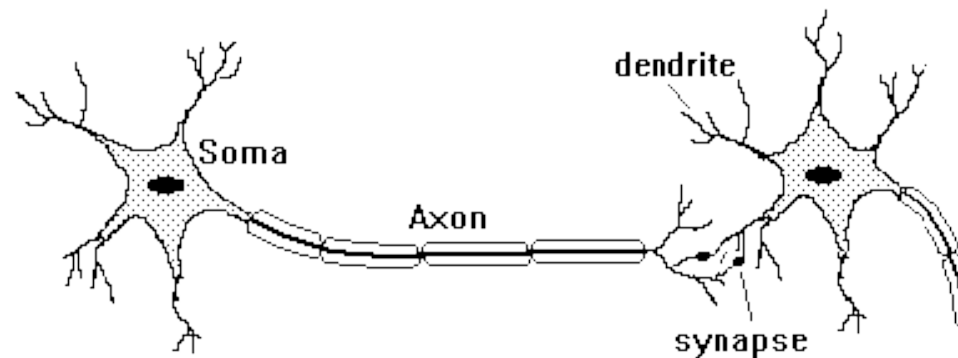
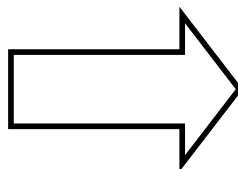
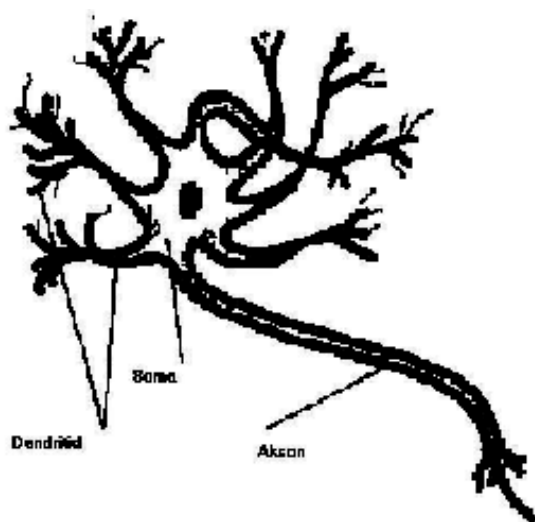


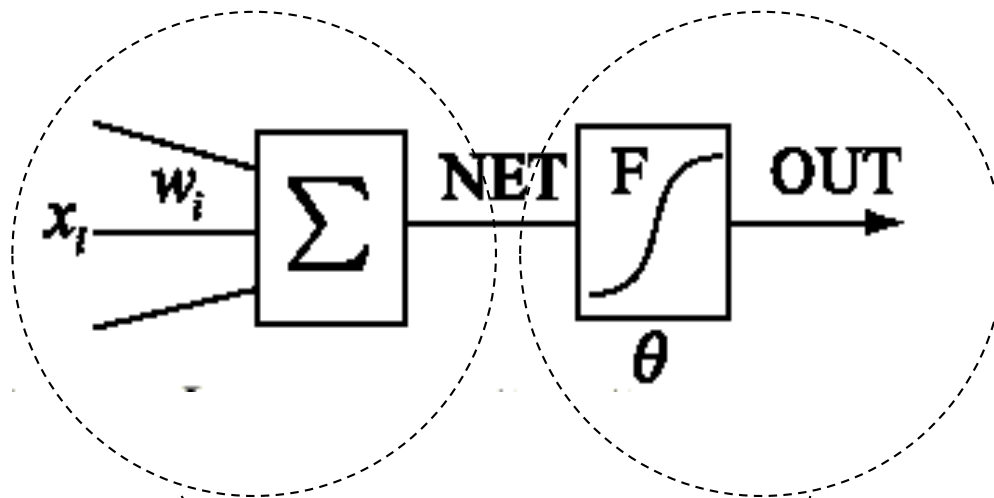
Tehisnärviõrgud ja nende rakendused

*Eduard Petlenkov,
Arvutisüsteemide instituut,
Arukate Süsteemide keskus*

Bioloogiline neuron ja bioloogilised närvivõrgud



Tehisneuron



kaalitud summaator

mittelineaarne element

tehisneuroni sisendite vektor: $X = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$

tehis neuroni kaalukoefitsientide vektor:

$$W = [w_1 \dots w_n]$$

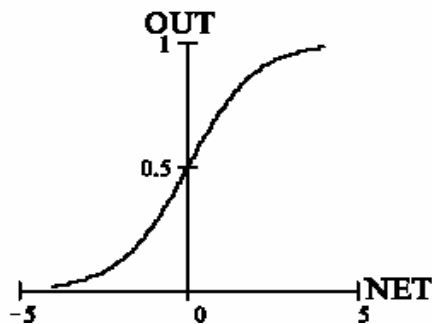
summatori funktsioon:

$$NET = W \cdot X = [w_1 \dots w_n] \cdot \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = w_1 x_1 + \dots + w_n x_n$$

Aktiverimisfunktsioonid (1)

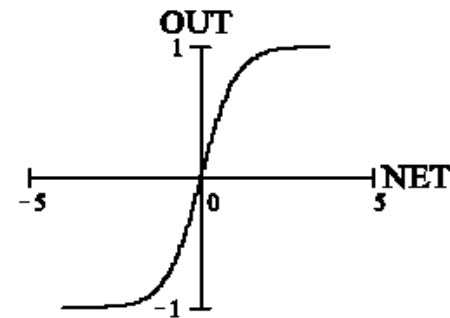
$$OUT = f(NET)$$

Sigmoid funktsioonid on ülemise ja alumise raja (0 ja 1 või -1 ja 1) vahel monotoonselt kasvavad pidevad funktsioonid.



1 Logistiline funktsioon

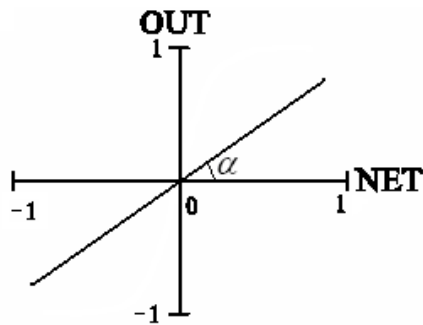
$$OUT = \frac{1}{1 + e^{-NET}}$$



2 Hüperboolne tangens

$$OUT = \frac{e^{NET} - e^{-NET}}{e^{NET} + e^{-NET}}$$

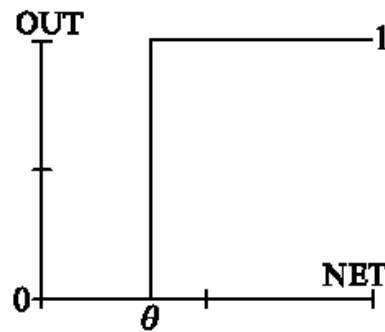
Aktiverimisfunktsioonid (2)



3 Lineaarne funktsioon

$$OUT = k \cdot NET$$

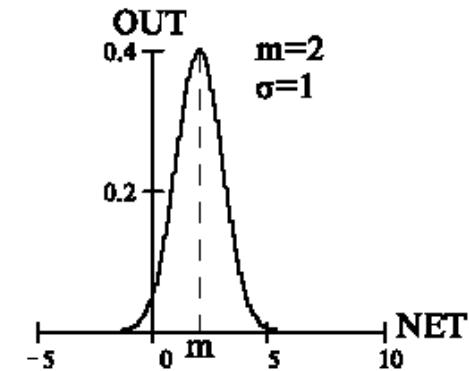
$$k = \tan(\alpha) = const$$



4 Astmefunktsioon

$$OUT = \begin{cases} 0, & kui \quad NET < \theta \\ 1, & kui \quad NET \geq \theta \end{cases}$$

$$OUT = \begin{cases} -1, & kui \quad NET < \theta \\ 1, & kui \quad NET \geq \theta \end{cases}$$



5 Gaussi funktsioon

$$OUT = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(NET-m)^2}{2\sigma^2}}$$



Aktiverimisfunktsiooni valik

Üldjuhul, aktiveerimisfunktsiooni valik sõltub:

- konkreetsest ülesandest (sisendite ja väljundite väärtusteest ja füüsilisest interpretatsioonist)
- närvivõrgu realiseerimisviisist (arvutil või elektrilise skeemi kujul)
- kasutatavast õppimisalgoritmist

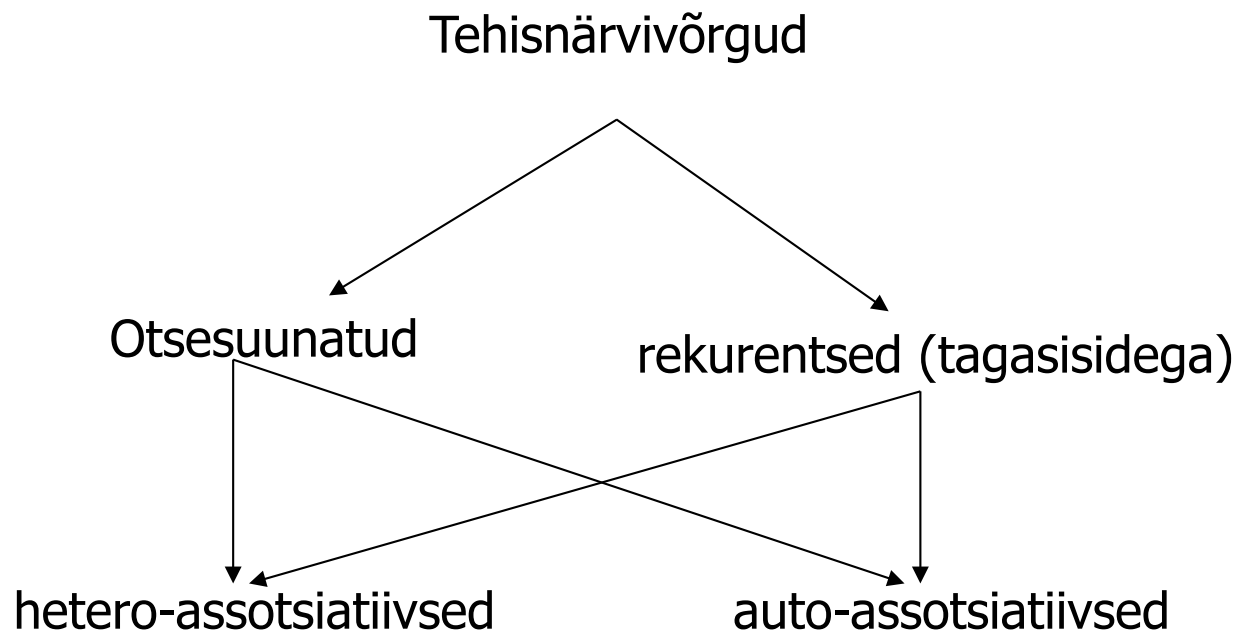


Tehisnärvivõrgud (1)

Tehisnärvivõrk on bioloogiliste närvivõrkude mudelite kogum.

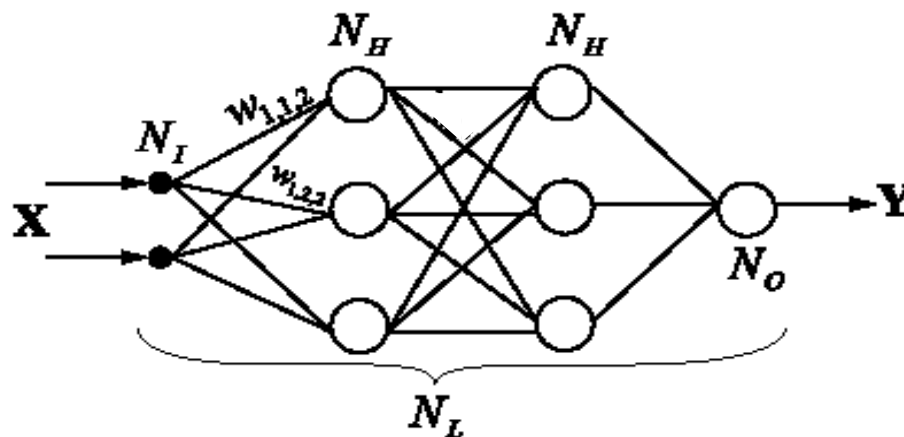
Närvivõrk on andmetöötlus süsteem, mis koosneb suurest arvust lihtsatest ja omavahel tugevalt seotud tehisneuronitest. Tehisneuronid on ühendatud arhitektuuri, mis on võetud inimese ajukoorest.

Tehisnärvivõrgud (2)



Otsesuunatud närvivõrgud ja mitmekihiline pertseptron

Otsesuunatuks nimetatakse närvivõrku, milles iga neuroni väljund võib olla seotud ainult järgmisel kihil oleva neuroni sisendiga.



“igaiüks igaihega”

N_I - sisendkiht
 N_O - väljundkiht
 N_H - peidetud kihid

w_{ijl} - kaalukoefitsiendid, kus

i - on neuroni sisendi number

j - neuroni järjekorra number

l - kihi number

Kahekihilise pertseptroni matemaatiline funktsioon

$$X = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}; \quad Y = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}; \quad W_1 = \begin{bmatrix} w_{111} & \cdots & w_{n11} \\ \vdots & \ddots & \vdots \\ w_{1k1} & \cdots & w_{nk1} \end{bmatrix}; \quad \Theta_1 = \begin{bmatrix} \theta_{11} \\ \vdots \\ \theta_{k1} \end{bmatrix}; \quad W_2 = \begin{bmatrix} w_{112} & \cdots & w_{k12} \\ \vdots & \ddots & \vdots \\ w_{1m2} & \cdots & w_{km2} \end{bmatrix}; \quad \Theta_2 = \begin{bmatrix} \theta_{12} \\ \vdots \\ \theta_{m2} \end{bmatrix};$$

F_1 - peidetud kihi neuronite aktiveerimisfunktsioon;

F_2 - väljund kihi neuronite aktiveerimisfunktsioon.

$$Y = F_2 \left(W_2 \left(\underbrace{F_1(W_1 X - \Theta_1)}_{\text{esimese kihi väljund}} \right) - \Theta_2 \right)$$

närvivõrgu väljund



Peidetud kihi neuronite arvu valik (1)

Neuronite arvu valik peidetud kihil sõltub:

- Lahendava ülesande keerukusest
- Andmete kogust ja kvaliteedist
- Nõutavast närvivõrgu sisendite ja väljundite arvust
- Närvivõrgu koostaja kogemustest



Peidetud kihi neuronite arvu valik (1)

Kui neuroneid või peidetud neuronite kihte on võrgus liiga vähe, siis:

- Võrgu õpetamisalgoritm ei koonu ja võrk töötab ebakorrektselt;
- Võrk ei reageeri aproksimeeritava funktsiooni järskvõnkumistele.



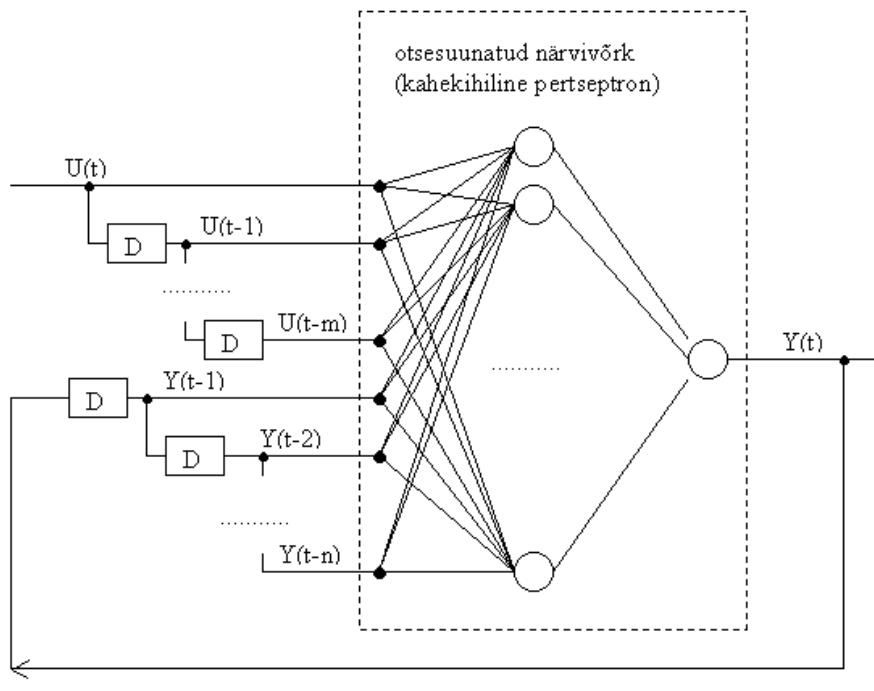
Peidetud kihi neuronite arvu valik (2)

Kui neuroneid või kihte võrgus on liiga palju, siis:

- Võrgu töökiirus võib olla väga madal ning vaja on väga suurt mälu mahtu;
- Võib tekkida nii nimetatud üleõppimise efekt: väljundvektor hakkab kajastama mitteolulisi detaile sõltuvuses, näiteks, müra ja ebatäpsed andmed;
- võrgu käitumine võib muutuda ebastabiilseks: väljund hakkab väga kõvasti ja ettearvamatult reageerima väikestele sisendvektori väärtuste muutustele;
- võrk võib kaotada võime üldistada: piirkonnas, kus õpetamisel on kasutada vähe ette teatud andmeid võib aproksimatsiooni täpsus olla kehv – väljundvektori väärtused on juhuslikud.

Rekurentsed närvivõrgud (1)

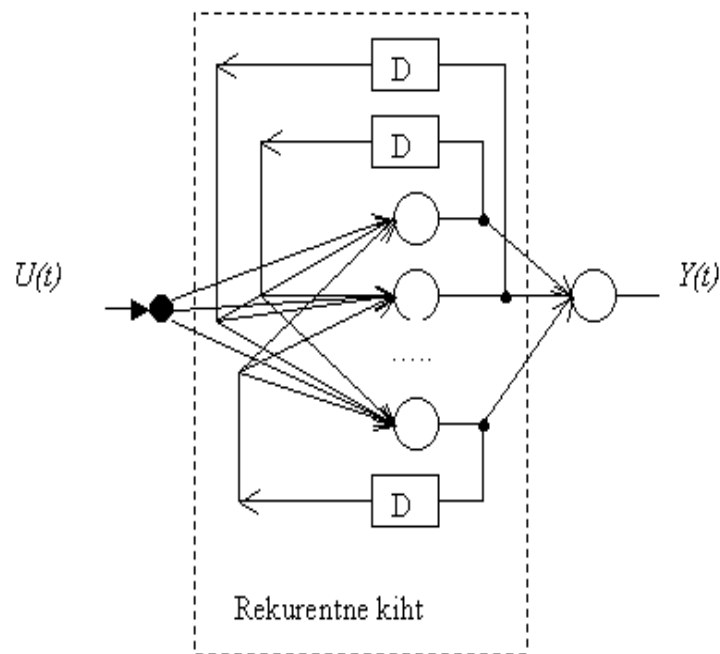
Rekurentseks ehk tagasisidestatuks nimetatakse närvivõrku, milles signaalid levivad nii sisendist väljundi poole, kui ka vastassuunas.



$$y(t) = F_2 \cdot (W_2 \cdot (F_1 \cdot (W_1 \cdot \begin{bmatrix} u(t) \\ \vdots \\ u(t-m) \\ y(t-1) \\ \vdots \\ y(t-n) \end{bmatrix} + \Theta_1) + \Theta_2) =$$

$$= f_{nn}(u(t), \dots, u(t-m), y(t-1), \dots, y(t-n))$$

Rekurentsed närvivõrgud (2)



Elman'i võrk

Rekurentseks nimetatakse peidetud kihti, millel iga neuroni väljundid on seotud kõikide selle kihi neuronite sisenditega.

$$W_r = \begin{bmatrix} w_{11} & \cdots & w_{1r} \\ \vdots & \ddots & \vdots \\ w_{r1} & \cdots & w_{rr} \end{bmatrix}$$

kus

r on rekurentse kihi neuronite arv;

$$\begin{cases} Y(t) = f_{nn}(U(t), X(t), W, B) \\ X(t) = Y_h(t-1) \end{cases}$$



Närvivõrkude õpetamine

Närvivõrgu sobivate parameetrite (konkreetse ülesande jaoks) valiku protsessi nimetatakse närvivõrgu õpetamiseks (või treenimiseks).

Stone-Weierstrassi teoreem:

Kahekihiline pertseptron sobiva neuronite arvuga peidetud kihil on võimeline aproksimeerima suvalist pidevat funktsiooni .

Sontag'i teoreem

Kahekihiline rekurentne närvivõrk sobiva neuronite arvuga peidetud kihil on võimeline aproksimeerima suvalist funktsiooni, millel on lõplik arv katkevuspunkte.



Õpetamine (Supervised learning)

Õpetamiseks nimetatakse meetodit, mis baseerub teadaolevatel sisend- ja väljundvektori väärtuste kogumil.

$$|Y_p - NN(X)| = |Y_p - Y| \rightarrow 0$$

X on sisendväärtuste vektor;

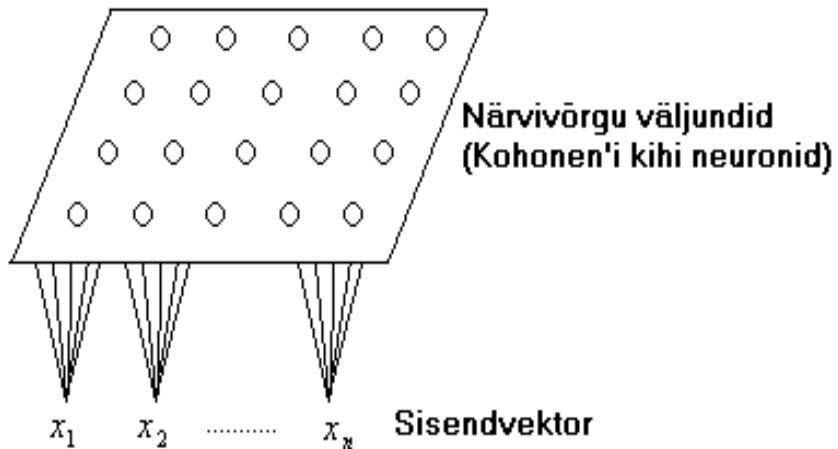
Y_p on nendele sisendväärtustele vastavate etalonväljundväärtuste vektor

Y on närvivõrgu väljundite vektor, mis vastab sisendile X

NN on närvivõrgu funktsioon ($Y=NN(X)$)

Iseorganiseeruvad närvivõrgud

Iseorganiseeruvaks nimetatakse närvivõrku, mis on võimeline häälestama oma kaalukoefitsiente lähtudes ainult sisendvektori väärtustest.



Kohonen'i võrk

$$d_j = \sum_{i=1}^N (x_i - W_{ij}(t))^2, \quad j = 1, \dots, M$$



Iseõppimine (Unsupervised learning)

Iseõppiv närvivõrk on võimeline häälestama oma kaalukoefitsiente lähtudes ainult sisendvektori väärtustest.

Iseõppiva võrgu korral fikseeritakse sihifunktsioon, mille ekstreemum tagatakse võrgu parameetrite muutmisega. Õigesti valitud sihifunktsiooni ekstreemumi saavutamine tagab ka võrgu väljundis õiged väljundvektori väärtused.



Õpetamine (1)

Kasutatakse kaht erinevat treenimisviisi:

1. pakett treenimine (batch-wise training) - kõik "treeninguks" vajalikud sisendandmed ja neile vastavad väljundvektori väärtuste jadad on esitatud ühe paketina. Võrgu parameetrite ümberarvutamise toimub kogu paketi alusel.
2. Sammhaaval treenimine (pattern-wise training) - võrgu parameetrite ümberarvutamine toimub peale igat sisendvektori töötlemist.



Õpetamine

Võrgu õpetamise protsess koosneb kolmest sammust:

1. võrgu väljundvektori väärtuste arvutamine olemasolevate parameetrite alusel;
2. võrgu vea arvutamine lähtudes õpetamismeetodi poolt määratud kriteeriumist (Näiteks, arvutatud võrgu väljundväärtuse ja etteantud etalonväärtuse vahe);
3. võrgu parameetrite väärtuse ümberarvutamine lähtudes õpetamismeetodi poolt määratud algoritmist.



Gradient vea pöördlevi meetod (1)

Veafunktsioon: $J(W, \Theta) = \sum_k (Y_p - Y_p^d)^2$

Y_p - närvivõrgu väljundvektor

$$Y_p = NN(X_p, W, \Theta)$$

X_p -treeninguks kasutatavad sisendvektori väärtused;

NN - närvivõrgu funktsioon;

Y_p^d - vastavad väljundvektori etalonväärtused.

Õpetamise (optimeerimise) ülesanne seisneb veafunktsiooni minimiseerimisel:

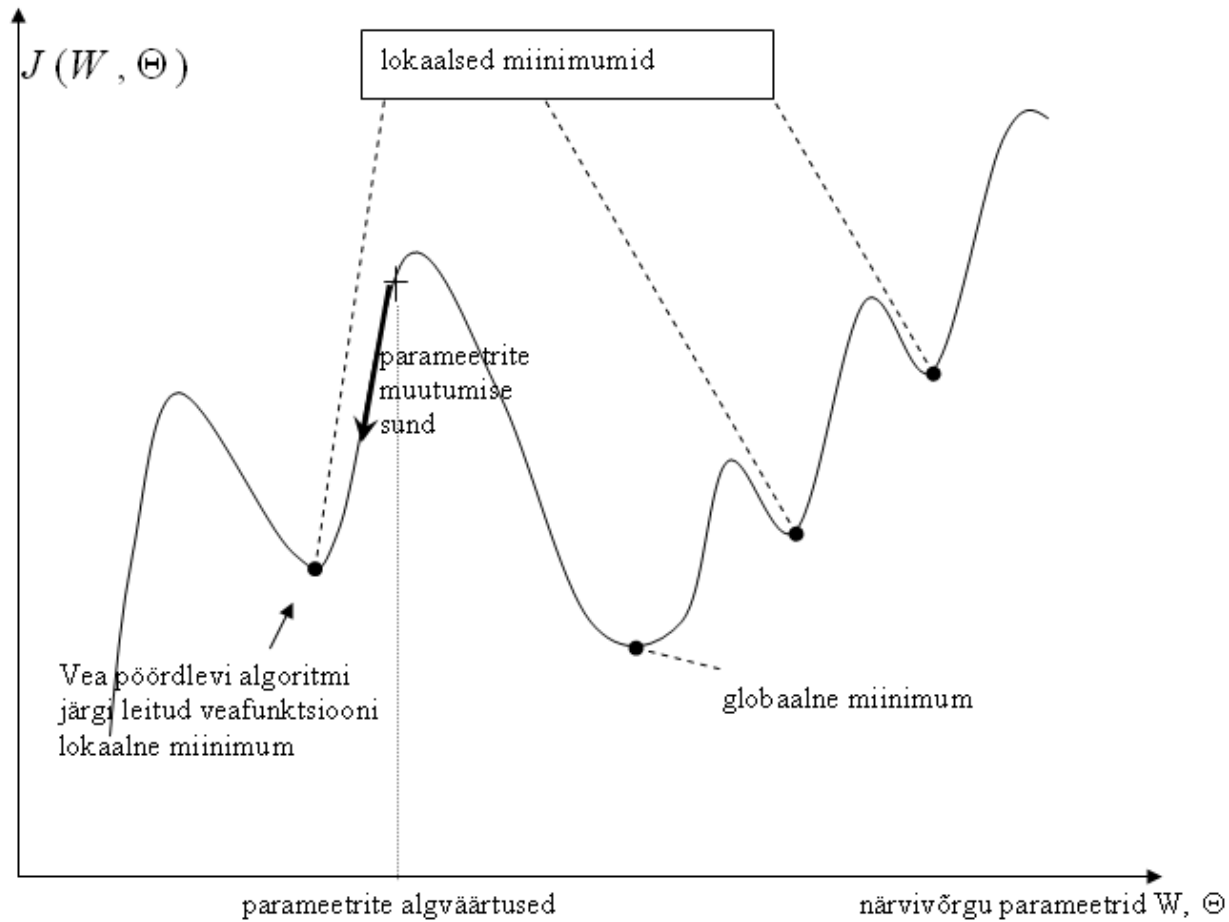
$$\min J(W, \Theta)$$

Gradient vea pöördlevi meetod (2)

$$F(x_1, \dots, x_n) \quad \Rightarrow \quad \begin{aligned} \nabla F(x_1, \dots, x_n) &= \begin{pmatrix} \frac{\partial F}{\partial x_1} & \dots & \frac{\partial F}{\partial x_n} \end{pmatrix} \\ -\nabla F(x_1, \dots, x_n) &= \begin{pmatrix} -\frac{\partial F}{\partial x_1} & \dots & -\frac{\partial F}{\partial x_n} \end{pmatrix} \end{aligned}$$

$$J(W, \Theta) \quad \Rightarrow \quad -\nabla J(W, \Theta) = \begin{pmatrix} -\frac{\partial J}{\partial W} & -\frac{\partial J}{\partial \Theta} \end{pmatrix}$$

Globaalse miinimumi leidmise probleem



Widrow-Hoff'i algoritm

Neuroni väljund: $I = \sum_{i=1}^n w_i x_i$

n – neuroni sisendite arv

$$\varepsilon^2 = (T - I)^2$$

$$\frac{\partial \varepsilon^2}{\partial w_i} = -2(T - I) \frac{\partial I}{\partial w_i} = -2(T - I)x_i, \quad i=1, \dots, n$$

$$\Delta w_i = -K \frac{\partial \varepsilon^2}{\partial w_i} = K \cdot 2(T - I)x_i = 2K\varepsilon x_i$$

$$\Delta w_i = \left[2K|X|^2 \right] \frac{\varepsilon x_i}{|X|^2} = \frac{\lambda \varepsilon x_i}{|X|^2}$$

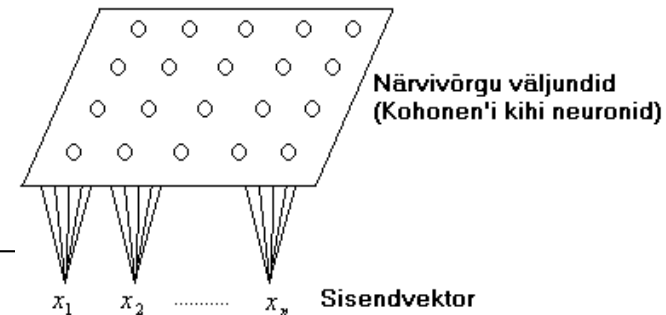
$$|X|^2 = x_1^2 + x_2^2 + \dots + x_n^2$$

$$\lambda = 2K|X|^2 - \text{õppimise kiirus}$$

Kohonen'i iseorganiseerumise algoritm (1)

1. Alguses ($t=0$) kõikidele kaalukoefitsientidele antakse juhuslikke algväärtusi ning määratakse raadiust R , mis määrab iga neuroni ümbrust $U(R)$;
2. Närvivõrgu sisenditele antakse sisendvektorit $X(t)=[x_1(t), \dots, x_N(t)]$;
3. iga neuroni j jaoks arvutatakse kaugust sisendvektori $X(t)$ ja kaalukoefitsientide vektori W_{ij} vahel:

$$d_j = \sum_{i=1}^N (x_i - W_{ij}(t))^2, \quad j = 1, \dots, M;$$



Kohonen'i iseorganiseerumise algoritm (2)

4. Valitakse neuronit j^* , mille kaugus on minimaalne:

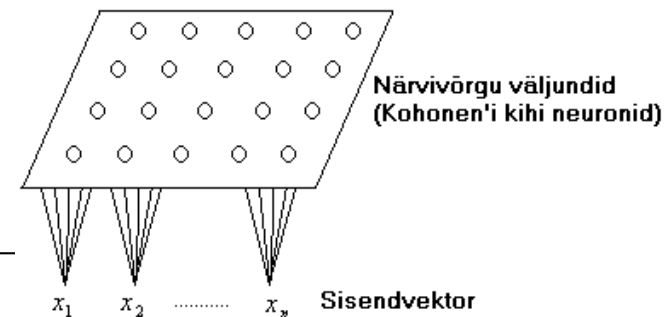
$$\min(d_1, d_2, \dots, d_M) = d_{j^*};$$

5. Muudetakse neuroni j^* ja tema ümbruses $U_{j^*}(R)$ asuvate neuronite kaalukoefitsiente:

$$W_{ij}(t+1) = W_{ij}(t) + \lambda(t)(x_i(t) - W_{ij}(t)),$$

$$\forall j \in U_j(R), \quad i = 1, \dots, N;$$

6. Järgmisel iteratsioonil ($t=t+1$) korratakse algoritmi teisest sammust uuesti.



Kahekihilise pertseptroni õpetamine vea tagasilevi meetodi abil (1)

$$yh_j(t) = F_1\left(\sum_i Wh_{ji}(t)x_i(t) + Bh_j(t)\right) \quad y_k(t) = F_2\left(\sum_i Wo_{kj}(t)yh_j(t) + Bo_k(t)\right)$$

i tähendab võrgu sisendi numbrit;

j - neuroni asukoht (järjekorranumber) peidetud kihil;

k - neuroni asukoht võrgu väljundis;

$x_i(t)$ on närvivõrgu sisendväärtused ajahetkel t ;

$Wh_{ji}(t)$ - peidetud kihi kaalukoefitsientide väärtused ajahetkel t ;

$Bh_j(t)$ - peidetud kihi nihete väärtused ajahetkel t ;

F_1 - peidetud kihi neuronite aktiveerimisfunktsioon

$yh_j(t)$ - peidetud kihi neuronite väljundid ajahetkel t ;

$Wo_{kj}(t)$ - väljundkihi kaalukoefitsientide väärtused ajahetkel t ;

$Bo_k(t)$ - väljundkihi nihete väärtused ajahetkel t ;

F_2 - väljundkihi neuronite aktiveerimisfunktsioon;

$y_k(t)$ - närvivõrgu väljundid ajahetkel t ;

Kahekihilise pertseptroni õpetamine vea tagasilevi meetodi abil (2)

Viga: $e(t) = y(t) - y_{etalon}(t)$

Gradiendid: $\frac{\partial J}{\partial W_{ij}} = \delta_j(t)x_i(t)$

$$\frac{\partial J}{\partial W_{kj}} = \delta_k(t)y_j(t)$$

$\delta_j(t)$, $\delta_k(t)$ on signaalid, mille abil edastatakse informatsioon vea kohta viimaselt kihilt esimeste kihtideni (siit ka pöördlevi meetodi nimetus):

$$\delta_k(t) = (y_k(t) - y_k^d(t)) \cdot F_2'$$

$$\delta_j(t) = F_1' \cdot \sum_k \delta_k(t) \cdot W_{kj}(t)$$

F_1' , F_2' on vastavate neuronite kihtide aktiveerimisfunktsioonide tuletised.

Kahekihilise pertseptroni õpetamine vea tagasilevi meetodi abil (3)

Kaalukoefitsientide uued väärtused:

$$W_{o_{kj}}(t+1) = W_{o_{kj}}(t) - \eta \cdot \frac{\partial J}{\partial W_{o_{kj}}} = W_{o_{kj}}(t) - \eta \delta_k(t) y_{h_j}(t)$$

$$W_{h_{ji}}(t+1) = W_{h_{ji}}(t) - \eta \cdot \frac{\partial J}{\partial W_{h_{ji}}} = W_{h_{ji}}(t) - \eta \delta_j(t) x_i(t)$$

η on koefitsient, mis iseloomustab õppimise kiirust (learning rate)

$$f(x) = \frac{1}{1 + e^{-x}} \Rightarrow f'(x) = \frac{e^{-x}}{(1 + e^{-x})^2} = \frac{1}{1 + e^{-x}} - \frac{1}{(1 + e^{-x})^2} = f(x) - f^2(x)$$



Närvivõrkude rakendused (1)

1. Lähendamise (Aproksimeerimine)

$$d = g(x)$$

Lähendamise ülesandeks on konstrueerida sellise närvivõrgu, et ta realiseeriks funktsiooni $g(x)$, st iga tema sisendi x puhul, närvivõrgu väljund d_{nn} peab olema võrdne funktsiooni $g(x)$ väärtusega d (või temast kuivõrd võimalikult lähedal):

$$g(x) \approx g_{nn}(x)$$

$g_{nn}(x)$ on närvivõrguga realiseeritav funktsioon.



Närvivõrkude rakendused (2)

2. Assotsiatsioon

a) autoassotsiatsioon

Närvivõrgu autoassotsiatsiooni ülesandeks on pidada meeles hulka vektoreid. Need vektorid antakse tema sisenditele järjestikult. Siis esitatakse närvivõrgule vektorid koos müraga (rikutud vektorid) ja närvivõrk peab leidma ja andma väljundile temale vastava originaalse vektori (ilma mürata).

b) heteroassotsiatsioon

Heteroassotsiatsioon erineb autoassotsiatsioonist selles, et igale sisendvektorile on vastavusse pandud oma väljundvektor, mis võib temast erineda.



Närvivõrkude rakendused (3)

3. Mustrite klassifitseerimine

Mustrite klassifitseerimise ülesande puhul peab olema etteantud fikseeritud klasside arv. Iga muster (sisendvektor) kuulub ühele (või mitmele) nendest klassidest. Närvivõrgu õppimiseks mustrite klassifitseerimiseks võib kasutada nii õpetamise kui ka iseõppimise algoritme.

Õpetamisalgoritmi kasutamisel mustrite klassifitseerimiseks iga õpetamisel kasutatava sisendvektori jaoks peab olema määratud temale vastav etalonväljund, mis ütleb millele klassile kuulub käesolev sisend. Kui niisugust *a priori*set informatsiooni klassidele kuulumise kohta ei ole, õpetamist kasutada ei saa. Klassifitseerida mustreid tuleb siis iseõppivate närvivõrkude abil.



Närvivõrkude rakendused (4)

4. Ennustamine

$x(t)$ - ajas muutuv protsess

$x(n-1), x(n-2), \dots, x(n-M)$ on teada

Ülesandeks on *ennustada* protsessi (funktsiooni) olekut (väärtust) käesoleval ajahetkel $x(n)$.

Ennustamise viga: $e(n) = x(n) - \hat{x}(n|n-1, \dots, n-M)$, kus

$x(n)$ on funktsiooni tegelik väärtus

$\hat{x}(n)$ on närvivõrguga ennustatud väärtus

$$e(n) \rightarrow 0$$



Närvivõrkude rakendused (5)

5. Juhtimine

Plant : $\{u(t), y(t)\}$

$u(t)$ on süsteemi juhtimissisend,

$y(t)$ on temale vastav süsteemi väljund.

Juhtimise ülesandeks on saavutada nõutavat süsteemi dünaamikat, mida kirjeldab etalonmudel (reference model):

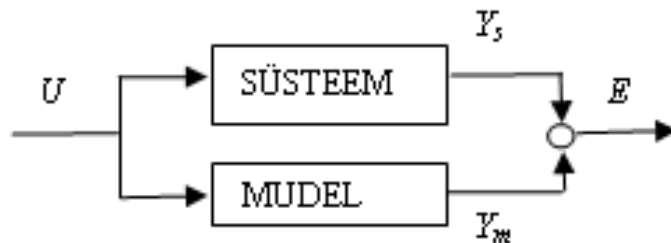
Reference model : $\{r(t), d(t)\}$

$r(t)$ on seadesuurus (juhtimissüsteemi sisend)

$d(t)$ on soovitud juhitava süsteemi väljund

$$\lim_{t \rightarrow \infty} |d(t) - y(t)| = 0$$

Modelleerimine tehisnärvivõrkudega (1)



$$E = Y_s - Y_m$$

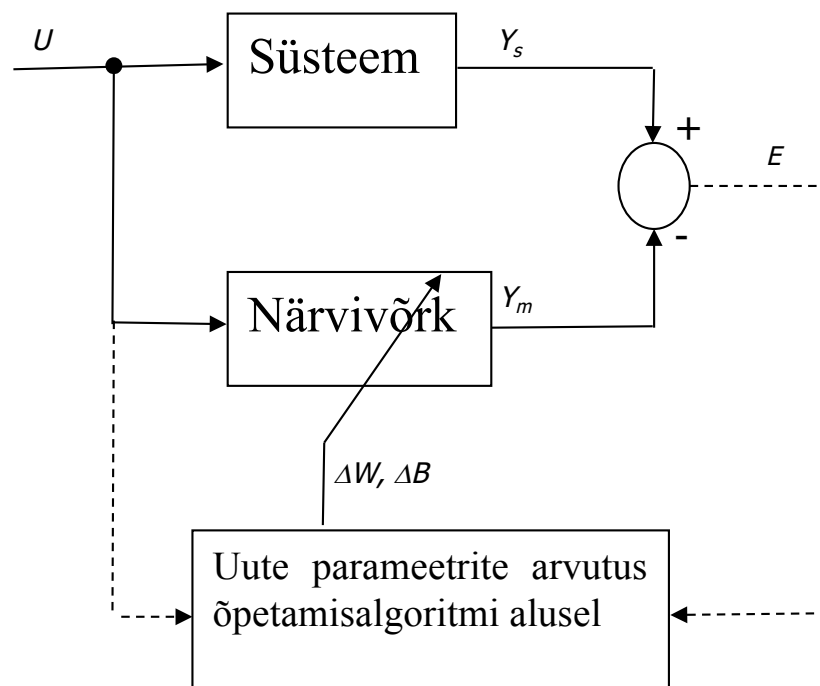
$$E \rightarrow 0$$

U on süsteemi ja mudeli sisendväärtus

Y_s on identifitseeritava süsteemi väljundväärtus

Y_m on identifitseeritava süsteemi matemaatilise mudeli väljundväärtus

Modelleerimine tehisnärvivõrkudega (2)





Modelleerimine tehisnärvivõrkudega (3)

1. Katseandmete kogumine: Identifitseeritava objekti sisendile antakse sisendväärtused (reeglina, need väärtused on juhuslikud). Objekti väljundis mõõdetakse nendele vastavaid väljundväärtusi.
2. Närvivõrku sobiva arhitektuuri valik: sisendite arv, väljundite arv, peidetud kihtide arv, neuronite arv peidetud kihtidel, iga kihi neuronite aktiveerimisfunktsioon.
3. Närvivõrgu kaalukoefitsientide ja nihete algväärtuste valik (reeglina valitakse juhuslikult).



Modelleerimine tehismärgivõrkudega (4)

4. Närvivõrgu väljundi arvutus etalon sisendväärtuste alusel.
5. Mudeli vea leidmine võrreldes närvivõrgu väljundeid objekti etalonväljunditega.
6. Uute parameetrite (kaalukoefitsientide ja nihete) arvutus valitud õpetamisalgoritmi alusel.

Sammud 4 – 6 korduvad iteratiivselt kasutaja poolt määratud õpetamise perioodide jooksul või seni kuni mudeli vajalik täpsus on saavutatud.