VILNIAUS UNIVERSITETAS

MATEMATIKOS IR INFORMATIKOS FAKULTETAS

DUOMENŲ MOKSLO STUDIJŲ PROGRAMA

Laboratorinis darbas

Dirbtinis neuronas

Ieva Prevelytė

Vilnius
2025

Iliustracijų sąrašas

Lentelių sąrašas

Turinys

Žymėjimai

Žymėjimai naudojami laboratoriniame darbe:

- $w1$,$w2$ žymi neurono svorius.
- b žymi poslinkį.
- a žymi sužadinimo reikšmę.
- $f(a)$ žymi aktivacijos(perdavimo) funkciją.

Įvadas

Dirbtinis intelektas yra vis plačiau minimas ne tik informatikos srityse, bet ir bendroje erdvėje. Ši technologija naudojama automatiniam vertimui, sintetinio turinio generavimui, veido atpažinimo funkcijoms ir panašiai [Aca]. Vienas iš labiausiai nagrinėjamų šios srities tipų yra neuroniniai tinklai. Kiekvieną tinklą sudaro struktūrinės dalys, vadinamos dirbtiniais neuronais. Šio laboratorinio darbo metu siekėme išnagrinėti šios struktūros architektūrą ir bendrą veiklą.

Darbo tikslas: Išanalizuoti dirbtinio neurono modelio veikimo principus.
Darbo uždaviniai:

- Sugeneruoti du dvimačių duomenų rinkinius, kurie būtų sudaryti iš 10 duomenų įrašų.
- Suteikti galimybę vartotojui pasirinkti aktyvacijos funkciją.
- Rankiniu būdu apskaičiuoti klasifikavimui tinkančius svorius ir poslinkį.
- Sukurti dirbtinį neuroną, kuris naudodamas išrinktus svorius ir poslinkius gebėtų klasifikuoti duomenis.
- Pavaizduoti atskiras klases ir jas skiriančias tieses grafike.

1. Duomenų generavimas

Šiam laboratoriniu darbui buvo generuojamos dvi duomenų klasės. Kiekviena klasė yra sudaryta iš 10 dvimačių taškų (Žr. 1 ir 2 lenteles). Visi duomenys suapvalinti iki 3 skaičių po kablelio.

1 lentelė. *Nulinės klasės sugeneruoti uždavinio duomenys*

X_1	-2,626	-2,268	-2,844	-2,942	-2,399	-2,979	-2,168	-2,818	-2,696	-2,568
X_2	0,951	0,599	0,156	0,866	0,708	0,969	0,212	0,183	0,525	0,291

2 lentelė. *Pirmos klasės sugeneruoti uždavinio duomenys*

X_1	3,612	3,292	3,456	3,199	3,594	3,608	3,065	3,966	3,305	3,684
X_2	0,139	0,366	0,785	0,514	0,047	0,171	0,949	0,809	0,098	0,440

Duomenys buvo generuojami naudojant funkciją **rand** iš **numpy.random** bibliotekos. Prieš naudojant šią funkciją buvo nustatytas **seed**, kuris užtikrina duomenų identišką generavimą. Tai reiškia, kad kiekvieną kartą paleidžiant kodą, yra generuojami tokie patys duomenys, išlaikant rezultatų pastovumą.

Pirmiausia, naudojant **rand** funkciją, buvo sugeneruoti 10 dvimačių taškų, priklausančių dviem skirtingoms klasėm. Dirbtinio neurono įvertinimui šios klasės turi būti tiesiškai atskiriamos, todėl buvo naudota **numpy.array** funkcija [Ope24]. Ji leido paslinkti sugeneruotus taškus į skirtingas x ašies puses: pirmoji klasė buvo paslinkta 3 vienetais į mažesnes reikšmes, o antroji klasė 3 vienetais į didesnes. Šis veiksmas užtikrino tiesiškai atskiriamas klases (Žr. 1 pav.).

```
29  seed(42)
30
31  c1 = rand(10, 2) + np.array([-3, 0])
32  c2 = rand(10,2) + np.array([3, 0])
33
34  print(c1)
35  print(c2)
36
37  x = np.vstack((c1, c2))
38  targets = [0]* 10 + [1]*10
```

1 pav. *Duomenų generavimo kodas*

Tolimesniam darbui abiejų klasių duomenys yra sujungiami į vieną bendrą rinkinį, išlaikant struktūrinį logiškumą. Viršutinėms reikšmėms priskiriami 0, o apatinėms 1.

2. Dirbtinio neurono realizacija

Dirbtiniai neuroniniai tinklai yra paremti biologinių neuronų veikimo principu. Kaip ir mūsų smegenyse, šiuos tinklus sudaro daug mažų neuronų. Šios struktūros yra taikomos klasifikavime, prognozavime ir t.t.[Sta14]. Kiekvienas tinklas yra sudarytas iš dalių vadinamų dirbtiniais neuronais. Dirbtinio neurono struktūra:

- Įėjimai – pateikiami duomenys neuronui.
- Sužadinimo reikšmės apskaičiavimas.
- Aktivacijos funkcija - jos reikšmė yra neurono išėjimo reikšmė.
- Išėjimas.

Šiame laboratoriniame darbe neurono realizacijai buvo naudojama **PyTorch** biblioteka. Realizuojant šią struktūrą, buvo sukurta klasė, kad dirbtinis neuronas būtų lengvai prieinamas darbo eigoje. Pirmiausia realizuojamas neuronas nurodant įėjimų ir išėjimų skaičių, o vėliau **def forward(self, x)** pritaiko pasirinktą aktyvacijos funkciją (Žr. 2 pav.).

```
12  class Neuron(nn.Module):
13      def __init__(self, input_size, activation_function):
14          super(Neuron, self).__init__()
15          self.neuron = nn.Linear(input_size, 1)
16          self.activation_function = activation_function
17
18      def forward(self, x):
19          return self.activation_function(self.neuron(x))
20
```

2 pav. *Dirbtinio neurono realizavimo kodas*

2.1. Įėjimai

Kadangi buvo generuojami duomenys su dvimačiais taškais, reiškias įėjimai bus **2**: x_1 ir x_2.

2.2. Sužadinimo reikšmė

Sužadinimo reikšmė yra vienas svarbiausių aspektų dirbtinio neurono realizacijos eigoje. Ši reikšmė yra perduodama į aktyvacijos funkciją ir taip randama dirbtinio neurono išėjimo reikšmė.

$$a = x_1 w_1 + x_2 w_2 + ... + x_n w_n + b \ [gee24]$$

Kintamieji, kurių reikalauja ši funkcija yra: x_k - duodami duomenys, kur k atitinka jų kiekį (pvz.: x_1, x_2), w_k - svoriai, b - poslinkis.

2.3. Aktyvacijos funkcijos

Pagal aktyvacijos funkciją yra randamos dirbtinio neurono išėjimo reikšmės. Neuroninių tinklų kontekste jų yra keletas. Tačiau, kadangi šio darbo tikslas yra klasifikuoti duomenis į dvi klases: 0 ir 1, tinkamiausios yra **Slenkstinė** ir **Sigmoidinė** funkcijos. Šio laboratorinio darbo metu vartotojui yra suteikiama galimybė pasirinkti, kurią aktyvacijos funkciją taikyti (Žr. 3 pav.).

```
45  choice = int(input("Pasirinkite kurią aktivavimo funkcija norėsite naudoti: (0 - Slenkstinė, 1 - Sigmoidinė)\n"))
46  if choice != 0 and choice != 1:
47      raise ValueError("Galimi pasirinkimai yra 0 - Slenkstinė ir 1 - Sigmoidinė aktivavimo funkcijos")
48
49  activation_function = step_function if choice == 0 else torch.sigmoid
50
```

3 pav. *Aktyvacijos funkcijos pasirinkimas*

2.3.1. Slenkstinė funkcija

Šios aktyvacijos funkcijos reikšmės gali būti tik 0 arba 1, todėl ji puikiai tinka laboratorinio darbo tikslui. Šios funkcijos atsakymas yra gaunamas priklausomai nuo sužadinimo reikšmės, kuri buvo aptarta praėjusiame skyrelyje. Jeigu ši reikšmė yra neigiama, gaunama išėjimo klasė yra 0. Priešingu atveju yra gaunamas 1.

$$f(a) = \begin{cases} 1, & \text{jeigu} \quad a \geq 0 \\ 0, & \text{jeigu} \quad a < 0 \end{cases} \quad [gee24]$$

Laboratorinio darbo metu, programiniame kode, buvo sukurta paprasta funkcija, kuri gražina 1, jeigu gauta reikšmė yra didesnė arba lygi nuliui (Žr. 4 pav.).

```
22
23  def step_function(x): #Slenkstinė funkcija
24      return (x >= 0).float()
25
```

4 pav. *Slenkstinės funkcijos realizacija*

2.3.2. Sigmoidinė funkcija

Ši funkcija yra S formos funkcija. Ji skiriasi nuo slenktstinės, kadangi gali įgyti bet kokias reikšmes nuo **0** iki **1**.

$$f(a) = \frac{1}{(1 + e^{-a})}, e \approx 2.718 \ [gee24]$$

Tačiau ją galima naudoti laboratorinio darbo tikslui įgyvendinti, suapvalinant reikšmes iki artimiausio sveikojo skaičiaus (Žr. 5 pav.).

```
85
86  if choice == 1:
87      output = (output >= 0.5).float()
88
```

5 pav. *Suapvalinimas iki artimiausio sveikojo skaičiaus*

2.4. Svorių ir poslinkio paieška

Dažniausiai svoriai ir poslinkis dirbtiniame neurone yra priskiriami apmokymo metu, tačiau norint ištirti kaip šis procesas veikia, laboratorinio darbo metu tai buvo daryta rankiniu būdu. Pirmiausia naudojant **uniform** funkciją sugeneruojami atsistiktiniai svoriai ir poslinkis. Apskaičiuojama sužadinimo reikšmė ir pagal pasirinktą aktyvacijos funkciją yra randama išėjimo reikšmė. Galiausiai yra sulyginamos išėjimo ir priskirtos klasės reikšmės. Jeigu reikšmės sutampa tai sugeneruoti svoriai ir poslinkis yra pridedami prie **solutions** sąrašo. Radus tris tinkamus sprendimus, funkcija yra nutraukiama (Žr. 6 pav).

```
55
56  solutions = []
57  attempts = 1000
58
59  for __ in range(attempts):
60      w = uniform(-5, 5, 2) # sugeneruoja du svorius
61      b = uniform(-5, 5)
62
63      a = np.dot(x, w) + b
64      y = activation_function(a)
65
66      if np.array_equal(y, targets):
67          solutions.append((w, b))
68          if len(solutions) == 3:
69              break
70
71  print(solutions)
72
```

6 pav. *Svorių ir poslinkio apskaičiavimas*

Naudojant skirtingas aktyvacijos funkcijas buvo rasti po 3 svorių ir poslinkio rinkinius (Žr. 3, 4 lenteles). Visi skaičiai yra suapvalinti iki 3 skaičių po kablelio.

3 lentelė. *Svorių ir poslinkio rinkiniai naudojant Slenkstinę funkciją*

Rinkiniai	w_1	w_2	b
1	4,093	-2,412	1,625
2	4,395	3,948	0,979
3	4,219	-4,115	-3,040

4 lentelė. *Svorių ir poslinkio rinkiniai naudojant Sigmoidine funkcija*

Rinkiniai	w_1	w_2	b
1	4,093	-2,412	1,625
2	4,395	3,948	0,979
3	4,219	-4,115	-3,040

Rezultatai naudojant dvi skirtingas aktyvacijos funkcijas išėjo tokie patys.

2.5. Svorių ir poslinkio rinkinių tiesių pavaizdavimas

Norint pastebėti, kaip skirtingi svoriai ir poslinkis perskiria erdvę. Tam padaryti buvo sugeneruotos trys tiesės su skirtingais rinkiniais. Pirmiausia buvo paimta 100 lygiai paskirstytų x ašies taškų,

kurie yra pasiskirstę nuo segeneruotų duomenų minimumo iki maximumo. Vėliau kiekvienai atskirai tiesei buvo apskaičiuoti y taškai, kuriuos radome naudojant išvestą funkciją (Žr. 7 pav.).

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

$$x_2 w_2 = -b - x_1 w_1$$

$$x2 = \frac{-b - x_1 w_1}{w_2}$$

Šiose funkcijose x_2 reprezentuoja y reikšmes.

```
104   x_vals = np.linspace(x[:,0].min(), x[:,0].max(), 100)
105
106 for i, (w, b) in enumerate(solutions):
107
108       y_vals = (-w[0] * x_vals - b)/w[1]
109       plt.plot(x_vals, y_vals, color=colors[i], label = (f'{i+1} rinkinio tiesė'))
110
```

7 pav. *Svorių ir poslinkio tiesių pavaizdavimo kodas*

2.6. Neurono paleidimas

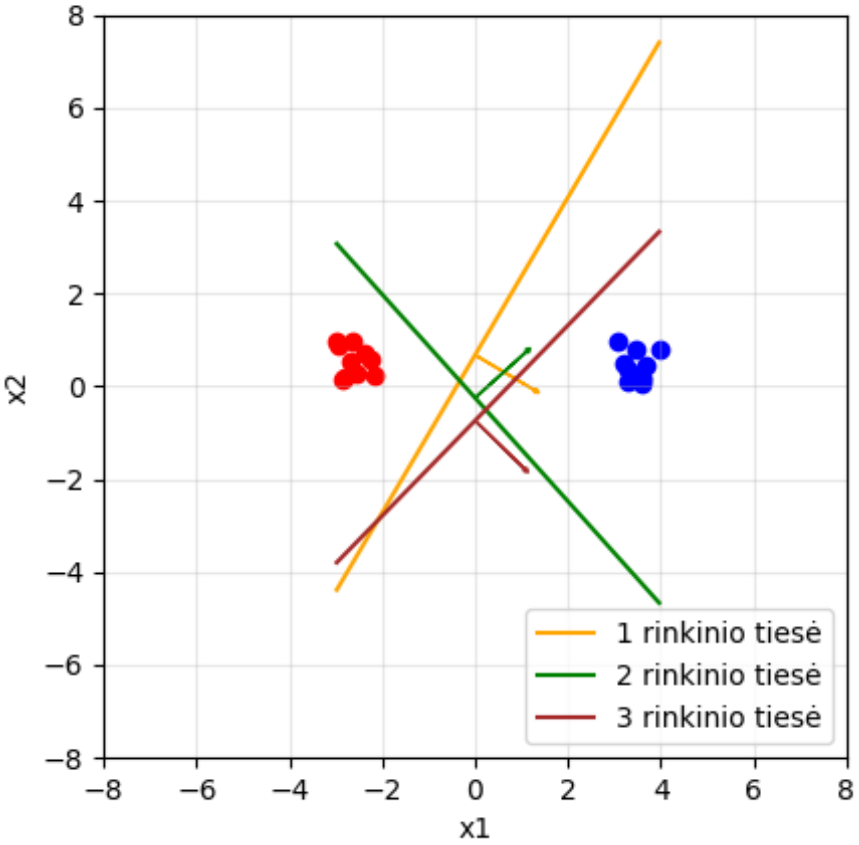
Kadangi buvo sukurta dirbtinio neurono klasė, ši klasė veliau yra panaudojama. Pirmiausia yra sukuriamas dirbtinis neuronas, kuris turi du įėjimus, parenkama aktyvacijos funkcija. Ankščiau buvo apibendrinta, kaip buvo surasti trys svorių ir poslinkio rinkiniai. Norint patikrinti ar neuronas teisingai klasifikuoja duomenis, naudojamas tik pirmas rinkinys. Šie parametrai rankiniu būdu priskiriami neuronui naudojant **PyTorch** biblioteką. Galiausiai paleidus neuroną su paduotomis reikšmėmis, gaunamos išėjimo reikšmės (Žr. 8 pav.).

```
76  # sukuriamas neuronas su dviem įėjimais ir specifikuojama aktivacijos funkcija
77  neuron = Neuron(2, activation_function=activation_function)
78
79  w, b = solutions[0] # Naudojamas pirmas rinkinys, kad patikrinti neurono veikimą
80  neuron.neuron.weight.data = torch.from_numpy(w.astype(np.float32)).unsqueeze(0)
81  neuron.neuron.bias.data = torch.tensor([b], dtype= torch.float32)
82
83  input = torch.tensor(x, dtype = torch.float32)
84  output = neuron(input)
85
```

8 pav. *Dirbtinio neurono paleidimo kodas*

Rezultatai ir išvados

Laboratorinio darbo metu buvo sugeneruotos dvi duomenų klasės, įgyvendinta dirbtinio neurono klasė, rankiniu būdu apskaičiuoti svorių ir poslinkio rinkiniai. Galiausiai neuronas buvo paleistas įgyvendinat laboratorinio darbo tikslą. Dvi sugeneruotos klasės yra tiesiškai atskiriamos (Žr. 9 pav.). Pirma klasė, kuriai buvo priskirti nuliai yra pažymėta raudona spalva. Antra klasė yra pažymėta mėlyna spalva. Naudojant tris skirtingų svorių ir poslinkio rinkinius, buvo suformuotos trys paviršių skiriančios tiesės. Iš jų galima pamatyti svoriu santykį ir nuspręsti, kuris svoris yra didesnis už kitą. Kaip matoma 8 pav., visos tiesės teisingai atskiria duomenis į priklausančias klases. Tai reiškia, kad net su skirtingais svorių ir poslinkio rinkiniais neuronas geba teisingai klasifikuoti duomenis.



9 pav. *Rezultatai*
[Ope24]

Žiūrėdami į rezultatų pavaizdavimą, galime padaryti kelias išvadas.

- Gali būti randami keli skirtingi svoriai ir poslinkis, kurie teisingai skiria paviršių.
- Iš tiesių krypties ir statumo galima daryti išvadas apie svorius.
- Pirmoji tiesė yra stačiausia, todėl galima spręsti, kad santykis tarp svorių yra didžiausias, palyginus su kitomis tiesėmis.

- Antroji tiesė vienintelė eina žemyn, iš to galima pasakyti, kad svoriai turi tapatį ženklą. Abudu svoriai yra teigiami arba neigiami.
- Vektoriai aiškiai parodo, kad klasė, kuriai priskiriami 1 yra dešinėje pusėje.

Šis laboratorinis darbas leidžia pamatyti, kaip veikia svorių ir poslinkių radimas rankiniu būdu, bendrą dirbtinio neurono struktūrą ir kaip viską sujungti į gerai klasifikuojantį dirbtinį neuroną. Šis darbas apsiribojo PyTorch bibliotekos naudojimu, tačiau yra ir kitų būdų įgyvendintį tikslą.

Literatūra ir šaltiniai

[Aca] C. Academy. *Dirbtinis intelektas: kas tai ir kaip jis keičia darbą ir laisvalaikį?* URL: https://codeacademy.lt/kas-yra-dirbtinis-intelektas-ir-kur-jis-naudojamas-pritaikymas-darbui-ir-laisvalaikiui/ (žiūrėta 2025-09-09).

[gee24] G. for geeks. *Activation functions*. 2024. URL: https://www.geeksforgeeks.org/engineering-mathematics/activation-functions/ (žiūrėta 2025-09-09).

[Ope24] OpenAI. *ChatGPT*. Buvo naudotas, kad vektoriai kordinačių ašyje atrodytų stačiai ir rodyklių (ang. arrrows) pavaizdavimui. Taip pat aiškinantis, kaip sugeneruoti duomenis, kurie yra tiesiškai atskiriami. 2024. URL: https://www.openai.com/ (žiūrėta 2024-06-06).

[Sta14] L. Stabingienė. *Dirbtiniai neuroniniai tinklai*. 2014. URL: http://www.ilab.lt/stabingiene/sk2_2.html (žiūrėta 2025-09-09).