4. a 5. prednáška

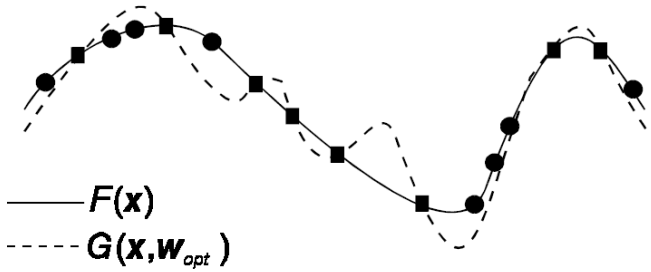
Backpropagation tips and tricks

Klasifikácia

- Kódovanie tried:
  - Kódovanie jeden z viacerých one-hot encoding
  - Binárne kódovanie
  - Kódovanie reálnym číslom

- Chyba klasifikácie
  - počet chybne zatriedených a nezatriedených vzorov
  - $o_k > 0.9 \rightarrow o_{ck} = 1$,
  - $o_k < 0.1 \rightarrow o_{ck} = 0$
  - alternatívna podmienka zastavenia

Aproximácia funkcií



Pre $\forall \varepsilon > 0$ platí, že $\exists G(\bar{x}) = f\left(\sum_{j=1}^{J} w_j f\left(\sum_{i=1}^{I} v_{ji} x_i\right)\right)$ také, že $\sum \left| F(\bar{x}_p) - G(\bar{x}_p) \right| < \varepsilon$

,kde že $F : R^n \rightarrow (0,1)$

F: funkcia spojitá, definovana na A_{train}

$G(\bar{x})$ aproximuje funkciu $F(\bar{x})$ s presnosťou $E = \frac{1}{2}\sum_{p=1}^{P}(d_p - o_p)^2 = \frac{1}{2}\sum_{p=1}^{P}(F(\bar{x}_p) - G(\bar{x}_p))^2$

Problém preučenia (Overfitting)

Metóda skorého zastavenia = early stopping

1. Rozdelenie dát na $A = A_{train}\ (75\%) \cup A_{val}\ (25\%)$, $A_{train} \cap A_{val} = \phi$

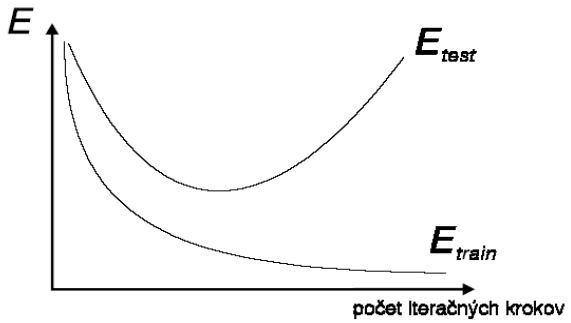
2. Väčší počet skrytých neurónov (J=20)

3. Malé počiatočné váhy

4. Malá rýchlosť učenia

5. Po každom cykle (epoche) vyhodnotiť E_{train} a E_{val}

6. Zastaviť trénovanie keď E_{val} začne rásť



- $A_{train} \cap A_{val} \cap A_{test} = \phi$
- rovnomerné zastúpenie bodov

Výber optimálneho modelu

Výber optimálneho počtu skrytých neurónov J
- viaceré modely s rôznym počtom skrytých neurónov
- výber modelu s najmenšou E_{test}

Rozdelenie dát na A_{train} a A_{test}
- veľa trénovacích dát – lepší model
- veľa testovacích dát – lepšie ohodnotenie modelu
- dáta často náročné na získanie – 1/3 až 1/10 z dát na testovanie
- rovnomerné zastúpenie tried (bodov) v množinách (stratification)

k-násobná cross validácia (k-fold cross validation)
- rozdeliť dáta do k-podmnožín rovnakej veľkosti
- trénovať na k-1 podmnožinách, testovať na zvyšnej podmnožine
- opakovať k krát, vystriedať všetky podmnožiny na testovanie
- vypočítať priemernú výkonnosť
- všetky dáta na trénovaní aj testovaní
- typicky k=10
- rovnomerné zastúpenie tried (bodov) v množinách (strat. k-fold cross valid.)

validácia vynechaním jedného (leave one-out cross validation)
- k-násobná cross validácia, pričom k = n (počet dostupných vzoriek)
- maximálne použitie dát na trénovanie
- deterministické, bez stochastického výberu dát do množín
- nerealizovateľné pre veľké množiny

bootstrap
- podmnožiny vytvárať výberom s vrátením (opakovaním)
- 63% prvkov z A sa ocitne v A_{train}

Bagging – bootstrap agregation
- trénovanie viacerých sietí
- trénovacie množiny vybrané technikou bootstrap
- výstup modelu je tvorený priemerovaním výstupov jednotlivých sietí

„Rastúce“ siete (Growing networks)

- zvoliť model s menším počtom skrytých neurónov
- trénovať pokiaľ klesá chyba
- modifikovať architektúru pridaním skrytých neurónov
- trénovať pokiaľ klesá chyba
- opakovať posledné 2 kroky podľa potreby

Kaskádová korelácia (Cascade Correlation)
- natrénovať jednoduchý model, iba vstupné a výstupné neuróny (nie backprop.)
- vložiť skrytý neurón a prepojiť ho na všetky ostatné skryté a vstupné neuróny
- nastaviť vstupné váhy aby výstup maximálne koreloval s chybou siete
- pripojiť neurón na výstupný neurón
- opakovať pridanie neurónu pokiaľ sa chyba zmenšuje

„Čistenie“ sietí (Pruning)

- natrénovať veľkú, husto prepojenú sieť
- overiť relatívny význam prepojení = váh v natrénovanej sieti
- odstrániť najmenej významné prepojenia
- dotrénovať sieť
- opakovať odstraňovanie najmenej významných váh podľa potreby
- Optimal Brain Damage, Optimal Brain Surgeon

Modifikácie algoritmu spätného šírenia chyby

Inkrementálne trénovanie (incremental learning)
- úprava váh po jednotlivých vzorkách

Kumulatívne, dávkové trénovanie (batch learning)
- úprava váh po prezentovaní všetkých vzoriek

Učenie strmosti aktivačnej funkcie

Použite techniky momenta
- zrýchlenie trénovania
- vyhýbanie sa lokálnym minimám

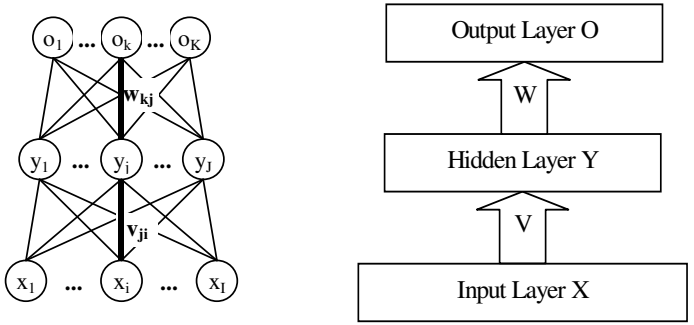
Zmena rýchlosti učenia
- linenárne znižovanie (monotónne klesanie)
- rôzna rýchlosť učenia pre skryté a výstupné neuróny
- „bold driven“
  - ak chyba za epochu klesá, zväčšenie rýchlosti učenia, inak zmenšenie
    $\Delta E(t) = E(t) - E(t-1)$,
    $\alpha(t+1) = \rho\alpha(t+1)$ ak $\Delta E(t) < 0$ pričom $\rho > 1$ napr. $\rho = 1.1$
    $\alpha(t+1) = \upsilon\alpha(t+1)$ ak $\Delta E(t) > 0$ pričom $\upsilon < 1$ napr. $\upsilon = 0.9$

- „delta-bar-delta“
  - pre každú váhu iná rýchlosť učenia
  - $g_i^p = \frac{\partial E_p}{\partial w_i^p}$, $\bar{g} = (1-\beta)g_i^p - \beta g_i^{p-1}$ kde $0 < \beta < 1$
  - $\alpha_i(t+1) = \alpha_i(t) + \kappa$ ak $\bar{g}_i(t-1)g_i(t) > 0$
  - $\alpha_i(t+1) = (1-\gamma)\alpha_i(t)$ ak $\bar{g}_i(t-1)g_i(t) < 0$
  - $\alpha_i(t+1) = \alpha_i(t)$ inak
  - náročné určiť parametre β, κ, γ
  - nepoužiteľné pre inkrementálne trénovanie

Quickpropagation

- $g_i^p = \frac{\partial E_p}{\partial w_i^p}$; $\Delta w_i^{p+1} = \frac{g_i^p}{g_i^{p-1} - g_i^p} \Delta w_i^p$

Algoritmizácia doprednej neurónovej siete



- dvojvrstvová dopredná neurónová sieť
- dopredné šírenie signálu
- spätné šírenie chybového signálu

Dopredné šírenie

- vstupný vektor $\bar{x} = (x_1 ... x_I)$,
- vypočítaný výstupný vektor $\bar{o} = (o_1 ... o_K)$

- pre skrytý neurón j sa vypočíta jeho „net výstup“ $\tilde{y}_j$ ako $\tilde{y}_j = \sum_{i=1}^{I} v_{ji} x_i$ a vyprodukuje výstup $y_j = f(\tilde{y}_j)$.

- pre výstupný neurón k sa vypočíta jeho „net výstup“ $\tilde{o}_k$ ako $\tilde{o}_k = \sum_{j=1}^{J} w_{kj} y_j$ a vyprodukuje výstup $o_k = f(\tilde{o}_k)$.

- v_{ji} je váhové prepojenie spajajúce skrytý neurón j so vstupom i
- w_{kj} je váhové prepojenie spajajúce výstupný neurón k so skrytým neurónom j
- f je aktivačná funkcia
- α rýchlosť učenia

Spätné šírenie

- očakávaný = želaný výstupný vektor, $\bar{d} = (d_1 ... d_K)$
- minimalizujeme chybu $E = \frac{1}{2} \sum_{k=1}^{K} (d_k - o_k)^2$

- zmeny výstupných váh určíme podľa $\Delta w_{kj} = -\alpha \frac{\partial E}{\partial w_{kj}} = \alpha \delta_k y_j$
- chybový signál δ_k výstupného neurónu k je definovaný ako $\delta_k = f'(\tilde{o}_k)(d_k - o_k)$

- zmeny skrytých váh určíme podľa $\Delta v_{ji} = -\alpha \frac{\partial E}{\partial v_{ji}} = \alpha \delta_j x_i$
- chybový signál δ_j skrytého neurónu j je definovaný ako $\delta_j = f'(\tilde{y}_j) \sum_{k=1}^{K} w_{kj} \delta_k$

- úprava váhových prepojení $v_{ji} = v_{ji} + \Delta v_{ji}$ a $w_{kj} = w_{kj} + \Delta w_{kj}$

Sigmoidálna aktivačná funkcia

- často používaná aktivačná funkcia $f(x)=\frac{1}{1+e^{-x}}$
- jej derivácia môže byť jednoducho určená vzťahom $f'(x)=f(x)(1-f(x))$
- chybové signály môžu byť vyjadrené ako $\delta_k=o_k(1-o_k)(d_k-o_k)$ a $\delta_j=y_j(1-y_j)\sum_{k=1}^{K}w_{kj}\delta_k$

Momentum

- úprava váh podľa vzťahov $\Delta v_{ji}(t)=\alpha\delta_j x_i+\beta\Delta v_{ji}(t-1)$ a $\Delta w_{kj}(t)=\alpha\delta_k y_j+\beta\Delta w_{kj}(t-1)$
- β je momentum

Kódovanie podľa Werbosa

- možnosť zakódovať sieť do jednoduchej štruktúry (Werbos notation)
- ľubovoľne poprepájaná sieť (sieť bez cyklov)
- jednoduchá implementácia algoritmu backpropagation
- jednoduchá implementácia algoritmov pre trénovanie rekurentných sietí (Backpropagation Through Time, Real-time recurrent learning)

neuróny

- očíslované, vstupné, skryté, výstupné (špec. prahový n. nastavený vždy na 1)
- neurón má prepojenia iba z neurónov s nižším indexom

prepojenia (váhy)

- index zdrojového neurónu
- index cieľového neurónu
- hodnota prepojenia
- čas oneskorenie prepojenia, ak rekurentná sieť
- postupne pre všetky neuróny po poradí (poradie určené indexom cieľového a potom zdrojového neurónu)

Štruktúra váhových prepojení

Index	Source	Dest.	Delay	Value
WI	SRC	DST	TD	W
0	0	6	0	
1	1	6	0	
2	2	6	0	
3	3	6	0	
4	4	6	0	
5	5	6	0	
6	0	7	0	
7	1	7	0	
8	2	7	0	
9	3	7	0	
10	4	7	0	
11	5	7	0	
12	0	8	0	
13	1	8	0	
14	2	8	0	
15	3	8	0	
16	4	8	0	
17	5	8	0	
18	0	9	0	
19	1	9	0	
20	2	9	0	
21	3	9	0	
22	4	9	0	
23	5	9	0	

Index	Source	Dest.	Delay	Value
WI	SRC	DST	TD	W
24	0	10	0	
25	6	10	0	
26	7	10	0	
27	8	10	0	
28	9	10	0	
29	0	11	0	
30	6	11	0	
31	7	11	0	
32	8	11	0	
33	9	11	0	
34	0	12	0	
35	6	12	0	
36	7	12	0	
37	8	12	0	
38	9	12	0	

Štruktúra neurónov

Index	First W.	Last W.	Type	Act. F.	Act.
UI	FW	LW	UT	AF	O
0			T		
1			I		
2			I		
3			I		
4			I		
5			I		
6	0	5	H	SGM	
7	6	11	H	SGM	
8	12	17	H	SGM	
9	18	23	H	SGM	
10	24	28	O	LIN	
11	29	33	O	LIN	
12	34	38	O	LIN	

Dopredné šírenie

- od neurónov s nižším indexom k vyšším, pre každý neurón i

$\tilde{o}_i=\sum_{j=FW(i)}^{LW(i)}w_j o_{SRC(j)}$

$o_i=f(\tilde{o}_i)$

Spätné šírenie

$\Delta w_i=\alpha\delta_{DST(i)}o_{SRC(i)}$

$\delta_i=f'(\tilde{o}_i)\left[(d_i-o_i)+\sum_{j=i+1}^{nu}\delta_j w_{FindBySrcDest(i,j)}\right]$

- inicializujeme δ_i na (d_i-o_i) pre všetky výstupné neuróny a ak je žel. výstup def.
- od neurónov s vyšším indexom k nižším, pre každý neurón i
  - dopočítame $\delta_i:=f'(\tilde{o}_i)\delta_i$
  - cez všetky jeho váhy j od $LW(i)$ až po $FW(i)$ prešírili chybový signál neurónu $\delta_{SRC(j)}:=\delta_i w_j$

Zápis algoritmu v pseudojazyku

Dátové štruktúry a premenné

`NW`	- počet váhových prepojení
`wSource[0..NW-1]`	- indexy zdrojových neurónov
`wDest[0..NW-1]`	- indexy cieľových neurónov
`wValue[0..NW-1]`	- hodnoty váhových prepojení
`NU`	- počet neurónov (aj vstupné neuróny, aj špec. neurón pre prahové prepojenia)
`uFirstWeight[0..NU-1]`	- indexy prvých váhových prepojení pre daný neurón
`uLastWeight[0..NU-1]`	- indexy posledných váhových prepojení pre daný neurón
`uType[0..NU-1]`	- typy neurónov (`TRESHOLD, INPUT, HIDDEN, OUTPUT`)
`ACT[0..NU-1]`	- aktivity neurónov
`ACTD[0..NU-1]`	- derivácie aktivít neurónov
`DE_DNA[0..NU-1]`	- chybové spätne šírené signály
`DLT_W[0..NW-1]`	- zmeny hodnôt váhových prepojení
`Sgm(nact)`	- aktivačná funkcia, vstup `nact` - „net output“ („net activity“)
`SgmDer(x)`	- derivácia aktivačnej funkcie
`Input(ui)`	- vráti hodnotu vstupu do neurónovej sieti pre zadaný vstupný neurón `ui`
`Output(ui, act)`	- nastaví výstupnú hodnotu z neurónovej siete na výstupnom neuróne `ui` na `act`
`Target(ui)`	- vráti želanú hodnotu výstupu z neurónovej siete na výstupnom neuróne `ui`
`alfa, beta`	- rýchlosť učenia, momentum

Inicializácia

- inicializacia štruktúry siete, napr. podľa strany 19
  (`NW=39`, `NU=13`, polia `wSource` až `uType`)
- vynulovanie poľa zmien váh (`for wi=0 to NW-1 do DLT_W[wi] := 0.0`)

Dopredné šírenie

```
for ui=0 to NU-1 do
begin
    if uType[ui] = TRASHOLD then ACT[ui] := 1.0;
    else if uType[ui] = INPUT then ACT[ui] := Input(ui);
    else
    begin
        nact := 0.0;
        for wi := uFirstWeight[ui] to uLastWeight[ui] do
            nact := nact + wValue[wi]*ACT[wSource[wi]];
        ACT[ui] := Sgm(nact);
        ACTD[ui] := SgmDer(nact);
    end;
    if uType = OUTPUT then Output(ui, ACT[ui]);
end;
```


Spätné šírenie

```
for ui=NU-1 downto 0 do DE_DNA[ui] := 0.0;

for ui=NU-1 downto 0 do
begin
    if uType[ui]=INPUT then break;
    if uType[ui]=OUTPUT then DE_DNA[ui]:=DE_DNA[ui] + (Target(ui)-ACT[ui]);
    DE_DNA[ui]:=DE_DNA[ui]*ACTD[ui];
    for wi:=uLastWeight[ui] downto uFirstWeight[ui] do
        DE_DNA[wSource[wi]]:=DE_DNA[wSource[wi]] + wValue[wi]*DE_DNA[ui];
end;
```

Zmena váh

```
for wi:=0 to NW-1 do
begin
    DLT_W[wi] := beta*DLT_W[wi] + alfa*DE_DNA[wDest[wi]]*ACT[wSource[wi]];
    wValue[wi] := wValue[wi] + DLT_W[wi];
end;
```


Poznámky k algoritmu

- dopredné šírenie
  - výpočet `ACTD[ui]` (derivácie akt. funkcie) iba ak bude prebiehať adaptácia váh, nemá zmysel vo fáze používania siete

- spätné šírenie
  - šírenie chybového signálu aj na vstupné neuróny nemá zmysel, nutná optimalizácia algoritmu a ukončenie šírenia pre váhu zo vstupu napr.:

```
    …
    …
    DE_DNA[ui]:=DE_DNA[ui]*ACTD[ui];
    for wi:=uLastWeight[ui] downto uFirstWeight[ui] do
    begin
        if uType[wSource[wi]] = INPUT then break;
        DE_DNA[wSource[wi]]:=DE_DNA[wSource[wi]] + wValue[wi]*DE_DNA[ui];
    end;
end;
```