Bio-nätverk: Feldriven inlärning

729G83

LINKÖPINGS UNIVERSITET

Översikt

- Hur går feldriven inlärning till
- Begreppet ”fel”
- När och hur beräknas felet
- Kombinerad inlärning
  - Oövervakad + övervakad samtidigt!

Hur går feldriven inlärning till

Minus-fas (gissnings-fas)

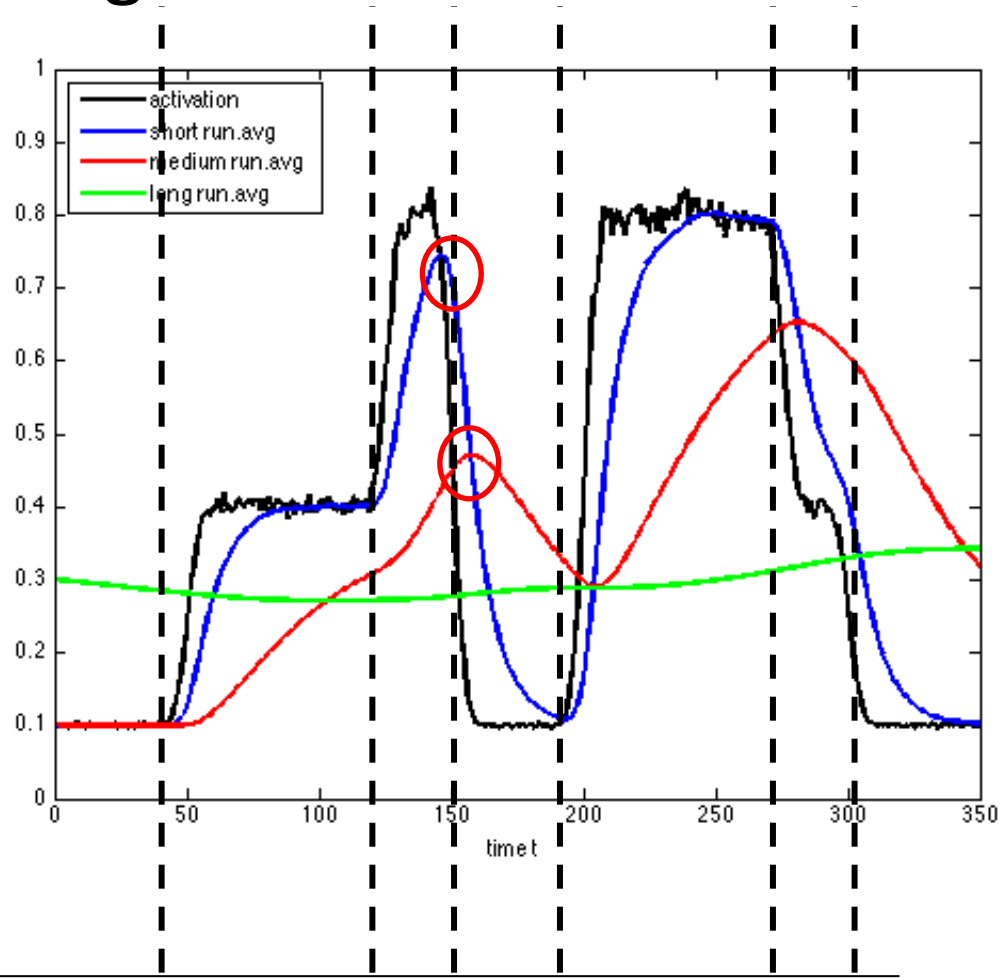
- Nätet presenteras input
- Skickar aktiveringar framåt och bakåt
- Nätet sätter sig i ett stabilt aktiveringsläge
  - Dvs. alla noder i alla lager ”nöjda” med sin aktivering, givet alla andras aktiveringar
- Man läser av ut-aktiveringarna -> ActM

Plus-fas (korrekt svar avslöjas)

- Nätet presenteras input och output
- Skickar aktiveringar framåt och bakåt
- Nätet sätter sig i ett nytt stabilt aktiveringsläge
  - Dvs. alla noder i alla lager ”nöjda” med sin nya aktivering, givet alla andras nya aktiveringar
- Man läser av nya ut-aktiveringarna -> ActP

Vill ha diff mellan två lägen i slutet av faserna

- När nätet får även se korrekt output
  - Sätter sig i aktivering som bättre återspeglar korrekt output

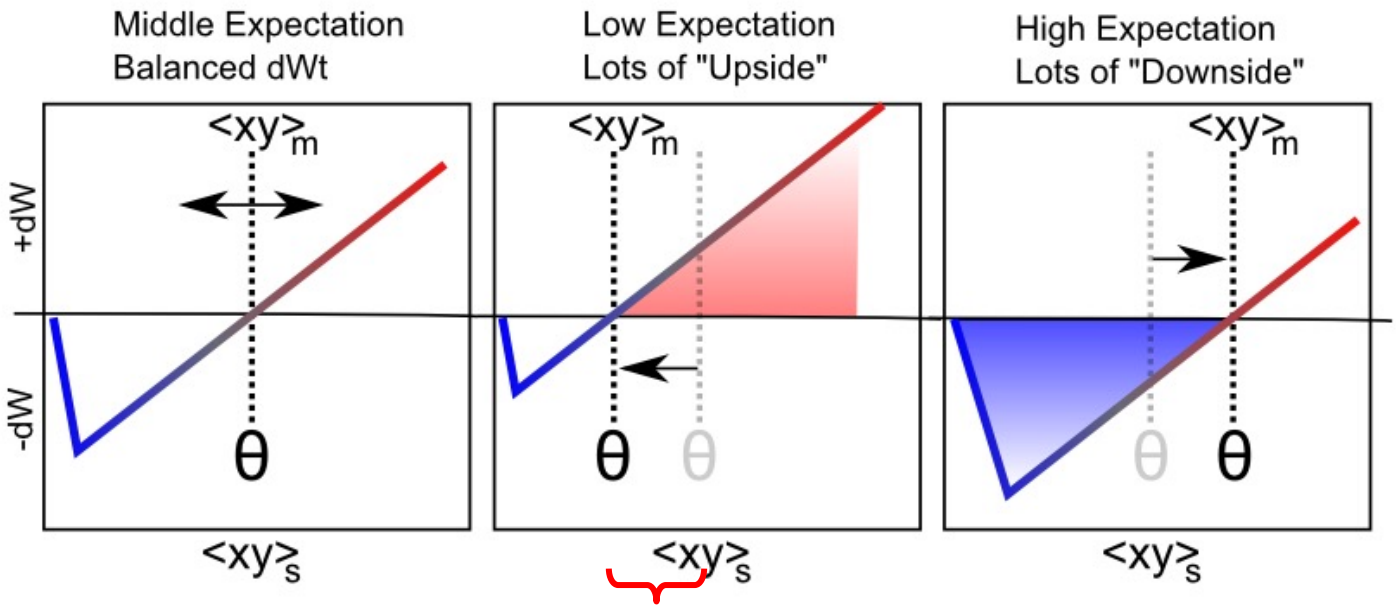


Beräkning av fel

- Varje nod beräknar sin ActP – ActM
- Mer korrekt beskrivet:
  - $\delta \approx \langle x \rangle_s - \langle x \rangle_m$
  - δ är ’lilla delta’ -- delta betecknar alltid diff av något slag, i detta fall felet i varje nod

Justering av vikter

- Enligt XCAL, dvs.
  - $\Delta w \approx \langle xy \rangle_s - \langle xy \rangle_m$

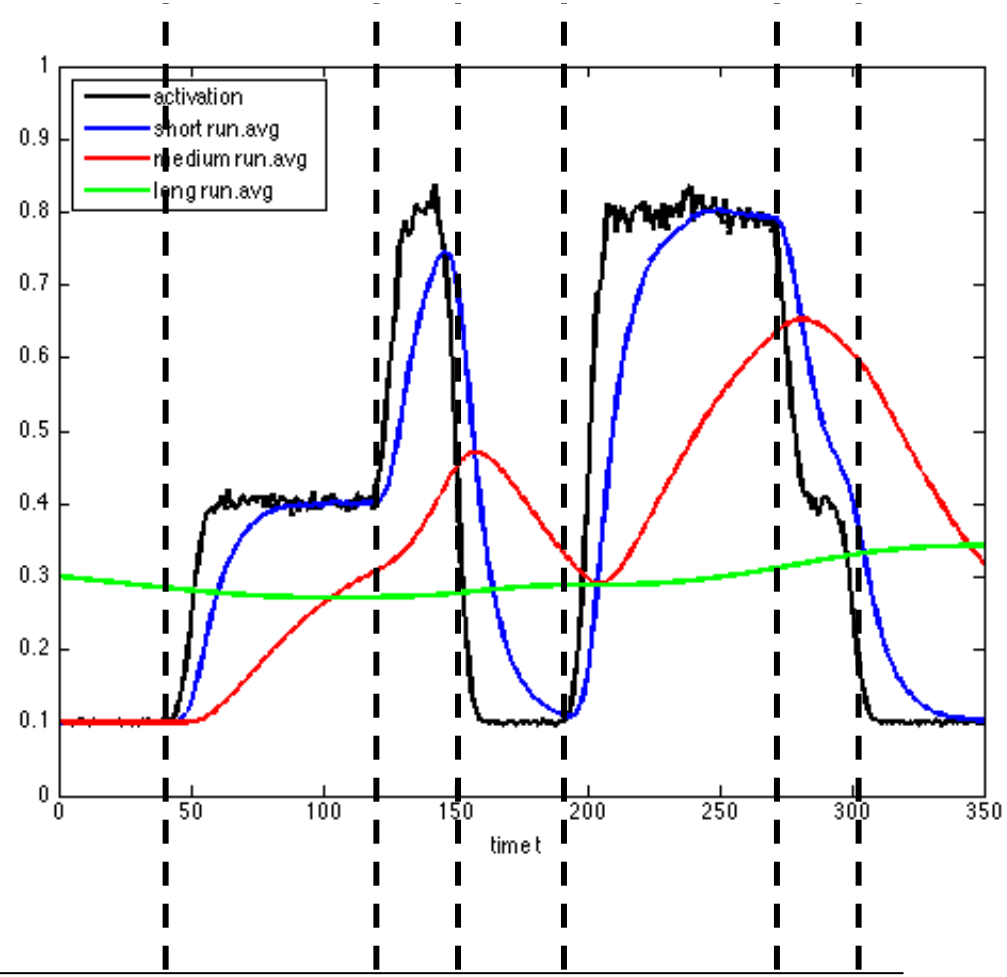


Lutningen på linjen är alltid 1, så dW = xy där det finns en positiv lutning

När sker inlärning

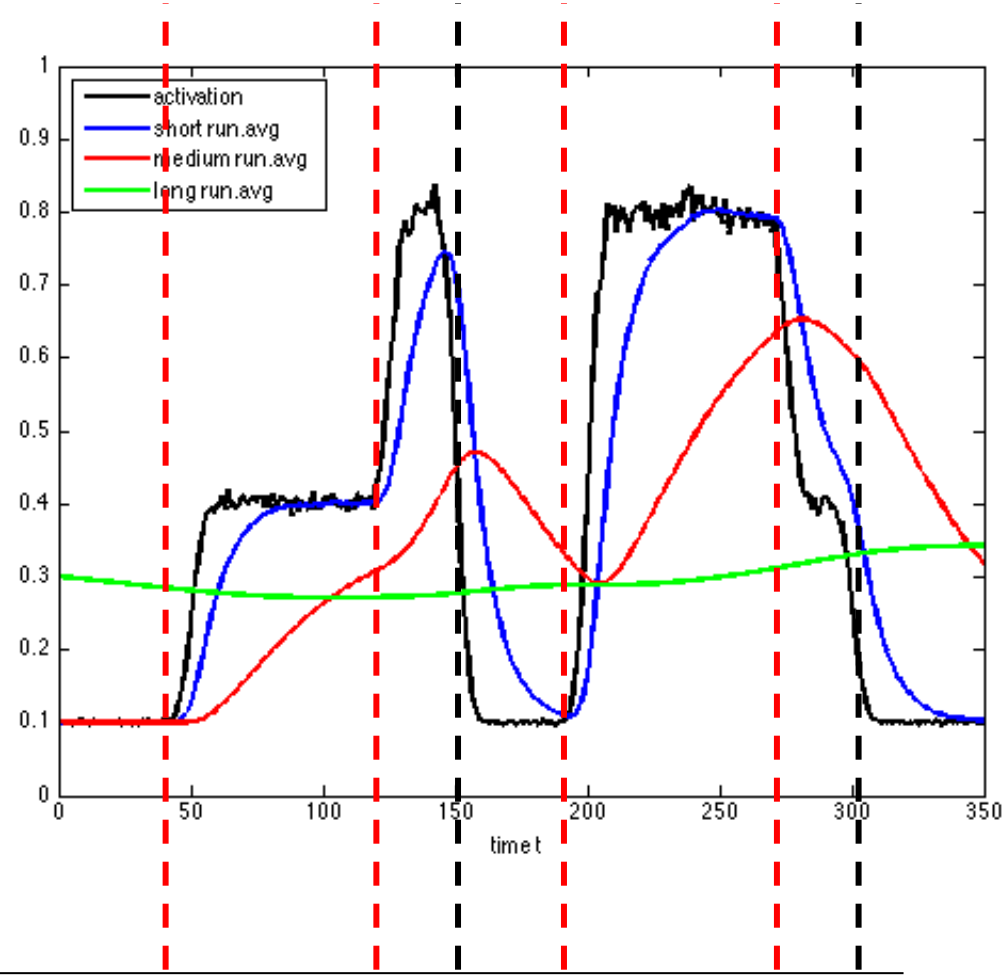
När beräknas felet

Quarter ~ 25 ms



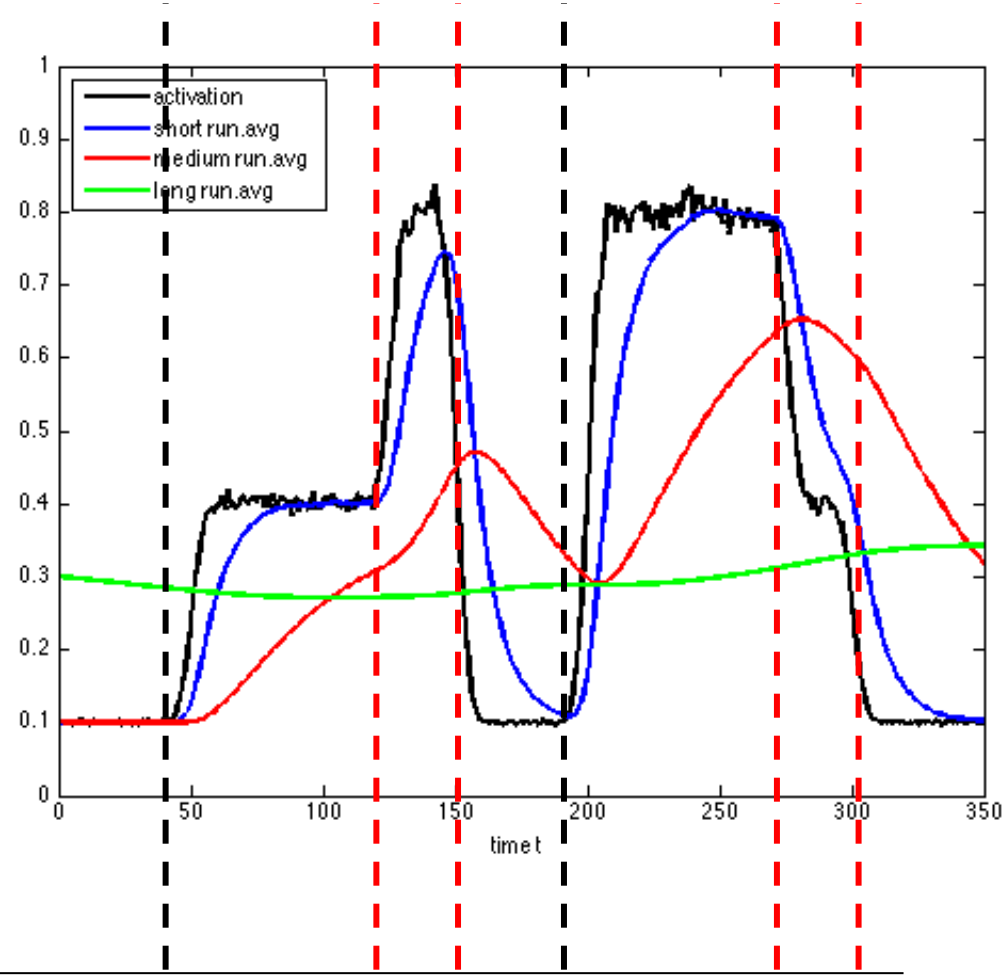
Tre quarters ~ 75 ms

- ”Gissningsfas”
  - Nätet får producera sin output



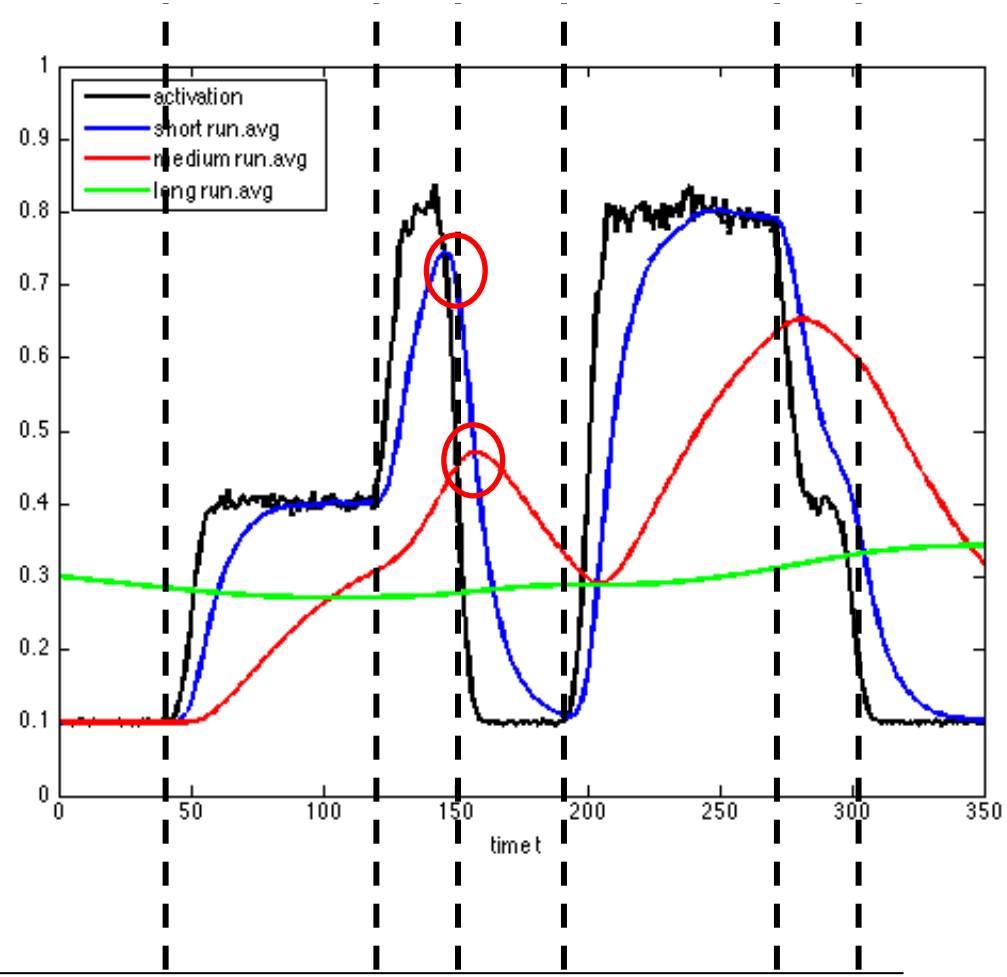
Fjärde quarter ~ 25 ms

- Nätet får även se korrekt output
  - Nätet sätter sig i aktivering som bättre återspeglar korrekt output



Fjärde quarter ~ 25 ms

- Nätet får även se korrekt output
  - Nätet sätter sig i aktivering som bättre återspeglar korrekt output

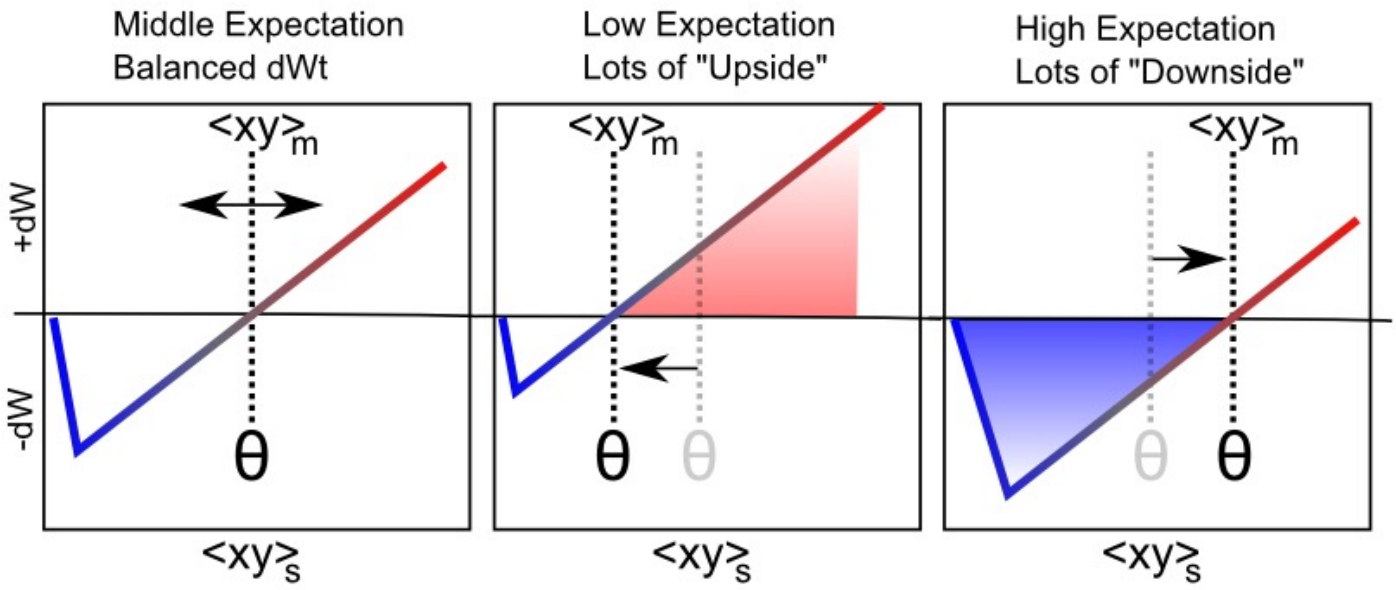


För att beräkna felet

- Vill ha diff mellan aktivering i senaste quarter och de tre tidigare quarters
- Dvs. vill ha
  - $\delta \approx \langle x \rangle_s - \langle x \rangle_m$
  - $\langle x \rangle_s$ är snittaktivering över senaste quarter:n
  - $\langle x \rangle_m$ är snitt över alla fyra quarters (en trial)

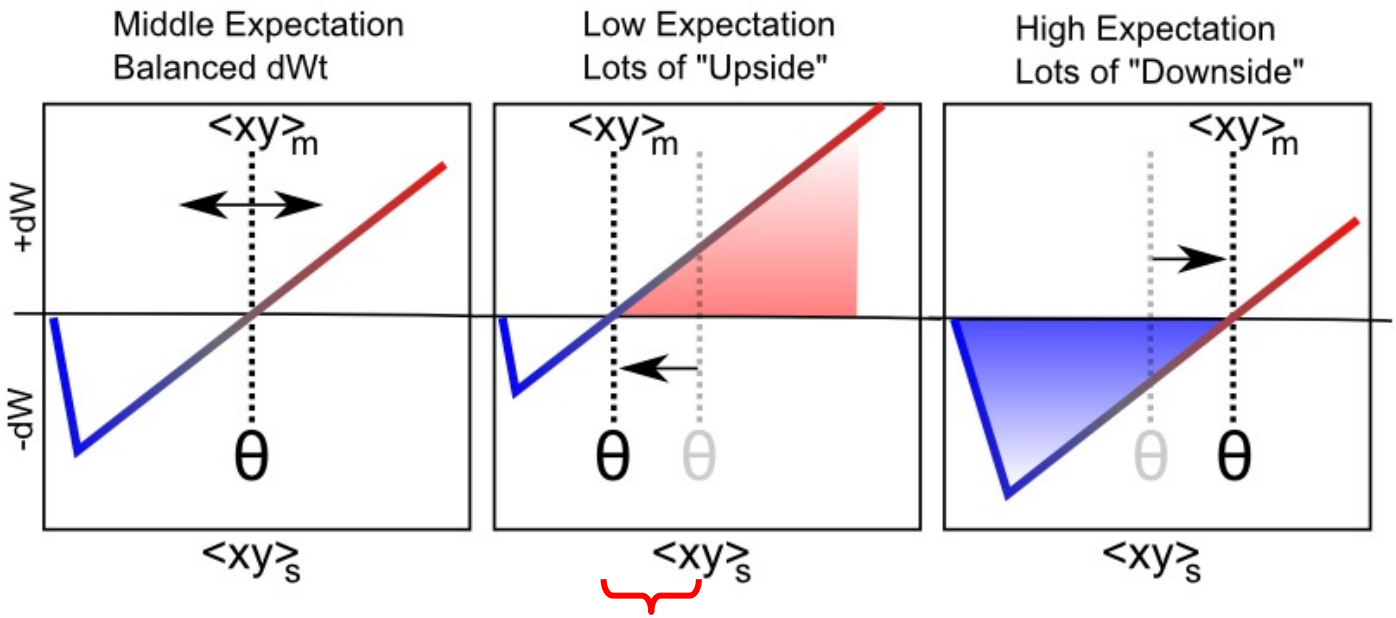
Ny slags tröskel

- Tröskel baseras på medium time scale average av $x * y$



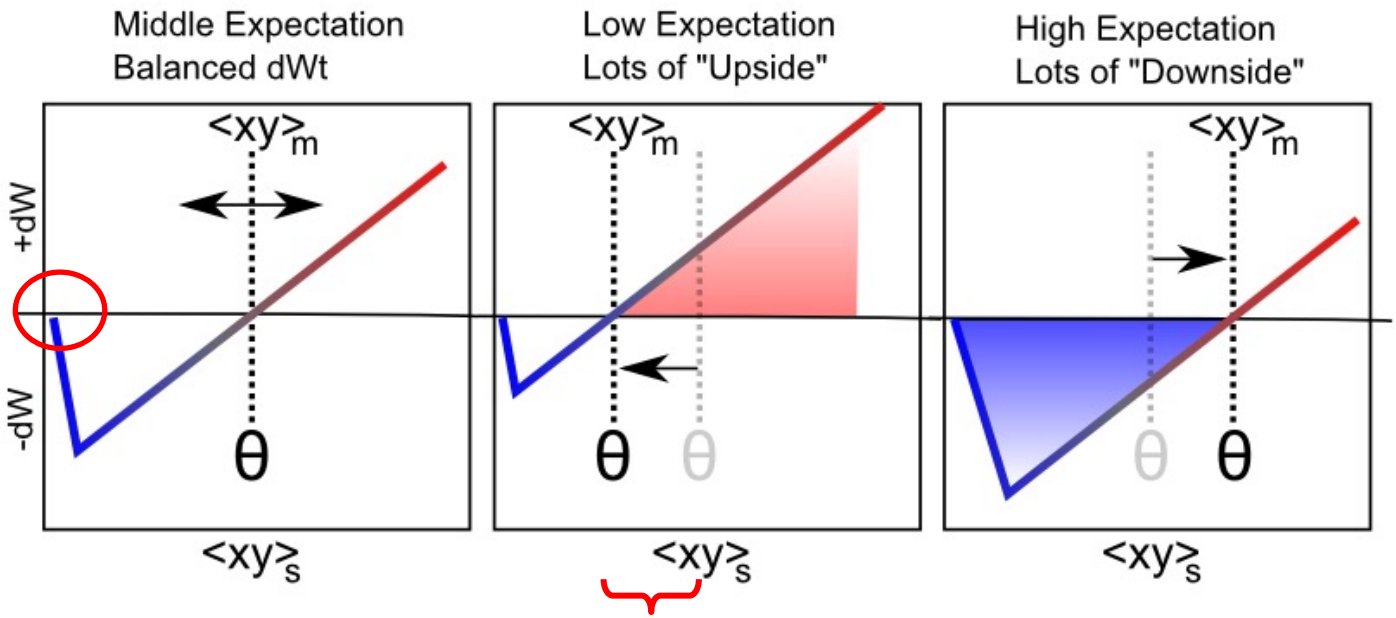
x-axel fortfarande …

- Short time scale average av $x * y$



Ingen feldriven inlärning när $x * y = 0$

- Precis som för oövervakad inlärning: båda enheter måste vara aktiva



När man läser av grafen

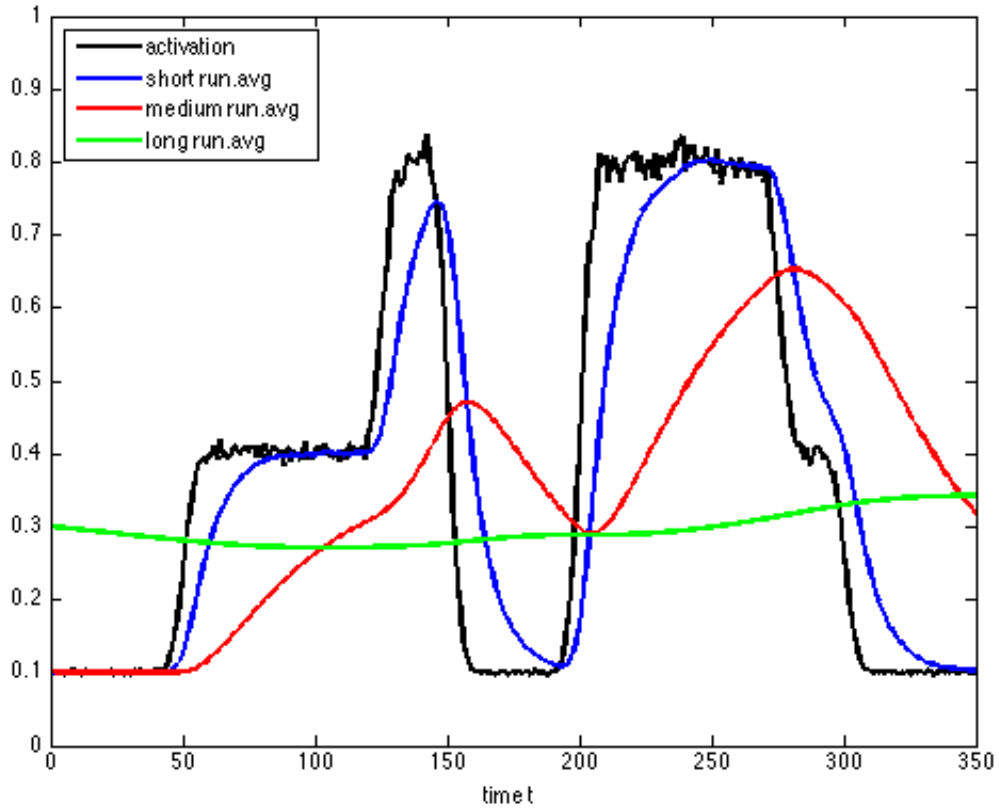
- $\Delta w \approx \langle xy \rangle_s - \langle xy \rangle_m$

- Viktuppdatering beräknas i slutet av en trial, när korrekt output (samt ursprunglig input) presenteras för nätet
  - $\langle xy \rangle_s$ är då senaste quarter:s aktivering
  - $\langle xy \rangle_m$ avspeglar hela trial:s aktivering

- Detta gäller även för oövervakad inlärning (Hebb)

Trial = 4 quarters ~ 100 ms

- Minus-fas
  - 3 första quarters
  - ”Gissningsfas”
    - Nätet gissar output
- Plus-fas
  - Sista quarter
  - Nätet matas med facit (top-down)
  - Får jämföra aktiveringar…



Epok

- Kör igenom alla events i data
  - Ofta i slumporder (eller helst permuterat)

Träning

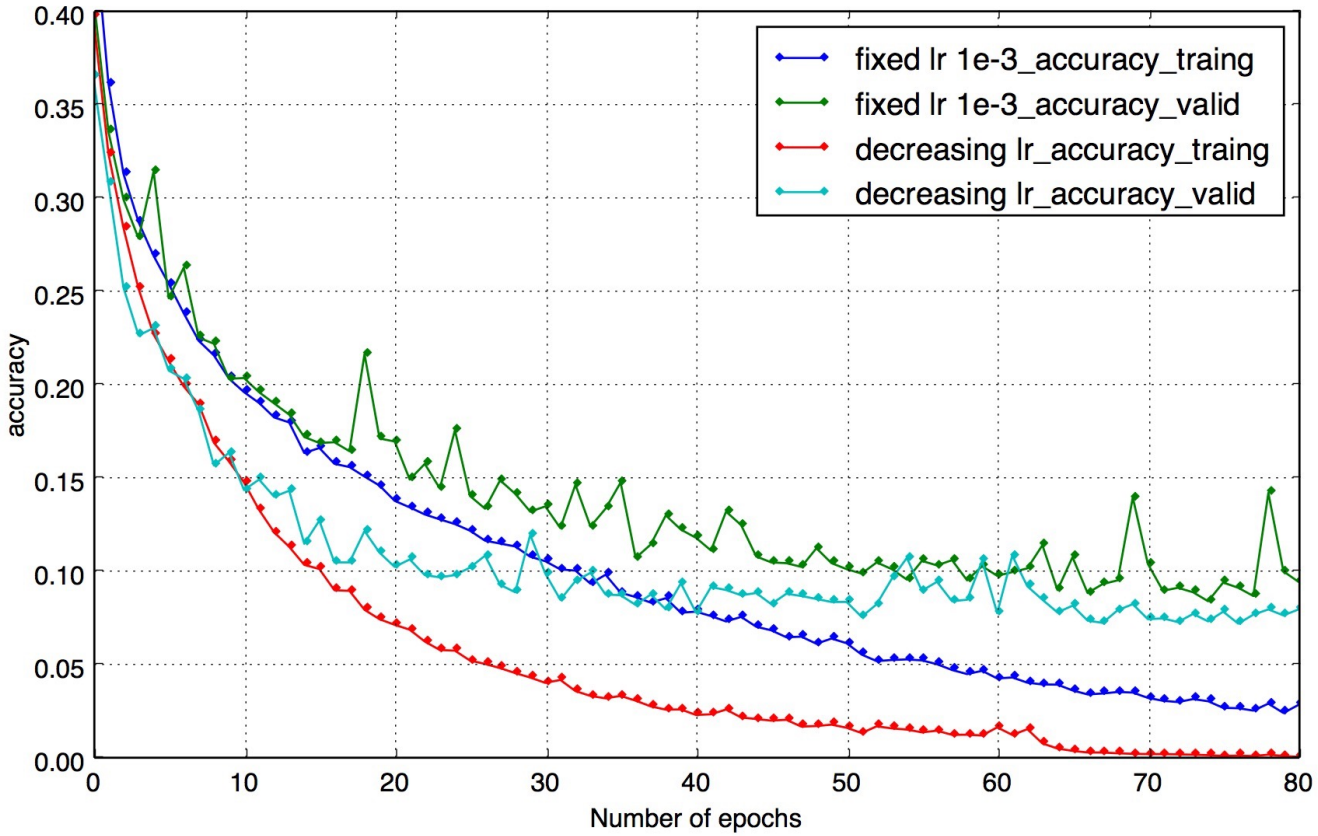
- Köra epok efter epok
  - Varje epok, alla events i ny slumporder

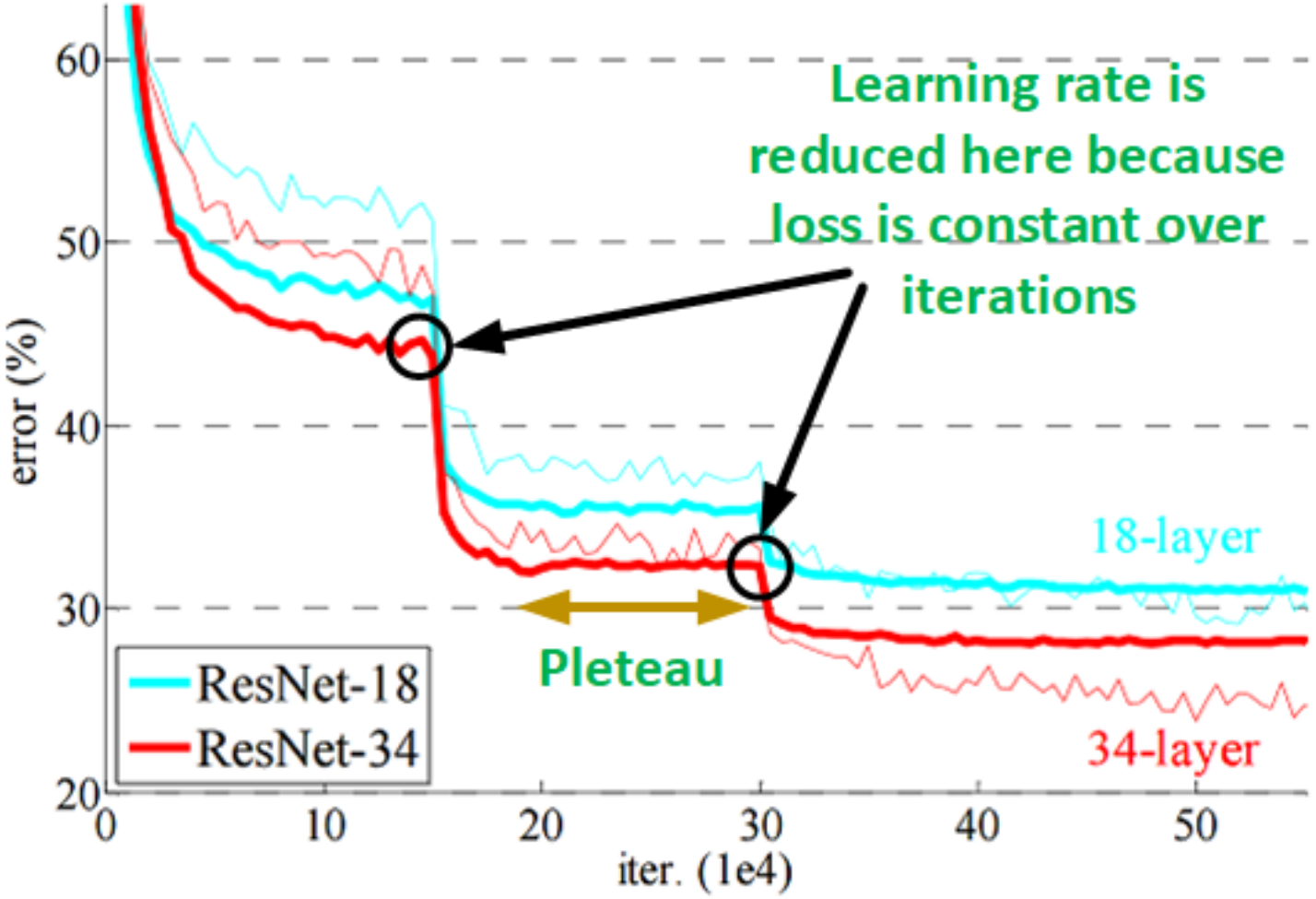
- … nöta på…
- Tills det valda fel-måttet är tillräckligt lågt
- Eller tills N epoker har körts

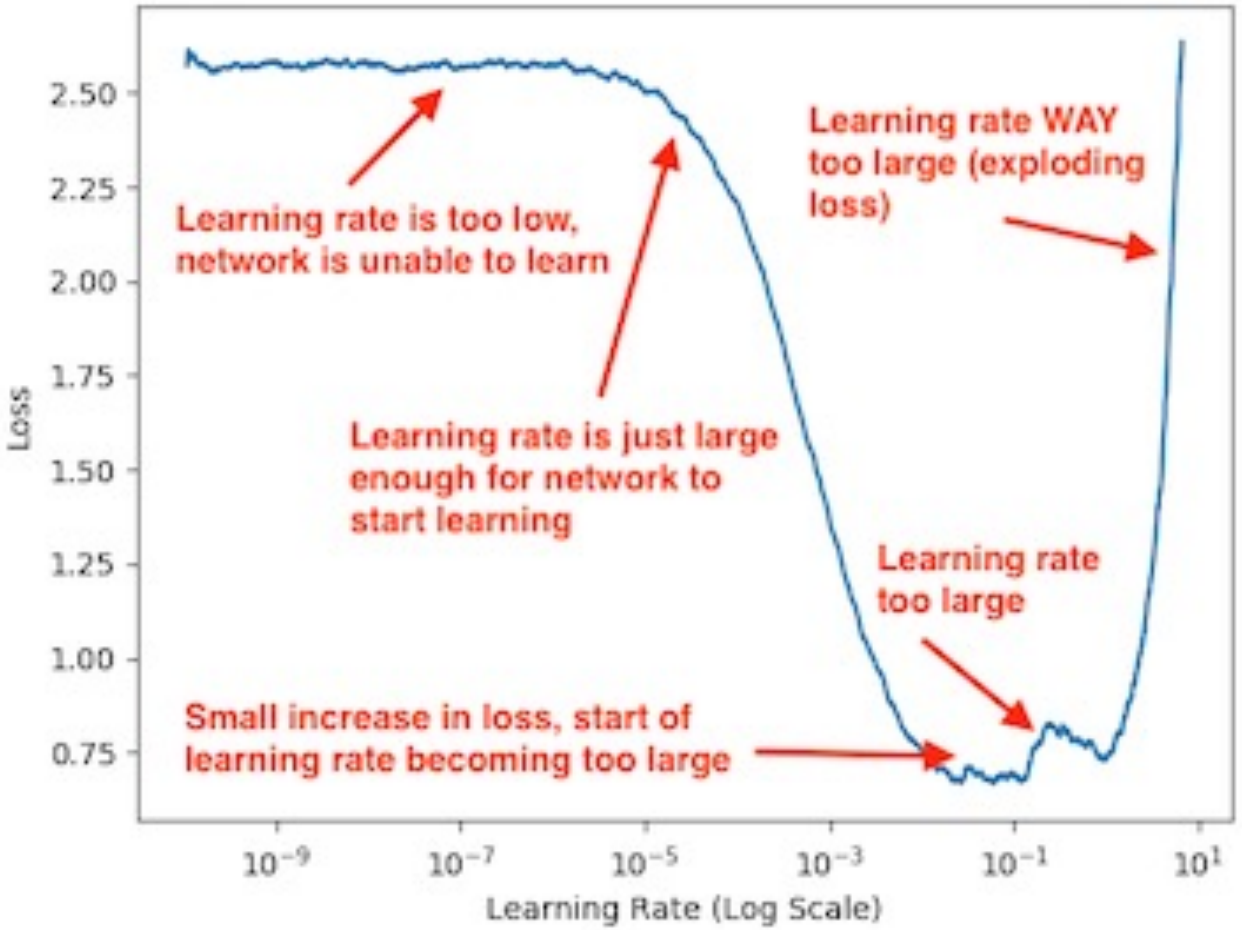
Learning rate

- För hög learning rate ger ryckig inlärning
  - Nätet kastas mellan olika viktuppsättningar
  - *För* anpassat till senast sedda input:en
- För låg learning rate ger långsam (eller ingen) inlärning
- Vill ofta starta med lite högre och sedan sänka var x:te epok

Fix vs. minskande learning rate




Learning rate is reduced here because loss is constant over iterations
Pleteau
18-layer
34-layer
ResNet-18
ResNet-34
error (%)
iter. (1e4)
60
50
40
30
20
0
10
20
30
40
50


Learning rate is too low, network is unable to learn
Learning rate is just large enough for network to start learning
Learning rate WAY too large (exploding loss)
Learning rate too large
Small increase in loss, start of learning rate becoming too large
Loss
2.50
2.25
2.00
1.75
1.50
1.25
1.00
0.75
10^{-9}
10^{-7}
10^{-5}
10^{-3}
10^{-1}
10^{1}
Learning Rate (Log Scale)

Test

- Kör en epok med testdata
  - Ingen viktuppdatering
- Testdata är data som har lagts undan
  - Har inte använts under träning

Batch (i emergent)

- Kör flera träningar
- Ta snitt, så att man får tillförlitliga resultat

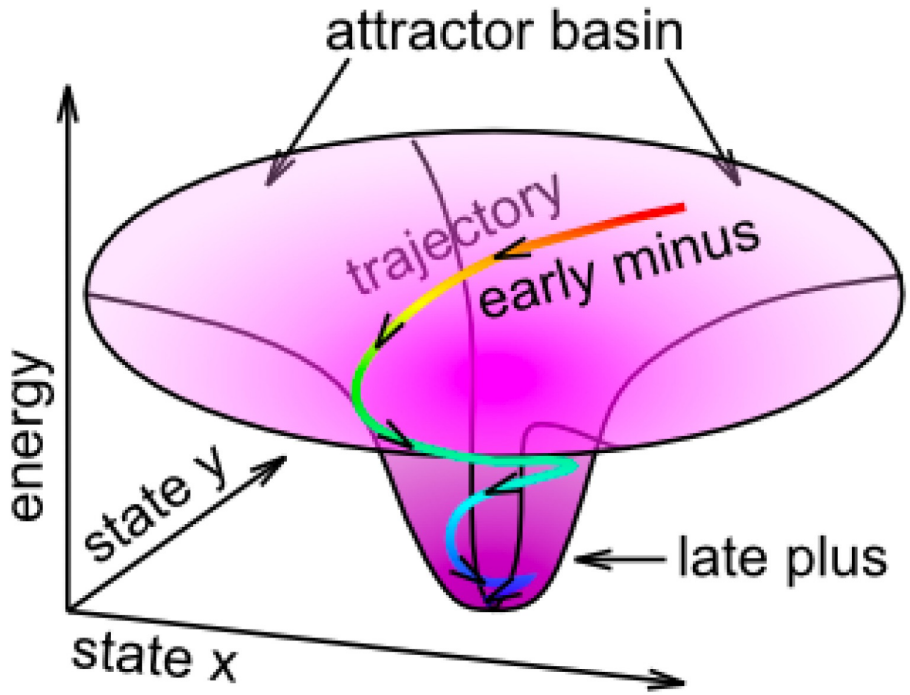
- (Batch, mer generellt = gruppering av körning eller av input för att ta fram snitt)

Kombinera oövervakad och övervakad inlärning

Bra att kombinera oövervakad och övervakad inlärning

- I emergent kombineras dessa i varje enhet
  - Tröskeln justeras för varje enhet
- Ger snabbare, stabilare inlärning
  - Viktigt dock att oövervakad viktuppdatering sker när nätet befinner sig i ”korrekt” aktiveringstillstånd (dvs. i fjärde quarter, dvs. i slutet av trial)

När ska oövervakad tillämpas?



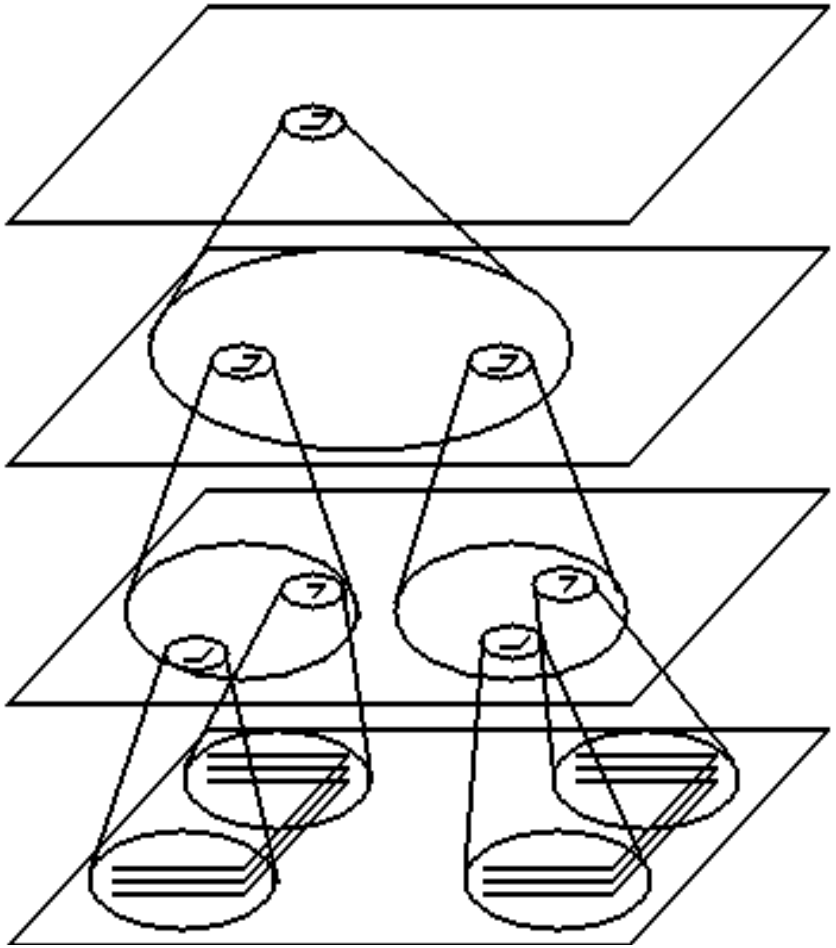
Nu två faktorer som påverkar tröskeln

- Låt en konstant bestämma relativ viktning mellan dessa faktorer
  - $\theta_p = \lambda\langle y\rangle_l + (1-\lambda)\langle xy\rangle_m$
- Eller, helt enkelt beräkna viktuppdatering enl. Hebb, och enl. error-driven
  - Sedan kombinera
  - $\Delta w = \lambda_l f_{self-org} + \lambda_m f_{err-driven}$

Bra att kombinera

- Ett annat sätt att kombinera är att arbetsfördela:
  - Oövervakad mönsterigenkänning i de första (nedersta/tidigaste) lagren
  - Övervakad, dvs. feldriven inlärning i de övre lagren
- T.ex.
  - Känna igen linjer med mönsterigenkänning
  - Kombinera linjer och andra särdrag för att kunna kategorisera, dvs. klassificera objekt

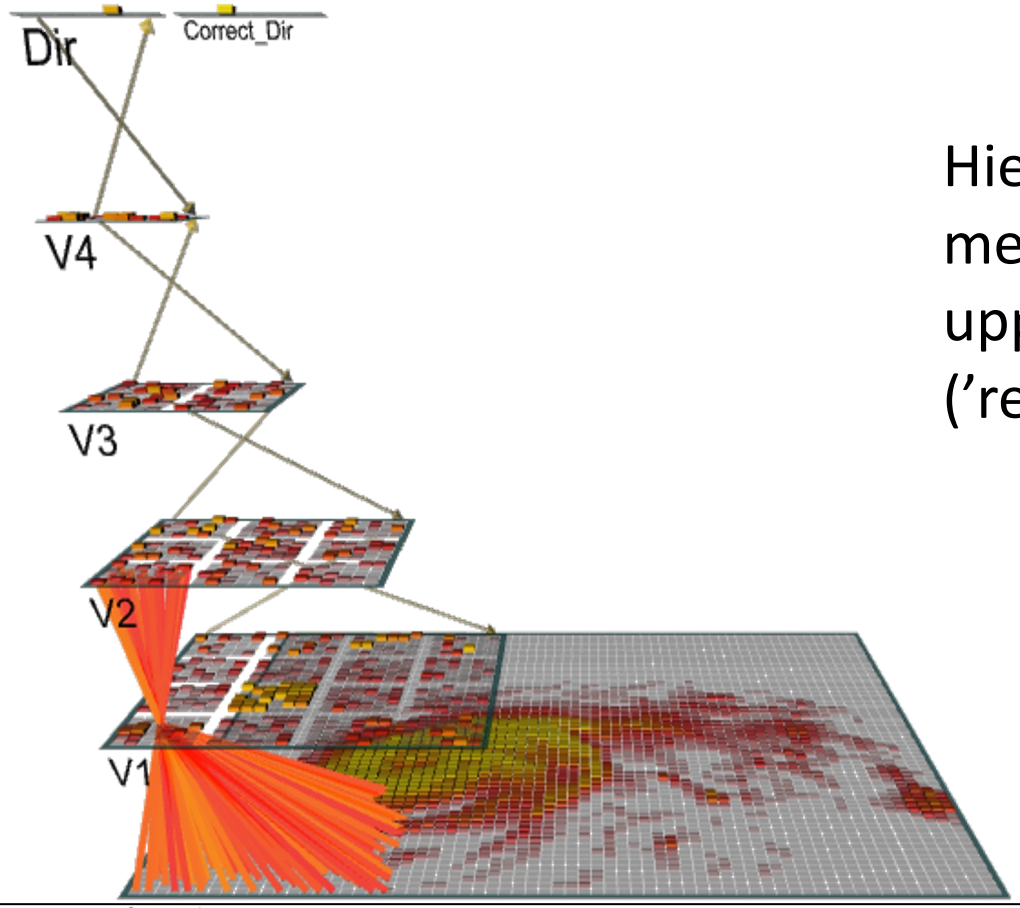
Kombinera enkla särdrag



Övervakad inlärning

Oövervakad inlärning

Nät för bildbehandling



Hierarki av neuroner med allt större upptagningsområde (’receptive field’)

Klassificering

- Inlärningsuppgift som går ut på att producera korrekt klass (dvs. kategori) för varje input
  - T.ex. dela in bilder i 2 klasser: katt – hund

Hebb (oövervakad) inlärning

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

Fel-driven inlärning

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

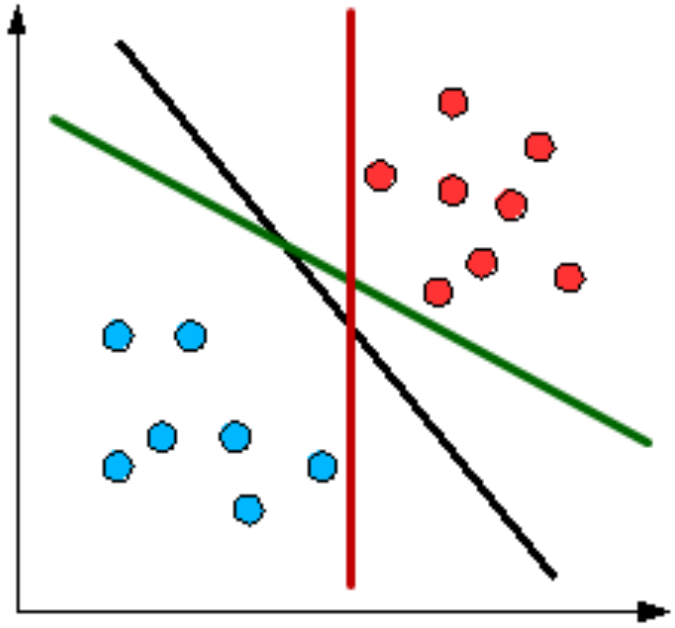
Byråkratisk:

- Problem för algoritmen att bestämma vilka vikter som ska göra vad då alla är beroende av varandra → Inlärningen tar (mycket) lång tid

”Impossible” problems

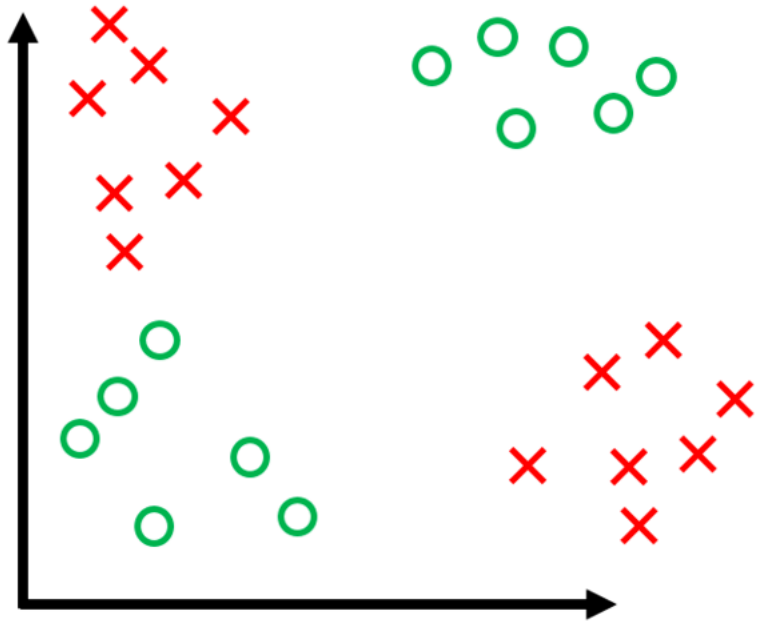
Vad neurala nätverk gör

- Varje enhet drar ett streck, dvs. ”skär med en kniv” i aktiveringsrymden



Finns ”omöjliga” problem

- Kan inte delas upp i två klasser m.h.a. en linje



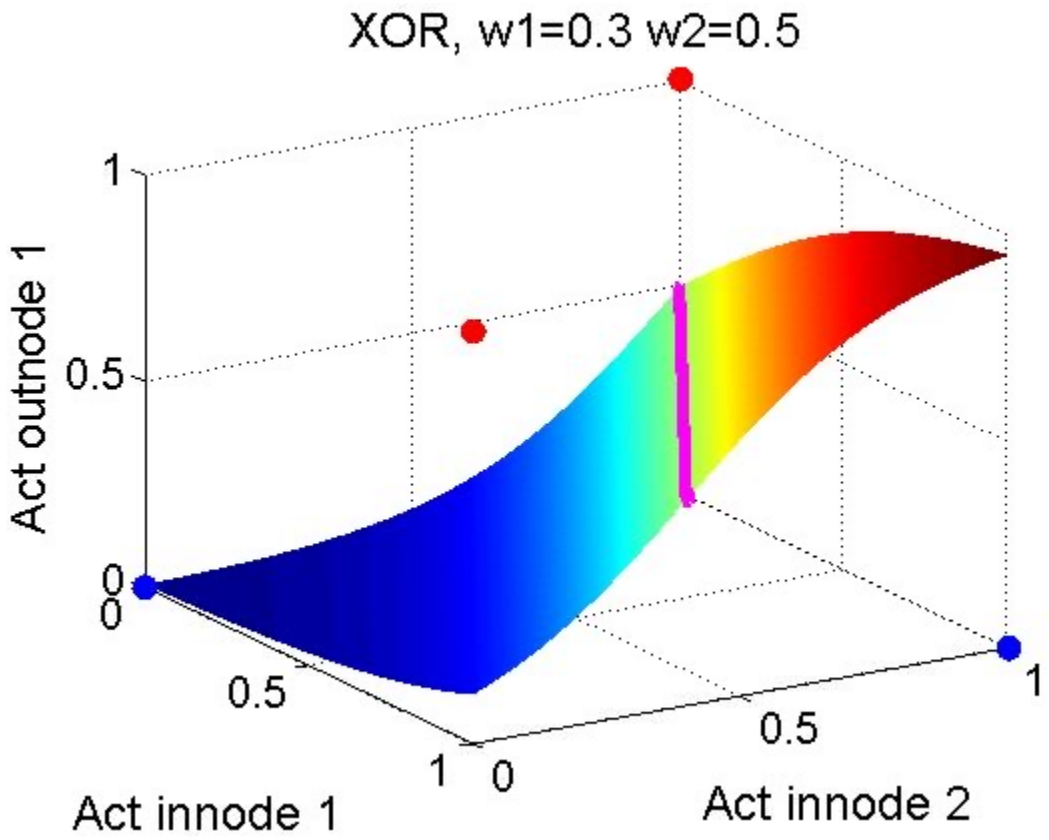
Exempel: XOR-problemet

- Exclusive OR:

0, 0	0
0, 1	1
1, 0	1
1, 1	0

- För icke linjärt separerbara problem behövs minst ett dolt lager

XOR (eXclusive OR)



”Fel” vs. loss

Fel

- Skillnad mellan korrekt (target), dvs. förväntad output och erhållen output (predicted)
  - $\delta = t - o$
- Måste beräknas för varje enhet i utlagret
  - $\delta = \sum(t_i - o_i)$
- Om n enheter i utlagret
  - $\delta = \sum_{i=1}^{n}(t_i - o_i)$

Fast, …

- … vad händer om vissa av delfelen är negativa?
- Vi vill inte att delfel ska kunna kvittas mot varandra
  - Vanligt att kvadrera termerna i summan
  - Kvadrering av ett negativt tal ger ett positivt tal
    - T.ex. $(-2)^2 = -2 * -2 = 4$
- Alltså: Squared Error
  - $\delta = \sum_{i=1}^{n}(t_i - o_i)^2$

MSE: Mean Squared Error

- Vi måste beräkna ’squared error’ för varje input-target-par i vårt data set
  - Ett sätt är att ta snitt på felet över alla input-target-par
- Går även att summera: SSE, dvs. Summed Squared Error

Fel vs. MSE

- Felet är ”rå-felet”, dvs. diffen utan krumelurer
- MSE (eller andra mått) är en värdering av ”rå-felet”
  - T.ex. kvadrering
    - Distorterar på olika sätt
    - Förminskar små ”rå-fel”: $-1 <$ ”rå-fel” < 1
    - Förstorar andra fel
    - ”Små fel gör ingenting, större fel är farliga…”

rita.kovordanyi@liu.se

www.liu.se

LINKÖPINGS UNIVERSITET