Kunskapsbaserade system, tentamen 2000-03-10
Institutionen för teknik
PBN

Tentamen i Kunskapsbaserade system, 5p, Data 3

Datum: 2000-03-10 Tid: 8.00 - 13.00 Lärare: Pontus Bergsten, 3365

Hjälpmedel: Miniräknare

Uppgifterna ska lösas på separat papper. Skriv namn och personnummer på samtliga inlämnade blad. Skriv läsligt och rita tydliga figurer. Alla eventuellt egna införda variabler skall tydligt förklaras. Otydlig eller tvetydig lösning ger poängavdrag.

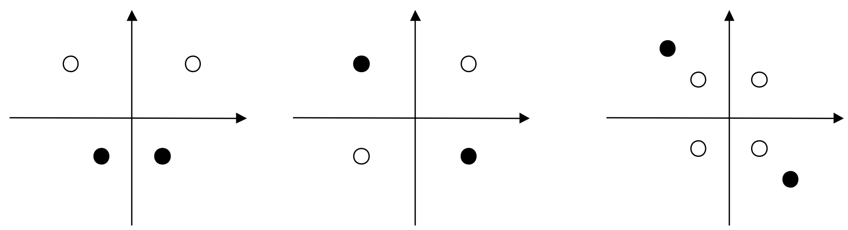
---

1 a) Gör en jämförelse mellan en biologisk neuron och en artificiell neuron. (2)

b) Ge exempel på två fundamentala skillnader mellan algoritmisk problemlösning och problemlösning baserad på neurala nätverk. (1)

---

2 a) Beskriv det *minsta* artificiella neurala nät bestående av perceptroner (bipolär aktiveringsfunktion) som kan klara av att skilja på de två klasserna nedan i de olika fallen nedan (Klass 1 - fylld, Klass 2 - ofylld. Ange ett nät för varje fall (3 st) nedan . Du behöver inte räkna ut några vikter.



(1.5)

---

3 a) Härled Widrow-Hoffs träningsregel för en Adaline. (Anta kvadratisk felfunktion, $e = (d - y)^2$, där d är önskad utsignal och y är aktuell utsignal). (2)

b) Anta att ett antal träningsvektorer finns tillgängliga för träning av en Adaline: $\{(\mathbf{x}^1, t^1), (\mathbf{x}^2, t^2), \ldots, (\mathbf{x}^p, t^p)\}$.Visa hur dessa kan användas för att bestämma vikterna genom att använda pseudoinversen av en matris. (1)

---

4 a) Redogör för hur adaptiv ekoutsläckning vid telefoni kan göras med en Adaline. En bild över hur signalerna överförs skall visas. Redogör också för exakt vad varje signal representerar såsom: eko, störd signal, filtrerad signal och fel. (4)

b) Visa med en detaljerad bild hur en Adaline kan användas för adaptiv systemidentifiering genom att förutsäga systemets utsignal $y(t)$, med hjälp av de tre tidigare utsignalerna $y(t\text{-}1)$, $y(t\text{-}2)$ och $y(t\text{-}3)$? (1)

---

5 Visa att ett artificiellt neuralt nät med *flera* lager bestående av linjära neuroner inte är kraftfullare en ett nät bestående av *ett* lager med linjära neuroner. (2)

---

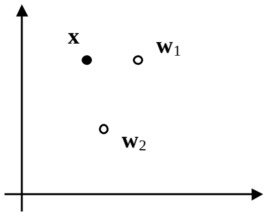
6 a) Beskriv hur proceduren *early-stopping* används för att motverka överträning av ett feed-foward nät. (1)

b) Beskriv proceduren hur *hold-one-out* går till vid träning av ett neuralt nät samt vilken information metoden ger om den valda nätverks konfiguration. (2)

c) Beskriv den utökade felfunktionen som används för träning av ett feed-forward nät med regularisering. Beskriv också det speciella fallet för träning med *weight decay*. (2)

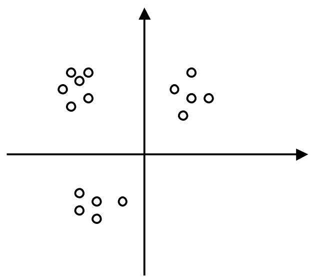
---

7 a) Ett självorganiserande nät med två neuroner skall tränas med competitive learning (winner-takes-it-all) till att utföra vektorkvantisering. I bilden nedan visas ett steg i träningen när träningsvektorn **x** presenteras för nätet. Vilken neuron vinner och hur kommer dess viktvektor (**w**) att uppdateras?



(2)

b) Visa hur den troliga Voronoi mosaiken (tesselation) beskriven av ett enlagers nät med 3 neuroner kommer att se ut efter att nedanstående datamängd har använts för competitive learning (winner-takes-it-all) av nätet.



(1)

---

8 a) Designa en minsta distans klassificerare för tre klasser med följande träningsvektorer:

Klass 1: $\begin{bmatrix} -1 \\ -0.5 \end{bmatrix}, \begin{bmatrix} -1 \\ -1.5 \end{bmatrix}$ Klass 2: $\begin{bmatrix} 1 \\ 0.5 \end{bmatrix}, \begin{bmatrix} 1 \\ -0.5 \end{bmatrix}$ Klass 3: $\begin{bmatrix} -1 \\ 0.5 \end{bmatrix}, \begin{bmatrix} -1 \\ 1.5 \end{bmatrix}$

Klassificera sedan vektorn $[0.5\ -1]^T$ med din designade klassificerare. Ta också fram den resulterande beslutslinjen mellan Klass 2 och Klass 3. (2)

b) Ge den generella beslutsfunktionen för en optimal klassificerare med 0-1 förlust funktion? (1)

---

9 a) Beskriv utförligt följande begrepp som uppkommer i samband med genetiska algoritmer: urval, mutation, crossover och schema. (3)

b) Hur skulle du kunna använda en genetisk algoritm för träning av ett neuralt nät? Beskriv också hur du skulle koda kromosomerna och vilken fitness funktion du skulle använda. (1.5)

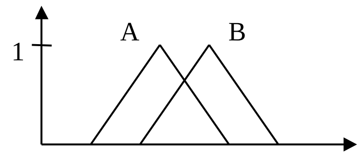
---

10 Vad blir den resulterande fuzzymängden från fuzzy påståendena p_1 och p_2 (en mängd för varje påstående, använd lämplig t-norm resp. t-conorm)

p_1: X is A and X is B

p_2: X is A or X is B

Fuzzymängderna A och B är definierade som enligt figur nedan. Rita om figurerna på eget papper för de två olika påståendena.



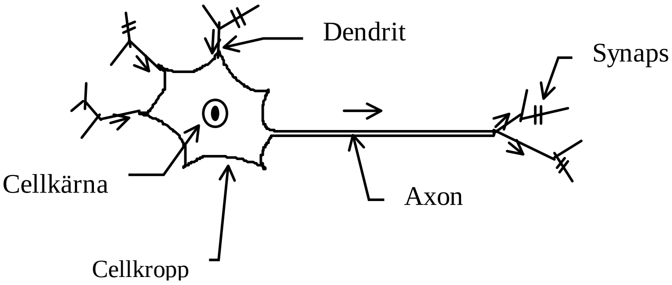
(2)

Kunskapsbaserade system, lösningar till omtentamen 2000-03-10
Observera att lösningarna kan vara kortfattade.

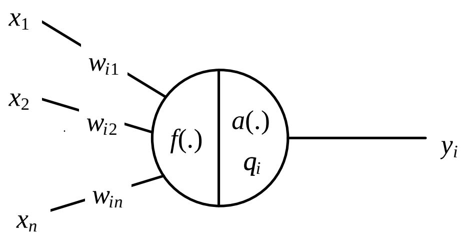
---

1 a)

Biologisk neuron



Neuronmodell



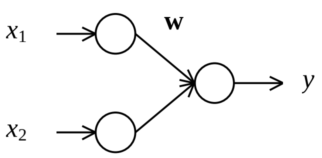
Biologisk neuron	Neuronmodell
Dendriter - nervbanor in till neuronen	Kopplingar mellan neuronerna, inputs x_1, x_2, …
Synapser - bestämmer kopplingens styrka,stimulerande eller hämmande	Motsvaras av vikterna, w_{ij}
Ackumulering av spänning i neuronen	Integreringsfunktionen f
Tröskel att överstiga innan avfyrning av signal genom axonen sker	Tröskeln q_i
Form och amplitud på pulsen som avfyras	Aktiveringsfunktionen a

b) Inlärning- ett neuralt nät lär sig genom exempel, en algoritmisk metod för att lösa ett problem måste ha exakt definition av problem. Parallellitet - ett neuralt nät är parallellt till sin struktur. Att parallellisera en sekvensiell algoritm kan vara mycket svårt.

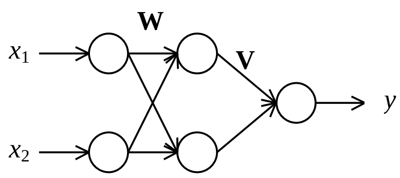
---

2 Figur från vänster till höger (första neuronlagret är ett buffertlager och har ingen egentlig funktion)

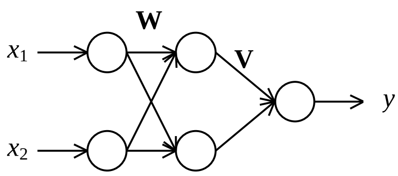
1: Linjärt separabelt



2: Ej linjärt separabelt, två linjer krävs i figuren



3: Ej linjärt separabelt, två linjer krävs i figuren



---

3 a) Den iterativa varianten kan tas fram genom att derivera felfunktionen för en presentation av data.

Utsignal: $$y = \mathbf{w}^T \mathbf{x} = \sum_{i=1}^{n} w_i x_i$$

Felfunktion: $$e = \left(d - \sum_{i=1}^{n} w_i x_i \right)^2$$, d är önskad utsignal för **x**.

Derivering m.a.p. w_i: $$\frac{\partial e}{\partial w_i} = -2\left(d - \sum_{i=1}^{n} w_i x_i \right) x_i$$

Uppdatering i derivatans negativa riktning med en inlärnings konstant:

$$w_i^{new} = w_i^{old} + h2\left(d - \sum_{i=1}^{n} w_i x_i \right) x_i$$

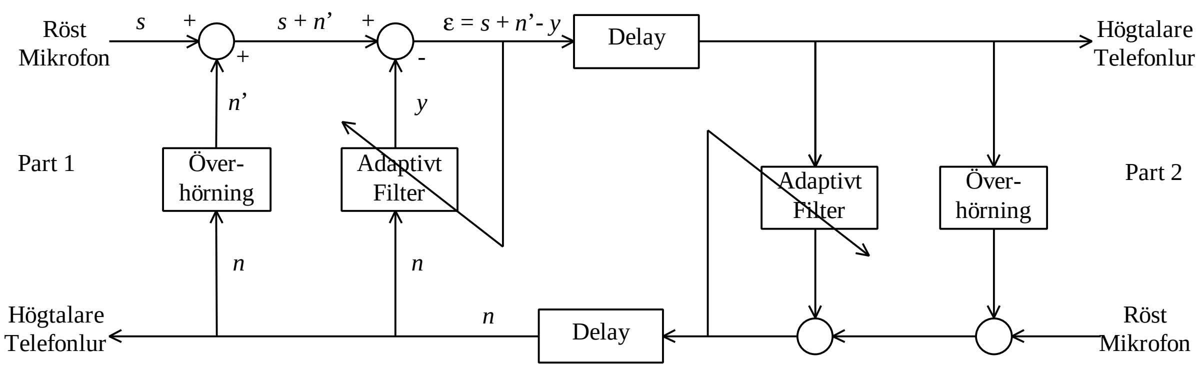
b) För *en* Adaline kan sambandet mellan träningsvektorer och utdata skrivas som

$$\mathbf{Xw} = \mathbf{t} \text{ där } \mathbf{X} = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_n^1 \\ x_1^2 & x_2^2 & \dots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^p & x_2^p & \dots & x_n^p \end{bmatrix}, \ \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} \text{ och } \mathbf{t} = \begin{bmatrix} t^1 \\ t^2 \\ \vdots \\ t^p \end{bmatrix}.$$

Vikterna, **w**, blir lösningen av ekvationssystemet enligt: $\mathbf{w} = \tilde{\mathbf{X}}^{-1}\mathbf{t}$, där $\tilde{\mathbf{X}}^{-1}$ är pseudoinversen av **X**.

4

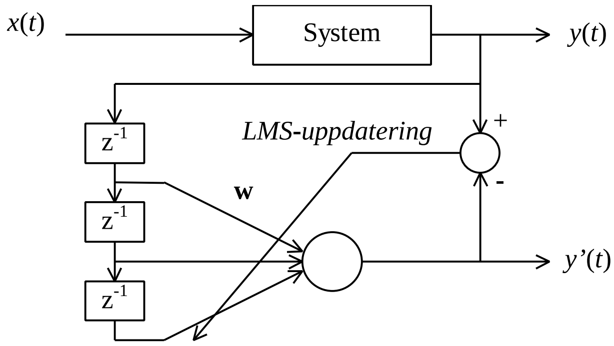
a)



s	- Part 1's tal, original signal som part 2 höra).
n'	- Eko (dämpad version av part 2's tal)
$s+n'$	- Störd signal (part 1 + part 2)
y	- Utsignal från adaline
ε	- felet mellan adalines utsignal och den störda signalen, detta är också den filtrerade signalen.

---

4 b)



---

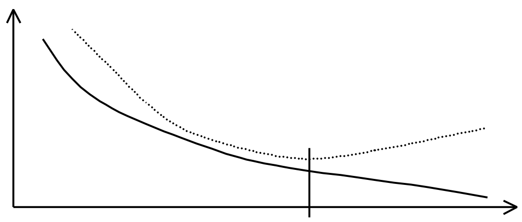
5 Ex. ett gömt lager:

Gömda lagret: **y** = **Wx**
Utlagret: **z** = **Vy** = **VWx** = **Qx**

z kan alltså fås genom transformation med en matris, alltså ekvivalent med ett lager.

---

6 a) *Early stopping*. Två datamängder används vid träningen, en träningsmängd och en testmängd. Träningsmängden används vid träningen (med tex backprop). Totalt kvadratsumma fel beräknas för tränings resp. test mängd efter varje epok. En graf över kvadratsummafelet som funktion av antal epoker ser då sannolikt ut som följer (heldragen – träningsmängd, streckad – testmängd).



Det är uppenbart att fortsättning av träningen när testmängdens fel ökar ger ett övertränat nät.

b) Hold-one out träning används för att skaffa information om en klassificerares generaliseringsförmåga. Klassificeraren tränas med hela träningsmängden utom en vektor. När träningen är klar klassificeras den vektor som hölls utanför. Proceduren görs om för varje vektor i träningsmängden. Om varje vektor som hållits utanför kan klassificeras korrekt så har man fått information om att nätets generaliserings-förmåga och att datamängden konsistens förmodligen är OK

c) Utökad felfunktion: E = E' + $\nu\Omega$. Där E' är den vanliga felfunktionen, ν är en liten konstant och Ω är en regulariseringsfunktion. Det har visat sig att stora värden på vikterna i ett FF-nät kan försämra generaliseringen. Regulariseringsfunktionen för träning med *weight decay* kan definieras som $\Omega = \frac{1}{2}\sum_{i=1}^{N}(w_i)^2$, alltså bestraffning av stora värden på vikterna (weight decay).

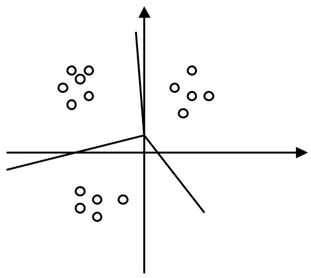
---

7 a) Vikt vektor w1 som ligger närmast x kommer att vinna och får därmed uppdatera enligt

$\mathbf{w}_1 = \mathbf{w}_1 + \alpha(\mathbf{x} - \mathbf{w}_1)$

$\mathbf{w}_1$ flyttas alltså i riktning mot $\mathbf{x}$.

b) Winner-takes-it-all algoritmen delar in rummet optimalt i celler (Voronoi mosaik) och skulle kunna se ut som



---

8 a) Designa klassificeraren genom att räkna ut medelvektorerna för varje klass:

$\mathbf{m}_1 = [-1\ -1]^T$, $\mathbf{m}_2 = [1\ 0]^T$, $\mathbf{m}_3 = [-1\ 1]^T$. Beslutsfunktionen är: $d_i(\mathbf{x}) = \mathbf{x}^T\mathbf{m}_i - 0.5\mathbf{m}_i^T\mathbf{m}_i$.

Klassning av $\mathbf{x}_{test}=[0.5\ -1]^T$: $d_1(\mathbf{x}_{test}) = [0.5\ -1][-1\ -1]^T - 0.5[-1\ -1][\ -1\ -1]^T = -0.5$
$d_2(\mathbf{x}_{test}) = [0.5\ -1][1\ 0]^T - 0.5[1\ 0][\ 1\ 0]^T = 0$
$d_3(\mathbf{x}_{test}) = [0.5\ -1][-1\ 1]^T - 0.5[-1\ 1][\ -1\ 1]^T = -2.5$

d_2 gav högst värde $\rightarrow \mathbf{x}_{test} \in$ Klass 2

Beslutslinje mellan klass 2 till 3: Sätt beslutsfunktionerna lika med varandra $d_2(\mathbf{x}) = d_3(\mathbf{x})$:

$$[\mathbf{x}_1\ \mathbf{x}_2][\ 1\ 0]^T - 0.5[1\ 0][\ 1\ 0]^T = [\mathbf{x}_1\ \mathbf{x}_2][\ -1\ 1]^T - 0.5[-1\ 1][\ -1\ 1]^T$$

Multiplicera ihop vektorerna och bryt ut $\mathbf{x}_2$ vilket då definierar beslutslinjen mellan klasserna.

b) Generell beslutsfunktion (för en klass w_j) för en statistiskt optimal klassificerare med 0-1 beslutsfunktion ges av

$d_j(\mathbf{x}) = p(\mathbf{x} \mid w_j)P(w_j)$

där $p(\mathbf{x} \mid w_j)$ är värdet på sannolikthetstäthetsfunktionen för $\mathbf{x}$ (mönstervektor) givet klass w_j och $P(w_j)$ är sannolikheten att klass w_j skall uppkomma.

---

9 a) *urval* **-** Avgör vilka individer som skall leva vidare till nästa generation i algoritmen. Ju större fitness desto större chans att leva vidare.

mutation – En bit i kromosomen ändrar slumpartat sitt värde. Om binär kodning används negeras biten. Mutation kan förhindra att algoritmen fastnar i ett lokalt minima.

crossover – Två kromosomer byter gener med varandra i hopp om att avkomman skall få ett bättre fitness värde, ex med crossover punkt efter fjärde genen:

```
0101|011100   ger   0101101010
1111|101010         1111011100
```

schema – Ett schema är en mängd kromosomer som delar vissa egenskaper, t.ex. representerar $\delta_1\delta_2{*}\,{*}\,\delta_3$ ett schema där δ_1, δ_2 and δ_3 har fasta värden men * kan vara vad som helst. En kromosom tillhör detta schema om den har gener som överensstämmer exakt med δ_1, δ_2 and δ_3.

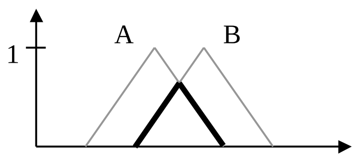
b) Anta att strukturen på nätet är fast med ett visst antal neuroner och kopplingar. En speciell nätverkskonfigurationen ger då ett fast antal vikter. Dessa vikter skulle kunna bestämmas genom att träna nätet med en genetisk algoritm. Om nätet har *N* vikter kan kromosomerna för de olika individerna i den genetiska algoritmen kodas som

Individ 1:	w_1	w_2	w_3	…	w_N
Individ 2:	w_1	w_2	w_3	…	w_N

…

Varje individ representerar alltså alla vikter i ett neuralt nät. Varje vikt i varje kromosom kan t.ex. kodas som ett binärtal med fixpunkts kodning som spänner inom ett lämpligt intervall (som bör bero på hur överförings funktionerna ser ut i de olika neuronerna). En lämplig fitnessfunktion skulle kunna vara kvadratsummafelet över alla träningsdata, men negerat.

10 p_1: t-norm = min



p_2: t-conorm = max