pro  $i = 1, 2, \dots, N$ , kde  $\mathbf{x}(i)$  představuje  $i$ -tý vektor trénovací množiny a  $y(i)$  určuje jeho příslušnost do jedné ze tříd. Během trénování jsou v jednotlivých iteracích získávány výstupy neuronové sítě  $\hat{y}(i)$ , které se liší od požadovaného  $y(i)$ . Cílem učení je nastavit synaptické váhy jednotlivých vrstev sítě tak, aby rozdíl mezi aktuálním výstupem  $\hat{y}(i)$  a požadovaným výstupem  $y(i)$  byl minimální.

Nové synaptické váhy jsou vypočítávány jako

$$w_j^r(nova) = w_j^r(stara) + \Delta w_j^r, \quad (1.10)$$

kde  $w_j^r$  je váha  $j$ -tého neuronu v  $r$ -té vrstvě sítě a  $\Delta w_j^r$  značí přírůstek, který je definován

$$\Delta w_j^r = -\mu \frac{\partial J}{\partial w_j^r}, \quad (1.11)$$

kde  $\mu$  je konstanta učení (parametr algoritmu). Cílem algoritmu zpětného šíření je minimalizace nákladové funkce  $J$  s předpisem:

$$J = \sum_{i=1}^N \mathcal{E}(i), \quad (1.12)$$

kde lze za  $\mathcal{E}(i)$  zvolit rozdíl čtverců

$$\mathcal{E}(i) = \frac{1}{2} \sum_{i=1}^N (y(i) - \hat{y}(i))^2. \quad (1.13)$$

Přírůstek  $\Delta w_j^r$  ze vzorce 1.10 je stanoven vyřešením rovnice

$$\frac{\partial \mathcal{E}(i)}{\partial w_j^r} = 0. \quad (1.14)$$

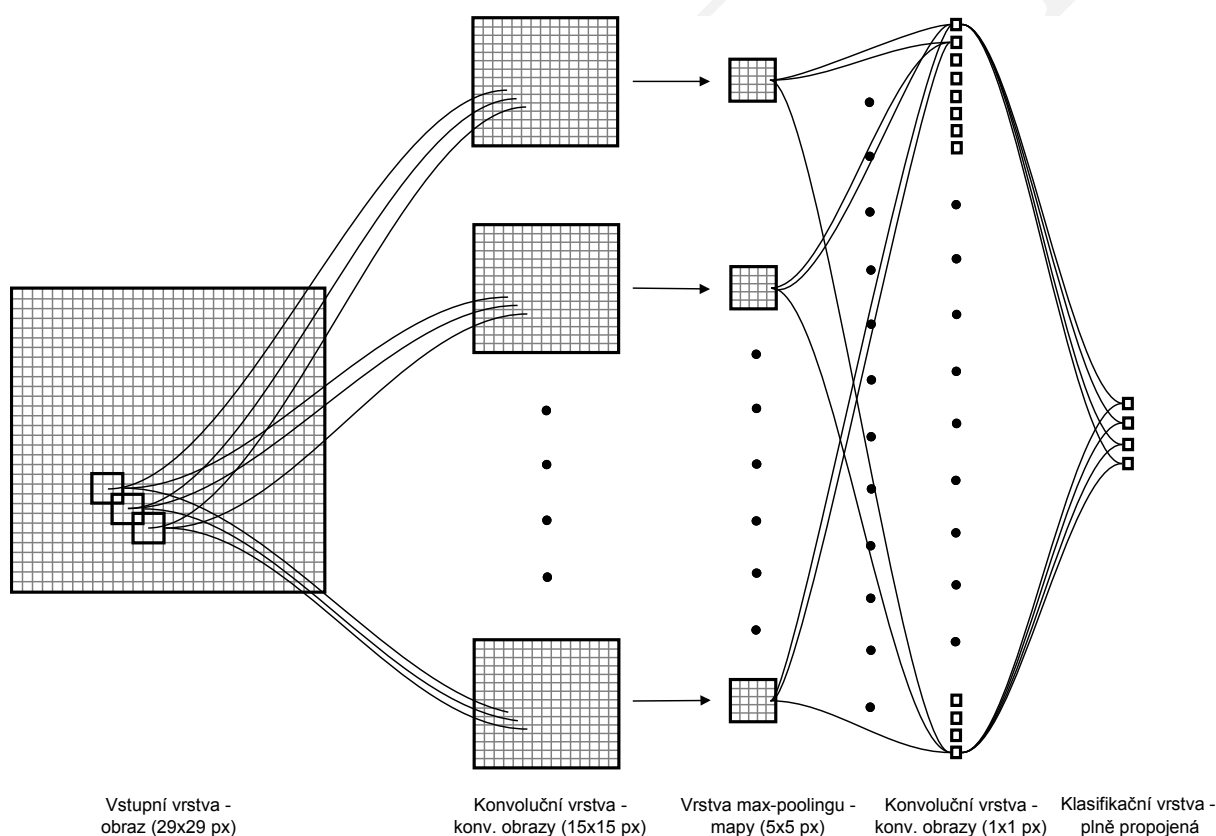
Algoritmus učení neuronové sítě probíhá v několika krocích. Nejprve jsou náhodně nastaveny synaptické váhy  $w_j^r$  pro jednotlivé neurony (obvykle se doporučuje váhy volit v intervalu  $\langle -1, 1 \rangle$ ) a je zvolena hodnota konstanty učení  $\mu > 0$  a maximální chyby  $\mathcal{E}_{max} > 0$ . Tento krok se nazývá inicializace. Poté je spočtena odezva vstupního vektoru  $\hat{y}(i)$  a dojde k aktualizaci vah podle vzorce 1.10. Obdobně se postupuje pro všechny prvky  $x(i) \in X$  z trénovací množiny. Jestliže je na konci trénovacího cyklu výsledná chyba  $\sum_{i=1}^N \mathcal{E}(i) \leq \mathcal{E}_{max}$  algoritmus skončí, v opačném případě dojde k nové inicializaci plynoucí z řešení rovnice 1.14 a algoritmus se zopakuje.

Algoritmus zpětného šíření představuje jednoduchý způsob učení neuronové sítě, který ovšem velmi kvalitně vystihuje princip takového učení. V odborné literatuře se lze setkat i se složitějšími algoritmy učení, mezi které patří algoritmy založené na určení směru gradientu funkce 1.12 či algoritmy využívající kvazi-Newtonovu metodu.

## 1.6 Konvoluční neuronové sítě

Za účelem klasifikace obrazových dat jsou momentálně v praxi nejhojněji využívány tzv. konvoluční neuronové sítě (CNN - *Convolutional Neural Networks*), které mají podobnou strukturu jako běžné neuronové sítě, ale výstupy jednotlivých vrstev nejsou získávány pomocí aktivační funkce neuronů, nýbrž pomocí konvoluce vstupu a příslušného konvolučního jádra.

CNN se od klasické neuronové sítě liší již svým vstupem. Vstupem totiž není příznakový vektor obrazu, ale obraz samotný (viz Obr. 1.6). Nad vstupním obrazem je provedena konvoluce, jejíž výsledek představuje výstup konvoluční vrstvy. Ta může být následována vrstvou max-poolingu, nebo další konvoluční vrstvou. Konvoluce probíhá pouze nad definovaným prostorem vstupu, tzn. že konvoluční jádro nesmí zasahovat mimo okraje vstupního obrazu. Zřetězením těchto operací je postupně redukována velikost vstupního obrazu až do bodu, kdy je výstupem pouze jednodimenzionální příznakový vektor. Ten je v nejvyšší vrstvě sítě (v tzv. klasifikační vrstvě) zařazen do jedné klasifikační třídy.



Obrázek 1.6: Architektura konvoluční neuronové sítě; velikost konvolučních jader je  $3 \times 3$ , hodnota posunu je nastavena na 1 a klasifikace probíhá do čtyř tříd.

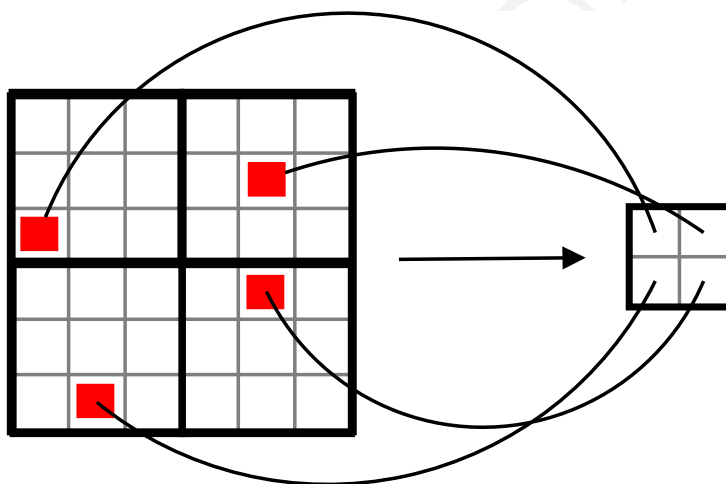
### Konvoluční vrstva

Konvoluční vrstva je určena počtem a velikostí konvolučních jader a hodnotou posunu.

Počet konvolučních jader  $M$  určuje počet výstupů vrstvy, které posléze vstupují do vrstvy následující. Hodnota posunu,  $S_x$  a  $S_y$ , definuje o kolik pixelů se jádro během konvoluce po vstupním obraze posouvá ve směru osy  $x$  a  $y$ . Velikost konvolučního jádra  $(K_x, K_y)$  spolu s hodnotou posunu určují velikost výstupních konvolučních obrazů  $(M_x, M_y)$ .

## Vrstva max-poolingu

Max-pooling představuje proces podvzorkování vstupního konvolučního obrazu. Tento proces je do CNN zařazen z důvodu, že urychluje konvergenci sítě a vede k nalezení robustnějších obrazových příznaků. Max-pooling pracuje na principu detekce maxim v nepřekrývajících se regionech vstupního obrazu. Tyto regiony jsou definovány velikostí konvolučního jádra  $(K_x, K_y)$ . Výstupem vrstvy max-poolingu je tedy mapa obsahující nalezená maxima a rozměr mapy odpovídá podvzorkování vstupního obrazu o faktor  $K_x$ , resp.  $K_y$ , podél obou os (viz Obr. 1.7).



Obrázek 1.7: Ukázka max-poolingu; maxima dílčích regionů vstupního obrazu jsou vyznačena červeně a regiony mají rozměr  $3 \times 3$ .

## Klasifikační vrstva

Tato vrstva představuje nejvyšší vrstvu CNN a obsahuje přesně tolik neuronů, kolik je klasifikačních tříd. Bývá také označována jako plně propojená vrstva, jelikož na vstup každého jejího neuronu jsou přivedeny všechny výstupní konvoluční obrazy vrstvy předešlé. Jak již bylo poznamenáno, výstupy předcházející vrstvy představují jednodimenzionální příznakový vektor vstupního obrazu, u kterého je následně rozhodnuto, do jaké náleží třídy.

Učení CNN probíhá obdobným způsobem jako učení klasické neuronové sítě. Jen s tím rozdílem, že nedochází k aktualizaci vah, ale jsou upravovány formy jednotlivých konvolučních jader.

## 1.7 Zdroje

- [1] Richard O. Duda, Peter E. Hart, David G. Stork, , *Pattern Classification (2nd Edition)*, Wiley-Interscience, 2000.
- [2] Breiman L., „Bagging Predictors“, *Machine Learning*, Vol. 24, pp. 123-140, 1996.
- [3] Breiman L., „Random Forests“, *Machine Learning*, Vol. 45, pp. 5-32, 2001.
- [4] Quinlan J.R., *C4.5: Programs for Machine Learning*, Morgan Kaufmann, 1993.
- [5] Theodoridis S., Koutroubas K., *Pattern Recognition*, Elsevier Inc., 2008.