BP sítě - analýza modelu

Minule

- Techniky pro zrychlení učení a pro zlepšení konvergence a aproximačních schopností BP-sítí
  - Adaptivní parametr učení, učení s momentem
  - Pseudonewtonovké metody
  - Metody konjugovaných gradientů
  - Relaxační metody

    ...

Srovnání těchto algoritmů

- Matlab → User Guide → Backpropagation → Speed and Memory Comparisons
- bylo za domácí úkol

Různé metody učení BP-sítě

Algoritmus zpětného šíření a jeho modifikace I.

- *traingd* ... algoritmus zpětného šíření (BP)
- *traingdm* ... BP s momentem
- *traingda* ... BP s adaptivním parametrem učení
- *traingdx* ... BP s adaptivním parametrem učení a momentem

Pseudonewtonovské metody

- *trainlm* ... Levenberg-Marquardtův algoritmus
- *trainloss* ... Quickprop
- *trainbfg*, *trainbfgc* ... další pseudonewtonovské metody

Různé metody učení BP-sítě

Algoritmus zpětného šíření a jeho modifikace I. Metody konjugovaných gradientů

- *trainscg* ... Moeller ...Metoda škálovaných konjugovaných gradientů
- *traincgf* ... Fletcher-Reeves
- *traincgp* ... Polak-Ribiere
- *traincgb* ... Powell-Beale

Relaxační metoda

- *trainrp* ... Resilent method

Řešení domácí úlohy

Příklad

- Uvažujeme síť s x skrytými neurony a stejný příklad (data) jako minule.
- Napiště skript (nebo i pomocné funkce), který naučí k (=100) sítí 4 z těchto metod (pro vhodně zvolené parametry trainParam) a spočítá:
  - průměr (mean) a směrodatnou odchylku (std) počtu cyklů a času (tr.best_epoch, tr.time)
  - průměr (mean) a směrodatnou odchylku (std) výsledné chyby na trénovací, testovací a validační množině (tr.perf, tr.vperf, tr.tperf pod indexem tr.best_epoch)
- Alternativně lze použít k-násobnou křížovou validaci jako v minulém dom. úkolu (pro k=10) - získáme rychlejší, ale méně stabilní odhad.
- Jak dopadlo srovnání jednotlivých metod?

Řešení domácí úlohy

```
for i=1:k
 net = newff(p,t,[5],{'tansig','purelin'},method);
 net.inputs{1}.processFcns = {};  % ... atd.
 net.trainParam.epochs = 1500; net.trainParam.max_fail = 100;
 % net.trainParam.lr=0.6; net.trainParam.mc=0.6;
 % net.trainParam.lr_dec=0.6;
 [net1,tr] = train(net,p,t);
 v(i,1) = tr.time(1,end); % cas
 v(i,2) = tr.best_epoch; % pocet cyklu
 v(i,3) = tr.perf(1,tr.best_epoch+1); % chyba trenovaci
 v(i,4) = tr.vperf(1,tr.best_epoch+1); % chyba validační
 v(i,5) = tr.tperf(1,tr.best_epoch+1); % chyba testovací
end
vysl = reshape([mean(v); std(v)],1,10); % vysledek v poli
```

Řešení domácí úlohy - pro 5 skrytých neuronů

Příklad výsledku experimentu - čas

metoda	čas	počet cyklů
traingd	22.0 ± 2.1	1451.7 ± 167.7
traingdm	21.9 ± 3.5	1454.5 ± 251.7
traingda	18.2 ± 7.0	1217.2 ± 520.2
traingdx	16.4 ± 7.6	1083.2 ± 564.9

Příklad výsledku experimentu - chyba

metoda	E_{tr}	E_v	E_t
traingd	0.043 ± 0.062	0.049 ± 0.072	0.047 ± 0.069
traingdm	0.052 ± 0.083	0.053 ± 0.080	0.056 ± 0.086
traingda	0.010 ± 0.035	0.011 ± 0.035	0.014 ± 0.041
traingdx	0.005 ± 0.021	0.013 ± 0.042	0.011 ± 0.034

Pro statisticky významné srovnání:

- alespoň **k=100** opakování nebo 10-násobná křížová validace

Řešení domácí úlohy - pro 5 skrytých neuronů, pokračování

Příklad výsledku experimentu - čas

metoda	čas	počet cyklů
traingd	22.0 ± 2.1	1451.7 ± 167.7
trainlm	27.2 ± 12.4	1203.2 ± 596.8
trainscg	26.9 ± 12.9	991.4 ± 535.7
trainrp	19.5 ± 9.2	1086.7 ± 572.8
traincgf	5.0 ± 2.2	1083.2 ± 92.1

Příklad výsledku experimentu - chyba

metoda	E_{tr}	E_v	E_t
traingd	0.0431 ± 0.062	0.049 ± 0.072	0.047 ± 0.069
trainlm	0.0022 ± 0.014	0.009 ± 0.034	0.396 ± 3.252
trainscg	0.0005 ± 0.002	0.004 ± 0.019	0.025 ± 0.090
trainrp	0.0044 ± 0.022	0.008 ± 0.030	0.007 ± 0.032
traincgf	0.0017 ± 0.009	0.010 ± 0.029	0.006 ± 0.017

Řešení domácí úlohy - pro 2 skryté neurony

Příklad výsledku experimentu - čas

metoda	čas	počet cyklů
traingd	21.3 ± 4.4	1421.2 ± 318.6
traingdm	20.7 ± 5.7	1371.4 ± 409.6
traingda	11.4 ± 6.6	701.1 ± 481.5
traingdx	9.4 ± 7.1	566.9 ± 525.8

Příklad výsledku experimentu - chyba

metoda	E_{tr}	E_v	E_t
traingd	0.156 ± 0.112	0.161 ± 0.124	0.159 ± 0.122
traingdm	0.165 ± 0.111	0.170 ± 0.116	0.174 ± 0.115
traingda	0.118 ± 0.085	0.115 ± 0.086	0.122 ± 0.093
traingdx	0.098 ± 0.086	0.098 ± 0.089	0.106 ± 0.097

Řešení domácí úlohy - pro 2 skryté neurony, pokračování

Příklad výsledku experimentu - čas

metoda	čas	počet cyklů
traingd	21.3 ± 4.4	1421.2 ± 318.6
trainlm	16.7 ± 13.3	730.2 ± 660.5
trainscg	10.9 ± 11.3	341.3 ± 465.5
trainrp	9.5 ± 8.1	820.3 ± 618.8
traincgf	4.2 ± 2.3	103.3 ± 88.4

Příklad výsledku experimentu - chyba

metoda	E_{tr}	E_v	E_t
traingd	0.156 ± 0.112	0.161 ± 0.124	0.159 ± 0.122
trainlm	0.043 ± 0.079	0.049 ± 0.091	4.402 ± 13.994
trainscg	0.055 ± 0.078	0.061 ± 0.086	0.084 ± 0.125
trainrp	0.098 ± 0.085	0.095 ± 0.090	0.109 ± 0.099
traincgf	0.073 ± 0.086	0.074 ± 0.090	0.077 ± 0.090

Schopnost sítě zobecňovat

Bias-Variance Tradeoff (dilema)

- MSE vyjadřuje odchylku mezi skutečnou a požadovanou odezvou sítě:
  $E = \frac{1}{2N} \sum_p \sum_i (d_i^p - y_i^p)^2 \sim E_{p,i}[d_i^p - y_i^p]^2$
  - d_i^p je požadovaný i-tý výstup sítě
    y_i^p je skutečný i-tý výstup sítě
    i je index přes výstupní neurony
    p je index přes možné trénovací vzory
- MSE mohu rozložit na Variance a Bias:

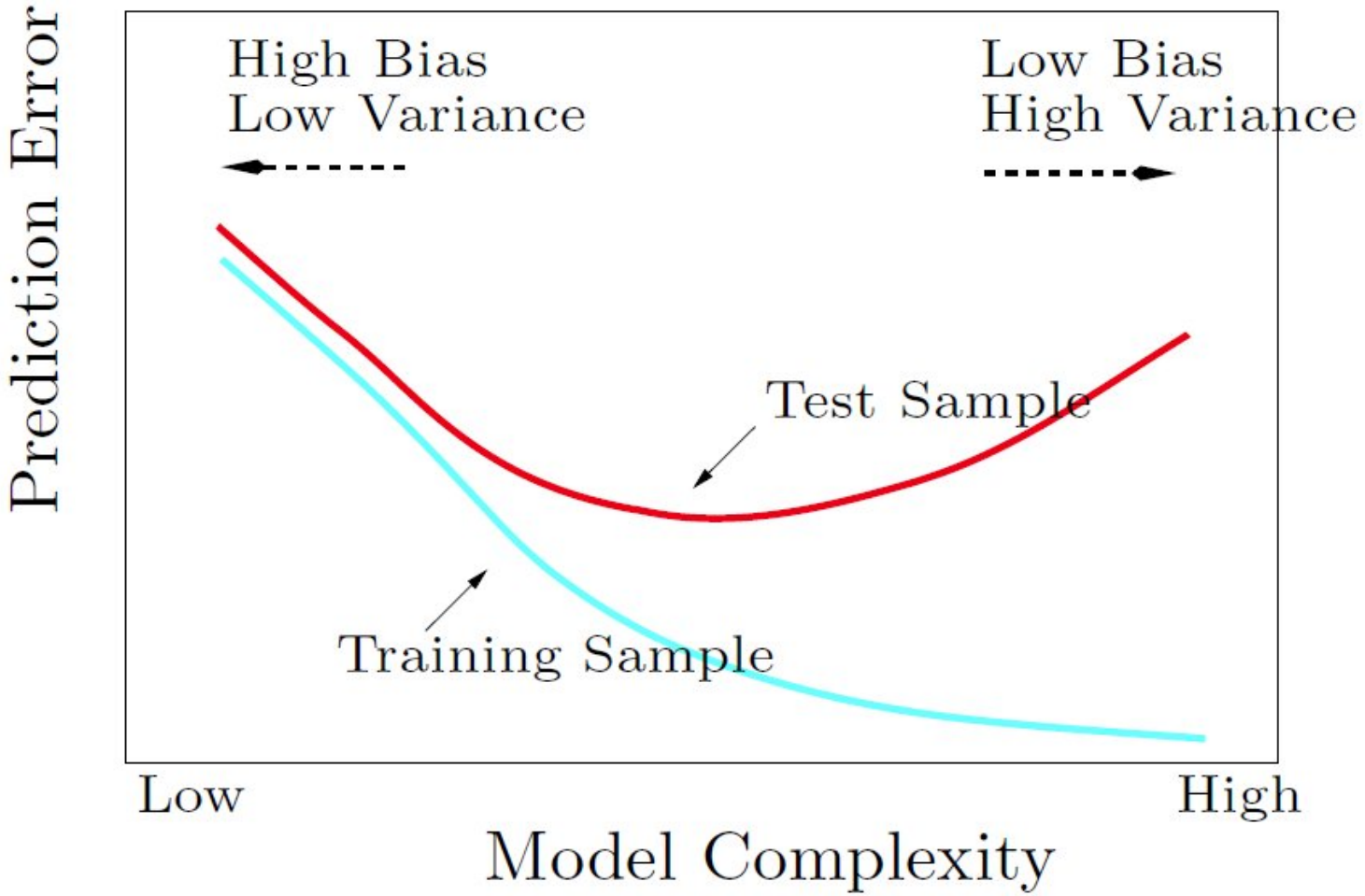
$$\begin{aligned} E &= E_{p,i}[d_i^p - y_i^p]^2 \\ &= E[y_i^p - E[y_i^p]]^2 + [E[y_i^p] - d_i^p]^2 \\ &= Var(y_i^p) + Bias^2(y_i^p) \end{aligned}$$

Bias-Variance Tradeoff

Vztah mezi vychýlením (Bias) a rozptylem (Variance)

- Záleží na architektuře sítě (na počtu vrstev a neuronů) - tj. na počtu parametrů modelu
- Bias a Variance jdou proti sobě - hledáme kompromis
- Malá síť
  - velký Bias, nízký rozptyl
  - potenciálně chybné, ale stabilní predikce (pro různé trénovací množiny a počáteční hodnoty vah)
  - hrozí *underfitting* (síť se nenaučila správně)
- Velká síť
  - velký rozptyl, nízký Bias
  - hrozí *overfitting* - síť se přeučila, špatně zobecňuje

Bias-Variance Tradeoff



Vapnik – Chervonenkisova dimenze (VC–dimenze)

Definice

Nechť $C = \{f_i\}$ je množina funkcí. Množinu m trénovacích vzorů $\{x_k\}, k = 1, \ldots, m$ lze rozčlenit pomocí C, jestliže pro každé ze 2^m možných označení těchto vzorů 1 / 0 , existuje alespoň jedna funkce fi , která tomuto označení vyhovuje.

Definice

VC-dimenze V množiny funkcí C je definována jako největší m , pro které existuje množina m rozčlenitelných trénovacích vzorů.

Příklad

- VC-dimenze množiny lineárních indikačních funkcí tvaru $\{\sum_{i=1}^{n} w_{x_i} + b\}$ v n-dimenzionálním prostoru je n+1.
- VC-dimenze množiny funkcí tvaru $\{sin(\alpha x)\}$ je nekonečná.

Vapnik – Chervonenkisova dimenze (VC–dimenze)

- VC-dimenze je důležitá pro správné zobecňování - velká VC-dimenze vede obvykle k horšímu zobecňování
- VC-dimenze množiny funkcí obecně nezávisí na počtu parametrů - aby síť dobře zobecňovala, může mít mnoho parametrů, ale měla by mít malou VC-dimenzi.
- Pokud je VC-dimenze C rovna nekonečnu, je takový problém „nenaučitelný“

Vapnik – Chervonenkisova dimenze (VC–dimenze)

Pravidla

- Pro síť s počtem vah W a počtem neuronů H a s omezením pro generalizační chybu ϵ, je počet trénovacích vzorů N potřebných pro správné zobecňování: $N \geq \frac{W}{\epsilon} log_2(\frac{H}{\epsilon})$.
- Vrstevnatá síť s 1 skrytou vrstvou nemůže dobře zobecňovat, jestliže $N \geq \frac{W}{\epsilon}$
- Pro požadovanou přesnost alespoň 90 % je třeba vybrat alespoň 10 ·W vzorů

Jak zajistit, aby síť dobře zobezňovala?

Jak snížit VC-dimenzi?

- Najít "optimální" architekturu
- Samplovací techniky
  - Early stopping - včas zastavit učení za použití validační množiny (už jsme probírali).
  - Krosvalidace (už jsme probírali)
- Regularizační techniky
- Učení s nápovědou
- Prořezávání

Regularizační techniky

Základní princip

- Přidávají k chybové funkci (MSE) další penalizační členy: $E = c_{mse}E_{mse} + c_A E_A + c_B E_B + \ldots$
- **Occamova břitva:** Menší sítě s jednodušší, hladší funkcí lépe zobecňují (VC-dimenze)
- Existuje celá řada jednoduchých i sofistikovaných penalizačních členů.

Regularizační techniky

Weight decay (Werbos, 1988)

- asi nejznáměnší a nejpoužívanější penalizační člen:

$$E = \beta E_{mse} + (1 - \beta)\frac{1}{N}\sum_{i=1}^{N} w_i^2$$

  i je index přes všechny váhy a prahy v síti
  $\beta \in < 0, 1 >$ udává váhu jednotlivých chybových členů
- V průběhu učení se snižují hodnoty vah v absolutní hodnotě
- Prevence paralýzy sítě
- Je možné ze sítě odstranit hrany s příliš malými vahami
- Bohužel často příliš nezlepšuje zobecňování (ale existují "lepší" reg. členy)

Jak je to v Matlabu

Implementované techniky pro zlepšení zobecňování
User Guide → Backpropagation → Improving Generalization

Rozdělení dat mezi trénovací, validační a testovací množinu

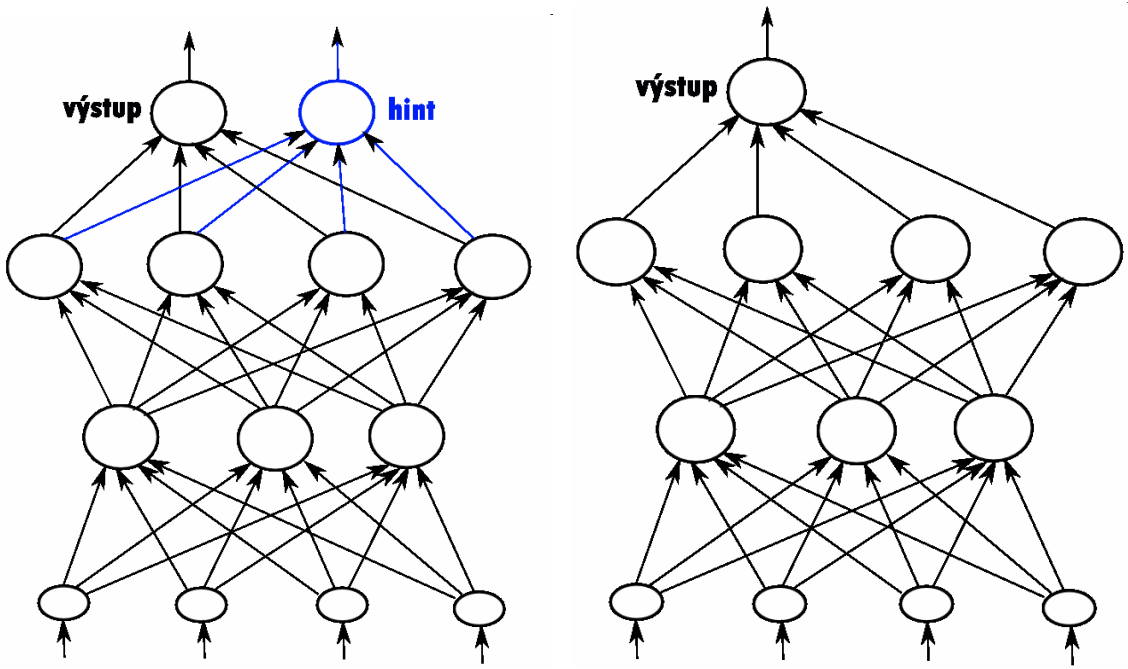
- lze nastavit *net.divideFcn* a jí příslušné *net.divideParam*
- *dividerand* ... náhodně (*trainRatio*, *valRatio*, *testRatio*)
- *divideint* ... bez přeuspořádání (*trainRatio*, *valRatio*, *testRatio*)
- *divideind* ... indexy zadá uživatel (*trainInd*, *valInd*, *testInd*)

Regularizační techniky - Weight decay

- *net.performFcn* = *'msereg'*
- *net.performParam.ratio* ... váha chybového členu

Učení s nápovědou

(Mostafa,1993;Suddarth,1990)



(a) Síť během učení (b) Síť po naučení

- Zvyšuje schopnost sítě zobecňovat a zrychluje učení (VC-dimenze).
- Vede k hladší funkci sítě, podporuje prořezávání.

Prořezávání sítě

Myšlenka:

- Vyhodnocení, které části BP-sítě jsou důležité
  - hrany
  - skryté neurony
  - vstupní příznaky
- Odstranění zbyečných částí BP-sítě

Důvody:

- Zrychlení výpočtu, snížení prostorové náročnosti.
- Zlepšení schopnosti sítě zobecňovat, detekce a řešení problému s přeučením (overfitting)
- Vytvoření sítě s jasnou strukturou.
- Automatická detekce důležitých vstupních příznaků.

Prořezávání sítě

Algoritmus:

1. Naučíme BP-síť s dostatečně velkou architekturou.
2. Dokud klesá chyba na validační množině (nebo dokud nepřekročí určitou mez):
   1. Spočteme relevanci skrytých neuronů nebo hran.
   2. Odstraníme nejméně relevantní neuron či hranu (neurony či hrany).
   3. Doučíme síť.

Problémy:

- Výpočet relevance skrytých neuronů nebo hran.
- Strategie, jak odstraňovat neurony / hrany.

Prořezávání sítě

Používané míry relevance (pro skryté neurony):

- **Goodness factor**: $G_i = \sum_p \sum_j (y_i^p w_{ij})^2$
  Zvýhodňuje neurony, ze kterých vedou hrany s velkou vahou a které aktivní jsou pro vetšinu vstupních vzorů.
- **Consuming energy**: $E_i = \sum_p \sum_j y_i^p w_{ij} y_j$
  Zvýhodňuje neurony, které jsou často aktivní, a to společně s neurony v následující vrstvě.
- **Součet vah**: $W_i = \sum_j (w_{ij})^2$.
  Nejjednodušší, ale účinná strategie.
- **Citlivostní koeficienty**

  ...

Citlivostní analýza [Zurrada at al. 1994]

Jak zjistit, které vstupní příznaky jsou pro neuronovou síť důležité? ... Citlivostní analýza

- Obecně: měří, jak moc je výstup neuronu (typicky výstupního) citlivý na malou změnu některých parametrů (typicky vah nebo vstupů)
- Umožňuje identifikovat důležité a naopak nedůležité části sítě.
- Umožňuje poznat, jak moc je který vstupní příznak důležitý pro výpočet sítě.

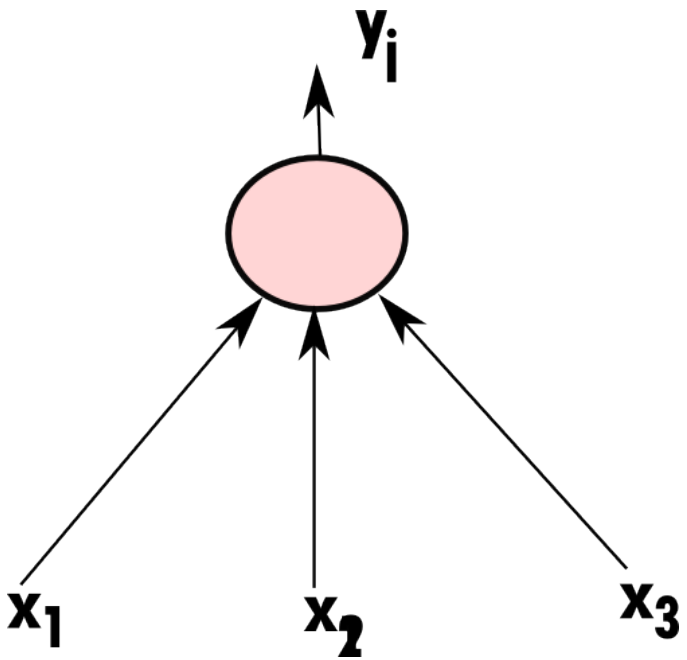
Obecná míra relevance:

- $S_{ij} = \frac{\partial y_j}{\partial x_i}$
  y_j ... j-tý výstup sítě (neuronu)
  x_i ... i-tý vstup sítě (neuronu)

Citlivostní analýza

Výpočet citlivostních koeficientů

- Citlivost výstupu neuronu j na jeho i-tý vstup:
  $S_{ij} = \frac{\partial y_j}{\partial x_i} = f'(\xi_j) w_{ij}$.

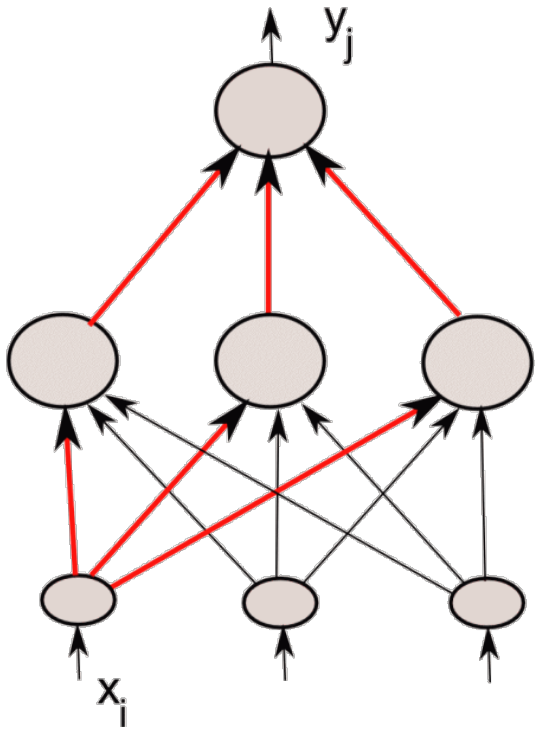


Citlivostní analýza

Výpočet citlivostních koeficientů

- Citlivost výstupu neuronu j na výstup neuronu i:
  $S_{ij} = \frac{\partial y_j}{\partial y_i} = \sum_k S_{kj} S_{ik} =$
  $\sum_k f'(\xi_j) w_{kj} S_{ik}$.

- Citlivost výstupu (výstupního) neuronu j na vstup sítě x_i:
  $S_{ij} = \frac{\partial y_j}{\partial x_i} = \sum_k f'(\xi_j) w_{kj} S_{ik}$.



Citlivostní analýza

Poznámky

- Citlivostní analýzu lze použít k prořezání sítě (hlavně vstupů - těch s nízkou citlivostí).
- Jak se zbavit výpočtu derivací? ... "zašuměním" (malinko pozměním jeden vstup a podívám se, jak moc se změnil výstup)

Citlivost sítě

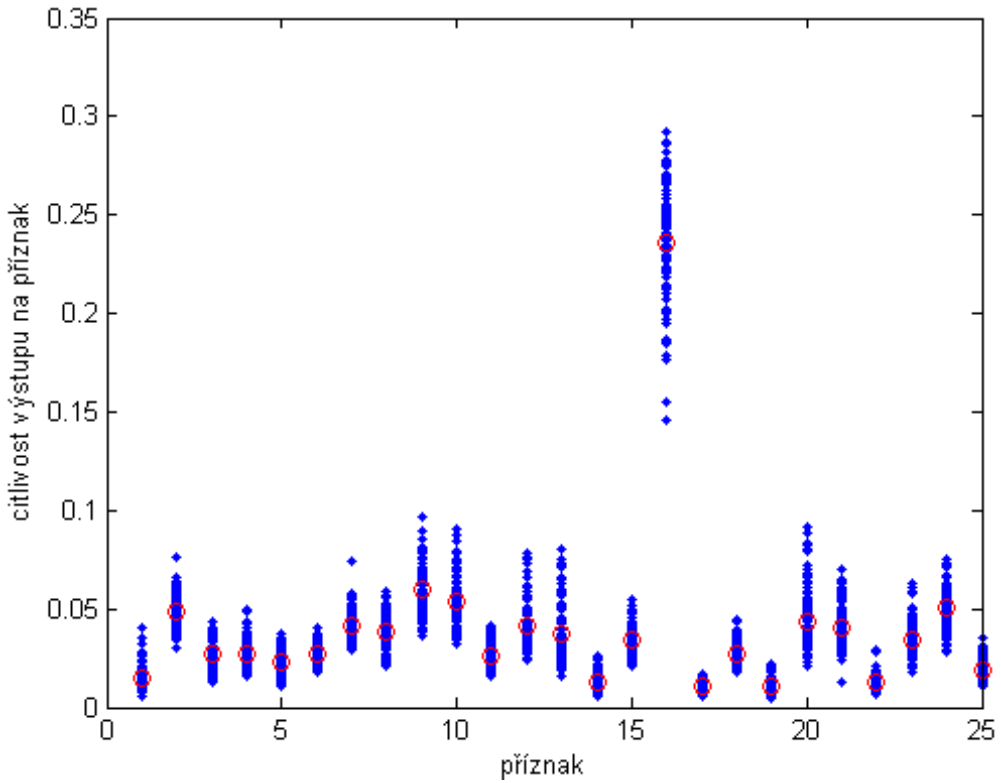
- Chování sítě na zašuměných datech ... indikátor, jak dobře síť zobecňuje.

Příklad: Data ze Světové banky - Indikátory světového vývoje

1	PPP konverzní faktor, místní měna vzhledem k mezinárodnímu $
2	Míra PPP konverzního faktoru vzhledem k oficiálnímu směnnému kurzu
3	Aktuální hodnota státního dluhu, % vývozu zboží, služeb a příjmů
4	Krátkodobý dluh, % celkového dluhu
5	Krátkodobý dluh, % vývozu zboží, služeb a příjmů
6	Celkový státní dluh, % HNP
7	Gini index
8	Příjem státního rozpočtu z daní, % HDP
9	Daně z příjmu, zisku a investic, % státních příjmů
10	Daně ze zboží a služeb, % příjmů
11	Daně z mezinárodního obchodu, % příjmů
12	Sociální příspěvky, % příjmů
13	Výdaje na výzkum a školství, % HDP
14	Deflace HDP, % růstu
15	Úmrtnost novorozenců, celkem (porody na ženu)
16	Vlastníci pevných a mobilních telefonů, na 1000 obyvatel
17	Růst HDP, ročně, %
18	Vývoz vyspělé technologie, % vývozu průmyslového zboží
19	Inflace, deflace HDP (ročně, %)
20	Uživatelé internetu, na 1000 obyvatel
21	Průměrná délka života, roky
22	Výdaje na zbrojení, % HDP
23	Počet obyvatel na hranici národní míry chudoby, % obyvatelstva
24	Aktuální hodnota státního dluhu, % HNP
25	Celkový státní dluh, % vývozu zboží, služeb a příjmů
26	HNP - Parita kupní síly

Příklad: Data ze Světové banky - Indikátory světového vývoje

Citlivost BP-sítě na vstupní příznaky



Důležité příznaky:

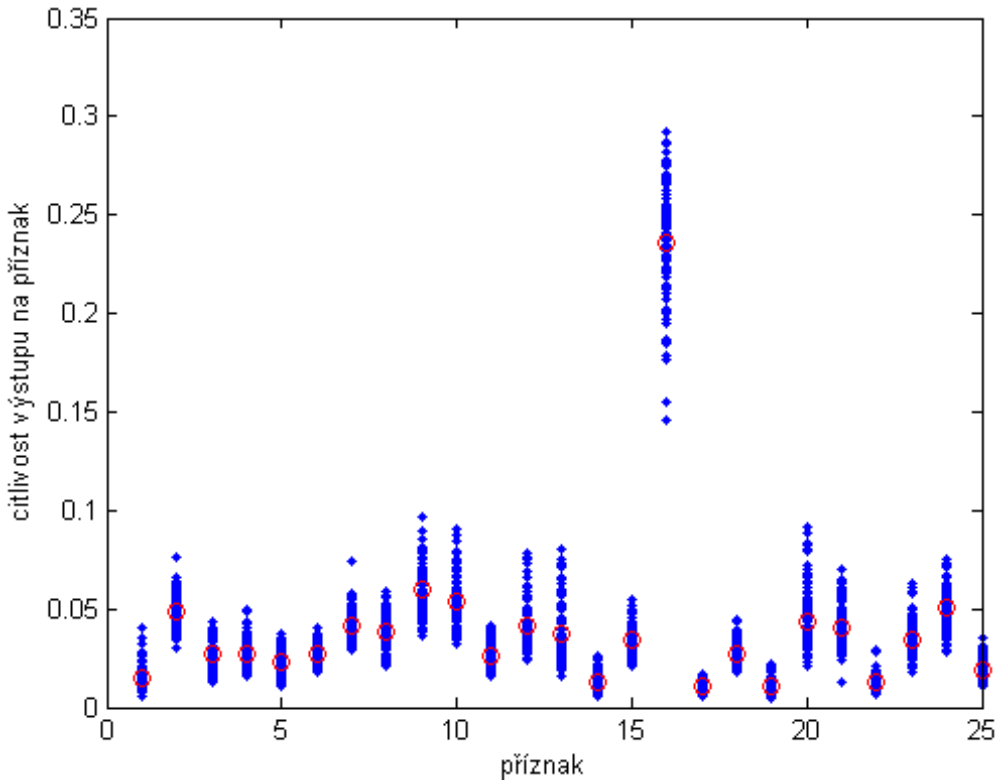
16	Vlastníci pevných a mobilních telefonů
9	Daně z příjmů
10	Daně ze zboží a služeb
20	Uživatelé internetu
24	Aktuální hodnota státního dluhu
2	PPP konverzní faktor
7	GINI index

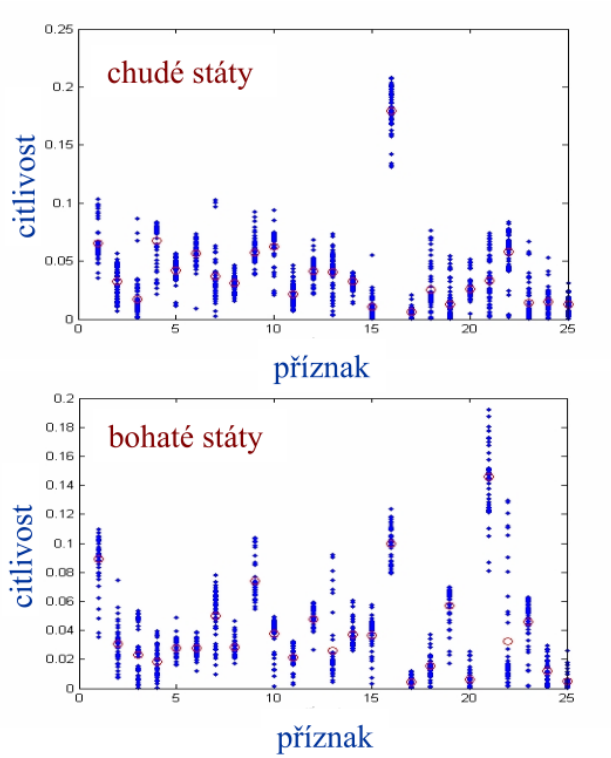
Nevýznamné příznaky:

17	Růst HDP
19	Inflace
22	Výdaje na zbrojení
14	Deflace HDP
1	PPP konverzní faktor
5	Krátkodobý dluh
25	Celkový státní dluh)

Příklad: Data ze Světové banky - Indikátory světového vývoje

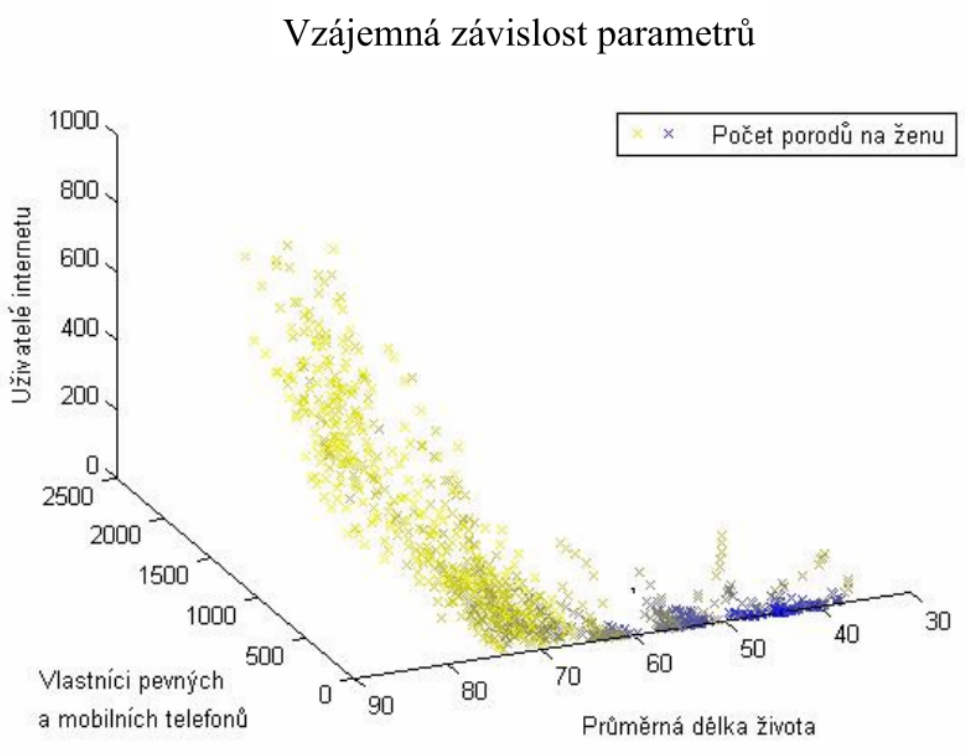
Citlivost BP-sítě na vstupní příznaky





Příklad: Data ze Světové banky - Indikátory světového vývoje

Vzájemná závislost parametrů



Další praktické poznámky

Klasifikace do více tříd pomocí BP-sítě

- Obvykle pro každou třídu jeden výstup (indikátor třídy). ... *cv7_klasifikace.m*

Predikce časových řad

- Obvykle predikce další hodnoty na základě předchozích. ... *cv7_sin.m*

Prokletí dimenzionality

- Pro velký počet vstupních parametrů mohu mít dat kolik chci, a stejně jich bude málo $\rightarrow$ redukce počtu parametrů (např. selekce)

Domácí úkol - dobrovolný

- Nahradí jeden předchozí chybějící.
- Najdětě si nějakou vlastní fotografii (obrázek) a upravte skript cv7_bitmapa.m tak, aby výrazně pozměnil její barevnost (ale jinak, než v původním skriptu) (zvolit jinak budete asi muset především architekturu sítě a trénovací vzory "navíc").
- Pošlete mi pozměněný skript i původní a několik výsledných fotografii (obrázků).