Neprivalomos užduotys papildomam balui (I dalis, 0,5 b.)
Sprendimai, paaiškinimai

1. Turime neuroninį tinklą, kuris susideda iš kelių pilnai jungių (angl. *fully connected*) sluoksnių ir *ReLU* aktyvacijos funkcijų. Kokios būtų pasekmės, jei visi neuroninio tinklo parametrai inicializacijos metu būtų nuliai? (0,1 b.)

- didžioji dauguma svorių nesimokytų, nes išvestinė būtų lygi nuliui (visų, išskyrus *bias*, pvz., nuostolių f-jos išvestinė pagal parametrą a lygtyje $z(x) = \text{relu}(ax + b)$ bus lygi 0, tačiau pagal b nebus lygi nuliui, dėl to svoriai b vis dar keistųsi);
- visi parametrai tame pačiame sluoksnyje įgytų tas pačias išvestines, dėl to parametrai keistųsi lygiai taip pat; t.y., faktiškai gautume tą patį rezultatą, jei sluoksnyje būtų tik vienas neuronas.

2. Sprendžiame navikų klasifikavimo uždavinį. Turime dvi klases ir jų pasiskirstymus duomenų rinkinyje: pavojaus nekeliantis (angl. *benign*) – 97% ir piktybinis (angl. *malignant*) – 3%. Koks yra paprasčiausias klasifikatorius, kuris išgautų bent 90% tikslumą šiame duomenų rinkinyje? Ar tai geras klasifikatorius? Kodėl? (0,1 b.)

- paprasčiausias klasifikatorius – visada spėti, kad tai pavojaus nekeliantis navikas, nepriklausomai nuo įvesties požymių;
- tai nebūtų geras klasifikatorius, net jei tikslumas viršija 90%, nes norėtume aptikti ir piktybinius navikus; tikslumas (angl. *accuracy*) dažniausiai nėra geras matas; šiuo atveju vertėtų naudoti kitą matą, pvz. F1 statistiką;
- reikia nepamiršti galimybės įvertinti modelius kokybiškai (angl. *qualitative evaluation*); pvz. teksto generavimo uždaviniuose kiekybinis įvertis (angl. *quantitative evaluation*) gali būti labai geras – pvz. aukštas perpleksiškumas, tačiau įvertinus modelį kokybiškai (t.y., rankiniu būdu peržiūrėjus kelis sugeneruotus pavyzdžius) tekstai gali būti beprasmiški.

3. Kodėl įprastai padaliname duomenis į tris duomenų rinkinius? T.y., kodėl naudojame ne tik mokymo ir validacijos rinkinį, bet papildomai ir testavimo duomenų rinkinį? (0,1 b.)

- mokymo duomenų rinkinys skirtas tiesiogiai iš jo mokytis (t.y., dalyvauja parametrų paieškoje, iš jo skaičiuojami gradientai); validacijos duomenų rinkinys leidžia mokymo metu įvertinti mokymosi kokybę ir keisti hiperparametrus; tačiau reikalingas trečiasis – testavimo – duomenų rinkinys, kad galėtume nešališkai įsivertinti galutinį rezultatą, t.y., gauti įsivertinimą, kuris nepriklausytų nuo mokymo/*tuning* proceso;
- jei po įsivertinimo su trečiuoju duomenų rinkiniu dar keistume modelį, modelis taptų „priderintas“ prie jo, ir įvertis būtų šališkas.

4. Ar įmanoma su konvoliucijos sluoksniu atlikti sutelkimą suvidurkinant (angl. *average pooling*)? Pateikite pavyzdį. (0,1 b.)

- taip, įmanoma; pavyzdžiui, jei norime atlikti 2×2 sutelkimą suvidurkinant su konvoliucija, turėtume naudoti 2×2 konvoliucijos filtrą K su paslinkimu/žingsnio dydžiu (angl. *stride*) $s = (2, 2)$, kurio visos reikšmės yra $\frac{1}{2 \cdot 2} = 0.25$:

$$Y = K * X\,, \quad K = \begin{bmatrix} 0.25 & 0.25 \\ 0.25 & 0.25 \end{bmatrix},$$

  čia X – įvestis prieš sutelkimą, Y – sutelkimo rezultatas;
- bendru atveju – norint atlikti $a \times b$ sutelkimo operaciją su konvoliucija, reiktų imti $a \times b$ dydžio filtrą, parinkti *stride* $s = (a, b)$, ir visas filtro reikšmes nustatyti $\frac{1}{ab}$.

5⋆. Neuroninių tinklų parametrai dažniausiai surandami automatiškai, iš duomenų, gradientinio nusileidimo pagalba. Kodėl šių metodų nenaudojame optimalių hiperparametrų (sluoksnių sk., mokymo greičio (angl. *learning rate*), etc.) radimui? (0,05 b.)

- įsivertinti vienos hiperparametrų kombinacijos reikšmę prilygsta pilnai apmokyti neuroninį tinklą, dėl to gali būti itin didelės skaičiavimo sąnaudos;
- visų įmanomų hiperparametrų kombinacijų erdvė yra per didelė; pvz. visų įmanomų architektūrų erdvė, nuostolių funkcijų erdvė yra eilėmis didesnė, nei neuroninio tinklo svorių reikšmių erdvė;
- ⋆ dauguma hiperparametrų nėra diferencijuojami (pvz. sluoksnių skaičius), t.y., yra diskrečios arba kategorinės reikšmės, netolydžios;
- ⋆ tačiau, kai kurie metodai, pvz. *DARTS: Differentiable Architecture Search* (Liu et al., 2019) leidžia automatiškai iš duomenų surasti tinkamą architektūrą iš nedidelio skaičiaus pasirinkimų.

6⋆. QR kodai išrasti 1994 metais. Jau tais laikais barkodų skenavimo algoritmai buvo itin greiti, skenavo akimirksniu. Kokį metodą pasirinktumėte kodų skaitytuvui? Kodėl? (0,05 b.)

- ⋆ tokį, kuris nėra vien tik paremtas statistika arba giliojo mokymosi metodais – išmokti atpažinti bet kokią užkoduotą informaciją būtų itin sunku;
- ⋆ kodų skaitytuvas turėtų būti greitas – kelių transformacijų pagalba turėtume išgauti aptiktą kodo regioną ir tiesiog naudoti dekodavimo algoritmą.