



Vilnius universitetas
Matematikos ir informatikos fakultetas
Informatikos katedra

Skaitmeninis intelektas ir sprendimų priėmimas (tiesioginio sklidimo dirbtiniai neuroniniai tinklai)

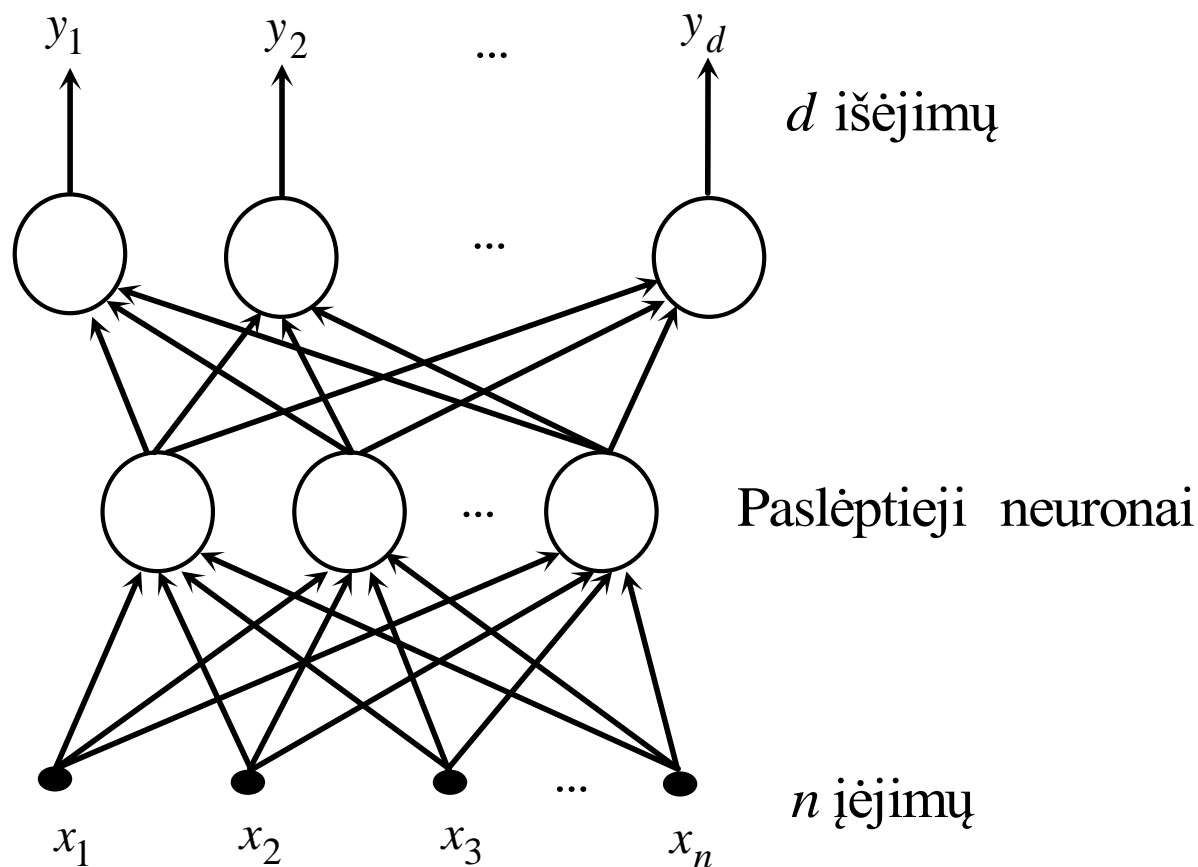
doc. dr. Olga Kurasova
Olga.Kurasova@mii.vu.lt

2017

Daugiasluoksniai tiesioginio sklidimo neuroniniai tinklai

- Turintys daugiau nei vieną neuronų sluoksnį tinklai, kuriuose galimi tik ryšiai į priekį iš įėjimų į išėjimus, yra vadinami
 - **daugiasluoksniais perceptronais** (*multilayer perceptrons*),
 - arba **daugiasluoksniais tiesioginio sklidimo neuroniniais tinklais** (*multilayer feedforward neural networks*).

Tiesioginio sklidimo DNT (1)



Kiekvienas neuronų sluoksnis turi po papildomą įėjimą ir jo jungtis su to sluoksnio neuronais, tačiau paprastumo dėlei jie paveiksle nėra pavaizduoti.

Tiesioginio sklidimo DNT (2)

- Tegul turime daugiasluoksnį neuroninį tinklą, kuriame yra **L sluoksnių**, pažymėtų $l = 0, 1, \dots, L$, čia
 - sluoksnis $l = 0$ žymi **įėjimus**,
 - $l = 1, \dots, L-1$ žymi **paslėptus sluoksnius**,
 - o $l = L$ paskutinį (**išėjimų**) sluoksnį.
- Kiekviename sluoksnyje l yra n_l neuronų.

Tiesioginio sklidimo DNT (3)

- Pirmojo sluoksnio j -ojo neurono išėjimo reikšmė y_j yra apskaičiuojama pagal formulę.

$$y_j = f(a_j) = f\left(\sum_{k=0}^n w_{jk} x_k\right), \quad j = 1, \dots, d$$

- čia w_{jk} yra jungties iš k -ojo įėjimo į j -ąjį neuroną svoris.
- Įėjimai į neuronus l -ajame sluoksnyje yra neuronų išėjimai $(l - 1)$ -ajame sluoksnyje.

Tiesioginio sklidimo DNT (4)

- Todėl kiekvieno j -ojo neurono išėjimo reikšmė l -ajame sluoksnyje yra apskaičiuojama taip:

$$y_j = f(a_j) = f\left(\sum_{k=0}^{n_{l-1}} w_{jk} y_k\right), \quad j = 1, \dots, n_l$$

- čia $f(\)$ yra neuronų aktyvacijos funkcija,
- w_{jk} – svoriai jungčių, kurios jungia k -ąjį neuroną $(l - 1)$ -ajame sluoksnyje su j -uoju neuronu l -ajame sluoksnyje,
- n_{l-1} – neuronų skaičius $(l - 1)$ -ajame sluoksnyje, $y_0 = 0$.
- Kairiojoje lygybės pusėje y_j yra l -ojo sluoksnio j -ojo neurono išėjimo reikšmė,
- o dešiniojoje – y_k yra $(l - 1)$ -ojo sluoksnio k -ojo neurono išėjimo reikšmė.

„Klaidos skleidimo atgal“ mokymo algoritmas (1)

- Taikant vienasluoksnio perceptrono mokymo idėją daugiasluoksniame neuroniniame tinklui, **būtina žinoti paslėptųjų neuronų išėjimų reikšmes.**
- Jei paklaidos ir aktyvacijos funkcijos yra diferencijuojamos, ieškant minimalios paklaidos gali būti **naudojama gradientinio nusileidimo strategija.**
- Algoritmas, kuris realizuoja gradientinio nusileidimo mokymo strategiją daugiasluoksniame tiesioginio sklaido neuroniniame tinklui, vadinamas **„klaidos skleidimo atgal“ algoritmu** (*back-propagation learning algorithm*).

„Klaidos skleidimo atgal“ mokymo algoritmas (2)

- Algoritmą sudaro du žingsniai:
 - įėjimų reikšmių „**skleidimas pirmyn**“ iš įėjimų į išėjimų sluoksnį;
 - paklaidos „**skleidimas atgal**“ iš išėjimų į įėjimų sluoksnį.
- Tiek perceptrono mokyme, tiek „klaidos skleidimo atgal“ algoritme naudojama **mokymo su mokytoju strategija**.

„Klaidos skleidimo atgal“ mokymo algoritmas (3)

- Pirmame algoritmo žingsnyje **įėjimų** vektoriui X_i apskaičiuojamas **išėjimų** vektorius $Y_i = (y_{i1}, y_{i2}, \dots, y_{id})$.
- Įvertinama **paklaidos funkcija** $E_i(W)$ išėjimų sluoksnyje L .

$$E_i(W) = \frac{1}{2} \sum_{j=1}^d (y_{ij} - t_{ij})^2$$

- čia t_{ij} – norimo j -ojo išėjimo reakcija į vektorių X_i .
- Tuo baigiama „**skleidimo pirmyn**“ fazė.

„Klaidos skleidimo atgal“ mokymo algoritmas (4)

- Jei paklaidos funkcija $E_i(W)$ nelygi nuliui, reikia keisti jungčių svorius.
- Panašiai kaip ir vienasluoksniame perceptrone, visi svoriai w_{jk} jungčių, kurios jungia k -ąjį neuroną $(l - 1)$ -ajame sluoksnyje su j -uoju neuronu l -ajame sluoksnyje, keičiami naudojantis formule

$$\Delta w_{jk}^i = -\eta \frac{\partial E_i}{\partial w_{jk}}$$

„Klaidos skleidimo atgal“ mokymo algoritmas (5)

- Dalinės išvestinės išreiškiamos taip:

$$\frac{\partial E_i}{\partial w_{jk}} = \frac{\partial E_i}{\partial a_{ij}} \cdot \frac{\partial a_{ij}}{\partial w_{jk}}$$

- Kadangi $a_{ij} = \sum_{k=0}^{n_l-1} w_{jk} y_{ik}$, tai $\frac{\partial a_{ij}}{\partial w_{jk}} = y_{ik}$.
- Tegul $\delta_{ij} = \frac{\partial E_i}{\partial a_{ij}}$, tuomet $\frac{\partial E_i}{\partial w_{jk}} = \delta_{ij} y_{ik}$
- ir $\Delta w_{jk}^i = -\eta \delta_{ij} y_{ik}$,
- čia j -asis neuronas priklauso l -ajam sluoksniui, k -asis neuronas priklauso $(l - 1)$ -ajam sluoksniui.

„Klaidos skleidimo atgal“ mokymo algoritmas (6)

- Išėjimų sluoksnyje

$$\delta_{ij} = \frac{\partial E_i}{\partial a_{ij}} = f'(a_{ij})(y_{ij} - t_{ij})$$

- čia j -asis neuronas priklauso išėjimų sluoksniui L .
- Turime rasti $\delta_{ij} = \frac{\partial E_i}{\partial a_{ij}}$ **paslėptiems neuronas**,
t. y. kai j -asis neuronas priklauso l -ajam sluoksniui ir $l < L$.

„Klaidos skleidimo atgal“ mokymo algoritmas (7)

- Naudojantis dalinėmis išvestinėmis, bendruoju atveju galima parašyti

$$\delta_{ij} = \frac{\partial E_i}{\partial a_{ij}} = \sum_{s=1}^{n_{l+1}} \frac{\partial E_i}{\partial a_{is}} \cdot \frac{\partial a_{is}}{\partial a_{ij}}$$

- čia n_{l+1} žymi neuronų $(l + 1)$ -ajame sluoksnyje skaičių.
- Išraiška $\frac{\partial E_i}{\partial a_{is}}$ yra lygi dydžiui δ_{is} , apibrėžtam s -ajam neuronui $(l + 1)$ -ajame sluoksnyje.

„Klaidos skleidimo atgal“ mokymo algoritmas (8)

- Turint $\frac{\partial a_{is}}{\partial a_{ij}} = f'(a_{ij})w_{is}$

- Paslėptųjų j -ųjų neuronų

$$\delta_{ij} = f'(a_{ij}) \sum_{s=1}^{n_{l+1}} w_{is} \delta_{is}$$

- čia j -asis neuronas priklauso sluoksniui $l < L$,
 s -asis neuronas priklauso $(l + 1)$ -ajam
sluoksniui.

„Klaidos skleidimo atgal“ mokymo algoritmas (9)

Apibendrinimas:

- Iš pradžių reikia apskaičiuoti δ_{ij} reikšmes **išėjimų sluoksnyje** L .
- Tada palaipsniui skaičiuoti δ_{ij} reikšmes **paslėptiems neuronams** tarpiniuose sluoksniuose $l < L$ naudojantis $(l + 1)$ -ųjų sluoksnių δ_{ij} reikšmėmis.
- Kai visi svoriai pakeičiami, **į tinklą pateikiamas sekantis mokymo vektorius** ir procesas kartojamas iš naujo.

Algoritmo **sustojimo kriterijus** yra

- arba iš anksto nustatyta **paklaidos funkcijos slenksčio reikšmė**,
- arba atitinkamas **atliktų iteracijų** (mokymo žingsnių) skaičius.

„Klaidos skleidimo atgal“ mokymo algoritmas (10)

- Norint pagreitinti mokymo procesą, yra **koreguojama mokymo taisyklė** ir gaunama tokia svorių pokyčio formulė:

$$\Delta w_{jk}^i(t) = -\eta \delta_{ij}(t) y_{ik}(t) + \alpha \Delta w_{jk}^i(t-1)$$

- čia t yra iteracijos numeris,
- η – yra teigiamas daugiklis, kuris vadinamas **mokymo greičiu** (*learning rate*).
- α – teigiama konstanta ($0 < \alpha \leq 1$), vadinamoji **momento konstanta** (*momentum constant*).