



Översikt

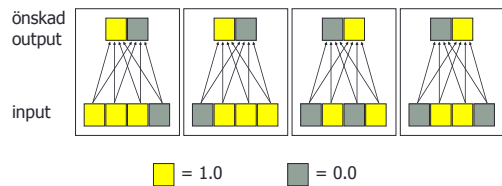
- Hebbiansk inlärning (CPCA) kan inte lära sig vad som helst
- Deltaregeln och backpropagation skulle vara en lösning, men...
 - Ej biologiskt trovärdig
- Biologiskt plausibel motsvarighet till backprop (CHL)
- Kombination av oövervakad och övervakad inlärning

Återblick

- CPCA + KWTa
 - CPCA = Conditional Principal Component Analysis (oövervakad inlärning)
 - Kategoriserar input, bygger modell av omgivningen
 - Arbetsfördelning genom självorganisering mha inbördes tävlan (KWTa)
- Men, om endast CPCA
 - Inlärningen (de slutliga vikterna) bestäms av korrelationer i indata
 - Kan inte lära sig godtyckliga funktioner

Exempel

Träningsdata utgörs av 4 input-output händelser (events)



Styrning av aktivering i nätet

- Input och output presenteras tillsammans
- Alla input-output-par presenteras, sedan en gång till, en gång till... (dvs. flera epoker)
 - Brukar presenteras i olika ordning för varje epok
- Aktivering i nätet får svänga in sig
 - Styrts av dessa yttre krav på sig, dvs. input och output-värdena ('hard clamped')
 - Vid fler än två lager måste finnas kopplingar från output-lagret till tidigare lager, så att aktivering kan sprida sig bakåt/neråt i nätet
- Korrelationerna som lärs in blir mellan input och output

Testning av nätet

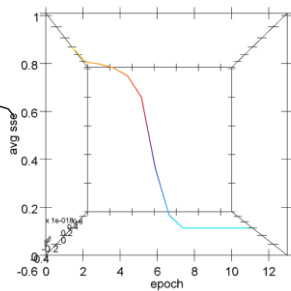
- Kör igenom alla events en gång (dvs. en test-epok)
- Input presenteras i varje event och nätet får själv producera output
 - Output kommer nu att vara beroende av inlärdade vikter

Typisk inlärningskurva

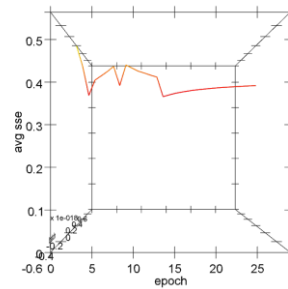
Fel = Diff mellan gissad
och korrekt aktivering i
varje ut-nod

Summed Squared Error
(SSE)

Snitt över alla events, epok
för epok

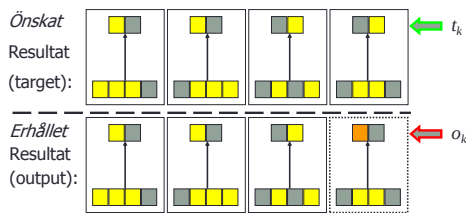


Träningsresultat med CPCA



Hebbiansk "uppgiftsinlärning" (CPCA)

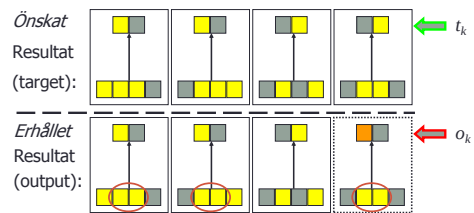
Efter träning:



$$\text{träningsfel} = (t_k - o_k) \neq 0.0$$

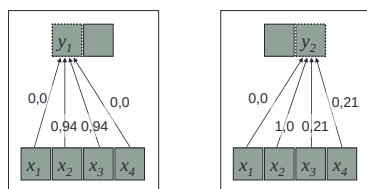
Hebbiansk uppgiftsinlärning (CPCA)

Efter träning:



$$(t_k - o_k) \neq 0.0 = \text{träningsfel}$$

Hebbiansk uppgiftsinlärning



vikterna efter träning (300 epoker)

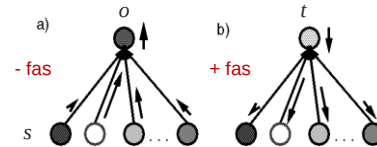
Uppgiftsinlärning

Uppgiftsinläring

- In- och utdata är på förhand kända
- Hitta viktsuppsättning för önskad mappning mellan in- och utdata
 - Oberoende av korrelationer
- Träningsfelet δ (delta)
 - Skillnad mellan önskad output och nätets output för varje event
 - Utnyttjas för att styra viktförändringarna

Två faser

- Gissa output (o) givet inputsignal (s)
- Beräkna felet ($t-o$), och förändra vikterna

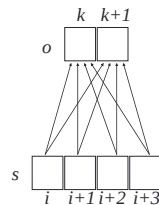


Deltaregeln

$$\Delta w_{ik} = \underbrace{\varepsilon(t_k - o_k)}_{\delta_k} s_i$$

t_k = önskad (target) aktivering
 o_k = erhållen (output) aktivering
 δ_k = felet

Δw_{ik} = viktförändringen
 ε = inlärningsfaktor
 s_i = sändande nodens aktivering



Deltaregeln

$$\Delta w_{ik} = \varepsilon(t_k - o_k)s_i$$

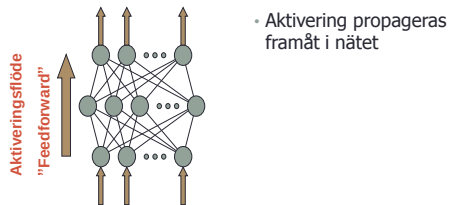
- Minska felet varje gång event:et presenteras för nätet
 - Om ε , δ_k eller $s_i = 0.0 \rightarrow \Delta w_{ik} = 0$
 - Om o_k borde varit större $\rightarrow \Delta w_{ik} > 0$, w_{ik} ökar i *proportion* till hur aktiv s_i var
 - Om o_k borde varit mindre $\rightarrow \Delta w_{ik} < 0$, w_{ik} minskar i *proportion* till hur aktiv s_i var
- 'Credit assignment' (eller blame assignment)
 - Viktförändringarna bestäms av hur mycket respektive sändarnod bidrog till felet

Problemet med deltaregeln

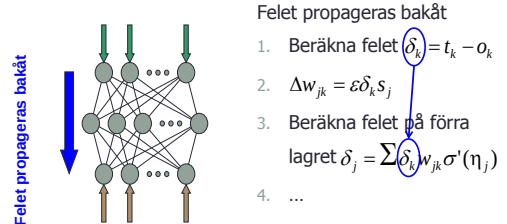
- Deltaregeln kan inte hantera nätverk med fler än två lager
- Dolda lager är dock viktiga
 - För de flesta beräkningsproblem måste minst tre lager finnas
- Måste därför kunna träna de dolda lagren i nätet

Backpropagation of error

Framåtflöde av aktivering



Backpropagation of error



Synaptiska kopplingar

- Antingen excitatoriska eller inhibitoriska
 - Kan inte ändra typ av signalsubstans
 - Förblir excitatorisk el. inhibitorisk
- Enkelriktad
 - Impulsen går endast från den sändande neuronen till mottagaren
 - Ingen information kan förmedlas bakåt

Problemet med backprop

- Bp är inte biologiskt plausibel!
 - Bp kräver att δ -värdet för varje lager propageras (förflyttas) bakåt i nätet
 - Inga neurofysiologiska data ger stöd för Bp
- En biologiskt plausibel variant av Backprop är: the Generalized Recirculation Algorithm (GeneRec)

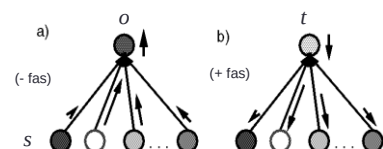
GeneRec

Biologiskt plausibel motsvarighet till backprop

Två faser

Deltaregel förutsätter två faser (beräkningssteg)

- a) Beräkna output (o) givet input (s)
- b) Beräkna felet, och förändra vikterna



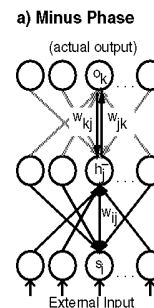
Två faser

- Minus-fas
 - Input presenteras till nätet
 - Nätets aktivering får svänga in sig = gissning av output
- Plus-fas
 - Input och output presenteras tillsammans
 - Nätets aktivering får svänga in sig med dessa yttre krav på sig
- OBS! Måste finnas kopplingar från output-lagret till tidigare lager, så att aktivering kan sprida sig bakåt/neråt i nätet
 - Direkta eller indirekta kopplingar (via andra lager)

GeneRec

Minusfasen

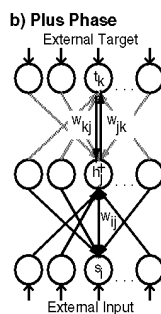
- Extern input ('clamped')
- Uppdatera nätet i cykler tills jämvikt uppnåtts
- Läs av *alla* aktiveringar (**act_m**)
- Alla noder lagrar sina act_m



GeneRec

Plusfasen

- Extern input och korrekt output presenteras
- Nätet körs i cykler tills jämvikt uppnåtts
- Läs av aktiveringarna (**act_p**)
- Varje nod kan nu jämföra sin act_m med sin act_p



GeneRec

- Aktiverings-skillnad mellan de bägge faserna i varje enskild nod utgör felet = $\text{act_p}_{\text{nod}} - \text{act_m}_{\text{nod}}$
- Styr nodens viktuppdatering

OBS! Återkoppling krävs för att korrekta aktiveringar ska kunna spridas i nätet

GeneRec jämfört med deltaregeln

- Deltaregeln: $\Delta w_{ik} = \varepsilon(t_k - o_k)s_i$
- GeneRec: $\Delta w_{ij} = \varepsilon(y_j^+ - y_j^-)x_i^-$

y_j = aktivering av mottagande nod

x_i = aktivering av sändande nod

GeneRec $\leftrightarrow$ CHL

Två justeringar i GeneRec:

- Medelvärdet av *sändamodens* aktivering mellan - och + fasen
- Medelvärdet av fram- och bakåtvikt (se ekv. 5.39)

$$\Delta w_{ij} = \varepsilon(y_j^+ - y_j^-)x_i^- \quad \Rightarrow \quad \Delta w_{ij} = \varepsilon(x_i^+ y_j^+ - x_i^- y_j^-)$$

- Förändrad GeneRec: "Contrastive Hebbian Learning" (CHL)

Uppgiftsinläring: CHL

- Biologisk plausibilitet
 - Ingen mystisk bakåt-kommunikation längs synaptiska kopplingar
 - Aktiveringsbaserad, inget felvärde behöver skickas
 - Vi vet att det finns återkopplade signalvägar i hjärnbarken
 - Inlärningsalgoritmen tenderar att skapa symmetri i vikterna
 - Viktuppdatering från x_i till y_j lika stor från y_j till x_i

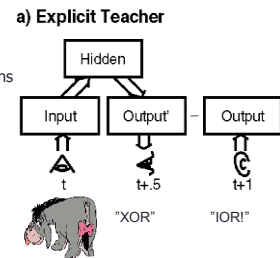
Övervakaren?

Extern feedback

- Vi hör vad vi själva säger
 - När detta auditiva stimulus skiljer sig från vad andra säger (eller vad vi förväntar oss att höra) kommer denna skillnad att fungera som felsignal
- Insikt
 - Språkförståelse → språkproduktion
 - Vi måste först ha hört en massa uttryck, hur de låter, och kopplat dem till olika begrepp
 - Vilket det finns mycket stark evidens för

Vi säger "XOR" och blir korrigerade

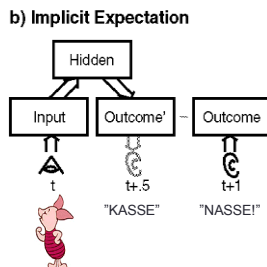
- Vid tidpunkten:
 - t , visuellt stimulus
 - $t+.5$, (– fas) verbal respons (+ auditiv feedback)
 - $t+1$, (+ fas) auditiv stimulus (korrigering)



Vi förväntar oss att höra "KASSE"...

Vid tidpunkten:

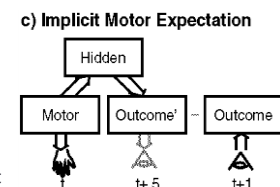
- t , visuellt stimulus
- $t+.5$, *förväntat* auditiv stimulus
- $t+1$, auditiv stimulus
 - Ett visuellt stimulus → ett förväntat auditiv stimulus
 - Skillnaden mellan förväntat och faktiskt stimulus = felsignalen



Vi förväntar oss att handen är här ...

Vid tidpunkten:

- t utförs en motorisk rörelse.
- $t+.5$ (– fas) *förväntat* visuellt stimulus.
- $t+1$ (+ fas) visuellt stimulus.
 - Motorisk handling → ett förväntat visuellt stimulus
 - Skillnaden mellan förväntat och faktiskt stimulus = felsignalen

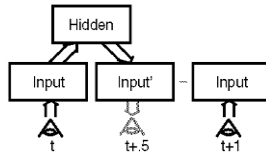


Vi förväntar oss att se X men får se Y

Vid tidpunkten:

- t , visuellt stimulus
- $t+.5$, *förväntat* visuellt stimulus
- $t+1$, visuellt stimulus
 - Ett visuellt stimulus $\rightarrow$ ett förväntat visuellt stimulus
 - Skillnaden mellan förväntat och faktiskt stimulus = felsignalen

d) Implicit Reconstruction



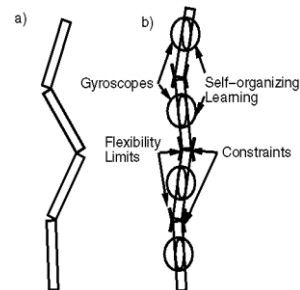
Kombinerad inlärning

Kombinerad inlärning

• Inlärning i två steg:

1. Prova
2. Få extern feedback och justera synaptisk effektivitet (vikter)
 - CPCA och CHL beräknas var för sig
 - Viktuppdatering (CPCA resp. CHL) summeras för varje nod

Fördelar med att kombinera



Uppgiftsinlärning

Icke-lokal: viktförändringarna beror av felsignaler från perifera delar av nätet

Uppgiftsinriktad: felsignalen återspeglar hela nätets prestation och alla vikter kan påverkas för att bättre lösa uppgiften

Samordnande: vikterna anpassas med hänsyn till övriga vikter

Byråkratisk:

1. Problem för algoritmen att bestämma vilka vikter som ska göra vad då alla är beroende av varandra $\rightarrow$
2. Inläringen tar (mycket) lång tid

Modellinlärning

Lokal: viktförändringen baseras på aktivering hos de omedelbart närmaste noderna

Autonom: vikten förändras utan någon felsignal

Tillförlitlig: kommer att extrahera de statistiska regelbundenheterna i indata

Girig: det enda som spelar någon roll är den lokala korrelationen mellan resp. input till en nod

Kortsynt: vikterna förändras oberoende om det är till gagn för den uppgift som nätet är tänkt att lösa eller inte

Kombinerad inlärning

$$\Delta w_{ij} = \varepsilon \left[\overbrace{k_{hebb} (y_j^+ (x_i^+ - w_{ij}))}^{\Delta_{hebb}} + (1 - k_{hebb}) \overbrace{(x_i^+ y_j^+ - x_i^- y_j^-)}^{\Delta_{err}} \right]$$

Δ_{hebb} beräknas i +fasen (andra fasen);
det vore kontraproduktivt att nätet lär sig korrelationer
som är i konflikt med önskad output

K_{hebb}

- Vid 50% Hebb (CPCA) dominerar modellinlärning totalt över uppgiftsinlärning
 - Uppgiften löses förmodligen inte
 - Hebb har svårt att "backa" när den väl har börjat arbetsfördela mellan noderna på fel sätt (dvs. tvärsen mot uppgiften)
- 1% Hebb (CPCA) funkar oftast bra (satt som default)
 - CHL kommer att hitta en korrekt viktuppsättning, om arkitekturen och initiala vikter är adekvata
 - Hebb (CPCA) ser till att vikterna blir tydliga (mot 1 eller 0)

Poängen med Hebb (CPCA)

- CPCA styr upp
 - Underlättar/påskyndar arbetsfördelning i början av inlärningen
 - Ökar nätets generaliseringsförmåga
 - Delar upp input i särdrag som kan omkombineras
 - Nätet kommer att kunna känna igen även helt nya kombinationer av dessa särdrag, dvs. kommer att kunna generalisera till ny input
 - Hjälper även till med icke-kombinatorisk generalisering