Keskikenttäoppimiseen perustuva algoritmi ohjaamattomille neuroverkoille

perustuen L. Saulin ja M. Jordanin artikkeliin
“A Mean Field Learning Algorithm for Unsupervised Neural Networks”

Vesa Siivola, 28. marraskuuta 2001

MLP-verkot

MLP-verkkojen (Multi Layer Perceptron) etuja:

- Monikäyttöisiä, yleistävät hyvin tavalliset ongelmat
- Helppoja opettaa (backpropagation)
- Oppivat hajautetun esityksen ongelmasta

MLP-verkkojen heikkouksia

- puuttuvan datan käsittely
- ohjaamaton oppiminen
- piiloyksiköillä ei ole tilastollista (/mitään) tulkintaa

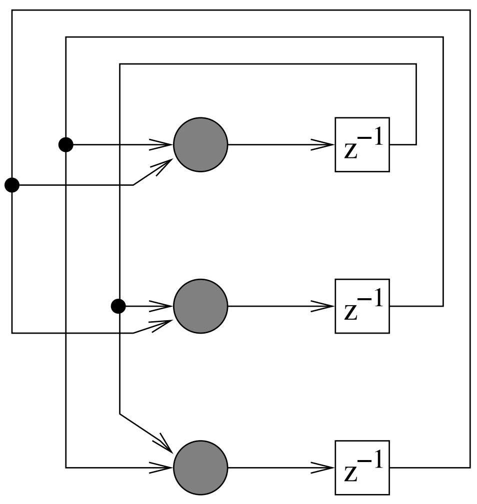
MLP-verkot tilastollisina malleina

- Yksiköt satunnaismuuttujia, joiden statistiikat koodattuna yhteistodennäköisyysjakaumassa (joint probability distribution)
- Ajatellaan, että mallinnetaan tiheysjakaumaa systeemissä, missä on latentteja muuttujia (vrt. funktion approksimointi)
- Tavoitteena datan uskottavuuden maksimointi
- Saadaan tavoitteet myös piiloyksiköille

Aikaisempia neuroverkkorakenteita

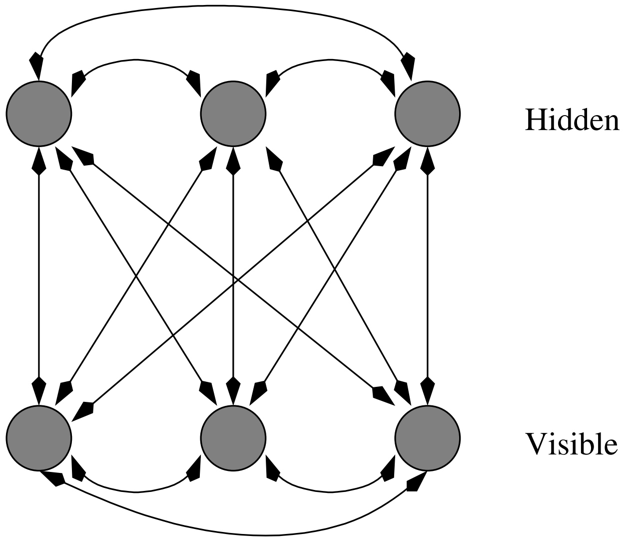
Hopfield-verkko

- Opetetaan hahmoilla, näistä tulee verkon minimienergiapisteitä
- Verkosta saadaan etsittyä opittuja hahmoja syöttämällä korruptoituneita hahmoja. Verkko hakeutuu lähimpään minimienergiapisteeseen.



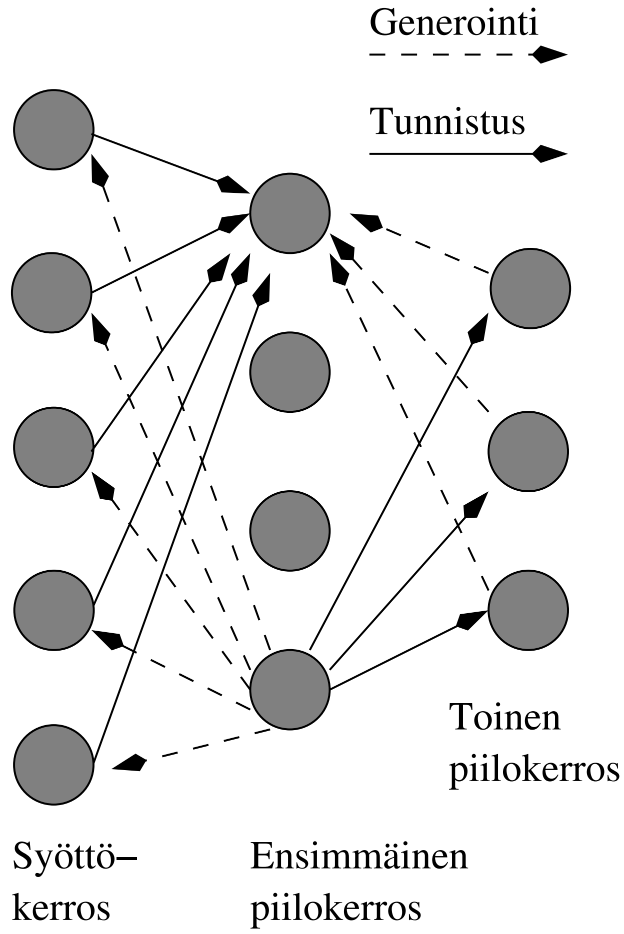
Boltzmannin kone

- Hopfield-verkko, jossa myös piiloyksiköitä
- Pystyy periaatteessa mihin tahansa kuvaukseen
- Ongelmia opetuksessa tarvittavien tilastojen arvionnissa
  - Analyyttinen ratkaisu kovin hankalaa
  - Gibbsin sämpläys liian hidas
  - Keskikenttäteoria liian köyhä esitys ohjaamattomaan oppimiseen



Neuroverkot kerrostettuina uskomusverkkoina

- Todennäköisyystulkinta sunnattuina graafisina malleina
- Suunnatuille verkoille saadeaan laskettua uskottavuuden alaraja → saadaan helpommat oppimissäännöt



Helmholtzin kone

- Yksinkertainen stokastinen opetussääntö (hereillä–nukkumassa, wake–sleep)
- Nopea, "alhaalta ylös" -tunnistus
- Ei osaa käsitellä puuttuvaa dataa ja ilmiöitä, joiden selittämiseen vaadittaisiin myös "ylhäältä alas" -prosessointia

Sigmoidinen uskomusverkko

- Näytteet binäärivektoreita
- Tavoita: mallintaa, miten vektorit on generoitu
- Todennäköisyys, että yksikkö on päällä

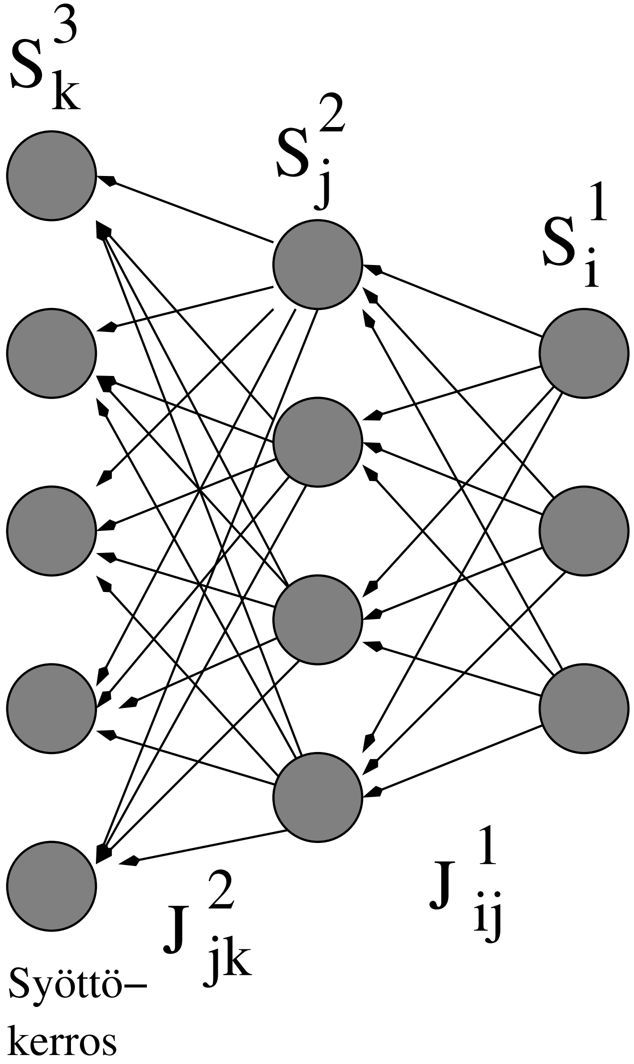
$$P(S_i^l = 1 | S^{l-1}) = \sigma(\sum_j J_{ij}^{l-1} S_j^{l-1} + h_i^l) = \sigma_i^l \quad (1)$$

missä
$$\sigma(z) = \frac{1}{1 - e^{-z}} \quad (2)$$

- Siispä koko mallin yhdistetty PDF on

$$P(S) = \prod_l \prod_i (\sigma_i^l)^{S_i^l} (1 - \sigma_i^l)^{1 - S_i^l} \quad (3)$$

- Tavoitteena on siis mallintaa alimman kerroksen data. Piiloyksiköille pitää keksiä sopivat tavoitearvot.



ML-estimaatti

- Lasketaan siis näytteiden todennäköisyys

$$P(V) = \sum_H P(H, V) \tag{4}$$

- Kun yhtälöstä 3 ratkaistaan uskottavuuden gradientit, saadaan päivityssäännöt painoille ja biaksille.

$$\Delta J_{ij}^l \propto E[(S_i^{l+1} - \sigma_i^{l+1}) S_j^l] \tag{5}$$

$$\Delta h_i^l \propto E[S_i^l - \sigma_i^l] \tag{6}$$

- Näiden muutosten suora ratkaisu ei onnistu
- Sämpläämällä ratkaiseminen on kovin hidasta

Keskikenttäteoria

- Tilastollisesta fysiikasta (kuten myös simuloity jäähdytys ja monet sämpläysmenetelmät)
- Menetelmä useiden korreloivien muuttujien tilastojen arvioimiseen
- Verkossa, jossa on paljon neuroneita, tutkitaan vaan kunkin neuronin keskimääräistä tilaa
- Arvioidaan siis, että joku funktio $Q(H|V) \approx P(H|V)$ ja saadaan alarajaksi

$$\ln P(V) \geq \sum_H Q(H|V) \ln \left[\frac{P(H,V)}{Q(H|V)} \right] \tag{7}$$

- Mitä lähempänä $Q(H|V)$ on $P(H|V)$:tä, sitä tarkempi alaraja. Valitaan faktoroitu jakauma

$$Q(H|V) = \prod_l \prod_{i \in H} (\mu_I^l)^{S_i^l} (1 - \mu_i^l)^{1 - S_i^l} \tag{8}$$

- Jos ajatellaan, että $\mu_i^l \approx E(S_i^l)$ ja $\xi_i^l \approx E(\sigma_i^l)$, seuraavilla keskikenttäyhtälöillä on intuitiivinen tulkinta

$$\begin{aligned}
\mu_i^l &= \sigma\Big[\sum_j J_{ij}^{l-1}\mu_j^{l-1} + h_i^l \sum_j J_{ij}^{l-1}(\mu_j^{l-1} - \xi_j^{l+1}) \\
&\quad -\frac{1}{2}(1-2\mu_i^l)\sum_j (J_{ji}^l)^2 \xi_j^{l+1}(1-\xi_j^{l+1})\Big] \qquad (9) \\
\xi_i^l &= \sigma\Big[\sum_j J_{ij}^{l-1}\mu_j^{l-1} + h_i^l - \frac{1}{2}(1-2\xi_i^l)\sum_j (J_{ji}^{l-1})^2 \mu_j^{l-1}(1-\mu_j^{l-1})\Big] \qquad (10)
\end{aligned}$$

- Nämä parametrit iteroimalla kukin verkon yksikkö tarpeeksi monta kertaa. Tämä on approksimaation laskennallisesti raskain osa.

Keskikenttäteoria, päivityssäännöt

- Gradienttiratkaisu

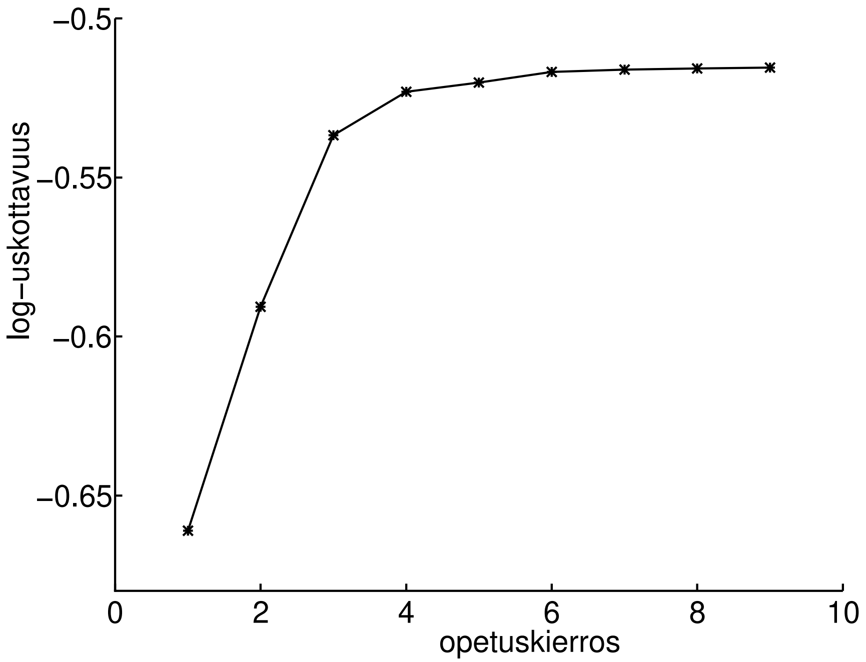
$$\Delta J_{ij}^{l} \propto \left[(\mu_i^{l+1} - \xi_i^{l+1}) \mu_j^{l} - J_{ij}^{l} \xi_i^{l+1} (1 - \xi_i^{l+1}) \mu_j^{l} (1 - \mu_j^{l}) \right] \tag{11}$$

$$\Delta h_i^{l} \propto (\mu_i^{l} - \xi_i^{l}) \tag{12}$$

- Hyvin samanlaiset kuin ML-estimaatit (yhtälöt 5 ja 6), mutta laskettavissa.
- Erona yhtälön 11 lopussa oleva termi, jota voi ajatella modifioituna painojen hiipumisena (weight decay).
- Jos yksiköt korreloivat paljon (μ_j^l tai ξ_i^{l+1} lähellä nollaa tai ykköstä), painojen ei hiivuteta. Painot ovat palaneet kiinni (burned in).

Kokeet

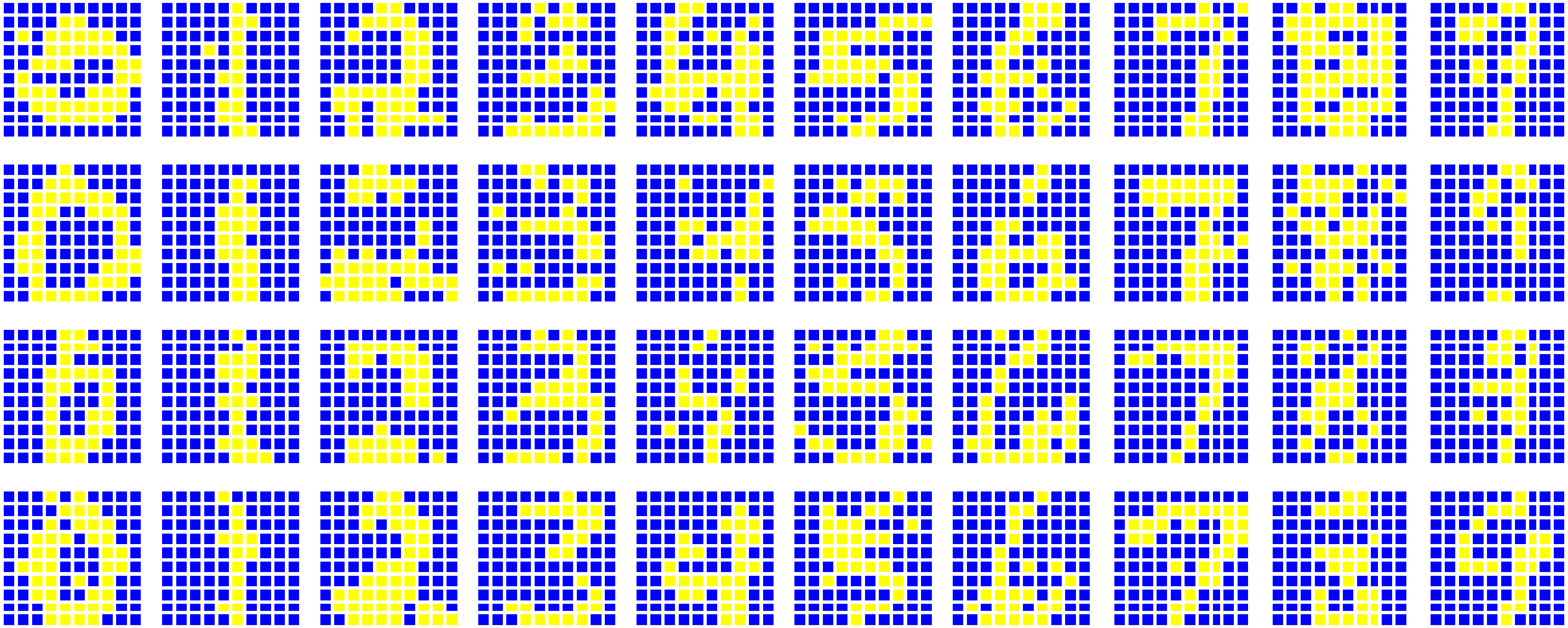
- Data: Käsinkirjoitetut numerot 0-9 suoristettuina, downsämplättyinä ja kynnystettyinä 10×10 pikselin binäärikuviksi
- Verkkona nelikerroksinen $4 \times 12 \times 36 \times 100$ verkko
- Kuvassa log-uskottavuuden muutos opetuskierrosten funktiona



- Kun opetettuja verkkoja käytettiin luokitukseen, saatiin n. 5.0% luokitteluvirhe (KNN:lle 5.4%)

Kokeet, mallikuvien generointi

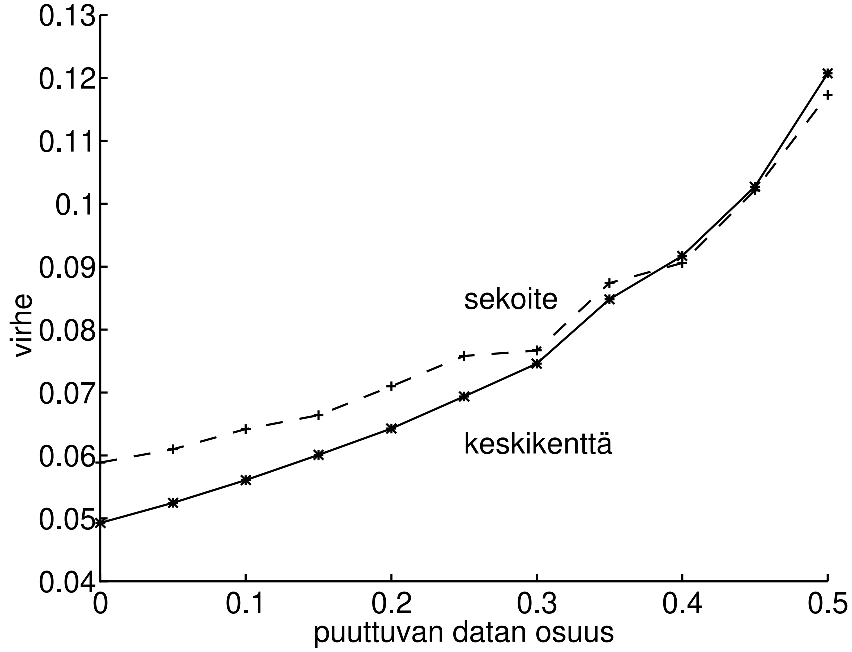
- Kun verkkoa käytettiin generoimaan satunnaisia hahmoja, saatiin seuraavanlainen kuva



- Nähdään, että verkko on oppinut jonkinlaisen järkevän esityksen

Kokeet, puuttuva data

- Verkkoja kokeiltiin myös merkitsemällä osa pikseleistä puuttuvaksi
- Tulosta veratiin 40 komponenttiseen sekoitemalliin, joka oli opetettu EM-algoritmilla
- Vertailu viereissä kuvassa. Molemmissa malleissa on suurinpiirtein yhtä paljon vapaita parametrejä → vaikka hierarkkiset mallit ovat vaikeampia opettaa, siitä on myös vastaavaa hyötyä



Yhteenveto

- Formulointi probabilistisessä kehyksessä
- Informaatio siirtyy paikallisena viestinvälityksenä – voidaan miettiä kytköksiä biologiaan
- Päivitys yhdistää delta-säännön, painojen hiipumisen ja painojen kiinnipalamisen
- Algoritmi kehityy koko ajan: vanhemmassa versiossa oli vähemmän approksimointeja ja se oli paljon hitaampi tarjoten suurinpiirtein saman oppimiskyvyn
- Voisi käyttää eksoottisempia verkkokonfiguraatioita (esim. lokaalimpia kytkentöjä)
- Jatkuvien arvojenkäyttö binäärivektoreiden sijaan
- Miten saadaan prioritiedot koodattua verkkoon ?
- Jonkinnäköinen feedback-mekanismi kuitenkin tarpeen ?