Strojové učení

Úloha strojového učení

- je dána množina trénovacích dat: vstupní a výstupní vzory (požadované výstupy)
- chceme, aby model co nejlépe aproximoval neznámou funkci $\rightarrow$ aby pro každý předložený vstupní správně predikoval hodnotu výstupu
- generalizace = zobecňování – model by měl dát správný výstup i pro data, která nejsou v trénovací množině

Základní způsoby učení

Typy úloh

- regrese – predikujeme numerickou hodnotu
- klasifikace – predikujeme diskrétní hodnotu

Metody učení

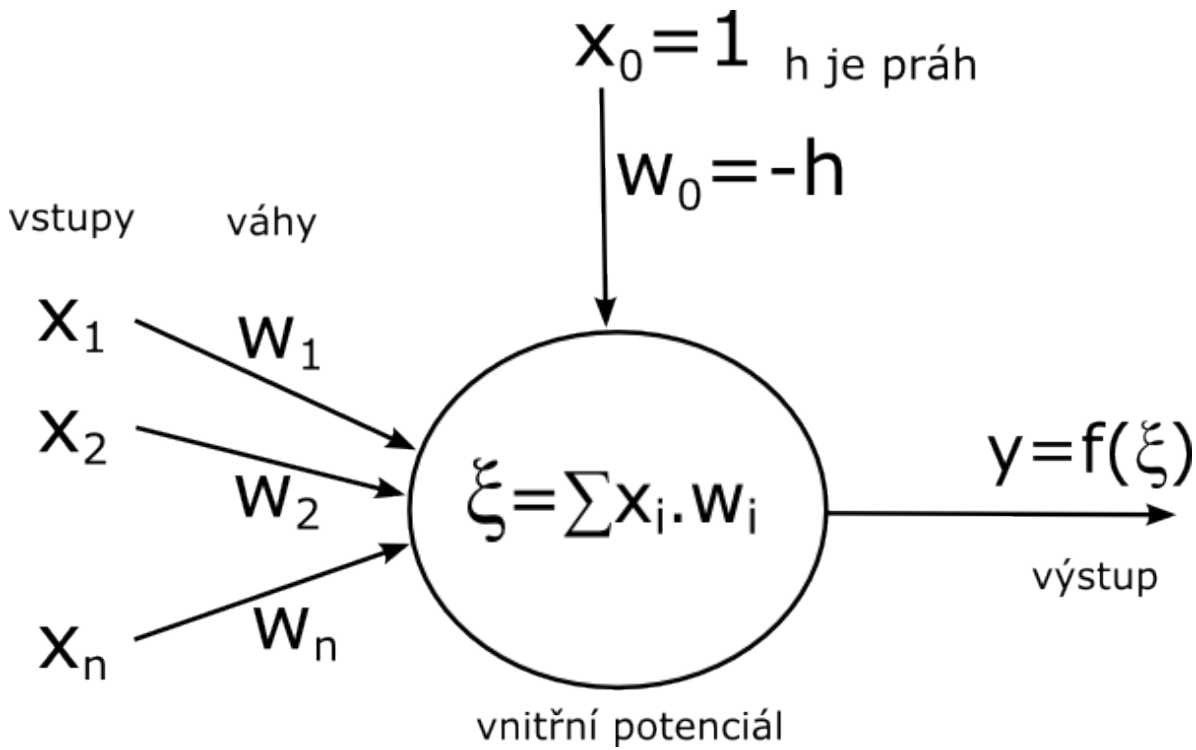
- Učení s učitelem (supervised learning)
  - trénovací množina ve tvaru [*vstup*, *požadovaný výstup*]
- Učení bez učitele (unsupervised learning, samoorganizace)
  - trénovací množina ve tvaru [*vstup*]
- Zpětnovazebné učení (reinforcement learning)

Strojové učení

Postup řešení úlohy

- připravit data – např. vybrat příznaky
- naučit model
  - jaký typ modelu? (záleží na problému)
  - jaký model daného typu? (volba vhodných parametrů)
- otestovat model – nejlépe na nových datech

Formální (matematický) model neuronu



- vnitřní potenciál $\xi = \sum_{i=1}^{n} w_i.x_i - h = \sum_{i=0}^{n} w_i.x_i = \vec{w}.\vec{x}$

Formální (matematický) model neuronu

Definice

- parametry neuronu:
  - vektor vah $\overrightarrow{w} = (w_1, ..., w_n) \in \Re^n$,
  - práh h
  - přenosová funkce $f : \Re^{n+1} \rightarrow \Re$
- neuron pro vstup $x \in \Re^n$ spočte výstup $y \in \Re$ jako hodnotu přenosové funkce $f_{\overrightarrow{w},h}(x)$

Formální (matematický) model neuronu

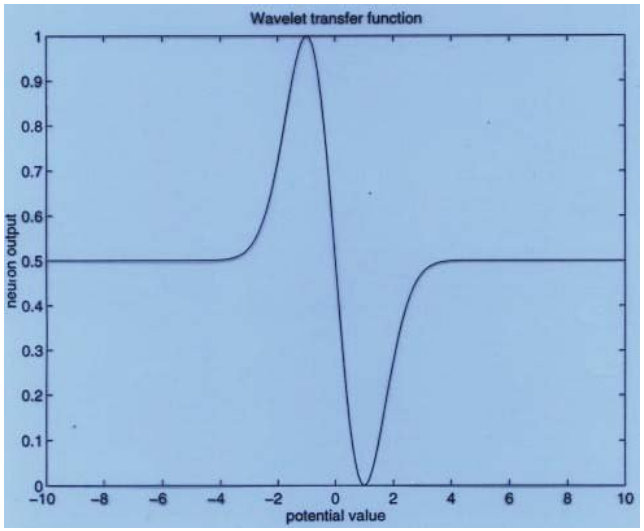
Definice

- parametry neuronu:
  - vektor vah $\overrightarrow{w} = (w_1, ..., w_n) \in \Re^n$,
  - práh h
  - přenosová funkce $f : \Re^{n+1} \rightarrow \Re$
- neuron pro vstup $x \in \Re^n$ spočte výstup $y \in \Re$ jako hodnotu přenosové funkce $f_{\overrightarrow{w},h}(x)$
  - $f_{\overrightarrow{w},h}(x) = 1$ ... neuron je aktivní
  - $f_{\overrightarrow{w},h}(x) = \frac{1}{2}$ ... neuron je tichý
  - $f_{\overrightarrow{w},h}(x) = 0$ ... neuron je pasivní

Přenosové funkce

- Diskrétní
  - Skoková (hard-limit) ... *hardlim*
  - Kvantovaná
- Spojitá
  - Lineární ... *purelin*
  - Lineárně saturovaná ... *satlin*
  - Sigmoidální ... *logsig*
  - Radiální (RBF) ... *radbas*
  - Waveletová
- Lokální
- Globální
- Pologlobální

Waveletová funkce



Přenosové funkce

Typické požadavky

- $f(y)$ definovaná na $(-\infty, \infty)$
- $lim_{y \to -\infty} f(y) = m < M = lim_{y \to \infty} f(y)$
- $f(y) \approx m$ ... neuron je inhibován (pasivní)
- $f(y) \approx M$ ... neuron je excitován (aktivní)

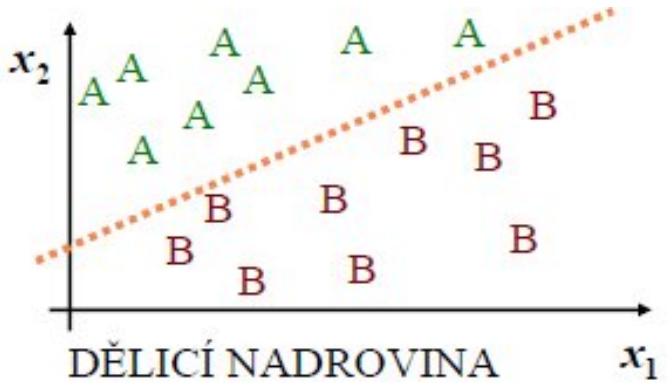
Typické modely

- $(m, M) = (0, 1)$ ... binární model ... *hardlim, logsig*
- $(m, M) = (-1, 1)$ ... bipolární model ... *hardlims, tansig*

Skoková přenosová funkce

hardlim, hardlims

- $f(\xi) = 1$ pro $\xi \geq 0$ ... třída A ... neuron je aktivní
- $f(\xi) = 0$ pro $\xi < 0$ ... třída B ... neuron je pasivní
- popř. varianta $f(\xi) = 0.5$ pro $\xi = 0$ ... neuron je tichý



- dělící nadrovina $x_2 = -\frac{w_1}{w_2}x_1 + \frac{h}{w_2}$
- $0 = w_1x_1 + w_2x_2 - h$

Spojité přenosové funkce

Lineární přenosová funkce

- $f(\xi) = \xi$ ... purelin

Sigmoidální funkce a hyperbolický tangens

- $f(\xi) = \frac{1}{1+e^{-\lambda\xi}}$ ... logsig
- $f(\xi) = \frac{1-e^{-\lambda\xi}}{1+e^{-\lambda\xi}}$ ... tansig

Radiální funkce (RBF)

- $f(\xi) = e^{-\frac{\xi^2}{\alpha}}$ ... radbas
- $\xi = \frac{|\vec{x}-\vec{w}|}{\beta}$

Stochastický model neuronu

- $f(\xi) = 1$ s pravděpodobností $P(v)$
- $f(\xi) = 0$ s pravděpodobností $1 - P(v)$

$P(v)$ je nejčastěji sigmoidální funkce:

- $P(v) = \frac{1}{1+e^{-\frac{v}{T}}}$
- T ... pseudoteplota

Neuronová síť

- Skládá se z neuronů, které jsou navzájem pospojovány synaptickými vazbami (hranami)
- Výstup jednoho neuronu může být vstupem jednoho nebo více dalších neuronů

Architektura (topologie) neuronové sítě

- Orientovaný graf, neurony představují uzly, synaptické vazby představují hrany
- Cyklická / acyklická (dopředná), vrstevnatá

Konfigurace neuronové sítě

- Váhy všech hran a prahy všech neuronů

Neuronová síť

Vstupní / výstupní neurony

- Nevedou do / z nich žádné hrany z / do jiných neuronů

Výstup (odezva) neuronové sítě

- Výstupy (aktivity) výstupních neuronů

Neuronová síť

Definice: Neuronová síť je šestice (N,C,I,O,w,t):

- N je konečná neprázdná množina neuronů,
- $C \subseteq N \times N$ je neprázdná množina orientovaných spojů mezi neurony
- $I \subseteq N$ je neprázdná množina vstupních neuronů
- $O \subseteq N$ je neprázdná množina výstupních neuronů
- $w : C \to R$ je váhová funkce
- $t : N \to R$ je prahová funkce

Učení a rozpoznávání

Učení (s učitelem)

- chceme, aby síť implementovala neznámou funkci $F : \Re^n \rightarrow Y^m$
- trénovací množina T tvaru $T = \{(x^1, t^1), ..., (x^N, t^N)\}$, $x^i \in \Re^n, y^i \in Y^m$
- (i-tý) trénovací vzor je uspořádaná dvojice tvaru (x^i, t^i) =(vstup,požadovaný výstup)
- cílem učení je adaptace (nastavení) vah a prahů neuronové sítě

Učení a rozpoznávání

Cílová (chybová) funkce

- např. $E = \sum_p \sum_i (y_i^p - d_i^p)^2$
  y je skutečný výstup, d je požadovaný výstup

Rozpoznávání

- předkládání vstupních vzorů x^j
- cílem rozpoznávání je získat skutečný výstup (odezvu) neuronové sítě $y^j = \varphi(x^j)$

Projekty - příklady úloh

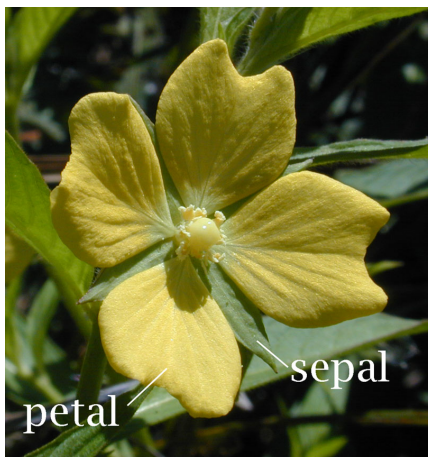
- predikce časové řady - predikce vývoje kurzu akcií (konkrétních), spotřeby elektrické energie, počasí (teplota či jiné údaje),...
- obrázky - rozpoznávání znaků - písmen nebo číslic, obličejů, redukce počtu barev, komprese obrázku, rozpoznávání psů a koček apod
- zvuky - rozpoznání (pohlaví) mluvčího, hudebního stylu, jazyka, tónů v akordech apod.
- odhad ceny - ojetý automobil, byt v Brně, PC sestava, akcie na trhu,...
- odhad čehokoliv (schopnost klienta banky splatit dluh, očekávaná délka nezaměstnanosti v Ostravě, příčina poruchy automobilu...)
- rozpoznání spamu, analýza diskusních příspěvků,...
- medicína - např. vliv tělesných charakteristik na EKG,...

Projekty - příklady úloh

- UCI Machine learning repository (http://archive.ics.uci.edu/ml/)
- http://www.kdnuggets.com/datasets/

Příklad: Iris DataSet

- 150 vzorů, 4 příznaky (sepal length in cm, sepal width in cm, petal length in cm, petal width in cm)
- klasifikace do 3 tříd (Setosa, Versicolour, Virginica)



Cvičení - Iris DataSet

- Prohlédněte si informace o datové sadě na UCI Machine learning repository a na Wikipedii
- Natáhněte data v Matlabu ('fisheriris')
- Zjistěte základní charakteristiky u jednotlivých příznaků (střední hodnota, rozptyl,...)
- Zkuste data vhodně zobrazit (boxplot, gscatter, gplotmatrix,...) a popište zjištěný charakter dat

Příprava dat

- výběr (vytvoření) relevantních příznaků
- převod dat na reprezentaci vhodnou pro daný model / algoritmus

Obvyklé problémy

- Příliš mnoho vzorů
- Příliš mnoho příznaků (atributů)
- Chybějící hodnoty, odlehlé hodnoty

Příliš mnoho vzorů

- Použít jen určitý vzorek (sample) dat
  - Vybrané objekty by měly co nejlépe vystihovat všechna data (např. podle shody rozdělení hodnot atributů ve vybraném vzorku i ve všech datech)
  - Je třeba zohlednit případnou nevyváženost původních dat $\rightarrow$
    $\rightarrow$ různé váhy pro různý typ chybného rozhodnutí
    $\rightarrow$ vybírat vzory z různých tříd s různou pravděpodobností
- Vytvořit více modelů a výsledky zkombinovat

Příliš mnoho příznaků

Redukce počtu příznaků

- za pomoci experta
- automaticky
  - transformací - z existujících atributů vytvoříme menší počet nových atributů (PCA, Kahunen-Loevův rozvoj,...) $\rightarrow$
    $\rightarrow$ nové příznaky nemají jasnou interpretaci
  - selekcí (výběrem)
    - metoda filtru - ke každému příznaku spočteme jeho vhodnost, příznaky uspořádáme podle vhodnosti a vybereme jen ty nejlepší (zdola nahoru nebo shora dolů) $\rightarrow$ nepostihuje současný vliv více příznaků
      (obvykle míry založené na entropii)
    - metoda obálky - za využití modelu neuronové sítě (zdola nahoru nebo shora dolů)

Chybějící hodnoty

- Ignorovat objekt s nějakou chybějící hodnotou
- Nahradit chybějící hodnotu novou hodnotou 'nevím'
- Nahradit chybějící hodnotu některou z existujících hodnot příznaku:
  - Nejčetnější hodnotou
  - Proporcionálním podílem všech hodnot
  - Libovolnou hodnotou
- Nahradit chybějící hodnoty na základě (použitého) modelu

Odlehlé hodnoty

- Ignorovat objekt s nějakou odlehlou hodnotou
- Vhodná normalizace dat

Cvičení - Normalizace dat

- Máme vstupní data $X = \{x^1, ..., x^N\}$

Obvyklé metody

- min-max normalizace na interval $< A, B >$ ... typicky $< -1, 1 >$ a $< 0, 1 >$
- normalizace podle směrodatné odchylky $std(X) = \sqrt{\frac{X - mean(X)}{N-1}}$
- sigmoidální normalizace
- logaritmická normalizace

Cvičení

- Odvoďte jednotlivé vzorečky
- Najděte odpovídající funkce v Matlabu (!! řádky vs. sloupce)
- Vyzkoušejte všechny funkce na *fisheriris* a srovnejte/zobrazte si výsledky
- Kdy se hodí která metoda?

Kategoriální data

1. Převod na numerické příznaky ... *grp2idx*
2. Další úpravy:
   - jeden příznak (k kategorií) $\rightarrow$ k příznaků (0/1)
   - jeden příznak (k kategorií) $\rightarrow$ jeden příznak ($x_i \in < 0, 1 >$)

Cvičení - Generování testovacích dat

- Často se hodí umět generovat "náhodná" data s danými vlastnostmi

Metody

- *rand(m,n)* ... matice $m \times n$ s rovnoměrným rozložením
- *randn(m,n)* ... matice $m \times n$ s normálním rozložením

Cvičení

- Vygenerujte matici *mxn* s rovnoměrným rozložením z intervalu $< A, B >$
- Vygenerujte matici *mxn* s normálním rozložením s danou střední hodnotou *s* a směrodatnou odchylkou *r*

Cvičení - Generování testovacích dat

Generování vektoru se dvěma shluky

- Napište funkci randv2n(n1,S1,R1,n2,S2,R2), která vygeneruje řádkový vektor obsahující n1 hodnot s normálním rozdělením se střední hodnotou S1 a rozptylem R1 a n2 hodnot s normálním rozdělením se střední hodnotou S2 a rozptylem R2. Vytvořte histogram vygenerovaných dat.

Generování shluků ve 2D

- Napište funkci randv2D(p), která vygeneruje n shluků podle daného předpisu. p je matice $n \times 5$. $p(i,1)$ je počet hodnot se střední hodnotou $p(i,2)$ a a rozptylem $p(i,3)$ v první souřadnici a se střední hodnotou $p(i,4)$ a a rozptylem $p(i,5)$ v druhé souřadnici. Zobrazte vygenerované shluky do 2D grafu.

Cvičení - Generování testovacích dat

Generování vzorků

- Navrhnětě funkci select(x,k), která z matice x náhodně vybere k řádků. Zobrazte původní a vybraná data do grafu rozdílnými barvami nebo značkami.
- Užitečná funkce *randperm(N)*