

2D1432

## Artificiella Neuronnät och andra lärande system

Lösningsförslag till Tentamen 2003-03-06

**Inga hjälpmedel.**

---

### Uppgift 1

(4p)

Vilka av följande påståenden är sanna? Korrigera de som är fel.

1. Potentialen i en nervcell definieras normalt som skillnaden i spänning mellan dess axon och dendrit.

*Fel. Potentialen man talar om är den mellan membranets insida och utsida.*

2. I en synaps överförs signalerna med hjälp av kemiska signalsubstanser.

*Rätt*

3. En aktionspotential varar normalt under några få minuter.

*Fel. Den varar några få millisekunder.*

4. En nervcell har normalt ett långt axon och flera kortare dendriter.

*Rätt*

### Uppgift 2

(3p)

Vilken eller vilka av följande tre avbildningar kan läras av en enlagersperceptron? Motivera ditt svar!

<b>a)</b>	$000 \rightarrow 1$	<b>b)</b>	$000 \rightarrow 0$	<b>c)</b>	$000 \rightarrow 1$
	$001 \rightarrow 1$		$001 \rightarrow 0$		$001 \rightarrow 0$
	$010 \rightarrow 0$		$010 \rightarrow 0$		$010 \rightarrow 0$
	$011 \rightarrow 0$		$011 \rightarrow 1$		$011 \rightarrow 0$
	$100 \rightarrow 1$		$100 \rightarrow 0$		$100 \rightarrow 0$
	$101 \rightarrow 1$		$101 \rightarrow 1$		$101 \rightarrow 0$
	$110 \rightarrow 0$		$110 \rightarrow 1$		$110 \rightarrow 0$
	$111 \rightarrow 0$		$111 \rightarrow 1$		$111 \rightarrow 1$

*a och b kan lösas, däremot inte c.*

### Uppgift 3

(4p)

*Perceptroninlärning* är en inlärningsalgoritm för enlagersnät. Beskriv hur algoritmen fungerar i fallet när man har fyra inoder och en utnod. Ange speciellt när och hur mycket vikterna ändras samt under vilka förutsättningar algoritmen konvergerar. Vad är skillnaden mot *delta-regeln*?

*Vikterna ändras endast när nätet svarar fel. Om det svarar 0 men borde svarat 1 ändras vikterna enligt  $\Delta\bar{w} = \eta\bar{x}$ . Om det svarar 1 men borde svarat 0 ändras vikterna enligt  $\Delta\bar{w} = -\eta\bar{x}$ . Konvergens får man med en ändlig linjärt separerbar träningsmängd. Delta-regeln minimerar felet innan trösklingen.*

### Uppgift 4

(3p)

Vid inlärning med hjälp av *backprop*-tekniken gör man dels ett pass framåt, dels ett pass bakåt genom nätet. Beskriv (formler krävs ej) vad det är man beräknar under vart och ett av dessa pass. När sker själva viktuppdateringen?

*Under framåtpasset beräknas aktiveringen av alla noder. Under bakåtpasset distribueras det generaliserade felet. Efter eller under bakåtpasset uppdateras vikterna.*

### Uppgift 5

(3p)

Vad menas med ett nätverks förmåga att *generalisera*? Hur påverkas normalt denna förmåga om antalet vikter i nätet ändras? Hur kan man i praktiken avgöra ifall ett nät generaliserar bra eller ej?

*Förmågan att ge lika bra svar för andra datapunkter än de som ingår i träningsmängden. Många vikter (frihetsgrader) ger sämre generalisering. Generaliseringsförmågan kan mätas med en separat testmängd.*

### Uppgift 6

(3p)

Vid användningen av *generaliserade deltaregeln* (backprop) kan man råka ut för att nätet inte lär sig att lösa uppgiften trots att det existerar en lösning. Vad beror detta på? Vad kan man göra åt det?

*Man har troligen hamnat i ett lokalt minimum. Enklaste åtgärden är att köra om inlärningen med ny startpunkt.*

### Uppgift 7

(3p)

För *radialbasnät* använder man ofta helt olika inlärningsmetoder för de olika lagren. Ge exempel på lämpliga inlärningsalgoritmer för de två lagren.

Första lagret (RBF'ernas position) tränas t.ex. med kompetitiv inläring. Andra lagret (kopplingar till utnoderna) tränas lämpligen med deltaregeln.

## Uppgift 8

(3p)

Antag att man vill lagra 30 slumpmässiga mönster i ett *Hopfield-nät*. Hur stort bör nätet vara för att man ska vara ganska säker på att lagringen lyckas? Motivera ditt svar. Vad kommer att hända ifall nätet är för litet?

*Det finns flera tumregler för slumpmässiga mönster. En säger att man kan lagra  $0.15 \times N$  vilket ger att vi behöver minst 200 noder. Ifall nätet är för litet så blir det för få attraktorer vilket gör att mönstren kommer att blandas ihop.*

## Uppgift 9

(3p)

Vad menas med *dead units*? I vilka typer av nät uppträder de? Under vilka förutsättningar? Vad kan man göra för att undvika dem?

*Det är noder vid kompetitiv inläring som råkar hamna så illa att de aldrig vinner och därför heller aldrig flyttas. Det finns många sätt att fixa detta: leaky learning, grannskap, m.m.*

## Uppgift 10

(3p)

Beskriv hur man kan använda ett nätverk för att få en reduktion av antalet dimensioner i indata som bevarar så mycket som möjligt av variansen. Hur är nätet uppbyggt? Hur ställer man in vikterna?

*Använd ett PCA (Principal Component Analysis) nätverk. Det är ett linjärt enlayersnät som tränas enligt  $\Delta W = \eta \bar{y}^T (\bar{x} - \bar{y}^T W)$ . Datapunkterna kommer av det färdiglärda nätet att projiceras på det underrum där variansen är som störst.*

## Uppgift 11

(3p)

Vad menas med termen *resonans* i Adaptive Resonance Theory (ART)? Hur påverkar den inläringen?

*Resonans innebär att indata är tillräckligt lik den prototyp (nod i utlagret) som för tillfället är aktiv. Inläring sker endast vid resonans och då ändras prototypens vikter så att den bättre matchar det visade mönstret.*

## Uppgift 12

(4p)

Antag att man vill använda en *genetisk algoritm* för att få fram ett bra schema för en skola. Beskriv vad dessa termer skulle få för konkret betydelse i denna tillämpning.

1. Population

*En mängd olika scheman som är de aktuella kandidaterna till den slutliga lösningen.*

2. Fitness-funktion

*En funktion som beräknar hur bra ett visst schema är. Omöjligheter bör bestraffas hårt, obekvämligheter mindre hårt.*

3. Kromosom

*Ett schema.*

4. Korsning (crossover)

*En operation där två scheman byter delar med varandra för att få fram nya scheman.*

## Uppgift 13

(5p)

Det vanliga periodiska systemet för grundämnena är en tvådimensionell "karta" där ämnen med liknande kemiska egenskaper ligger nära varandra. Antag att man behöver en liknande karta över ett antal kemiska föreningar och att du kommit på den lysande idén att man borde kunna använda SOFM (self-organizing feature maps) för att automatiskt skapa denna karta.

Beskriv hur du ska gå tillväga för att åstadkomma detta. Svara specifikt på följande frågor: Vad utgör lämpliga indata? Vad använder du för träningsmönster? Hur bör nätet vara uppbyggt (antal noder, lagerstruktur, antal kopplingar o.s.v.)? Hur genomför du inlärningen? Hur får du fram den slutliga kartan?

*Lämpliga indata är mönster, ett per ämne, sammansatta av de egenskaper som man vill grupper efter.*

*Alla dessa mönster utgör träningsdata.*

*Nätet har ett inlager med en nod per egenskap och ett utlager med gott om noder (absolut minst ett per ämne). Alla noder i inlagret är kopplade med justerbara vikter till alla noder i utlagret. Utlagret är organiserat i ett 2D rutmönster som definierar grannskapet i SOFM-algoritmen.*

*Inlärningen sker genom att ett mönster i taget visas. Vinnaren i utlagret identifieras och den och dess grannar justeras närmare mönstret. Man bör ha ett stort grannskap i början och minska det efterhand.*

*När inlärningen är över visar man ett mönster i taget och placerar motsvarande ämne på den plats där vinnarnoden finns. Detta gör att alla ämnen placeras i rutmönstret som nu utgör den slutliga kartan.*

## Uppgift 14

(3p)

Beskriv hur *belöningsfunktionen* bör vara definierad när man vill använda reinforcement learning i följande situationer:

1. ett styrsystem för en robot som ska lära sig att gå  
–1 när den ramlar, 0 annars.
2. ett program som ska spela luffarschack  
1 när den vinner, –1 när den förlorar, 0 annars.
3. ett program som ska hitta snabbaste vägen ut ur en labyrint  
–1 medan den är kvar i labyrinten, 0 när den kommer ut.

## Uppgift 15

(3p)

Antag att vi använder *kompetitiv inlärning* med tre prototypvektorer som vid ett viss tillfälle har följande värden:

$$\begin{aligned} &(0.2, -0.8, -0.8) \\ &(0.7, 0.65, -0.3) \\ &(0.7, -0.2, -0.7) \end{aligned}$$

Beskriv hur vikterna ändras när ett nytt indatamönster  $(0.6, 0.6, -0.6)$  presenteras för nätet.

*Indatamönstret är närmast prototypvektor 2 (vanlig euklidisk norm).*

*Vektor 2 ändras därför enligt*

$$\Delta \bar{w}_2 = \eta[0.6 - 0.7, 0.6 - 0.65, -0.6 - -0.3].$$

*De andra ändras inte (Winner-takes-all).*