



ADALINŲ TINKLAI



Paskaitos
Tūrinys

1. Adalina. Madalina

- Apibrėžimas
- Struktūra
- Loginių funkcijų realizavimo pavyzdys

2. Madalinos tinklo struktūros

- Daugiasluksnė madalinos tinklo struktūra
- Mišrios logikos daugiasluksnė madalinos tinklo struktūra

3. Madalinos tinklo mokymas

- Madalinos taisyklė I
- Madalinos taisyklė II

4. Madalinos tinklo atsakas

5. Konvergencija

6. Adalinų/madalinos tinklo savybės

- Privalumai
- Trūkumai
- Naudojimas



Pagrindinė
Literatūra

1. Simpson, P.K. (1990). *Artificial Neural Systems: Foundations, Paradigms, Applications, and Implementations*. Pergamon Press, pp. 106-111.
2. Widrow, B., Stearns, S. (1985). *Adaptive Signal Processing*. Englewood Cliffs: Prentice-Hall.
3. MATLAB ver. 4.0 (1994). *Neural Networks: Toolbox*. Math Corp., 4 chapter.

Skaidrės Nr.

1



Adalina. Madalina

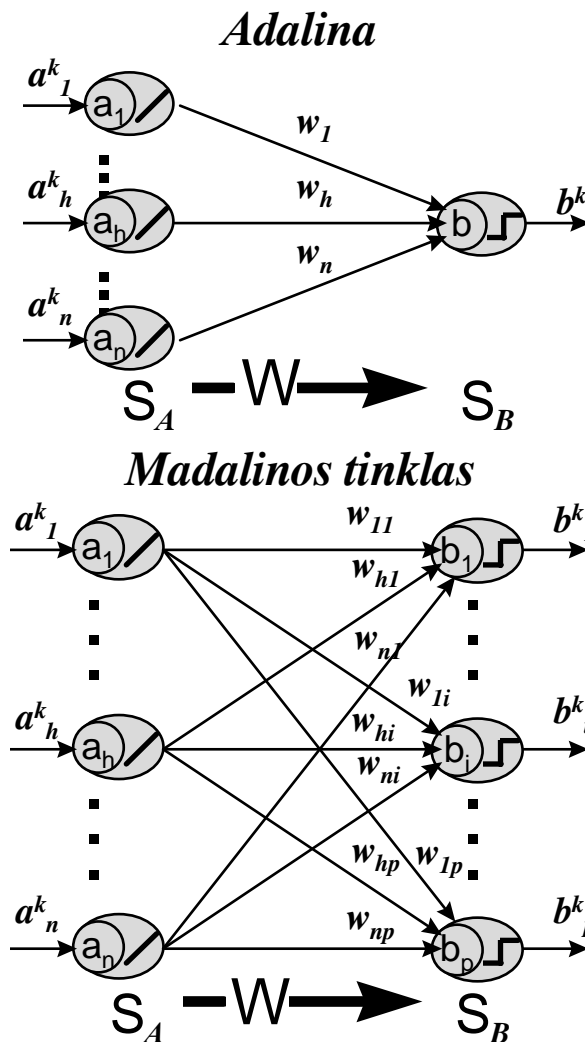
Apibrėžimas

Adalina (angl.k. - Adaline, nuo adaptive linear element) yra dviejų sluoksnių nuoseklus tinklas su n dirbtinių neuronų įėjimo sluoksnyje ir vienu dirbtiniu neuronu išėjimo sluoksnyje.

Madalina (angl. k. - Madaline, nuo multiple adaline) yra kelių adalinių tinklas, sudarantis įprastą dviejų sluoksnių nuoseklių ryšių struktūrą. Jis yra heteroasociatyvus, artimiausios reikšmės pavyzdžio atitikmens radimo įrenginys, kuris išimena pavyzdžių poras (A^k, B^k) , $k=1, 2, \dots, m$ naudodamas mažiausios vidutinės kvadratinės (MVK) klaidos korekcijos procedūrą, kur k -toji pavyzdžių pora pateikiama analoginės reikšmės vektoriumi $A^k = (a_1^k, \dots, a_n^k)$ ir dvejetainės $\{-1; +1\}$ reikšmės vektoriumi $B^k = (b_1^k, \dots, b_p^k)$.

Madalinos tinklas mokinamas nerealiame laiko mastelyje, veikia diskrečiame laike ir yra atvaizduojamas dviejų sluoksnių nuoseklių ryšių struktūra. Jo išėjimo signalai gali įgyti bet kokią reikšmę skirtingai nuo $\{-1; +1\}$ reikšmių, gaunamų perceptronų tinklu.

Struktūra



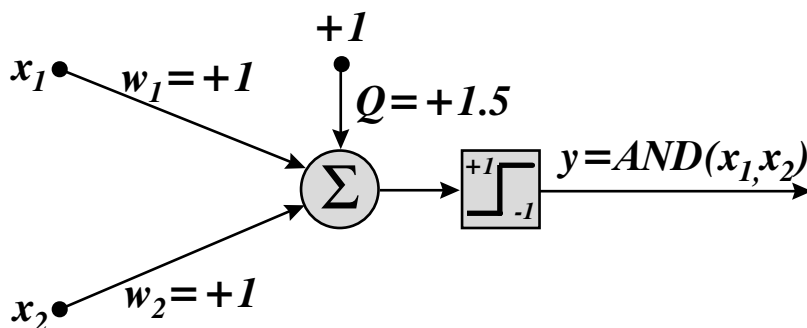
Skaidrės Nr.
2



Adalina. Madalina (tęsinys)

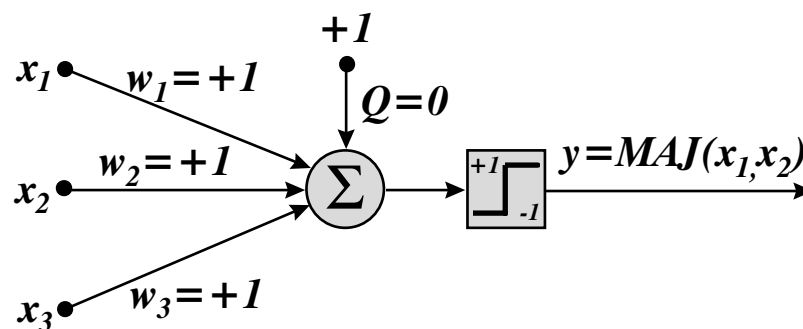
Loginių funkcijų realizavimo pavyzdys

Loginė sandauga



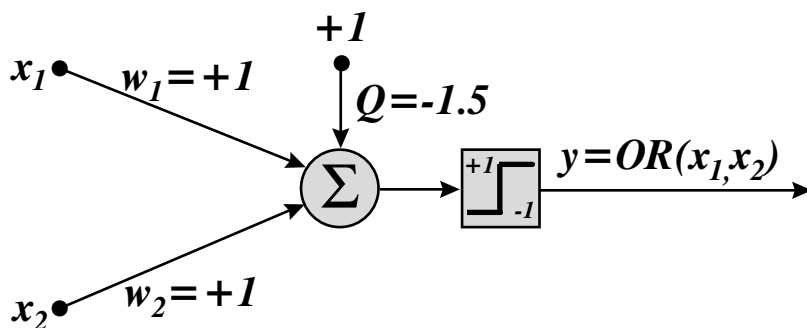
Pav. 5.3

Loginė dauguma



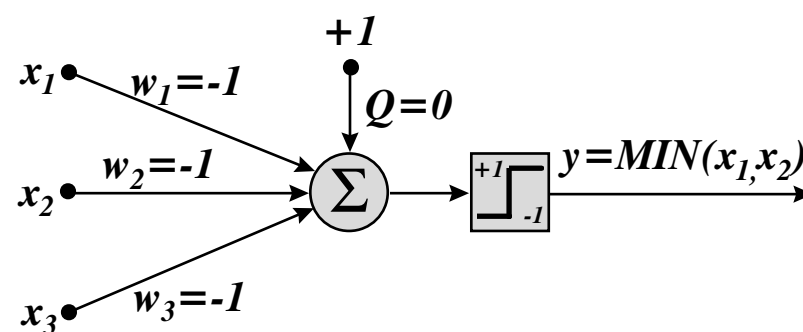
Pav. 5.4

Loginė suma



Pav. 5.5

Loginė mažuma



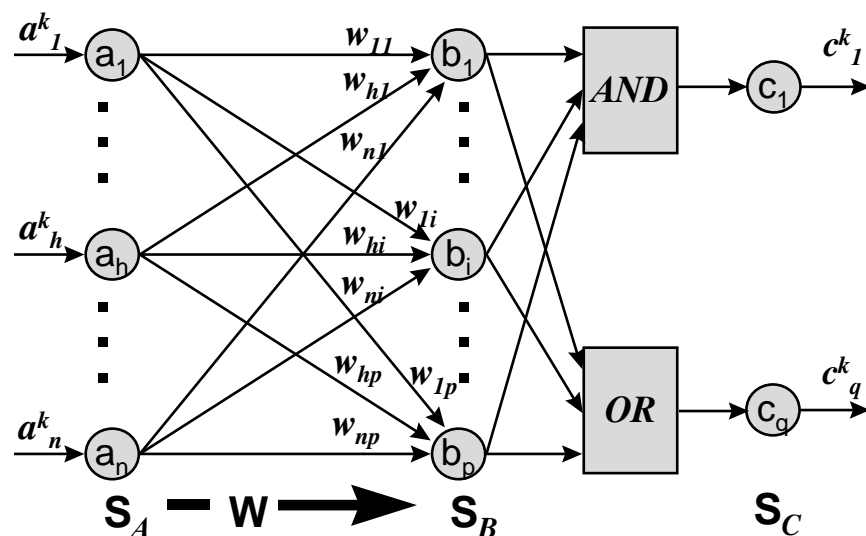
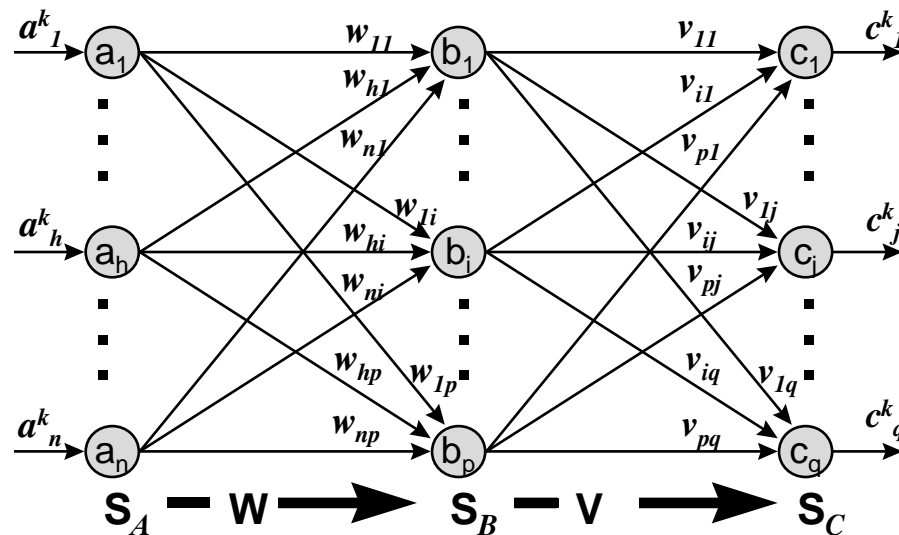
Pav. 5.6



Madalinos tinklo struktūros

*Daugiasluoksni
madalinos tinklo
struktūra*

*Mišrios logikos
daugiasluoksni
madalinos tinklo
struktūra*



Pav. 5.7, 5.8



Madalinos tinklo mokymas

Madalinos taisyklė I (mažiausios vidutinės kvadratinės (MVK) klaidos korekcijos procedūra)

A. Ryšių bei slenksčių inicializavimas

Visiems adaptuojamiems madalinos tinklo ryšių w_{ij} bei slenksčių Q_j svoriams priskiriamos atsitiktinės reikšmės (+1,-1) ribose.

B. Įėjimo pavyzdžių pateikimas

Įėjimo pavyzdžiai A^k , $k=1,\dots,n$ pateikiami madalinos tinklui ir į juos gaunami atsakai B^k :

$$b_j^k = \sum_{i=1}^n w_{ij}^k \cdot a_i^k - Q_j^k.$$

C. Tikrinimas

Skaiciuojama klaida D^k tarp esamų B^k ir norimų gauti B^* atsakų:

$$d_j^k = b_j'^k - b_j^k,$$

jei klaida norimai maža arba lygi nuliui, tai peršokama į algoritmo E žingsnį.

D. Ryšių bei slenksčių adaptavimas

Prie esamų ryšių bei slenksčių svorių reikšmių pridedama:

$$\Delta w_{ij}^k = \lambda \cdot a_i^k \cdot d_j^k \quad \text{ir} \quad \Delta Q_j^k = \lambda \cdot d_j^k, \quad \text{kur } \lambda - \text{mokymo greičio konstantė.}$$

Grįžtama prie algoritmo B žingsnio.

E. Pabaiga



Madalinos tinklo mokymas (tęsinys).

Madalinos tinklo atsakas

Madalinos taisyklė II (daugiasluoksnio tinklo mokymo procedūra)

- A. Ryšių ir slenksčių svorių inicializavimas.*
- B. Įėjimo pavyzdžių pateikimas ir tinklo atsako į juos radimas. Tinklo klaidos (skirtumo tarp norimo ir esamo atsako) skaičiavimas.*
- C. Jei klaida norimai maža, tai peršokimas į H algoritmo žingsnį.*
- D. Neuronų (pradedant nuo įėjimo sluoksnio), kurio išėjimo signalas arčiausiai 0, radimas ir jo ryšių svorių koregavimas, kad jo išėjimo signalas pakeistų ženklą.*
- E. Pakartotinas tinklo klaidos skaičiavimas. Klaidai nesumažėjus - paskutiniojo ryšių svorių koregavimo atmetimas.*
- F. Grįžimas į C algoritmo žingsnį, jei dar ne visi sluoksnio neuronai arba jų kombinacijos koreguoti.*
- G. Sekančio neuronų sluoksnio ryšių svorių koregavimas grįžtant į C algoritmo žingsnį, jei dar ne visi neuronų sluoksniai koreguoti.*
- H. Pabaiga.*

Skaidrės Nr.

6

Madalinos tinklo atsakas

Atsakas formuojamas naudojant lygtį:

$$b_j^k = F\left(\sum_{i=1}^n w_{ij}^k \cdot a_i^k - Q_j^k\right), \text{ kai } F(x) = \begin{cases} +1, & x > 0; \\ -1, & x \leq 0. \end{cases}$$



Konvergencija

Norint įrodyti MVK algoritmo globalią konvergenciją reikia išanalizuoti jo vidutinę kvadratinę paklaidą.

Kiekvieno dirbtinio neuroono klaida, kuri yra išreiškiama kaip skirtumas tarp realaus ir norimo atsako, yra randama naudojant šią lygtį:

$$d_j = b'_j - b_j = b'_j - \sum_{i=1}^n w_{ij} a_i - Q_j.$$

Įvedame kintamojo Q_j pakeitimą sandauga $w_{0j} a_0$, tada:

$$d_j = b'_j - \sum_{i=0}^n w_{ij} a_i \text{ arba } d_j = b'_j - A_k W_j^T.$$

Kvadratinė klaida išreiškiama sekančiais:

$$\begin{aligned} d_j^2 &= (b'_j)^2 - 2b'_j A_k W_j^T + [A_k W_j^T]^2 = \\ &= (b'_j)^2 - 2b'_j A_k W_j^T + W_j A_k^T A_k W_j^T. \end{aligned}$$

Priėmę, kad įėjimo signalai yra stacionarūs, absoliučius S_A ir S_B sluoksnių pavyzdžius pakeičiame jų vidurkiais ir tada gauname sekančią vidutinių kvadratų klaidos lygtį:

$E\{\}$ išreiškiama vidutinių operacija. Gautą lygtį galime išreikšti trumpiau įvedus sekančius pažymėjimus:*

$$P = E \{ b'_j A_k \}, \text{ kur}$$



Konvergencija (tęsinys)

P yra įėjimo sluoksnio S_A pavyzdžių ir norimo atsako koreliacijos vektorius, ir

$$R = E \{ A_k^T A_k \}, \text{ kur}$$

R yra įėjimo sluoksnio S_A pavyzdžių autokoreliacinė matrica. Supaprastinta MVK lygtis gaunama tokia:

$$E \{ (d_j)^2 \} = E \{ (b'_j)^2 \} - 2 P W_j^T + W_j R W_j^T.$$

Ieškant MVK lygties minimalaus sprendinio reikia šią lygtį diferencijuoti svorių vektoriaus atžvilgiu:

$$\frac{\partial}{\partial W_j} E \{ (d_j)^2 \} = 0 - 2 P + 2 W_j R.$$

Prilyginus gautą lygtį nuliui ir išsprendus gauname:

$$W_j = P R^{-1},$$

o tai yra ne kas kita kaip Wiener-Hopf lygties matricinė forma. Šis rezultatas įrodo, kad MVK algoritmas suras optimalų (mažiausių kvadratų prasme) klaidos svorių vektorių, jei egzistuoja inversinė R matrica.

Be to yra įrodyta, kad madalinos tinklas konverguos, jei mokymo koeficientas α tenkina šias sąlygas:

$$0 < \alpha < (\lambda_{max})^{-1}, \text{ kur}$$

λ_{max} yra didžiausia autokoreliacinės matricos R charakteringoji reikšmė.



Adalinių/madalinos tinklo sąvybės

Privalumai

- *galimybė išsaugoti dvigubai daugiau pavyzdžių negu tinklo įėjimo sluoksnio dydis*
- *lengvai suprantamas veikimas*
- *betarpiškas atsakas*

Trūkumai

- *nesugebėjimas mokytis realiame laiko mastelyje*
- *ilgas mokymas nerealiame laiko mastelyje*
- *sugebėjimas formuoti tik tiesinius (A^k , B^k) sąryšius*

Naudojimas

- *triukšmo slopinime*
- *pavyzdžių atpažinime*
- *adaptyviame apdorojime*
- *antenu sistemose*
- *vaizdo apdorojime*
- *valdymo uždaviniuose*



Pagrindiniai teiginiai

- ✎ *Adalinų/madalinos tinkluose naudojamos slenkstinės neuronų aktyvavimo funkcijos, o jie mokinami naudojant įvairias MVK algoritmo modifikacijas (madalinos taisyklė I, madalinos taisyklė II).*
- ✎ *Mišrios logikos daugiasluoksniame madalinos tinkle naudojami ir įprastos logines funkcijas vykdančys elementai.*
- ✎ *MVK klaidos korekcijos procedūros esmė: svorių pokyčiai yra tiesiogiai proporcingi tinklo klaidos dydžiui, kai tinklas atsaką formuoja be neuronų aktyvavimo funkcijų.*
- ✎ *Mokant adalinų tinklą, skirtingai negu perceptronų tinklų atveju, mažinama MVK klaida.*
- ✎ *Adalinų tinklų konvergencija yra įrodyta. Pagrindinis reikalavimas - reikia parinkti mokymo koeficientą mažesniu negu įėjimo pavyzdžių autokoreliacinės matricos didžiausia charakteringoji reikšmė.*
- ✎ *Pagrindinis adalinų tinklų privalumas - sąryšių formavimas mažinant MVK klaidą, didžiausias trūkumas - nesugebėjimas įsisąvinti netiesinius sąryšius.*
- ✎ *Jei mokymo žingsnis yra pakankamai mažas tai net neegzistuojant tiksliai sprendiniui adalinų tinklas minimuos vidutinę kvadratinę klaidų sumą. Šis tinklas ras labiausiai artimą sprendinį, apspręstą tiesine DNT architektūros prigimtimi*