Neuronske mreže i klasifikacija proteina

Runac, Borna

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Borna Runac

NEURONSKE MREŽE I KLASIFIKACIJA PROTEINA

Diplomski rad

Voditelj rada:
dr. sc. Pavle Goldstein

Zagreb, rujan 2020.

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Svojoj majci

Sadržaj

Uvod

Razvoj interneta i inflacija memorijskih kapaciteta u 21. stoljeću dovele su do velike količine neobrađenih podataka. Velike količine podataka su omogućile razvoj modernih metoda strojnog učenja i umjetne inteligencije. *Umjetna neuronska mreža* (engl. *Artificial Neural Network*) ili skraćeno *neuronska mreža* je jedan od najsofisticiranijih dostignuća strojnog učenja. Neuronske mreže nalaze svoju primjenu u autonomnim vozilima, prepoznavanju lica te obradi prirodnog jezika (engl. *Natural Language Processing*). Pomicanje granica u mogućnostima umjetne inteligencije je dovelo do kolektivnog oduševljenja (engl. *hype*) neuronskim mrežama.

Proteini ili *bjelančevine* su velike molekule koje grade čitav živi svijet. Proteini su po kemijskom sastavu dugački lanci aminokiselina. *Sekvenciranje proteina* je postupak kojim se kemijskim i fizikalnim reakcijama određuje niz aminokiselina koji grade određeni protein. U prirodi se pojavljuje 20 osnovnih aminokiselina, što znači da svaki protein možemo reprezentirati kao niz zapisan abecedom od 20 slova. Razvoj tehnika sekvenciranja proteina omogućio je digitalno skladištenje strukture proteina.

Svaki protein ima svoju svrhu te se različiti proteini pojavljuju u različitim organizmima. Zato proteine dalje dijelimo u skupine koje nazivamo *proteinske familije*. Kada biolozi otkriju i sekvenciraju neki novi protein, postavlja se pitanje klasifikacije proteina u određenu proteinsku familiju. *Klasifikacija* proteina i usporedba s ostalim proteinima pomaže u otkrivanju svrhe i podrijetla određenog proteina. Osim biološke klasifikacije u proteinske familije, u bioinformatici je važna i računalna klasifikacija proteina metodama strojnog učenja. Klasifikacija proteina pomoću metoda kao što su logistička regresija, metoda potpornih vektora (engl. *Support Vector Machine*, skraćeno SVM) ili k-sredine (engl. *k-means*) daje informacije o udaljenostima među proteinima. Možemo reći da su proteini iz iste klase sličniji, a proteini iz različitih klasa manje slični. Informacija o sličnosti ili udaljenosti proteina može pomoći odgonetnuti evolucijsko podrijetlo proteina, a samim time i evoluciju vrste.

Cilj ovog rada je istražiti mogućnosti neuronske mreže i metode potpornih vektora na problemu računalne klasifikacije proteina. Metoda potpornih vektora je relativno jednostavan model u odnosu na model neuronske mreže te nam služi kao orijentir (engl. *benchmark*) za ocjenu učinkovitosti neuronske mreže.

Htio bih zahvaliti svom mentoru profesoru dr. Pavlu Goldsteinu koji mi je ponudio rad na ovako zanimljivoj temi te bez čijeg truda i pomoći ovaj rad ne bi ugledao svjetlo dana. Također bih zahvalio kolegama Tomislavu Vlahu i Domagoju Iveku koji su odradili dio posla s neuronskim mrežama te mi ponudili svoje rezultate.

Slika 0.1: Niz aminokiselina proteina L0M9T1_ENTBF/23-59 iz familije *hemP*. Ovaj protein je građen od 14 različitih aminokiselina koje su posložene u lanac duljine 37.

LISSKLLLGDEGSVLIEHNGQHYQLRQTQSGKLILTK

Poglavlje 1

Osnovni matematički pojmovi

1.1 Linearna algebra

Definicija 1.1.1. *Neka je V neprazan skup i neka je $\mathbb{F}$ polje. Neka su zadane operacije zbrajanja vektora $+ : V \times V \to V$ i operacija množenja vektora skalarom $\cdot : \mathbb{F} \times V \to V$. Kažemo da je uređenja trojka $(V, +, \cdot)$ vektorski prostor nad poljem $\mathbb{F}$ ako vrijedi*

(1) $(x + y) + z = x + (y + z), \quad \forall x, y, z \in V,$

(2) $\exists 0 \in V, \quad 0 + x = x + 0 = x, \quad \forall x \in V,$

(3) $\forall x \in V, \exists -x \in V, \quad (-x) + x = x + (-x) = 0,$

(4) $x + y = y + x, \quad \forall x, y \in V,$

(5) $\alpha(\beta x) = (\alpha\beta)x, \quad \forall \alpha, \beta \in \mathbb{F}, \forall x \in V,$

(6) $(\alpha + \beta)x = \alpha x + \beta x, \quad \forall \alpha, \beta \in \mathbb{F}, \forall x \in V,$

(7) $\alpha(x + y) = \alpha x + \alpha y, \quad \forall \alpha \in \mathbb{F}, \forall x, y \in V,$

(8) $\exists 1 \in \mathbb{F}, \quad 1 \cdot x = x, \quad \forall x \in V.$

U tom slučaju elemente skupa V zovemo vektori, dok elemente polja $\mathbb{F}$ zovemo skalari.

Skup $\mathbb{R}^n = \{(x_1, \ldots, x_n) : x_1, \ldots, x_n \in \mathbb{R}\}$ s operacijama

$$(x_1, \ldots, x_n) + (y_1, \ldots, y_n) = (x_1 + y_1, \ldots, x_n + y_n)$$
$$\alpha(x_1, \ldots, x_n) = (\alpha x_1, \ldots, \alpha x_n), \quad \alpha \in \mathbb{R}$$

je vektorski prostor nad $\mathbb{R}$.

Definicija 1.1.2. *Neka je V vektorski prostor nad poljem $\mathbb{F}$ i S= $\{x_1, \ldots, x_k\}, k \in \mathbb{N}$ konačan podskup od V. Kažemo da je S linearno nezavisan skup ako vrijedi*

$$\sum_{i=1}^{k} \alpha_i x_i = 0 \quad \Longrightarrow \quad \alpha_1 = \cdots = \alpha_k = 0,$$

gdje su $\alpha_1, \ldots, \alpha_k \in \mathbb{F}$.

Definicija 1.1.3. *Neka je V vektorski prostor nad poljem $\mathbb{F}$ i S= $\{x_1, \ldots, x_k\}, k \in \mathbb{N}$ konačan podskup od V. Kažemo da je S sustav izvodnica za V ako vrijedi*

$$V = \left\{ \sum_{i=1}^{k} \alpha_i x_i : \alpha_1, \ldots, \alpha_k \in \mathbb{F} \right\}.$$

Definicija 1.1.4. *Neka je V vektorski prostor. Linearno nezavisan skup koji je i sustav izvodnica zovemo baza.*

Definicija 1.1.5. *Kažemo da je vektorski prostor V konačnodimenzionalan ako postoji neki konačan sustav izvodnica za V.*

Teorem 1.1.6. *Svaki konačnodimenzionalni vektorski prostor ima bazu.*

Teorem 1.1.7. *Sve baze konačnodimenzionalnog vektorskog prostora su jednakobrojne.*

Definicija 1.1.8. *Neka je $(V, +, \cdot)$ vektorski prostor nad $\mathbb{F}$ i M podskup od V. Kažemo da je M potprostor od V ako je $(M, +, \cdot)$ također vektorski prostor nad $\mathbb{F}$.*

Definicija 1.1.9. *Neka je $(V, +, \cdot)$ vektorski prostor nad poljem $\mathbb{F}$. Skalarni produkt na V je preslikavanje $\langle \cdot, \cdot \rangle : V \times V \to \mathbb{F}$ sa sljedećim svojstvima*

(1) $\langle x, x \rangle \geq 0, \quad \forall x \in V,$

(2) $\langle x, x \rangle = 0 \iff x = 0,$

(3) $\langle x + y, z \rangle = \langle x, z \rangle + \langle y, z \rangle, \quad \forall x, y, z \in V,$

(4) $\langle \alpha x, y \rangle = \alpha \langle x, y \rangle, \quad \forall \alpha \in \mathbb{F}, \forall x, y \in V,$

(5) $\langle x, y \rangle = \overline{\langle y, x \rangle}, \quad \forall x, y \in V.$

U tom slučaju vektorski prostor $(V, +, \cdot)$ zajedno sa skalarnim produktom zovemo unitarni prostor.

Euklidski skalarni produkt na vektorskom prostoru $\mathbb{R}^n$ je zadan s

$$\langle (x_1, \ldots, x_n), (y_1, \ldots, y_n) \rangle = \sum_{i=1}^{n} x_i y_i.$$

Definicija 1.1.10. *Neka je V vektorski prostor nad poljem $\mathbb{F}$. Funkciju $\|\cdot\| : V \to \mathbb{R}$ koja zadovoljava svojstva*

(1) $\|x\| \geq 0, \quad \forall x \in V,$

(2) $\|x\| = 0 \iff x = 0,$

(3) $\|\alpha x\| = |\alpha| \|x\|, \quad \forall \alpha \in \mathbb{F}, \forall x \in V,$

(4) $\|x + y\| \leq \|x\| + \|y\|, \quad \forall x, y \in V,$

zovemo norma na V.

Propozicija 1.1.11. *Neka je V unitarni prostor. Funkcija $\|\cdot\| : V \to \mathbb{R}$ definirana s*

$$\|x\| = \sqrt{\langle x, x \rangle}, \quad \forall x \in V$$

je norma.

Euklidska norma inducirana Euklidskim skalarnim produktom na $\mathbb{R}^n$ je zadana s

$$\|(x_1, \ldots, x_n)\| = \sqrt{\sum_{i=1}^{n} x_i^2}.$$

Definicija 1.1.12. *Neka je V vektorski prostor. Funkciju $d : V \times V \to \mathbb{R}$ koja zadovoljava svojstva*

(1) $d(x, y) \geq 0, \quad \forall x, y \in V,$

(2) $d(x, y) = 0 \iff x = y,$

(3) $d(x, y) = d(y, x), \quad \forall x, y \in V$

(4) $d(x, z) \leq d(x, y) + d(y, z), \quad \forall x, y, z \in V,$

zovemo udaljenost ili metrika na V.

Propozicija 1.1.13. *Neka je V vektorski prostor. Ako je $\|\cdot\|$ norma na V, tada je funkcija $d : V \times V \to \mathbb{R}$ definirana s $d(x, y) = \|x - y\|, \forall x, y \in V$ udaljenost na V.*

Definicija 1.1.14. *Neka je V unitarni prostor. Za vektore $x, y \in V$ kažemo da su ortogonalni ako vrijedi $\langle x, y\rangle = 0$. Kažemo da je konačan skup vektora $S = \{e_1, \ldots, e_k\}$ ortonormiran ako vrijedi $\langle e_i, e_j\rangle = \delta_{ij}$, gdje je δ_{ij} Kroneckerov delta simbol*

$$\delta_{ij} = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases}$$

Ortonormirani skup koji je ujedno i baza prostora V zovemo ortonormirana baza za V.

Definicija 1.1.15. *Neka je V unitarni prostor i M potprostor od V. Ortogonalni komplement potprostora M (oznaka $M^\perp$) je skup svih vektora iz V koji su ortogonalni s vektorima iz M.*

$$M^\perp = \{x \in V : \langle x, v\rangle = 0, \forall v \in M\}$$

Teorem 1.1.16. *Neka je V unitarni prostor i M potprostor od V. Za svaki vektor $x \in V$ postoje jedinstveni vektor $a \in M$ i $b \in M^\perp$ takvi da je $x = a + b$.*

Definicija 1.1.17. *Neka je V unitarni prostor i M potprostor od V. Neka je $x \in V$ i neka je $x = a + b, a \in M, b \in M^\perp$ rastav vektora x iz teorema 1.1.16. Vektor $a \in M$ zovemo ortogonalna projekcija vektora x na potprostor M.*

Teorem 1.1.18. *Neka je V unitarni prostor i neka je M potprostor od V. Tada postoji skup $S = \{e_1, \ldots, e_n\}$ i $k \leq n$ takav da je S ortonormirana baza prostora V, a $\{e_1, \ldots, e_k\}$ ortonormirana baza potprostora M. Ako je $x \in V$ vektor, tada je ortogonalna projekcija vektora x na potprostor M jednaka $\sum_{i=1}^{k} \langle x, e_i\rangle e_i$.*

Neka je zadan unitarni prostor $\mathbb{R}^n$ s Euklidskim skalarnim produktom. Neka su zadani $w \in \mathbb{R}^n$ vektor normale različit od 0, skalar $b \in \mathbb{R}$, vektor x_0 i hiperravnina $H = \{x \in \mathbb{R}^n : \langle w, x\rangle + b = 0\}$. Želimo naći formulu za udaljenost vektora x_0 od hiperravnine H.

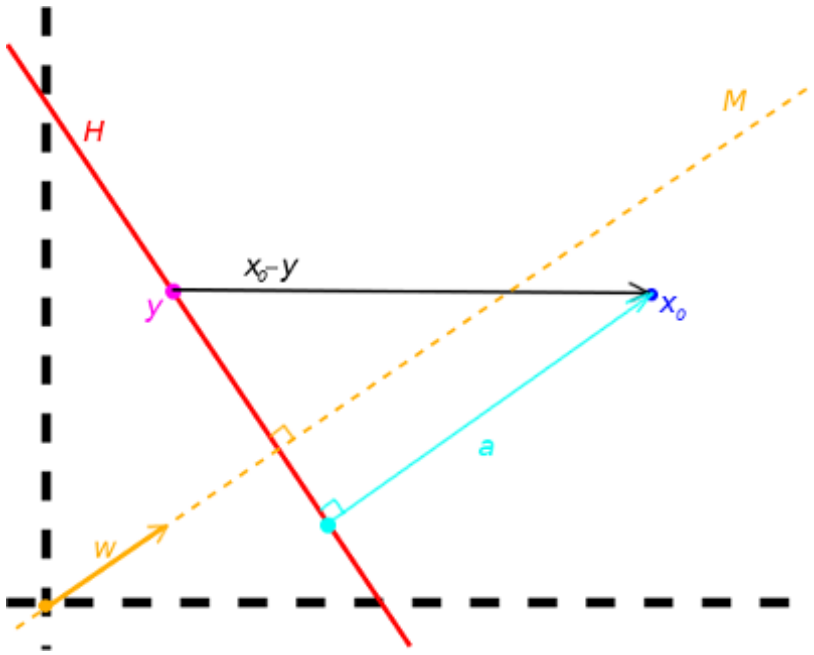
Neka je $y \in \mathbb{R}^n$ neki vektor hiperravnine H, tj. vrijedi $\langle w, y\rangle + b = 0$. Vektor $x_0 - y$ povezuje hiperravninu H i vektor x_0. Neka je $M = \{\alpha w : \alpha \in \mathbb{F}\}$ potprostor u smjeru normale w. Skup $\left\{\frac{w}{\|w\|}\right\}$ je ortonormirana baza za M. Prema teoremu 1.1.18 ortogonalna projekcija vektora $x_0 - y$ na potprostor M je jednaka

$$a = \left\langle x_0 - y, \frac{w}{\|w\|}\right\rangle \frac{w}{\|w\|} = \frac{\langle x_0 - y, w\rangle w}{\|w\|^2} = \frac{\langle w, x_0 - y\rangle w}{\|w\|^2} = \frac{(\langle w, x_0\rangle - \langle w, y\rangle)\, w}{\|w\|^2}.$$

Budući da smo izabrali vektor y tako da vrijedi $-\langle w, y\rangle = b$, imamo

$$a = \frac{(\langle w, x_0\rangle + b)\, w}{\|w\|^2}.$$

Slika 1.1: Izvod udaljenosti točke x_0 od hiperravnine H u $\mathbb{R}^2$. Hiperravnina u $\mathbb{R}^2$ je pravac.



Udaljenost točke x_0 od hiperravnine H je jednaka normi ortogonalne projekcije vektora $x_0 - y$ na potprostor M.

$$d(x_0, H) = \|a\| = \left\| \frac{(\langle w, x_0 \rangle + b)\, w}{\|w\|^2} \right\| = \frac{|\langle w, x_0 \rangle + b| \|w\|}{\|w\|^2} = \frac{|\langle w, x_0 \rangle + b|}{\|w\|}.$$

1.2 Optimizacija

Definicija 1.2.1. *Neka je zadana funkcija $f : X \subseteq \mathbb{R}^n \to \mathbb{R}$. Kažemo da funkcija f ima lokalni minimum u točki $x_0 \in X$ ako postoji $\varepsilon > 0$ takav da vrijedi*

$$\forall x \in X, \quad \|x - x_0\| < \varepsilon \implies f(x_0) \leq f(x).$$

Kažemo da funkcija f ima globalni minimum u točki $x_0 \in X$ ako vrijedi

$$f(x_0) \leq f(x), \quad \forall x \in X.$$

Neka su zadane funkcije $g_1, \ldots, g_m : \mathbb{R}^n \to \mathbb{R}$ te neka je

$$X = \{x \in \mathbb{R}^n : g_i(x) \leq 0, 1 \leq i \leq m\}.$$

Zadaću pronalaska minimuma funkcije $f : X \to \mathbb{R}$ nazivamo *optimizacijski problem*.

Definicija 1.2.2. *Neka je V vektorski prostor. Kažemo da je skup $C \subseteq V$ konveksan ako vrijedi*

$$\forall t \in [0,1], \quad x, y \in C \implies (1-t)x + ty \in C.$$

Definicija 1.2.3. *Neka je $X \subseteq \mathbb{R}^n$ konveksan skup u $\mathbb{R}^n$. Kažemo da je funkcija $f : X \to \mathbb{R}$ konveksna ako je*

$$\forall t \in [0,1], \quad x, y \in X \implies f((1-t)x + ty) \leq (1-t)f(x) + tf(y).$$

Definicija 1.2.4. *Neka je $X \subseteq \mathbb{R}^n$ konveksan skup u $\mathbb{R}^n$. Kažemo da je funkcija $f : X \to \mathbb{R}$ afina ako su postoji linearni operator $m : X \to \mathbb{R}$ takav da je*

$$m(x-y) = f(x) - f(y), \quad x, y \in X.$$

Definicija 1.2.5. *Neka je V vektorski prostor i neka je A neki njegov podskup. Konveksna ljuska skupa A (oznaka:* Convex *A) je najmanji konveksni skup u V koji sadrži A.*

Propozicija 1.2.6. *Neka je V vektorski prostor i A neki njegov podskup. Vrijedi*

$$\text{Convex } A = \left\{ \sum_{i=1}^{n} t_i x_i : n \in \mathbb{N}, t_1, \ldots, t_n \geq 0, \sum_{i=1}^{n} t_i = 1, x_1, \ldots, x_n \in A \right\}.$$

Teorem 1.2.7. *(Karush-Kuhn-Tuckerovi uvjeti) Neka je $K \subseteq \mathbb{R}^n$ otvoren i konveksan, $f, g_1, \ldots, g_m : K \to \mathbb{R}$ konveksne funkcije klase C^1 te $h_1, \ldots, h_l : K \to \mathbb{R}$ afine funkcije. Promotrimo problem*

$$(P) = \begin{cases} f(x) \to \min\limits_{x \in K} \\ g_i(x) \leq 0, \quad 1 \leq i \leq m \\ h_j(x) = 0, \quad 1 \leq j \leq l. \end{cases}$$

Pretpostavimo da problem ima Slaterovu točku*, tj. postoji točka $y \in K$ takva da je*

$$g_i(y) < 0, \quad 1 \leq i \leq m, \qquad h_j(y) = 0, \quad 1 \leq j \leq l.$$

Točka $x^* \in K$ je optimalno rješenje problem (P) ako i samo ako postoje $(\lambda^*, \mu^*) \in \mathbb{R}^m \times \mathbb{R}^l$ takvi da vrijedi

(1)

$$\lambda_i^* \geq 0, \quad 1 \leq i \leq m,$$

(2)

$$\lambda_i^* g_i(x^*) = 0, \quad 1 \leq i \leq m, \qquad h_j(x^*) = 0, \quad 1 \leq j \leq l,$$

(3)

$$\nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g(x^*) + \sum_{j=1}^{l} \mu_j^* \nabla h_j(x^*) = 0.$$

Poglavlje 2

Računalna klasifikacija

U ovom poglavlju uvodimo neke osnovne pojmove strojnog učenja potrebne za razumijevanje tehnika korištenih u radu. Opisani su načini klasifikacije koje smo koristili u istraživanju — metoda potpornih vektora i neuronska mreža.

2.1 Osnovni pojmovi strojnog učenja

Prvo ćemo definirati pojam *klasifikatora*.

Definicija 2.1.1. *Neka je $\Theta \subseteq \mathbb{R}^p$ parametarski skup, $\theta \in \Theta$ parametar, $X \subseteq \mathbb{R}^d$ prostor ulaznih podataka te neka je $k \in \mathbb{N}$ broj klasa. Klasifikator je funkcija $f_\theta : X \to \{1, 2, \ldots, k\}$.*

Broj d označava dimenziju ulaznih podataka. Klasifikacija može biti *binarna* ($k = 2$) ili *višestruka* ($k > 2$). Familiju svih klasifikatora $\{f_\theta : \theta \in \Theta\}$ nazivamo *model*. Obično svi klasifikatori iz modela imaju nekakvu zajedničku strukturu.

Definicija 2.1.2. *Skup za učenje ili skup za treniranje (engl.* training set*) je diskretan skup*

$$S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\},$$

gdje je $x_i \in X, y_i \in \{1, 2, \ldots, k\}$ za svaki indeks $1 \leq i \leq n$. Broj $n \in \mathbb{N}$ zovemo veličina skupa za učenje. Vektor x_i zovemo vektor značajki (engl. feature vector*), dok y_i zovemo oznaka (engl. label). Jedan par (x_i, y_i) zovemo primjer za učenje.*

Vektori značajki x_i su vektorske reprezentacije objekata koje želimo klasificirati. Oznaka y_i govori kojoj klasi pripada objekt reprezentiran vektorom značajki x_i. Glavni cilj problema klasifikacije je pronaći klasifikator iz modela $\{f_\theta : \theta \in \Theta\}$ koji najbolje klasificira podatke iz skupa za učenje S. Da bismo mogli uspoređivati različite klasifikatore, definiramo funkciju greške (engl. *loss function*) $L : \theta \to \mathbb{R}_+$. Funkcija greške ovisi o parametru

θ i skupu za učenje S. Vrijednost funkcije greške $L(\theta)$ mora poprimati vrijednost blisku 0 ako za većinu primjera za učenje $(x_i, y_i) \in S$ vrijedi $f_\theta(x_i) = y_i$, tj. ako klasifikator f_θ dobro klasificira podatke u odgovarajuće klase. U suprotnom, vrijednost funkcije greške $L(\theta)$ mora biti velika kako bismo kaznili klasifikatore koji loše klasificiraju podatke. Problem pronalaska najboljeg klasifikatora se tada svodi na pronalazak parametra θ za koji je funkcija greške $L(\theta)$ minimalna. Postupak pronalaska najboljeg parametra θ za zadani skup za učenje S nazivamo *treniranje modela* te je ekvivalentan problemu optimizacije funkcije greške.

Definicija 2.1.3. *Skup za testiranje (engl.* test set*) je diskretan skup*

$$T = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\},$$

gdje je $x_j \in X, y_j \in \{1, 2, \ldots, k\}$ *za svaki indeks* $1 \leq j \leq m$. *Broj* $m \in \mathbb{N}$ *zovemo veličina skupa za testiranje.*

Pretpostavimo da smo pronašli klasifikator f koji je najbolji među klasifikatorima iz modela $\{f_\theta : \theta \in \Theta\}$, tj. kažemo da smo trenirali model na skupu za učenje. Skup za testiranje služi za ispitivanje uspješnosti pronađenog klasifikatora. Skup za testiranje je najčešće disjunktan sa skupom za treniranje kako bismo mogli testirati kako se model ponaša na novim podacima. Da bismo ocijenili koliko je model zapravo uspješan, uvodimo pojam točnosti.

Definicija 2.1.4. *Neka su zadani skup za testiranje* T *i klasifikator* $f : X \to \{1, 2, \ldots, k\}$. *Točnost klasifikatora (engl.* accuracy*) je broj točnih klasifikacija primjera iz* T *podijeljen s veličinom skupa za testiranje, tj.*

$$\text{točnost} = \frac{1}{m} \sum_{j=1}^{m} \mathbf{1}_{y_j}(f(x_j)),$$

gdje je operator $\mathbf{1}_b(a)$ *jednak 1 ako je* $a = b$, *a inače 0.*

Točnost je broj između 0 i 1 i govori koliko je dobar klasifikator. Dobar model će proizvesti klasifikator koji će imati visoku točnost.

2.2 Metoda potpornih vektora

Metoda potpornih vektora je metoda nadziranog učenja (engl. *supervised learning*) koja rješava problem klasifikacije. Osnovna ideja metode potpornih vektora je pronaći hiperravninu koja najbolje razdvaja podatke. Prvo se bavimo binarnom klasifikacijom linearno odvojivih podataka.

Linearni klasifikator

Pretpostavimo da je zadan skup za učenje S te da su podaci podijeljeni u dvije klase ($k = 2$).

Definicija 2.2.1. *Linearni klasifikator je funkcija* $f : X \subseteq \mathbb{R}^d \to \mathbb{R}$ *oblika*

$$f(x) = \langle w, x \rangle + b, \quad x \in X$$

gdje je $\langle \cdot, \cdot \rangle$ *skalarni produkt, a* $w \in \mathbb{R}^d$ *i* $b \in \mathbb{R}$ *su parametri klasifikatora.*

Ako je zadan linearni klasifikator $f : X \to \mathbb{R}$ iz definicije 2.2.1, možemo konstruirati klasifikator iz definicije 2.1.1 tako da definiramo

$$f_\theta(x) = \begin{cases} 1, & f(x) < 0, \\ 2, & f(x) \geq 0. \end{cases} \tag{2.1}$$

Ovdje je sada parametar zadan s $\theta = (w, b)$, dok je parametarski prostor $\Theta \subseteq \mathbb{R}^d \times \mathbb{R}$. Parametarski prostor modela ima dimenziju $d + 1$ što je za jedan veće od dimenzije vektora značajki.

Jednadžba $\langle w, x \rangle + b = 0$ je zapravo jednadžba hiperravnine koja razdvaja podatke iz skupa za učenje. Vektor w predstavlja normalu hiperravnine, dok je b pomak od ishodišta.

Od linearnog klasifikatora želimo da točno klasificira podatke iz skupa za učenje S, tj. želimo da za svaki $1 \leq i \leq n$ vrijedi

$$f(x_i) < 0, \text{ ako je } y_i = 1,$$
$$f(x_i) > 0, \text{ ako je } y_i = 2.$$

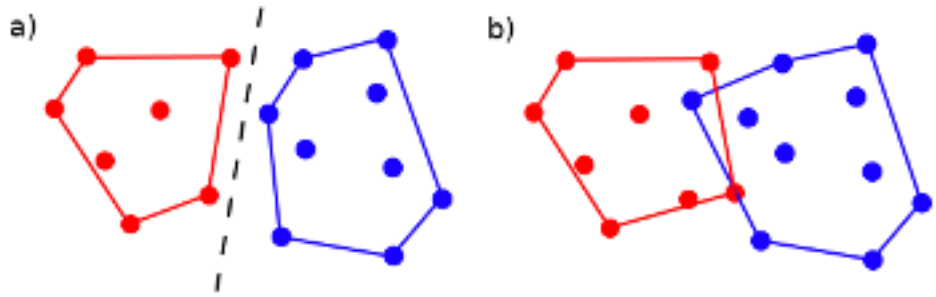
Postavlja se pitanje kako pronaći parametre w i b tako da klasifikator zadovoljava zadane uvjete. Takav klasifikator neće uvijek postojati.

Definicija 2.2.2. *Kažemo da su podaci iz skupa za učenje linearno odvojivi ako vrijedi*

$$\text{Convex } \{x_i : y_i = 1\} \cap \text{Convex } \{x_i : y_i = 2\} = \emptyset,$$

gdje je Convex *A konveksna ljuska skupa A.*

Slika 2.1: a) linearno odvojivi podaci b) linearno neodvojivi podaci



Možemo reći da su podaci iz skupa za učenje linearno odvojivi ako se konveksne ljuske vektora značajki iz različitih klasa ne sijeku. Slika 2.1 objašnjava definiciju 2.2.2. Ako podaci nisu linearno odvojivi, tada odvajajuća hiperravnina ne postoji jer sa jedne strane hiperravnine postoje podaci iz obje klase.

Kada imamo linearno odvojiv skup za učenje, tada postoji beskonačno mnogo hiperravnina koji razdvajaju podatke. Cilj je odabrati razdvajajuću hiperravninu koja je najudaljenija od konveksnih ljuski podataka iz različitih klasa.

Uvjet za razdvajajuću hiperravninu

Od sada koristimo oznake $y \in \{-1, 1\}$ kako bismo definirali pojam *margine*.

Definicija 2.2.3. *Neka je zadan linearni klasifikator f s parametrima (w, b). Definiramo marginu μ_i primjera za učenje $(x_i, y_i) \in S$ kao*

$$\mu_i = y_i f(x_i) = y_i \left(\langle w, x_i \rangle + b\right).$$

Ako klasifikator dobro klasificira zadani primjer za učenje $(x_i, y_i) \in S$, tada je margina μ_i tog primjera za učenje pozitivna jer y_i i $f(x_i)$ imaju isti predznak. Ako klasifikator dobro klasificira cijeli skup za učenje, tada je margina svakog primjera za učenje pozitivna. Možemo reći da razdvajajuća hiperravnina ima pozitivne margine za sve primjere za učenje. Ako je hiperravnina zadana s (w, b) ujedno i razdvajajuća hiperravnina, bez smanjenja općenitosti možemo pretpostaviti da vrijedi

$$\mu_i = y_i(\langle w, x_i \rangle + b) \geq 1, \quad 1 \leq i \leq n. \tag{2.2}$$

Ako je neka margina μ_i manja od 1 za neki primjer za učenje, možemo transformirati hiperravninu zadanu s (w, b) tako da dobijemo hiperravninu zadanu parametrima $(\lambda w, \lambda b)$,

gdje je $\lambda > 1$. Tada je nova margina jednaka

$$y_i(\langle \lambda w, x_i \rangle + \lambda b) = \lambda y_i(\langle w, x_i \rangle + b) = \lambda \mu_i > \mu_i.$$

Vidimo da odabirom odgovarajućeg parametra λ, možemo proizvoljno povećati margine, što opravdava pretpostavku $\mu_i \geq 1$ za svaki $1 \leq i \leq n$. Uvjet (2.2) je dovoljan da hiperravnina koju tražimo uistinu razdvaja podatke.

Optimizacijski problem

Uvodimo pojam *potpornog vektora* kako bismo izveli optimizacijski problem koji rješava problem pronalaska najbolje razdvajajuće hiperravnine.

Definicija 2.2.4. *Neka je zadan skup za učenje S i linearni klasifikator f s parametrima (w, b). Primjere za učenje $(x_i, y_i) \in S$ za koje vrijedi $\mu_i = 1$ nazivamo potporni vektori.*

Euklidska udaljenost između hiperravnine zadane s $\langle w, x \rangle + b = 0$ i točke x_0 je jednaka

$$\frac{|\langle w, x_0 \rangle + b|}{\|w\|}, \tag{2.3}$$

gdje je $\|\cdot\|$ Euklidska norma. Neka je (x_s, y_s) potporni vektor i neka je (x_i, y_i) primjer za učenje. Uočimo da vrijedi $|\langle w, x_i \rangle + b| = \mu_i$ jer je $|y_i| = 1$. Udaljenost potpornog vektora od hiperravnine zadane s (w, b) je tada manja od udaljenosti primjera (x_i, y_i) od iste hiperravnine.

$$\frac{|\langle w, x_s \rangle + b|}{\|w\|} = \frac{\mu_s}{\|w\|} = \frac{1}{\|w\|} \overset{(2.2)}{\leq} \frac{\mu_i}{\|w\|} = \frac{|\langle w, x_i \rangle + b|}{\|w\|}.$$

Zaključujemo da su potporni vektori primjeri za učenje najbliži razdvajajućoj hiperravnini te je ta udaljenost jednaka $\frac{1}{\|w\|}$. Kako bismo dobili što bolji klasifikator, želimo da je udaljenost potpornog vektora od razdvajajuće hiperravnine što veća. Dobili smo problem maksimizacije

$$\frac{1}{\|w\|} \to \max_{(w,b)} \tag{2.4}$$

koji je ekvivalentan sljedećem problemu minimizacije

$$\|w\|^2 \to \min_{(w,b)}. \tag{2.5}$$

Problem (2.5) i uvjet (2.2) zajedno daju kvadratni optimizacijski problem

$$\begin{cases} \|w\|^2 \to \min, \\ y_i\left(\langle w, x_i \rangle + b\right) \geq 1, \quad 1 \leq i \leq n, \\ (w, d) \in \mathbb{R}^d \times \mathbb{R}. \end{cases} \tag{2.6}$$

Rješenje problema (2.6) daje rješenje linearne metode potpornih vektora.

Linearno neodvojivi podaci

U prošlom odjeljku ponudili smo rješenje linearne metode potpornih vektora za linearno odvojive podatke. Pretpostavimo sada da podaci iz skupa za učenje nisu linearno odvojivi. Tada se problem (2.6) modificira u problem

$$\begin{cases} \|w\|^2 + C\sum_{i=1}^{n} \xi_i \to \min, \\ y_i\left(\langle w, x_i\rangle + b\right) \geq 1 - \xi_i, \quad 1 \leq i \leq n, \\ (w, d) \in \mathbb{R}^d \times \mathbb{R}, \end{cases} \tag{2.7}$$

gdje je $C > 0$ neka konstanta, a varijable $\xi_i, 1 \leq i \leq n$ *varijable opuštanja* (engl. *slack variables*). Svaku varijablu opuštanja pridružujemo jednom primjeru za učenje. Varijable opuštanja se definiraju s

$$\xi_i = \begin{cases} 0, & y_i\left(\langle w, x_i\rangle + b\right) \geq 0 \\ \dfrac{|\langle w, x_i\rangle + b|}{\|w\|}, & y_i\left(\langle w, x_i\rangle + b\right) < 0. \end{cases}$$

Vrijednost varijable opuštanja ovisi o tome klasificira li točno linearni klasifikator odgovarajući primjer za učenje. Prisjetimo se da ako linearni klasifikator dobro klasificira primjer za učenje (x_i, y_i), tada je vrijednost $y_i(\langle w, x_i\rangle + b)$ pozitivna. Varijabla opuštanja za primjer koji klasifikator točno klasificira je jednaka 0 pa nema razlike u starim i novim uvjetima za takve primjere.

S druge strane, varijabla opuštanja za primjer koji klasifikator krivo klasificira je jednaka udaljenosti vektora značajki od hiperravnine zadane s $\langle w, x\rangle + b$. Želimo da su takvi *devijantni* primjeri što bliži hiperravnini jer to znači da se ne izdvajaju previše od svojih klasa. Budući da minimiziramo sumu $\sum_{i=1}^{n} \xi_i$, minimizirat ćemo i udaljenosti devijantnih primjera od hiperravnine. Udaljenost devijantnog primjera je veća od jedan, jer bi inače taj primjer bio potporni vektor. Zato ima smisla uvjet $y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i$, jer su izrazi s obje strane nejednakosti negativni.

Višestruka klasifikacija

U dosadašnjem tekstu smo opisali linearnu metodu potpornih vektora primijenjenu na binarnu klasifikaciju pa ostaje pitanje kako se ta metoda koristi kod višestruke klasifikacije. Višestruka klasifikacija se u takvim slučajevima može rastaviti na više binarnih klasifikacija. Postoje dva načina na koji se višestruka klasifikacija može rastaviti na binarnu klasifikaciju. Pretpostavimo da imamo $k > 2$ klasa.

(a) **Jedna protiv ostalih** (engl. *One-vs-rest*)
Napravimo k binarnih klasifikacija za svaku od k klasa. Pri svakoj klasifikaciji pozitivnom klasom označimo primjere iz jedne klase, a negativnom klasom označimo primjere iz ostalih $k-1$ klasa. Na taj način dobijemo k linearnih klasifikatora $f_1, \ldots, f_k$ gdje klasifikator f_i poprima pozitivne vrijednosti za primjere iz klase i. Novi vektor značajki x pridružimo klasi c ako je vrijednost $f_c(x)$ maksimalna od svih vrijednosti $f_i(x), 1 \leq i \leq k$.

(b) **Jedna protiv jedne** (engl. *One-vs-one*)
Napravimo $\binom{k}{2}$ binarnih klasifikacija za svaki par klasa. Na taj način dobijemo isto toliko linearnih klasifikatora $f_{(i,j)}$, takvih da $f_{(i,j)}$ poprima pozitivne vrijednosti za primjere iz klase i, a negativne za primjere iz klase j. Neka je x novi vektor značajki koji želimo klasificirati. Definiramo vektor odluke $z \in \mathbb{R}^k$ sljedećim algoritmom.

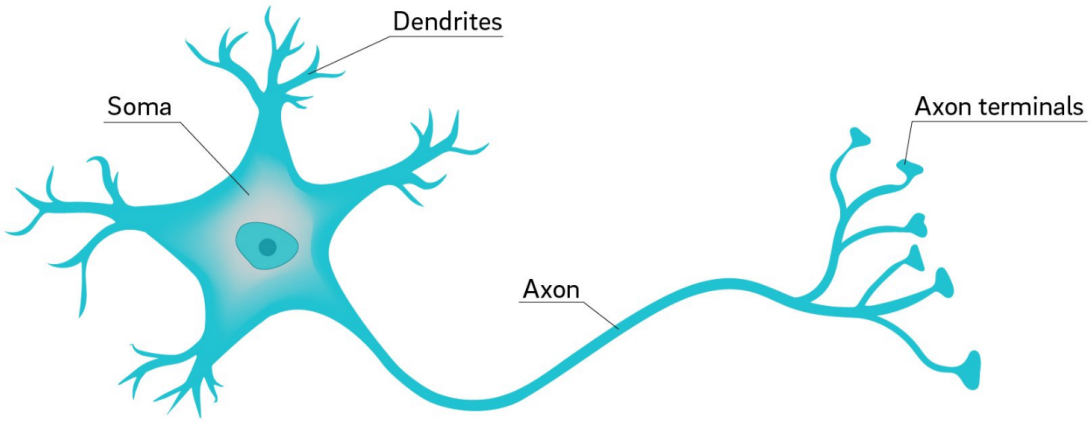
```
postavi z = (0,...,0)
za svaki klasifikator f[i,j]
    izračunaj p = f[i,j](x)
    ako je p > 0
        postavi z(i) = z(i) + p
    inače
        postavi z(j) = z(j) + abs(p)
```

Vektor odluke z na i-toj koordinati ima zbroj vrijednosti klasifikatora koje idu u prilog klasifikaciji primjera x klasi i. Primjer x pridružimo klasi c ako z ima maksimalnu vrijednost na c-toj koordinati.

2.3 Umjetna neuronska mreža

Umjetna neuronska mreža ili skraćeno neuronska mreža (engl. *artificial neural network*, kratica *ANN* ili *NN*) računalni je sustav koji oponaša živčani sustav ljudi i životinja. Živčani sustav se sastoji od živčanih stanica koje su međusobno povezane u mrežu koju zovemo *biološka neuronska mreža*. Da bismo opisali strukturu umjetne neuronske mreže, potrebno je prvo opisati strukturu biološke neuronske mreže.

Osnovna gradivna jedinica biološke neuronske mreže je *neuron* ili *živčana stanica*. Struktura neurona je prikazana na slici 2.2. Središnji dio neurona je *soma* ili *tijelo neurona*. Soma sadrži jezgru neurona u kojem se sintetiziraju proteini nužni za funkcioniranje stanice. *Dendriti* su veze koje se protežu iz tijela neurona. Dendriti dovode živčane impulse iz drugih živčanih stanica. *Akson* je relativno dugačka veza koja se proteže iz tijela neurona i služi za daljnji prijenos podražaja iz živčane stanice. Akson neurona se spaja s dendritima ostalih neurona. Na kraju aksona nalaze se *telodendroni* ili *aksonski terminali* koji povezuju stanicu s drugim neuronima. Možemo reći da dendriti dovode informacije, soma ih obrađuje, dok akson prosljeđuje informacije drugim neuronima.

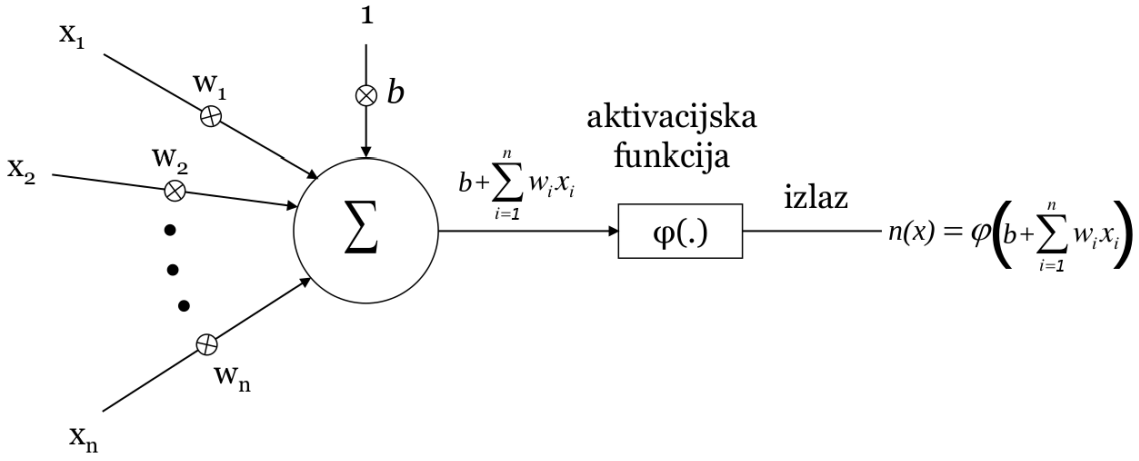


Slika 2.2: Struktura živčane stanice, tj. neurona. Dendriti dovode impulse u stanicu dok ih akson odvodi. Slika preuzeta iz [2].

Perceptron

Osnovna gradivna jedinica umjetne neuronske mreže je *perceptron*. Perceptron ima strukturu sličnu živčanoj stanici. Struktura perceptrona je prikazana na slici 2.3.

Slika 2.3: Struktura perceptrona, umjetnog neurona. Vrijednosti $x = (x_1, \ldots, x_n)$ predstavljaju izlazne informacije iz susjednih neurona, vrijednosti $b, w = (w_1, \ldots, w_n)$ parametre neurona, dok je $n(x)$ izlazna vrijednost neurona. Slika preuzeta iz [10].



Perceptron možemo matematički definirati kao funkciju $n : \mathbb{R}^n \to \{0, 1\}$ oblika

$$n(x) = \varphi\left(\langle w, x\rangle + b\right), \quad x \in \mathbb{R}^n,$$

gdje su $b \in \mathbb{R}, w \in \mathbb{R}^n$ parametri perceptrona, $\varphi : \mathbb{R} \to \mathbb{R}$ *aktivacijska funkcija* te $\langle \cdot, \cdot \rangle$ standardni Euklidski skalarni produkt. Aktivacijska funkcija je stvar izbora arhitekta neuronske mreže. Izbor aktivacijske funkcije uveliko utječe na ponašanje perceptrona. Jedan od mogućih izbora aktivacijske funkcije je

$$\varphi(y) = \begin{cases} 0, & y < 0, \\ 1, & y \geq 0. \end{cases}$$

Primijetimo da se perceptron s ovakvom aktivacijskom funkcijom ponaša jednako kao linearni klasifikator definiran s (2.1), samo što u ovom slučaju imamo skup vrijednosti $\{0, 1\}$ umjesto $\{1, 2\}$. Zaključujemo da je perceptron zapravo klasifikator iz definicije 2.1.1.

Učenje perceptrona

Postavlja se pitanje kako pronaći parametre tako da perceptron točno klasificira podatke. Neka je zadan primjer za učenje S i konstanta $\varepsilon > 0$. Perceptron se podešava sljedećim algoritmom.

```
postavi težine na nasumične vrijednosti
dok nisu svi uzorci ispravno klasificirani
    za svaki primjer za učenje (x,y) iz S
        izračunaj o = n(x)
        ako je o == y
            nastavi sa sljedećim primjerom za učenje
        inače
            postavi b = b + eps * (y-o)
            postavi w = w + eps * (y-o) * x
```

Zadnja linija pseudokoda je krucijalna za treniranje perceptrona jer u njoj modificiramo parametre perceptrona. Neka je (x, y) primjer za učenje i neka je $o = n(x)$ izlazna vrijednost perceptrona za taj primjer. Pretpostavimo da je izlaz o manji od oznake y, tj. $y - o > 0$. Tada želimo povećati buduću izlaznu vrijednost perceptrona za primjer (x, y).

(a) Ako je $x_i > 0$, tada će algoritam povećati w_i jer je tada $\varepsilon(y - o)x_i > 0$. Povećanje i-te komponente parametra w rezultira većoj vrijednosti $w_i x_i$ pa samim time i većoj budućoj izlaznoj vrijednosti perceptrona.

(b) S druge strane, ako je $x_i < 0$, tada će algoritam smanjiti w_i jer je tada $\varepsilon(y - o)x_i < 0$. Smanjenje i-te komponente parametra w rezultira većoj vrijednosti $w_i x_i$ pa samim time i većoj budućoj izlaznoj vrijednosti perceptrona.

Analogno razmišljanje vrijedi i za slučaj kada je $y-o < 0$, samo što tada želimo smanjiti buduću izlaznu vrijednost perceptrona.

Teorem 2.3.1. *(Minsky-Papert) Algoritam konvergira u konačnom broju koraka ako su primjeri za učenje linearno odvojivi i ako je stopa učenja $\varepsilon > 0$ dovoljno mala.*

Ponuđeni algoritam podešavanja perceptrona ne konvergira za podatke koji nisu linearno odvojivi. Zato uvodimo algoritam gradijentnog spusta (engl. *gradient descent*).

```
postavi težine na nasumične vrijednosti
dok nije zadovoljen kriterij zaustavljanja
    za svaki primjer za učenje (x,y) iz S
        izračunaj o = n(x)
        postavi db = eps * (t-o)
        postavi dw = eps * (t-o) * x
    postavi b = b + db
    postavi w = w + dw
```

Ovaj algoritam konvergira prema minimumu čak i kada podaci nisu linearno odvojivi.

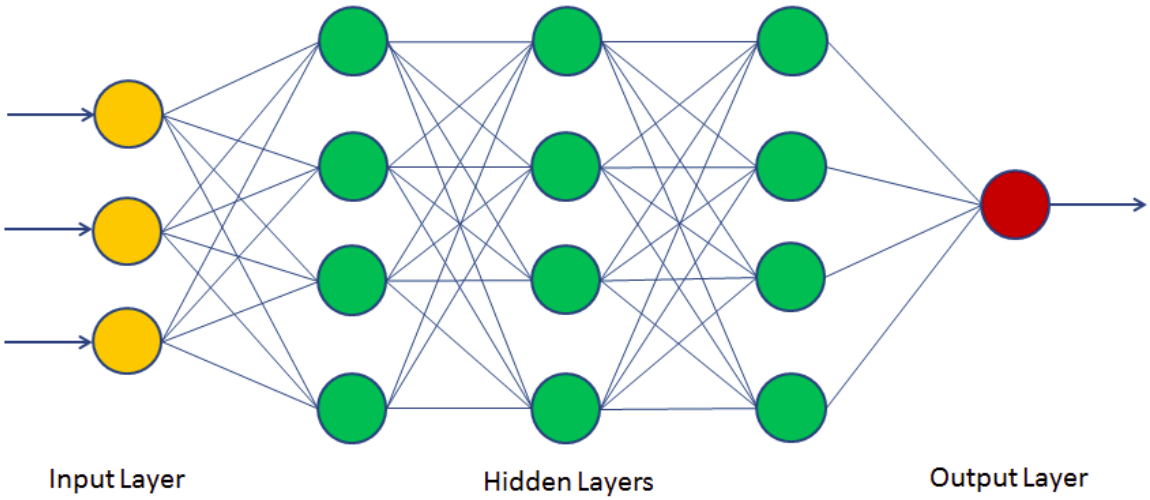
Teorem 2.3.2. *Neka je S skup za učenje veličine n. Neka je $\varepsilon > 0$ stopa učenja i neka su $b \in \mathbb{R}, w \in \mathbb{R}^d$ parametri perceptrona s aktivacijskom funkcijom $\varphi(y) = y$. Algoritam gradijentnog spusta asimptotski konvergira prema globalnom minimumu funkcije pogreške*

$$L(b, w) = \sum_{i=1}^{n} (y_i - n(x_i))^2, \quad b \in \mathbb{R}, w \in \mathbb{R}^d.$$

Topologija neuronske mreže

Uveli smo strukturu perceptrona koji imitira funkciju živčane stanice u živčanom sustavu. Umjetna neuronska mreža je struktura koju dobijemo kada povežemo više perceptrona zajedno u mrežu. Ako perceptrone shvatimo kao čvorove, tada je neuronska mreža zapravo usmjereni povezani graf koji se sastoji od *slojeva*. Slojevi neuronske mreže su prikazani na slici 2.4. Slojeve možemo zamisliti kao različite razine neuronske mreže. Perceptroni koji se nalaze u istom sloju nisu povezani vezama. *Dimenzija sloja* je broj perceptrona u sloju. Neuronska mreža ima *ulazni sloj*, *skrivene slojeve* te *izlazni sloj*.

Ulazni sloj ima dimenziju jednaku dimenziji vektora značajki. Svaki perceptron u ulaznom sloju odgovara jednoj komponenti vektora značajki. Kada želimo evaluirati vrijednost neuronske mreže za zadani vektor značajki, pošaljemo ga u ulazni sloj kao ulazne vrijed-



Slika 2.4: Struktura višeslojne acikličke potpuno povezane neuronske mreže. Žutom bojom je obojan ulazni sloj (engl. *input layer*), zelenom skriveni slojevi (engl. *hidden layers*), dok je crvenom obojan izlazni sloj (engl. *output layer*). Slika je preuzeta iz [6].

nosti odgovarajućih perceptrona u ulaznom sloju. Nakon što perceptroni izvrše sve račune, izlazne vrijednosti se šalju u sljedeći sloj.

Skriveni slojevi su slojevi koji nisu niti ulazni niti izlazni. U njima se događa većina računanja neuronske mreže. Nema nekog posebnog pravila za broj skrivenih slojeva ili broj čvorova u sloju te se njihova struktura određuje eksperimentalno ili u skladu s resursima. Kada bismo uzeli perceptron iz nekog skrivenog sloja i proučili njegove parametre, često ne bismo našli nikakvo posebno značenje. U literaturi se skriveni slojevi ponekad nazivaju i *crnom kutijom* (engl. *black box*) jer je jako teško shvatiti zašto je učenje neuronske mreže postavilo parametre na određenu vrijednost. Neuronska mreža koja ima puno skrivenih slojeva se naziva *duboka neuronska mreža* (engl. *deep neural network*).

Izlazni sloj je posljednji sloj u neuronskoj mreži. Ako neuronsku mrežu koristimo za klasifikaciju, izlazni sloj ima dimenziju jednaku broju klasa. Tada svaki perceptron u izlaznom sloju odgovara jednoj klasi. Novi primjer možemo pridružiti onoj klasi za koju je pripadni perceptron izbacio najveću vrijednost. Neuronske mreže se mogu koristiti i za vektorizaciju podataka tako da vrijednosti u izlaznom sloju predstavljaju vektor koji opisuje određeni podatak.

vrsta klasifikacije	**kategorije**	**opis**
povratne veze	*aciklička*	sve informacije teku slijeva nadesno, nema povratnih veza
	ciklička	informacije mogu teći iz sloja u prethodni sloj, postoje povratne veze
broj slojeva	*jednoslojna*	mreža ima samo ulazni i izlazni sloj, nema skrivenih slojeva
	višeslojna	mreža ima jedan ili više skrivenih slojeva
povezanost	*djelomično povezana*	postoje čvorovi koji nisu povezani s nekim čvorovima iz prethodnog sloja, broj veza nije maksimalan
	potpuno povezana	svi čvorovi u sloju su povezani sa svim čvorovima u sljedećem sloju, broj veza je maksimalan

Tablica 2.1: Vrste topologije neuronske mreže.

Topologija ili *arhitektura* mreže je način na koji povezujemo perceptrone (čvorove) i slojeve. Umjetne neuronske mreže se klasificiraju po različitim kategorijama ovisno o topologiji. Klasifikacija je prikazana u tablici 2.1. Engleski termin za acikličke neuronske mreže je *feedforward neural network* (kratica: *FNN* ili *FFNN*), dok se termin *recurrent neural network* (kratica: *RNN*) koristi za cikličke neuronske mreže.

Učenje umjetne neuronske mreže

Najpoznatiji algoritam učenja acikličke neuronske mreže na skupu za učenje se naziva *povratno širenje* (engl. *backpropagation*). Algoritam slijedi istu logiku kao algoritam gradijentnog spusta. Algoritam učenja cikličke neuronske mreže se naziva *povratno širenje kroz vrijeme* (engl. *backpropagation through time*, kratica *BPTT*) te je generelizacija algoritma povratnog širenja.

Prvi korak povratnog širenja je da se parametri neuronske mreže prvo inicijaliziraju na slučajne vrijednosti. Tada se za svaki primjer za učenje izračunaju vrijednosti u perceptronima izlaznog sloja. Parametri perceptrona u mreži se tada korigiraju unazad sloj po sloj počevši od izlaznog sloja, preko skrivenih slojeva pa sve do ulaznog sloja. Postupak se ponavlja za svaki primjer za učenje. Kada se evaluira cijeli skup za učenje, postupak se ponavlja dok se ne zadovolji određeni uvjet zaustavljanja. Više o povratnom širenju se može pročitati u [1].

Poglavlje 3

Eksperiment

Kako bismo usporedili metodu potpornih vektora i neuronsku mrežu, testirali smo obje metode na problemu klasifikacije proteina. Proteine smo izabrali iz 8 različitih proteinskih familija (tablica 3.1). Zapisi proteina su preuzeti iz javno dostupne baze podataka *Pfam* [7]. Svaka proteinska familije predstavlja različitu klasu proteina.

Tablica 3.1: Proteinske familije

ime proteinske familije	oznaka	podrijetlo
Upf2	PF04050	plijesan, biljke, svitkovci
G3P_antiterm	PF04309	želučane bakterije
Prolamin_like	PF05617	biljke (streptofiti)
hemP	PF10636	bakterije
KfrA_N	PF11740	bakterije
CABIT	PF12736	svitkovci, člankonošci
Ecm29	PF13001	svitkovci, gljive, biljke
S8_pro-domain	PF16470	svitkovci, člankonošci, oblići

Iz svake familije smo generirali uzorke od 5, 10, 15, 20 i 25 proteina koji čine skupove za treniranje te uzorke od 50 proteina koji čine skupove za testiranje. Uzorke smo generirali 10 puta da možemo više puta testirati modele.

Nakon što smo generirali skupove za treniranje i testiranje, trenirali smo modele na generiranim uzorcima. Vršili smo binarnu i višestruku klasifikaciju. Binarnu klasifikaciju smo proveli za sve moguće parove proteinskih familija, dok smo višestruku klasifikaciju proveli na svih 8 klasa istovremeno. Budući da postoji previše mogućih parova klasa (28),

odabrali smo 4 od 8 proteinskih familija za binarnu klasifikaciju da bismo dobili samo 6 različitih parova.

Metodu potpornih vektora koristili smo za binarnu i višestruku klasifikaciju, dok smo neuronskom mrežom proveli samo višestruku klasifikaciju. Za svaku klasifikaciju smo mjerili točnost klasifikatora na skupu za testiranje.

Svi algoritmi su implementirani u programskom jeziku *Python*.

Za metodu potpornih vektora koristili smo funkciju `sklearn.svm.SVC` iz biblioteke *Scikit-learn*. Postavili smo parametar `kernel` na opciju `"linear"` kako bismo dobili linearni klasifikator. Klasifikaciju metodom potpornih vektora je samostalno odradio autor.

Za klasifikaciju neuronskom mrežom koristili smo sustav *Transformer-XL* čija se dokumentacija može naći u [5]. Klasifikaciju neuronskom mrežom su odradili Tomislav Ivek i Domagoj Vlah.

Vektorizacija proteina

Kako bismo mogli primijeniti metodu potpornih vektora na proteine, trebamo transformirati nizove aminokiselina u vektore. Jedan od načina je da promatramo frekvencije četvorki aminokiselina. Budući da su proteini građeni od 20 aminokiselina, postoji $20^4 = 160\,000$ mogućih četvorki aminokiselina, što znači da jedan protein možemo reprezentirati vektorom dimenzije 160 000. Budući da naši proteini nemaju više od 600 aminokiselina u lancu, jasno je da će vektori frekvencija biti puni nula.

Slika 3.1: 20 osnovnih aminokiselina i njihove oznake

aminokiselina	oznaka	aminokiselina	oznaka
alanin	A	leucin	L
arginin	R	lizin	K
asparagin	N	metionin	M
asparaginska kiselina	D	fenilananin	F
cistein	C	prolin	P
glutamin	Q	serin	S
glutaminska kiselina	E	treonin	T
glicin	G	triptofan	W
histidin	H	tirozin	Y
izoleucin	I	valin	V

Osim apsolutnih frekvencija, možemo promatrati vektore relativnih frekvencija. Relativne frekvencije dobijemo tako da apsolutne frekvencije *normaliziramo*. Vektore frekvencija možemo normalizirati na više načina. Jedan od načina je da apsolutne frekvencije jednog proteina podijelimo s apsolutnim frekvencijama četvorki aminokiselina u svih 8 proteinskih familija. Drugi način je da apsolutne frekvencije jednog proteina podijelimo s ukupnim brojem četvorki aminokiselina u tom proteinu. Ako uzmemo logaritam relativnih frekvencija, dobili smo još jednu metodu vektorizacije proteina.

Poglavlje 4

Rezultati

U ovom poglavlju su prikazani rezultati svih provedenih klasifikacija. Pri svakoj klasifikaciji n označava veličinu skupa za treniranje, dok m označava veličinu skupa za testiranje.

Sve klase su jednako zastupljene u svim skupovima za treniranje i testiranje. U binarnoj klasifikaciji, pola primjera je iz jedne familije, a pola iz druge. Ako se radi o višestrukoj klasifikaciji s 8 klasa, svaka klasa je zastupljena u osmini primjera iz skupova za treniranje i testiranje.

Kod metode potpornih vektora, klasifikacije su provedene na vektorima koji su dobiveni različitim postupcima vektorizacije proteina (vektorizacije A, B, C i D). Vektorizacija A (oznaka *VektorA*) predstavlja rezultate klasifikacije vektora dobivenih brojanjem apsolutnih frekvencija. Vektorizacija B (oznaka *VektorB*) predstavlja vektorizaciju relativnim frekvencijama gdje je normalizirajući faktor vektor apsolutnih frekvencija u svih 8 proteinskih familija. Vektorizacija C (oznaka *VektorC*) predstavlja logaritme vektora s oznakom *VektorB*. Vektorizacija D (oznaka *VektorD*) predstavlja normalizirane relativne frekvencije takve da je zbroj relativnih frekvencija u jednom proteinu jednak 1.

Svaka klasifikacija je ponovljena na 10 različitih uzoraka za svaki par klasa i za svaku veličinu skupa za testiranje. Za svaku klasifikaciju su mjerene točnosti na skupovima za testiranje te je izračunat njihov prosjek.

4.1 Binarna klasifikacija

Binarnu klasifikaciju smo proveli linearnom metodom potpornih vektora. Uzeli smo u obzir 4 proteinske familije *PF04050, PF05617, PF11740* i *PF13001*, kako bismo dobili ukupno 6 različitih parova klasa za klasifikaciju. Prosječne točnosti na 10 generiranih uzoraka su prikazane u tablici 4.1.

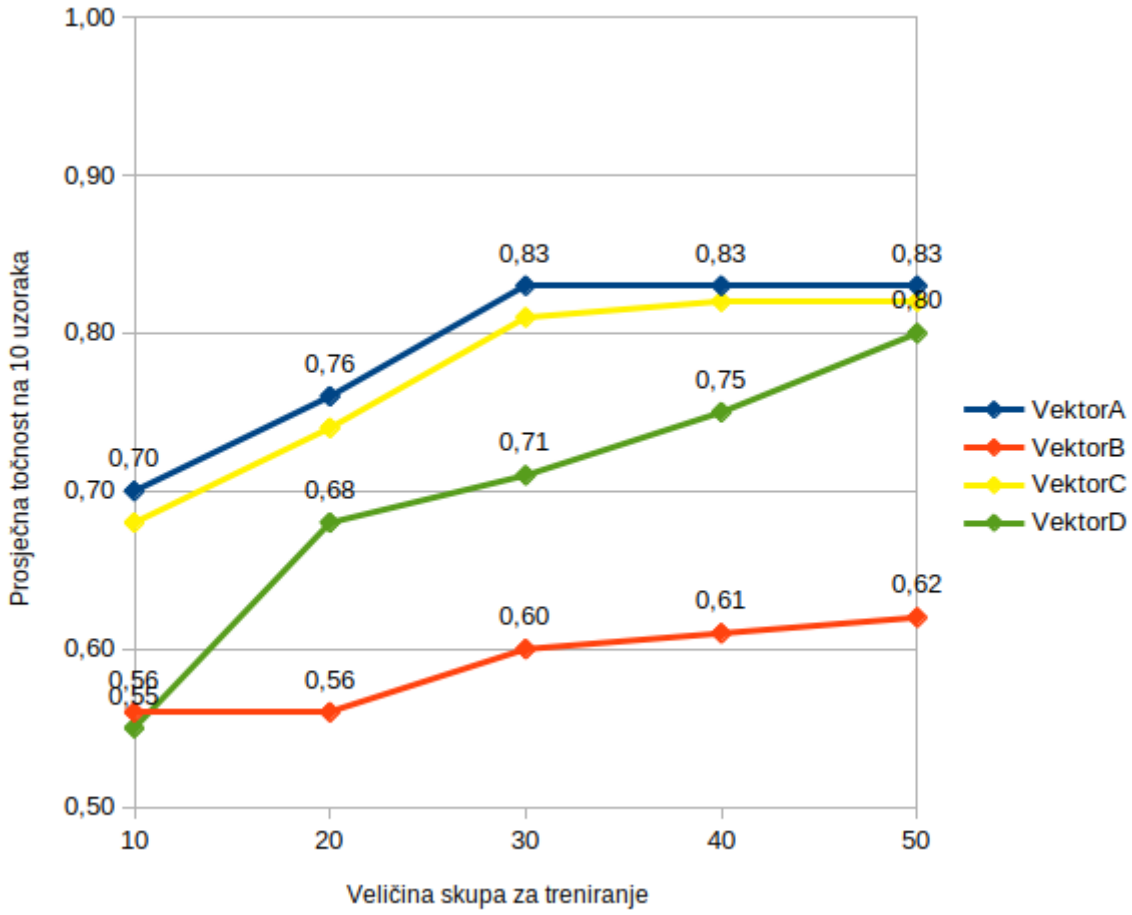
Iz tablica možemo primijetiti da je vektorizacija B poražavajuće loša. Klasifikatori vektorizacije B nisu imali prosječnu točnost veću od 0.66. Vektorizacije A i C u većini

Tablica 4.1: Prosječne točnosti binarne klasifikacije metodom potpornih vektora za različite n i načine vektorizacije proteina. Veličina skupa za testiranje je svugdje $m = 100$.

klasa 1	klasa 2	n	VektorA	VektorB	VektorC	VektorD
PF04050	PF05617	10	0.70	0.56	0.68	0.55
PF04050	PF11740	10	0.80	0.66	0.82	0.71
PF04050	PF13001	10	0.65	0.58	0.62	0.62
PF05617	PF11740	10	0.55	0.51	0.53	0.55
PF05617	PF13001	10	0.58	0.61	0.57	0.53
PF11740	PF13001	10	0.64	0.56	0.62	0.56
PF04050	PF05617	20	0.76	0.56	0.74	0.68
PF04050	PF11740	20	0.85	0.58	0.86	0.72
PF04050	PF13001	20	0.73	0.58	0.74	0.64
PF05617	PF11740	20	0.61	0.53	0.57	0.59
PF05617	PF13001	20	0.63	0.61	0.63	0.56
PF11740	PF13001	20	0.67	0.56	0.68	0.58
PF04050	PF05617	30	0.83	0.60	0.81	0.71
PF04050	PF11740	30	0.93	0.60	0.91	0.78
PF04050	PF13001	30	0.79	0.59	0.80	0.66
PF05617	PF11740	30	0.65	0.57	0.61	0.67
PF05617	PF13001	30	0.65	0.65	0.65	0.62
PF11740	PF13001	30	0.71	0.56	0.74	0.61
PF04050	PF05617	40	0.83	0.61	0.82	0.75
PF04050	PF11740	40	0.89	0.58	0.91	0.77
PF04050	PF13001	40	0.89	0.62	0.90	0.68
PF05617	PF11740	40	0.74	0.57	0.69	0.61
PF05617	PF13001	40	0.72	0.64	0.73	0.60
PF11740	PF13001	40	0.77	0.61	0.79	0.75
PF04050	PF05617	50	0.83	0.62	0.82	0.80
PF04050	PF11740	50	0.88	0.60	0.90	0.76
PF04050	PF13001	50	0.83	0.65	0.84	0.77
PF05617	PF11740	50	0.77	0.53	0.73	0.58
PF05617	PF13001	50	0.72	0.66	0.72	0.64
PF11740	PF13001	50	0.74	0.59	0.77	0.66

slučajeva daju značajno veću točnost od vektorizacija *B* i *D*. Kada bismo poredali vektorizacije po uspješnosti, možemo reći da su vektorizacije *A* i *C* najbolje vektorizacije, dok je vektorizacija *B* najgora od ove četiri vektorizacije.

Na slici 4.1 prikazana je ovisnost točnosti o veličini skupa za treniranje *n*. Na dijagramu su prikazani rezultati binarne klasifikacije klasa *PF04050* i *PF05617* za različite načine vektorizacije. Za ostale parove proteinskih familija situacija je analogna. Možemo primijetiti da vektorizacije *A* i *C* imaju gotovo pa jednaku točnost za sve veličine skupa za treniranje. Iz dijagrama se također vidi da vektorizacije *A* i *C* imaju značajno veću točnost od vektorizacija *B* i *D*. Povećanjem veličine skupa za treniranje razlika u točnosti između vektorizacija *A* i *B* se povećava, dok se razlika u točnosti između vektorizacija *A* i *D* smanjuje.



Slika 4.1: Ovisnost točnosti o veličini skupa za treniranje

Prosječna točnost se povećava povećanjem veličine skupa za treniranje. To je bilo za očekivati jer ako neki model *vidi* više podataka, više će informacija apsorbirati te će biti *pametniji*. Možemo primijetiti da nakon određene veličine skupa za treniranje (npr. $n = 30$ za vektorizacije A, B, C) točnost više ne raste te čak počne i blago padati. Ovaj fenomen možemo objasniti kao *overfitting* ili *pretreniranost modela*. Ako model treniramo na previše podataka, on će jako dobro opisivati skup za učenje, ali će se zato *slomiti* na novim podacima.

Prosječne točnosti klasifikacija nam mogu dati informaciju o sličnosti proteinskih familija. Ako je točnost binarne klasifikacije relativno niska (blizu 0.50), to znači da su te dvije klase teško odvojive pa samim time i vrlo slične. Ako je točnost binarne klasifikacije relativno visoka (blizu 1.00), to znači da su te dvije klase lako odvojive pa samim time i vrlo različite. U tablici 4.2 možemo vidjeti prosječne točnosti binarnih klasifikacija za različite parove proteinskih familija (za sve vrste vektorizacija). Vidimo da su familije *PF05617,PF11740* i *PF13001* slične jer su prosječne točnosti klasifikacija u kojima sudjeluju ove familije relativno niske. Familija *PF04050* se dobro razlikuje od ovih triju familija jer klasifikacije u kojima sudjeluje ta klasa imaju relativno visoku točnost.

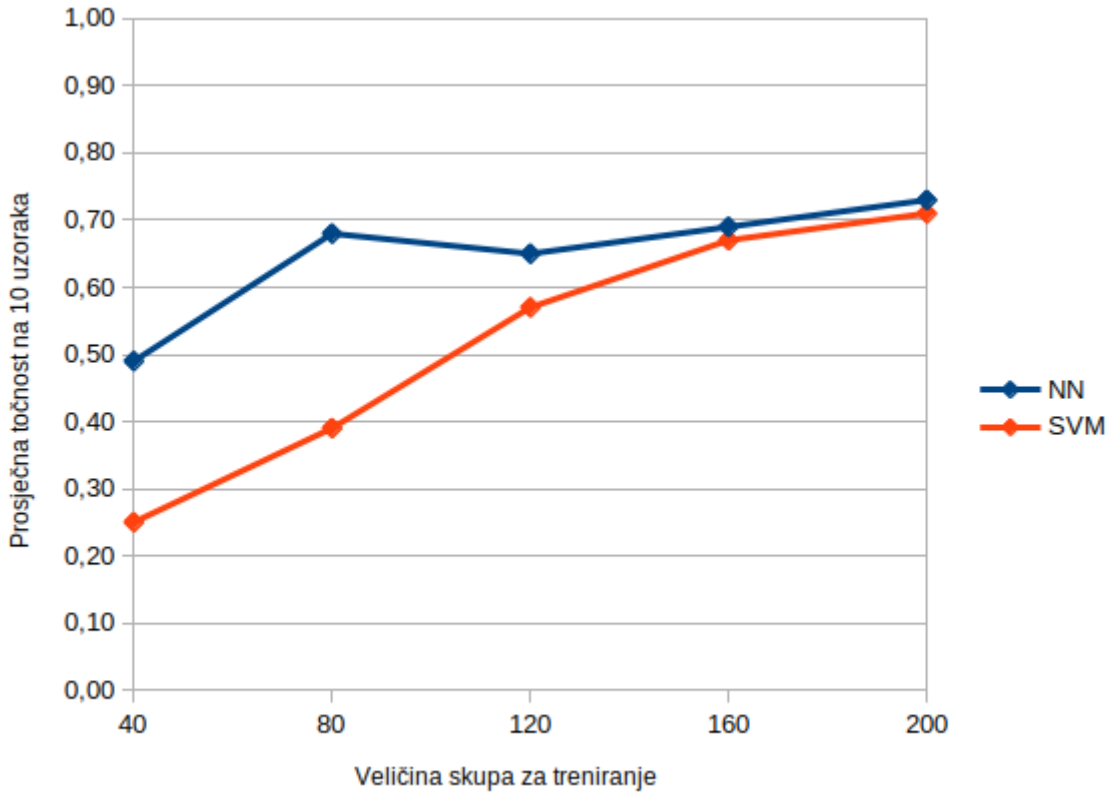
klasa 1	klasa 2	prosječna točnost
PF04050	PF05617	0.71
PF04050	PF11740	0.78
PF04050	PF13001	0.71
PF05617	PF11740	0.61
PF05617	PF13001	0.64
PF11740	PF13001	0.66

Tablica 4.2: Prosječne točnosti binarne klasifikacije metodom potpornih vektora za različite parove proteinskih familija.

4.2 Višestruka klasifikacija

Višestruku klasifikaciju smo proveli linearnom metodom potpornih vektora i klasifikacijom pomoću neuronske mreže. U primjeni metode potpornih vektora smo koristili samo vektorizaciju A jer se ona pokazala kao najbolja (uz vektorizaciju C) u binarnoj klasifikaciji. Uzeli smo u obzir svih 8 proteinskih familija pa je broj klasa jednak $k = 8$. Prosječne točnosti na 10 generiranih uzoraka su prikazane u tablici 4.3.

Na slici 4.2 prikazana je ovisnost točnosti o veličini skupa za treniranje n. Na slici su grafički prikazani rezultati iz tablice 4.3. Vidimo da je točnost neuronske mreže značajno



Slika 4.2: Ovisnost točnosti višestruke klasifikacije o veličini skupa za treniranje.

veća od točnosti metode potpornih vektora za male uzorke ($n = 40, 80$). Kako se veličina skupa za treniranje povećava, točnost metode potpornih vektora značajno raste te poprima točnost gotovo jednaku točnosti neuronske mreže. Možemo reći da neuronska mreža puno brže *upija* informacije iz podataka jer već na manjem skupu za treniranje ima zadovoljavajuću točnost. Metoda potpornih vektora sustiže neuronsku mrežu čim je skup za učenje dovoljno velik ($n \geq 120$). Neuronska mreža je jako kompliciran model (24 milijuna parametara), te je iznenađujuće da s puno jednostavnijim modelom metode potpornih vektora (160 tisuća parametara) možemo dostići gotovo jednaku točnost.

Tablica 4.3: Prosječne točnosti višestruke klasifikacije metodom potpornih vektora (oznaka *SVM*) i neuronskom mrežom (oznaka *NN*) za različite *n*.

k	*n*	*m*	NN	SVM
8	40	400	0.49	0.25
8	80	400	0.68	0.39
8	120	400	0.65	0.57
8	160	400	0.69	0.67
8	200	400	0.73	0.71

Poglavlje 5

Zaključak

Na temelju provedenog istraživanja možemo izvući sljedeće zaključke o klasifikaciji proteinskih familija.

(a) Povećanjem veličine skupa za učenje povećava se točnost klasifikatora, ali je zato rast točnosti sve sporiji.

(b) U metodi potpornih vektora najveću točnost daju vektorizacije apsolutnim frekvencija (vektorizacija A) te logaritmima relativnih frekvencija (vektorizacija C). Prednost dajemo vektorizaciji apsolutnim frekvencijama zbog komputacijske jednostavnosti.

(c) Umjetna neuronska mreža ima visoku točnost na malom skupu za učenje. Točnost umjetne neuronske mreže raste sporo povećanjem veličine skupa za učenje. Neuronska mreža loše upija informacije iz novih podataka.

(d) Točnost metode potpornih vektora raste povećanjem veličine skupa za učenje. Metoda potpornih vektora dobro upija informacije iz novih podataka.

(e) Neuronska mreža nema značajno veću točnost od metode potpornih vektora na velikim skupovima za učenje.

Možemo reći da umjetna neuronska mreža nije opravdala velik broj parametara modela. Metoda potpornih vektora je jednostavan te dovoljno dobar model za klasifikaciju proteinskih familija u odnosu na umjetnu neuronsku mrežu.

Bibliografija

[1] Ethem Alpaydın, *Introduction to Machine Learning, 2nd edition*, The MIT Press, 2010.

[2] David Baillot, *struktura neurona*, 2018., https://medicalxpress.com/news/2018-07-neuron-axons-spindly-theyre-optimizing.html, preuzeto u kolovozu 2020.

[3] Damir Bakić, *Linearna algebra*, Školska knjiga, 2008.

[4] Nello Cristianini i John Shawe-Taylor, *An Introduction to Support Vector Machines and other kernel-based learning methods*, Cambridge university press, 2000.

[5] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le i Ruslan Salakhutdinov, *Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context*, (2019), https://arxiv.org/abs/1901.02860.

[6] Leon Kwok Hing, *struktura neuronske mreže*, 2019., preuzeto u kolovozu 2020.

[7] European Bioinformatics Institute, *Pfam version 33.1*, 2019., https://pfam.xfam.org/, preuzeto u srpnju 2020.

[8] Gareth James, Daniela Witten, Trevor Hastie i Robert Tibshirani, *An Introduction to Statistical Learning with Applications in R*, Springer Texts in Statistics, 2017.

[9] Lavoslav Čaklović, *Geometrija linearnog programiranja*, Element, 2010.

[10] Tomislav Šmuc, *materijali kolegija Strojno učenje na Prirodoslovno-matematičkom fakultetu Sveučilišta u Zagrebu*, 2020., https://web.math.pmf.unizg.hr/nastava/su/materijali/, preuzeto u kolovozu 2020.

Sažetak

Cilj ovog rada je ispitati mogućnosti modela umjetne neuronske mreže na problemu klasifikacije proteinskih familija. U prvom poglavlju su predstavljeni osnovni matematički pojmovi iz linearne algebre i optimizacije potrebni za razumijevanje korištenih metoda strojnog učenja. U drugom poglavlju su uvedeni osnovni pojmovi strojnog učenja te je objašnjena matematička pozadina korištenih metoda. Metode koje su korištene u istraživanju su metoda potpornih vektora te umjetna neuronska mreža. U trećem poglavlju su predstavljena provedena istraživanja. Izvršena je binarna i višestruka klasifikacija na sekvencama proteina tako da proteinske familije predstavljaju različite klase. U četvrtom poglavlju su predstavljeni rezultati istraživanja te je napravljena usporedba mogućnosti umjetne neuronske mreže i metode potpornih vektora. U posljednjem poglavlju je ponuđen zaključak.

Summary

The goal of this paper is to examine the applicability of an artificial neural network model to the problem of protein classification. The first chapter provides a brief introduction to basic concepts of linear algebra and optimization. The second chapter introduces basic machine learning concepts and explains the mathematical background. Models used in research are support vector machine and artificial neural network. The third chapter presents the workflow of the conducted research. Research consists of binary and multiclass classification of protein sequences, with protein families representing different classes. The fourth chapter presents the main results of the thesis and compares the results of the two models. The last chapter presents the conclusions.

Životopis

Borna Runac rođen je 18. ožujka 1997. godine u Zagrebu kao dijete učiteljice i policajca. Svoje predškolsko razdoblje proveo je kao podstanar u raznim zagrebačkim kvartovima dok se njegova obitelj nije stalno preselila na Knežiju. Na Knežiji završava OŠ Matije Gupca te usporedno pohađa radionice Zagrebačkog računalnog saveza. Nakon završetka osnovne škole upisuje V. gimnaziju u Zagrebu. Tijekom školovanja sudjeluje na natjecanjima iz informatike, fizike i kemije od kojih treba izdvojiti državna natjecanja iz informatike i fizike u osnovnoj školi te državno natjecanje iz fizike u srednjoj školi. U srpnju 2015. upisuje preddiplomski studij Matematike na Prirodoslovno-matematičkom fakultetu u Zagrebu. Nakon završetka preddiplomskog studija 2018. godine upisuje diplomski studij Matematičke statistike na istom fakultetu. Iste godine postaje stipendist zagrebačkog ureda tvrtke Atos Convergence Creators d.o.o. Od područja interesa treba navesti podatkovne znanosti (engl. *Data Science*), složeno pretraživanje podataka (engl. *Data Mining*), strojno učenje (engl. *Machine Learning*) te bioinformatiku.