Vilniaus universitetas

Matematikos ir informatikos fakultetas

Informatikos katedra

Dirbtinis neuronas – perceptronas

prof. dr. Olga Kurasova

Olga.Kurasova@mii.vu.lt

2018

Dirbtiniai neuroniniai tinklai (DNT)

- Viena iš dirbtinio intelekto sričių yra **dirbtiniai neuroniniai tinklai**, kurie, jei aišku iš konteksto, vadinami tiesiog neuroniniais tinklais.
- Jie pradėti tyrinėti kaip **biologinių neuroninių sistemų modelis**, siekiant išsiaiškinti ir pritaikyti biologinių neuronų sąveikos mechanizmus efektyvesnėms informacijos apdorojimo sistemoms kurti.
- Neuroniniai tinklai **turi galimybę mokytis** iš pavyzdžių.
- Turint duomenų pavyzdžius ir naudojant **mokymo algoritmus**, neuroninis tinklas pritaikomas prie duomenų struktūros ir **išmoksta atpažinti naujus** duomenis, kurie nebuvo naudojami tinklo mokyme.

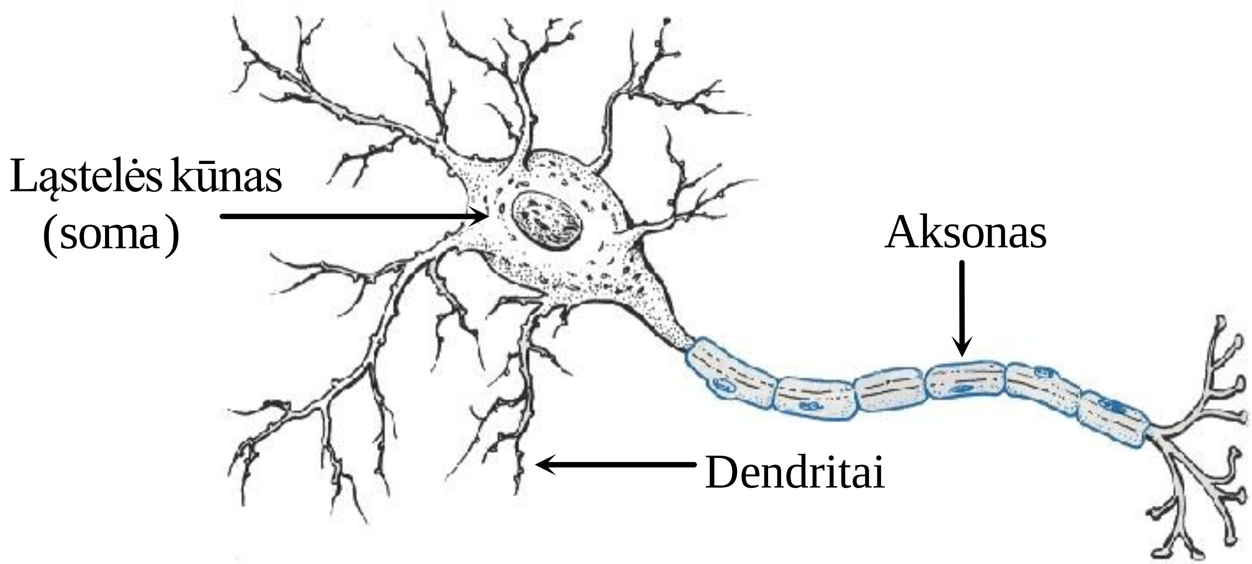
Biologinis neuronas

- Žmogaus **smegenys** susideda iš daugelio (apie 10^{13}) **neuronų**, sujungtų vienų su kitais.
- Kiekvienas neuronas turi vidutiniškai keletą tūkstančių **jungčių**.
- **Neuronas** – tai ląstelė, galinti generuoti elektrocheminį signalą.

Biologinis neuronas

Neuronas turi

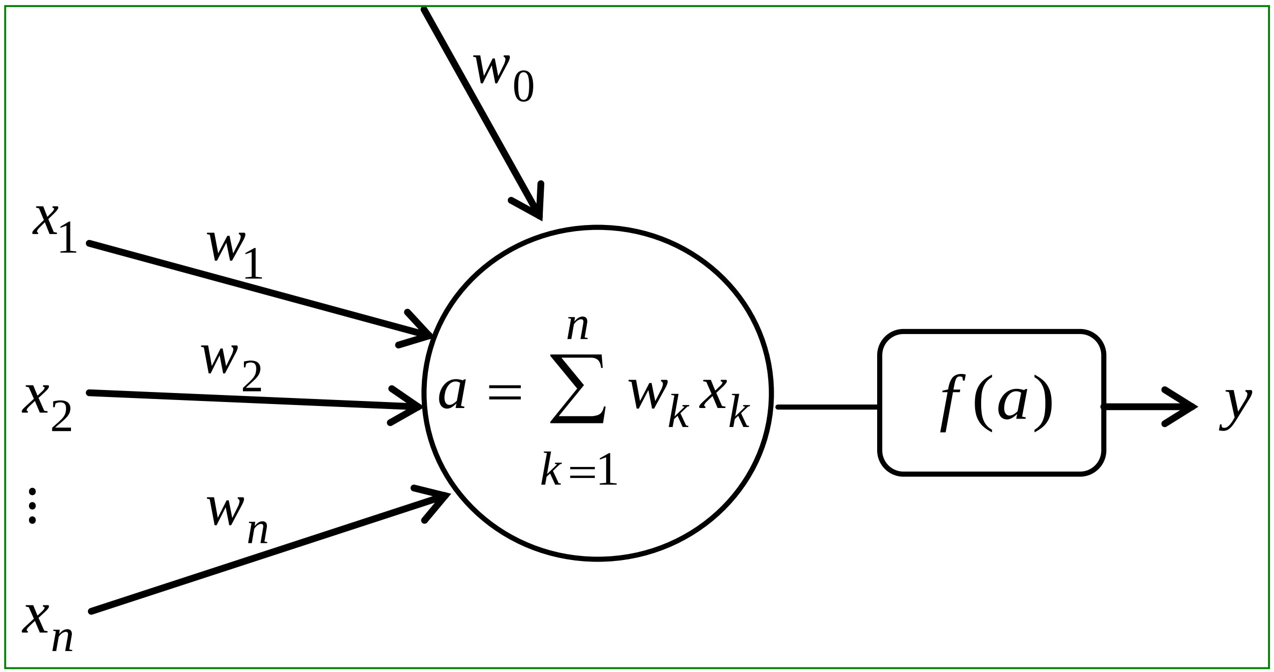
- išsišakojusią įėjimo struktūrą, vadinamuosius **dendritus**,
- ląstelės kūną, vadinamąją **somą**,
- ir besišakojančią išėjimo struktūrą – **aksoną**.



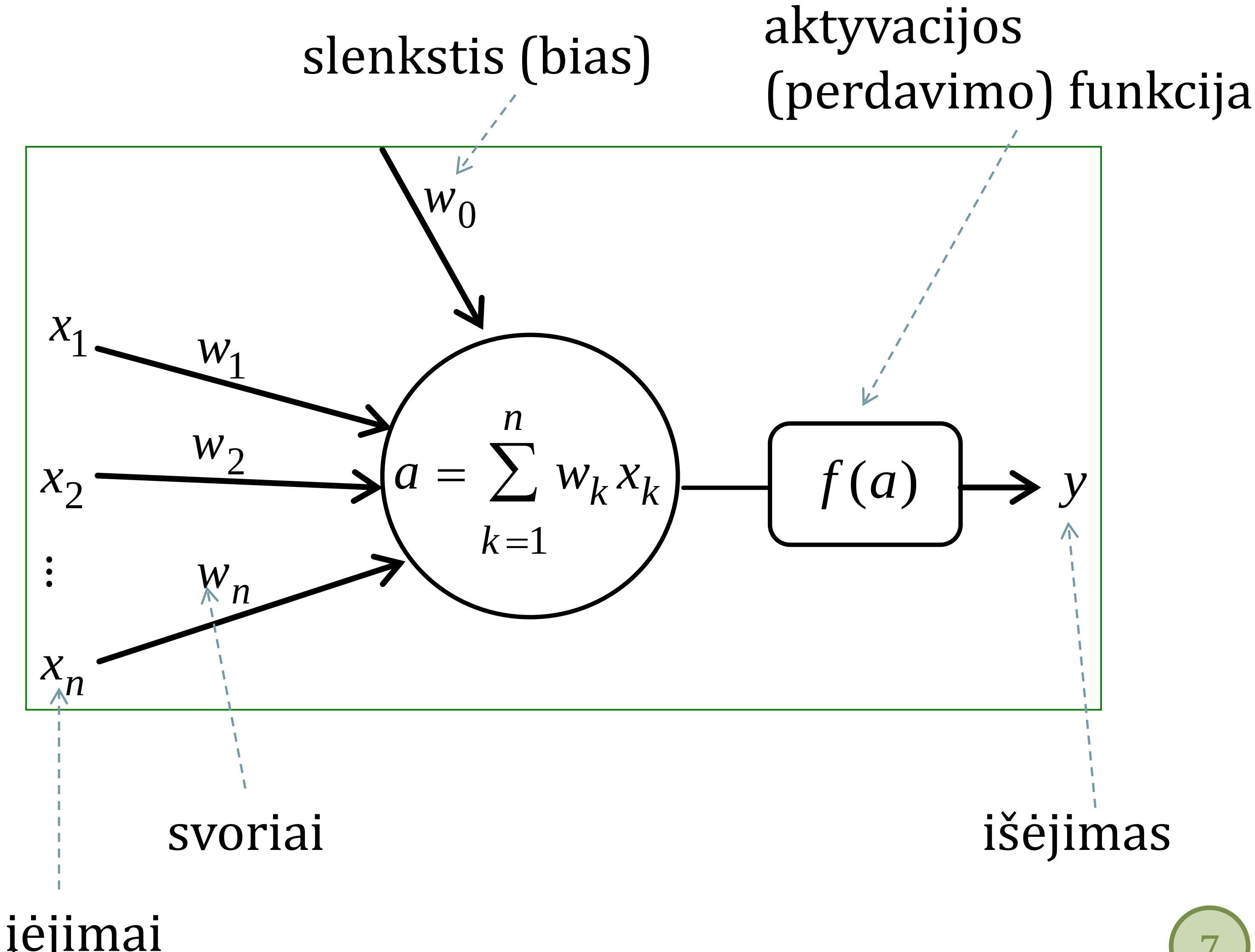
Biologinis neuronas

- Vienos ląstelės aksonas su kitos ląstelės dendritais jungiasi per **sinapses**.
- Kai sužadinama pakankamai neuronų, prijungtų prie neurono dendritų, tas **neuronas taip pat sužadinamas** ir generuoja elektrocheminį impulsą.
- **Signalas** per sinapses perduodamas kitiems neuronams, kurie vėl gali būti sužadinami.
- Neuronas sužadinamas tik tuo atveju, jei bendras dendritais gautas signalas viršija tam tikrą lygį, vadinamąjį **sužadinimo slenkstį**.
- Turint **didžiulį skaičių visiškai paprastų elementų**, kurių kiekvienas skaičiuoja svorinę įeinančių signalų sumą ir generuoja binarinį signalą, jei suminis signalas viršija tam tikrą lygį, **galima atlikti gana sudėtingas užduotis**.

Dirbtinio neurono modelis



Dirbtinio neurono modelis



Dirbtinio neurono modelis

- Neuronas turi keletą **įėjimų** x_1, x_2, ..., x_n.
- Kiekviena įėjimo x_k, $k = 1, ..., n$, **jungtis** turi savo perdavimo koeficientą (**svorį**) w_k, $k = 1, ..., n$.
- Įprastai įėjimų ir jungčių svorių reikšmės yra **realieji skaičiai**.
- Skaičiuojama įėjimo reikšmių ir svorių sandaugų **suma**

$$a = w_1 x_1 + w_2 x_2 + \ldots + w_n x_n = \sum_{k=1}^{n} w_k x_k$$

Dirbtinio neurono modelis

- Neuroną apibūdina **aktyvacijos (perdavimo) funkcija**

$$y = f(a) = f\left(\sum_{k=1}^{n} w_k x_k\right)$$

kurios reikšmė

$$f(a) = \begin{cases} 1, & \text{jei } a \geq w_0, \\ 0, & \text{jei } a < w_0, \end{cases}$$

vadinama **neurono išėjimo reikšme**.

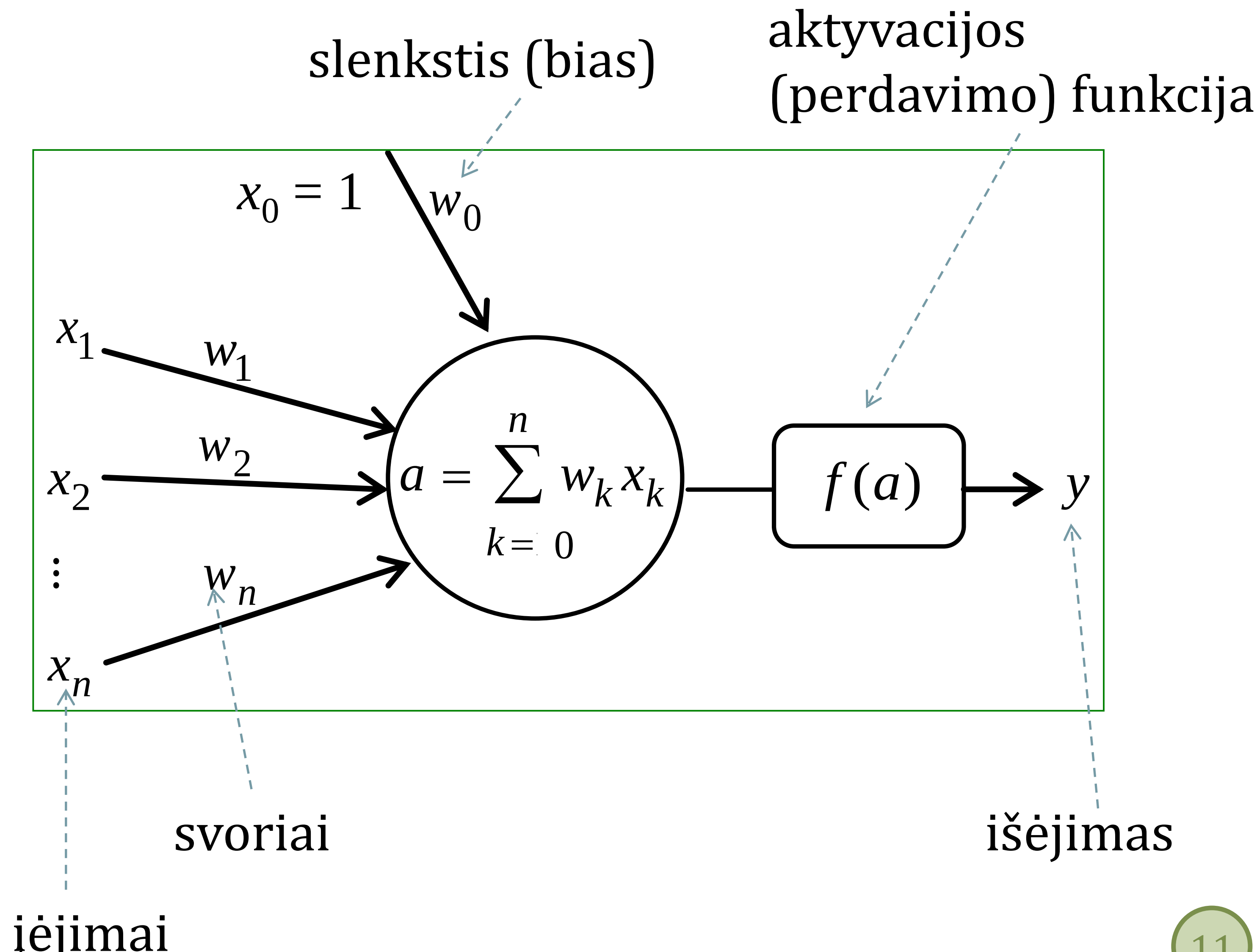
- Čia w_0 yra **slenksčio reikšmė** (bias).

Dirbtinio neurono modelis

- Dažnai yra įvedamas **nulinis įėjimas** x_0, kuris yra pastovus, $x_0 = 1$, o slenksčio reikšmė tampa **nuliniu svoriu** w_0.
- Tuomet

$$a = \sum_{k=0}^{n} w_k x_k$$

Dirbtinio neurono modelis



Aktyvacijos funkcijos

- Tiesinė

$$f(a) = \begin{cases} 1, \text{ if } a \geq 0 \\ 0, \text{ if } a < 0 \end{cases}$$

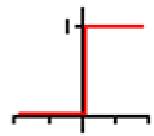
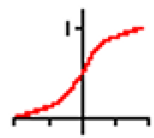
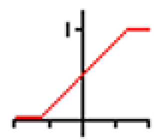
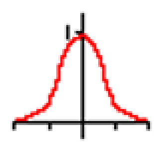
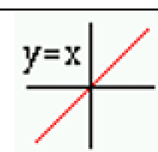
- Sigmoidinė

$$f(a) = \frac{1}{1 + \mathrm{e}^{-a}}$$

- Hiperbolinis tangentas

$$f(a) = \tanh(a) = \frac{\mathrm{e}^{a} - \mathrm{e}^{-a}}{\mathrm{e}^{a} + \mathrm{e}^{-a}}$$

Aktyvacijos funkcijos

Slenkstinė (angl. *unit step*)		$f(x) = \begin{cases} 0 \text{ if } 0 > x \\ 1 \text{ if } x \geq 0 \end{cases}$
Sigmoidinė (angl. *sigmoid*)		$f(x) = \dfrac{1}{1+e^{-\beta x}}$
Gabalais tiesinė (angl. *piecewise linear*)		$f(x) = \begin{cases} 0 & \text{if } x \leq x_{min} \\ mx+b & \text{if } x_{max} > x > x_{min} \\ 1 & \text{if } x \geq x_{max} \end{cases}$
Gauso (angl. *Gaussian*)		$f(x) = \dfrac{1}{\sqrt{2\pi\sigma}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$
Tiesinė (angl. *linear*)		$f(x) = x$

Neuronas duomenims klasifikuoti

- Dirbtinis **neuronas** dar vadinamas **perceptronu** arba **vienasluoksniu** perceptronu.
- Į **neurono įėjimus** paduodami objektus apibūdinančių požymių x_1, x_2, ..., x_n reikšmės.
- **Neurono išėjime** – duomenų klasių reikšmės.
- Tikslas – **rasti tokias svorių reikšmes** w_0, w_1, w_2, ..., w_n, kad apskaičiavus $a = w_0 x_0 + w_1 x_1 + \ldots + w_n x_n$ ir $f(a)$, **išėjime** y gautos reikšmės **sutaptų su duomenų klasių reikšmėmis**.
- Svorių reikšmės turi būti tokios, kad jos būtų **tinkamos visiems duomenims**.

Perceptrono mokymas

- Galimybė **mokytis** yra **esminė intelekto savybė**.
- Tinkamų svorių radimas vadinamas neurono (perceptrono) **mokymu**.
- Duomenų aibė, kuri bus naudojama neuronui mokyti, vadinama **mokymo aibe**.
- Tegul turime m mokymo aibės vektorių $X_i = (x_{i1}, x_{i2}, \ldots, x_{in})$, $i = 1, \ldots, m$, kuriuos vadinsime **įėjimų vektoriais**.
- Šie vektoriai yra susieti su **norima reikšme** t_i (*target*). Tai norima reakcija į vektorių X_i.
- Sprendžiant klasifikavimo uždavinį, **norimos reikšmės yra klasių numeriai**.

Perceptrono mokymas

- **Mokymo procese** svoriai $W = (w_0, w_1, w_2, \ldots, w_n)$ keičiami taip, kad tinklo išėjimo reikšmė y_i, gauta į įėjimą pateikus vektorių X_i, būtų kiek galima **artimesnė norimai reikšmei** t_i.
- Tai yra perceptrono veikimo **paklaida** būtų kiek galima **mažesnė**.
- Ši **paklaida** $E(W)$ gali būti apibrėžiama, kaip skirtumų tarp neurono išėjime gautų reikšmių ir norimų reikšmių sumos funkcija

$$E(W) = \sum_{i=1}^{m} (y_i - t_i)$$

Perceptrono mokymas

- Vadinasi perceptrono mokymo eigoje reikia **minimizuoti paklaidos funkciją** $E(W)$.
- Jeigu ši funkcija yra diferencijuojama pagal svorius, jos minimumą galima rasti **gradientiniais optimizavimo metodais**.
- Patogumo dėlei, dažnai minimizuojama **tokios išraiškos funkcija**.

$$E(W) = \frac{1}{2}\sum_{i=1}^{m}(y_i - t_i)^2$$

Perceptrono mokymas

- Iš pradžių **generuojamos atsitiktinės svorių** w_k reikšmės, įprastai intervale (0, 1).
- Tada gradientinio nusileidimo algoritmu judama antigradiento kryptimi, **svorių reikšmes keičiant** pagal iteracinę formulę

$$w_k(t+1) = w_k(t) + \Delta w_k(t)$$

- Čia t – iteracijos numeris,

$$\Delta w_k(t) = -\eta \frac{\partial E(W)}{\partial w_k}$$

- η – yra teigiamas daugiklis, kuris vadinamas **mokymo greičiu** (*learning rate*) ir kuriuo reguliuojamas gradientinio optimizavimo žingsnio ilgis.

Perceptrono mokymas

- Įprastai svoriai **pakeičiami pateikus vieną** įėjimo vektorių.
- Mokymo procesas kartojamas **daug kartų pateikiant** visus įėjimo vektorius.
- Mokymas **stabdomas** arba atlikus iš anksto nustatytą **iteracijų skaičių**, arba pasiekus norimą **mažą paklaidos reikšmę**.

Perceptrono mokymo pavyzdys

- Tegul mokymo duomenys – loginės funkcijos AND teisingumo lentelė.

	x_1	x_2	t
X_1	1	1	+1
X_2	1	0	– 1
X_3	0	1	– 1
X_4	0	0	– 1

- Dar reikia pridėti vieną stulpelį x_0=(1, 1, 1, 1)
- Svorių vektorių sudaro trys komponentės $W(w_0, w_1, w_2)$.

Perceptrono mokymo pavyzdys

- **Tikslas** – rasti tokias svorių reikšmes w_0, w_1, w_2, kad išėjime y_i, gautos reikšmės sutaptų su norimomis reikšmėmis t_i, t. y. paklaida $E(W) = 0$.
- Naudosime perceptroną, kurio **aktyvacijos funkcija** yra

$$y = f(a) = \begin{cases} +1, \text{jei } a > 0 \\ -1, \text{jei } a \leq 0 \end{cases}$$

- Pradinės svorių reikšmės lygios nuliui, $w_k = 0$.
- Paprastumo dėlei, tegul $\eta=1$.

Perceptrono mokymo algoritmas

WHILE (išėjimo reikšmės nelygios trokštamoms reikšmėms ir iteracijų skaičius neviršija nustatytąjį)

{ FOR (visiems įėjimo vektoriams X_i)

{

$a_i = w_0 x_0 + w_1 x_1 + w_2 x_2$

$y_i = f(a)$

IF $(y_i \neq t_i)$

w_k (naujas) = w_k (senas) + $\eta t_i x_{ik}$

}

}

Paklaidos minimizavimas

- **Minimizuojama** funkcija $E(W) = \frac{1}{2}\sum_{i=1}^{m}(y_i - t_i)^2$.
- Tai suma paklaidų **kiekvienam** įėjimų vektoriui: $E(W) = \sum_{i=1}^{m} E_i$
- Prisiminkime, kad $y_i = f(a_i) = f(\sum_{k=0}^{n} w_k x_{ik})$.
- Randama funkcijos **išvestinė** pagal w_k:

$$\frac{\partial E_i(W)}{\partial w_k} = (y_i - t_i) \times \frac{\partial f(a)}{\partial w_k} \times \frac{\partial a}{\partial w_k} = (y_i - t_i) x_{ik}$$

kai $f(\sum_{k=0}^{n} w_k x_{ik}) = \sum_{k=0}^{n} w_k x_{ik}$.

- Todėl bendru atveju neurono mokyme galima taikyti **taisyklę**:

$$w_k(t+1) = w_k(t) + \eta(y_i - t_i)x_{ik}$$

Skiriamasis paviršius

- Neuronas **padalina** sprendinių aibę į du regionus.
- Nagrinėkime atvejį, kai įėjimų yra tik du x_1, x_2 ir

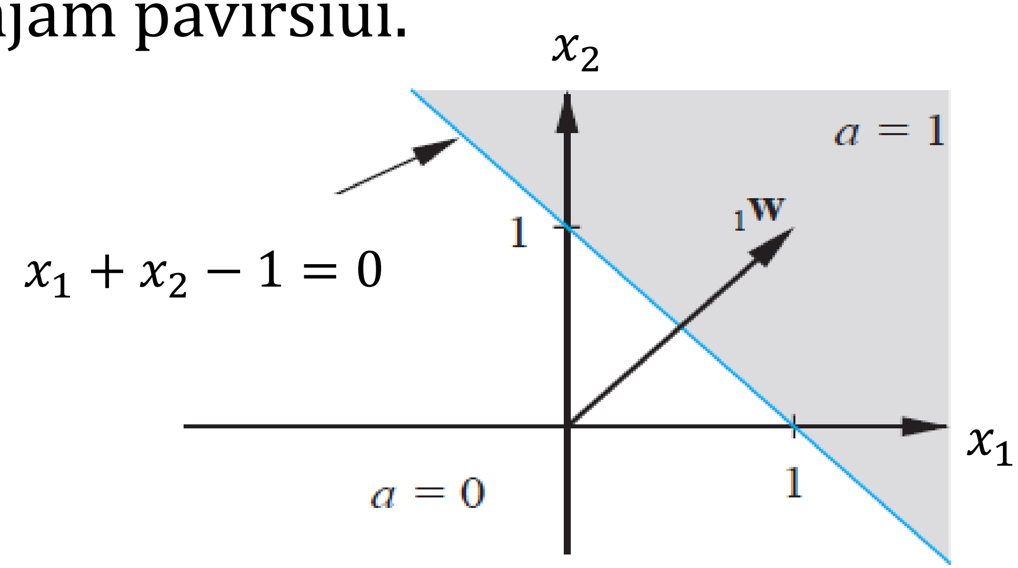
$$y = f(a) = \begin{cases} 1, \text{jei } a \geq 0 \\ 0, \text{jei } a < 0 \end{cases}$$

- Tuomet **skiriamasis paviršius** (*decision boundary*) (dviejų įėjimų atveju – tai teisė) apibrėžiamas įėjimų vektoriais, kuriems $a = 0$:

$$w_1 x_1 + w_2 x_2 + w_0 = 0$$

Pavyzdys

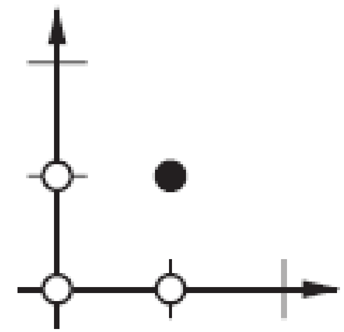
- Tarkime $w_1 = 1, w_2 = 1, w_0 = -1$. Tuomet **skiriamasis paviršius** $x_1 + x_2 - 1 = 0$.
- Šios tiesės vienoje pusėje, **neurono išėjimas** bus lygus 0, kitoje 1.
- Svorių vektorius turi būti **statmenas** (**ortogonalus**) skiriamajam paviršiui.



Skiriamasis paviršius loginei funkcijai AND (1)

- Nagrinėkime **pavyzdį**, kai funkcija AND apibrėžiama taip:

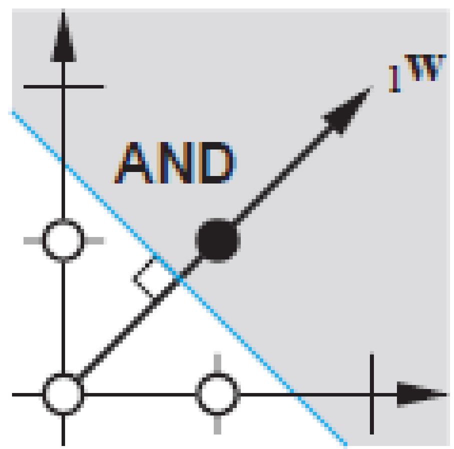
	x_1	x_2	t
X_1	0	0	0
X_2	0	1	0
X_3	1	0	0
X_4	1	1	1



- Tuščias apskritimas atitinka $t = 0$, skrituliukas $t = 1$.

Skiriamasis paviršius loginei funkcijai AND (2)

- Reikia nubrėžti skiriamąjį paviršių (melsva tiesė), kurios vienoje pusėje būtų apskritimai, kitoje – skrituliukas.



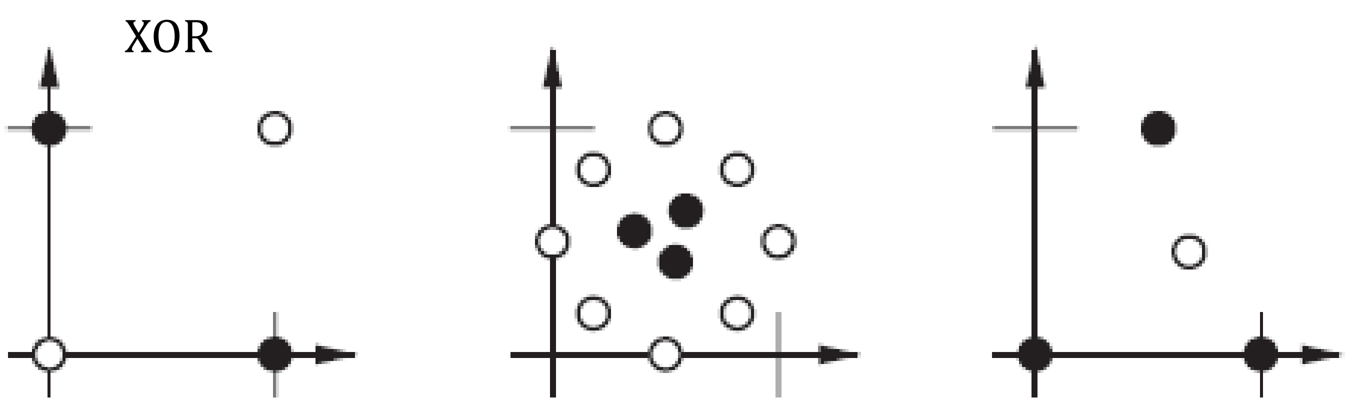
- Dabar reikia pasirinkti svorių vektorių, kuris būtų statmenas skiriamajai tiesei. Vienas variantų $W(w_1, w_2) = (2, 2)$.

Skiriamasis paviršius loginei funkcijai AND (3)

- Dabar belieka **rasti** w_0.
- Reikia **parinkti tašką** (x_1, x_2), esantį ant skiriamosios tiesės ir tenkinantį lygybę $w_1 x_1 + w_2 x_2 + w_0 = 0$.
- Galimas **taškas** $(x_1, x_2) = (1{,}5, 0)$.
- Tuomet $2 \times 1{,}5 + 2 \times 0 + w_0 = 0$. Iš čia $w_0 = -3$.

Tiesiškai neatskiriami atvejai

- Deja, daugelyje realių uždavinių **negalima** nubraižyti (suformuoti) **tiesiškai atskiriamo paviršiaus**.
- Tam reikia naudoti **sudėtingesnius neuroninius tinklus**.



„Kišeninis“ algoritmas (1)

- Perceptrono mokymo **„kišeninis“ algoritmas** (*pocket algorithm*) taikomas sprendžiant tiesiškai neatskiriamą uždavinį, ieškant **kiek galima geresnio teisiško atskyrimo**.
- Algoritmo metu „kišenėje“ saugomi gauti svoriai ir **naudojami rasti geriausi**. Svoriai keičiami, jei tik randami geresni.

„Kišeninis“ algoritmas (2)

```
pocket (training_list, max_iteration)
        w = randomVector()
        best_error = error(w)
        for i in range(0, max_iteration)
                x=misclassified_sample(w, training_list)
                w=vector_sum(w, x.y(x))
                if error(w) < best_error
                        best_w = w
                        best_error = error(w)
return best_w
```

Mokymo iteracija – mokymo epocha

- Neuronų mokyme naudojamos šios sąvokos „**mokymo iteracija**“, „**mokymo epocha**“.
- Koks **skirtumas** tarp jų?
- Mokymo **iteracija** – tai neuronų mokymo proceso dalis, kurios metu apdorojamas vienas įėjimų vektorius.
- Mokymo **epocha** – tai neuronų mokymo proceso dalis, kurios metu apdorojamas visas įėjimų vektorių rinkinys vieną kartą.
- Vienos mokymo epochos metu **įvyksta tiek iteracijų**, kiek yra įėjimo vektorių.

Dirbtinių neuronų ištakos

- **McCullogh-Pitts** (MCP, M-P) neurono modelis (1943)
  - Įėjimai tik (0, 1).
  - Tik slenkstinė aktyvacijos funkcija.
  - Vienintelė w_0 reikšmė visiems įėjimams.
  - Visiems įėjimams vienodi svoriai (teigiami skaičiai).
- **Rosenblatt** perceptronas (1958)
  - Visi svoriai nėra identiški (teigiami ir neigiami skaičiai).
  - Įvairios aktyvacijos funkcijos.
  - Yra mokymo taisyklė.

Mark I Perceptron mašina

- **Pirmoji perceptrono realizacija** buvo sukurta 1957 m. skaičiavimo mašinoje IBM 704 ne kaip programinė, bet **techninė įranga**.
- Ši mašina buvo skirta **vaizdų atpažinimo** (*image recognition*) uždaviniui spręsti.