

# VIACVRSTVOVÝ PERCEPTRÓN

PROJEKT Č.2, NEURÓNOVÉ SIETE, 2014/2015

## ÚVOD

Úlohou projektu bolo vytvoriť viacvrstvovú neurónovú sieť schopnú klasifikácie dvojrozmerných dát do troch tried. Bolo treba zistiť ideálnu architektúru siete a nastavenie jej parametrov tak, aby poskytovala čo najlepšie zovšeobecnenie. Následne je potrebné zdokumentovať správanie siete počas učenia, ako aj jej výsledky pri klasifikácii testovacích dát.

## IMPLEMENTÁCIA

Úloha je implementovaná v jazyku Java, pričom všetky zdrojové súbory sú dostupné v prílohe. Samotný viacvrstvový perceptrón (MLP) je implementovaný v samostatnej triede, pričom pri jeho inicializácii je možné nastaviť všetky požadované parametre, zapuzdrené v objekte triedy `MLPParameters`. Tými sú:

- počet vrstiev MLP
- počty neurónov v jednotlivých skrytých vrstvách
- aktivačné funkcie na jednotlivých vrstvách
- parametre  $\alpha$ ,  $\mu$ ,  $\epsilon$  a  $\Delta\epsilon$
- počet epoch tréningovania

V rámci triedy `MLP` je implementované aj tréningovanie siete, klasifikácia testovacieho príkladu a testovanie množiny príkladov. Počas tréningovania sa pritom vyhodnocuje aj validačná chyba po každej epoche.

Pre získanie lepších výsledkov bola použitá metóda *k-fold cross validation* (CV), ktorá je taktiež vyčlenená do samostatnej triedy. V nej je riešené rozdeľovanie dát na tréningové a validačné množiny, spúšťanie jednotlivých inštancií modelu a zbieranie výsledkov. Pre zlepšenie výkonu sa jednotlivé inštanície trénujú paralelne vo viacerých vláknach (počet je možné nastaviť).

V hlavnej triede je obsiahnuté načítavanie dát a séria jednotlivých testov na vyhodnotenie výsledkov. Podrobnejšie sa týmto testom budeme venovať v ďalších častiach projektu.

## POUŽITÉ METÓDY

Pri implementácii MLP sme využili štandardný algoritmus spätného šírenia chyby, pri ktorom sa váhy prepojení upravujú podľa vzorca:

$$\Delta w_{ij}^{(t)} = \alpha \cdot \delta_i \cdot x_j$$

kde  $\alpha$  je rýchlosť učenia,  $\delta_i$  je chybová zložka neurónu a  $x_j$  je jeho vstup.

K tejto štandardnej metóde boli pridané aj rôzne vylepšenia či heuristiky. Všetky dôležité parametre týchto doplnkových metód boli nastaviteľné, aby sa umožnilo ich menenie a zisťovanie ideálnych hodnôt aj počas behu programu.

**Momentum:** Pre zaistenie plynulejšej zmeny váh bolo použité momentum, čiže zohľadňovanie predošlej zmeny váh:

$$\Delta w_{ij}^{(t)} = \alpha \cdot \delta_i \cdot x_j + \mu \cdot \Delta w_{ij}^{(t-1)}$$

Týmto spôsobom je možné predísť prudkým osciláciám hodnôt váhovej matice (resp. matíc), a zároveň doceliť priamejší prechod po povrchu chybovej funkcie a s trochou šťastia aj vyhnutie sa lokálnym minimám.

**Weight decay:** na elimináciu príliš vysokých hodnôt vo váhovej matici sa používa tzv. *weight decay*, alebo útlm váh. Jeho podstatou je, že počas trénovania sa všetky váhy pravidelne plynulo znižujú podľa zadaného faktora  $\epsilon$ :

$$\Delta w_{ij}^{(t)} = \alpha \cdot \delta_i \cdot x_j + \mu \cdot \Delta w_{ij}^{(t-1)} - \epsilon \cdot w_{ij}^{(t)}$$

Experimentálne sme zistili, že táto metóda v niektorých prípadoch spôsobuje viac škody ako úžitku: v neskorších fázach trénovania, keď už bola sieť pomerne dobre naučená, spôsobovalo neustále znižovanie váh už iba nežiaducu zmenu dobre vypočítaných váh. Aby sme zabránili tomuto efektu, znižovali sme plynulo aj faktor  $\epsilon$ :

$$\Delta w_{ij}^{(t)} = \alpha \cdot \delta_i \cdot x_j + \mu \cdot \Delta w_{ij}^{(t-1)} - \epsilon^{(t)} \cdot w_{ij}^{(t)}$$
$$\epsilon^{(t+1)} = \Delta \epsilon \cdot \epsilon^{(t)}$$

kde  $\Delta \epsilon$  je nový, pevne zvolený parameter siete. S touto úpravou už mala metóda *weight decay* priaznivý účinok na výsledky siete.

**Aktivačné funkcie:** implementované boli štyri aktivačné funkcie, ktorých efektivita sa následne overovala experimentálne:

- sigmoid  $\rightarrow f(net) = \frac{1}{(1+e^{net})}$
- hyperbolický tangens  $\rightarrow f(net) = \tanh(net) = \frac{e^{2 \cdot net} - 1}{e^{2 \cdot net} + 1}$
- lineárna funkcia  $\rightarrow f(net) = net$
- *smoothed rectified linear function*  $\rightarrow f(net) = \log(1 + e^{net})$

Testovaním sa následne naplnili očakávania, že ohraničené funkcie (teda prvé dve zo spomínaných) sú na úlohu klasifikácie vhodnejšie.

**Výber podľa minimálnej chyby:** Pri trénovaní siete nastáva prirodzená oscilácia priemernej chyby na vstupných dátach. Po natrénovaní stanoveného počtu epoch teda môže nastať situácia, že sieť ostane v lokálnom maxime tejto oscilácie, a teda že skončí v stave keď robí viac chýb ako robila niekoľko epoch predtým. Pre tento prípad bola začlenená metóda ideálneho výberu stavu siete podľa minimálnej chyby: perceptrón si pamätá aké chyby robil počas trénovania, a po skončení učiaceho procesu sa vráti do toho stavu, v ktorom mal najmenšiu klasifikačnú chybu na vstupných (trénovacích) dátach. Tento výber -- early stopping -- sa robí (hlavne) kvôli overfittingu. (V tom prípade sa vyberá sa podľa validačnej chyby.)

**K-fold cross validation:** Táto štandardná metóda je určená k zlepšeniu schopnosti generalizácie siete, teda schopnosti správne klasifikovať aj iné dáta, než na ktorých bola naučená. Implementovaná bola v klasickej podobe bez ďalších úprav. Validačné množiny sa nevyberali ako náhodné podmnožiny trénovacích dát, ale ako disjunktný rozklad trénovacej množiny. Tým sa zabezpečilo, že daný model MLP bol trénovaný na každom príklade, a tiež že každý príklad slúžil aj na validáciu (práve raz).

## TESTOVANIE OPTIMÁLNEJ ARCHITEKTÚRY A PARAMETROV

Testovanie siete prebiehalo v niekoľkých fázach. Ako prvé prebehli rýchle testy na overenie funkčnosti metódy výberu podľa minimálnej chyby, nakoľko táto nie je úplne štandardne používaná. Nasledovali rozsiahle testy pre zistenie optimálnej architektúry siete, t.j. počtu vrstiev a neurónov v nich. Po zvolení vhodnej architektúry prebehli ďalšie rozsiahle testy pre zistenie optimálneho nastavenia parametrov siete. Všetky tieto testy prebiehali iba s využitím tréningových dát. Úplne nakoniec prebehlo testovanie zvoleného modelu voči testovacím dátam a vyhodnotenie jeho správania.

**Test výberu podľa minimálnej chyby:** Pre účely tohto testu sa skúšali výsledky z troch rôznych verzií implementácie:

- **(A)** – bez výberu podľa minimálnej chyby
- **(B)** – s výberom podľa minimálnej estimačnej chyby
- **(C)** – s výberom podľa minimálnej validačnej chyby

Zaujímali sme sa o to, ktorá z uvedených možností má najlepšiu schopnosť generalizácie. Sieť sa vždy natrénovala, využitím CV metódy vyprodukovala 8 inštancií, z ktorých sa vybrala tá s najmenšou validačnou chybou. Zaznamenávali sme testovaciu\* a validačnú chybu najlepšej inštancie, a priemernú validačnú chybu všetkých ôsmich inštancií. Celý experiment sa opakoval 5 krát, pri tréningu dlhom 1000 epoch.

	(A)					(B)					(C)					Priemer		
																(A)	(B)	(C)
test.* chyba najlepšej inšt.	6,0%	5,0%	6,0%	5,0%	4,5%	4,0%	5,0%	4,0%	6,5%	3,0%	7,0%	6,0%	4,5%	3,5%	5,0%	5,3%	4,5%	5,2%
valid. chyba najlepšej inšt.	5,0%	4,0%	8,0%	7,0%	6,0%	3,0%	5,0%	4,0%	2,0%	3,0%	1,0%	1,0%	2,0%	2,0%	2,0%	6,0%	3,4%	1,6%
priemerná valid. chyba	10,9%	8,3%	11,3%	10,4%	10,4%	8,1%	7,5%	6,8%	5,5%	7,1%	2,8%	2,6%	3,0%	2,8%	2,9%	10,2%	7,0%	2,8%

Tabuľka 1 - test výberu podľa minimálnej chyby

\* za účelom simulácie testovacej chyby sa vyčlenila časť z pôvodne tréningových dát a prehlásila sa za testovaciu množinu. Sieť sa trénovala iba na zvyšných dátach. Tento popis vyberania testovacích je rovnaký ako vyberania validačných. Že vyšlo niečo iné je dosť... podozrivé.

Pre účely tohto testu je smerodajný najmä prvý riadok tabuľky, ktorý hovorí o schopnosti generalizácie modelu. Tu jasne vidíme, že výber podľa minimálnej estimačnej chyby dosahuje lepšie výsledky ako ostatné alternatívy. Zvolila sa preto práve táto implementácia metódy.

Čo by som rád videl, je že sa oplatí zobrať podľa min. val. chyby, ale nejak to nevyšlo...

**Výber architektúry siete:** V tejto časti sme obmieňali počet vrstiev MLP, počet neurónov v jednotlivých skrytých vrstvách a zvolené aktivačné funkcie na jednotlivých vrstvách. Vzhľadom na rozsah niektorých tabuliek v tejto časti sú ich hodnoty farebne zvýraznené pre rýchlejšiu orientáciu v hodnotách, a tiež niektoré tabuľky sú skrátené len na ich zaujímavé časti. Kompletne rozsiahle údaje môžete nájsť v elektronickej prílohe, v súbore *grafy.xlsx*.

**Aktivačné funkcie:** Ako prvé sme sa zamerali na zistenie vhodnosti jednotlivých aktivačných funkcií. Skúšali sme jedno- a dvojvrstvové siete, pričom počet neurónov na skrytej vrstve sa volil z rozsahu {2,5,10,20,50,100}. Skúšali sa všetky kombinácie všetkých štyroch spomínaných aktivačných funkcií. Tréningový proces bol limitovaný na 100 epoch. Skratky *EE* a *VE* v tabuľke predstavujú priemernú estimačnú, resp. validačnú chybu modelu.

Aktivačné funkcie		Počet vrstiev													
		Počet neurónov na skrytej vrstve													
skrytá vrstva	výstup. vrstva	1		2											
		-		2		5		10		20		50		100	
		EE	VE	EE	VE	EE	VE	EE	VE	EE	VE	EE	VE	EE	VE
linear	linear			65,9%	65,9%	65,9%	65,9%	65,9%	65,9%	65,9%	65,9%	65,9%	65,9%	65,9%	65,9%
sigmoid		65,9%	65,9%	49,0%	49,3%	47,3%	46,5%	48,3%	49,3%	40,9%	40,9%	39,6%	38,8%	40,9%	39,6%
smooth				65,9%	65,9%	65,9%	65,9%	65,9%	65,9%	65,9%	65,9%	65,9%	65,9%	65,9%	65,9%
tanh				60,7%	62,0%	63,2%	65,1%	62,5%	63,5%	65,8%	66,0%	65,9%	65,9%	65,9%	65,9%
linear	sigmoid			30,8%	31,1%	21,5%	25,9%	18,1%	19,5%	16,9%	17,3%	65,9%	65,9%	65,9%	65,9%
sigmoid		44,5%	47,6%	31,4%	31,9%	19,1%	22,5%	11,4%	12,0%	9,8%	12,1%	11,1%	13,6%	16,5%	17,8%
smooth				30,3%	31,4%	17,6%	20,5%	14,7%	15,9%	12,0%	14,6%	64,4%	63,9%	52,9%	56,1%
tanh				30,3%	29,5%	20,4%	24,6%	14,7%	16,1%	11,6%	13,5%	12,9%	14,3%	39,1%	40,8%
linear	smooth			65,9%	65,9%	65,9%	65,9%	65,9%	65,9%	65,9%	65,9%	65,9%	65,9%	65,9%	65,9%
sigmoid		65,9%	65,9%	27,6%	28,3%	13,0%	15,6%	11,7%	13,3%	21,8%	24,1%	31,8%	33,0%	41,4%	43,4%
smooth				62,5%	61,4%	62,3%	61,9%	63,0%	63,9%	63,7%	64,0%	65,3%	65,4%	62,9%	63,9%
tanh				46,0%	45,6%	54,6%	56,1%	52,5%	57,5%	64,2%	64,1%	65,9%	65,9%	65,9%	65,9%
linear	tanh			39,7%	40,9%	38,6%	39,4%	38,5%	39,9%	65,9%	65,9%	65,9%	65,9%	65,9%	65,9%
sigmoid		44,2%	45,1%	34,4%	36,5%	21,6%	22,4%	20,3%	22,6%	20,0%	24,9%	24,3%	27,4%	28,4%	29,6%
smooth				34,1%	35,9%	22,4%	25,1%	21,1%	23,4%	49,6%	49,8%	65,9%	65,9%	65,9%	65,9%
tanh				36,1%	37,2%	32,5%	35,0%	31,5%	34,5%	36,2%	37,0%	42,9%	45,0%	54,5%	55,6%

Tabuľka 2 - test architektúry siete - aktivačné funkcie

Z tabuľky jasne vidíme, že lineárna a *smoothed rectified linear* funkcie dosahujú podstatne horšie výsledky než sigmoid a hyperbolický tangens. Vzhľadom na ich slabú perspektívu a snahu o ušetrenie testovacieho času sme ich preto z ďalších testov vynechali. (Na skrytej vrstve, na výstupnej asi nič.)

Neobmedzeným funkciám, ako lineárnej a rektifikátorom, by mohli pomôcť malé váhy von.

**Počet vrstiev a skrytých neurónov:** Druhý test bol pomerne rozsiahly, a zameral sa na určenie vhodného počtu vrstiev a neurónov v nich. Skúšali sa jedno- až štvorvrstvové siete, počty skrytých neurónov sa volili z rozsahu {2,5,10,20,50}. Použité boli všetky kombinácie dvoch zostávajúcich uvažovaných aktivačných funkcií. Ostatné parametre ostali nezmenené.

Tabuľka 3 (na ďalšej strane) zobrazuje výsledky tohto testu. Nakoľko týchto údajov je príliš veľa, tak časti tabuľky s veľmi zlými hodnotami nahrádzame iba priemerom z týchto častí (vyznačené kurzívou). Uvádzané sú hodnoty priemernej validačnej chyby.

Prvé, čo je z výsledkov tohto testu jasné je fakt, že štyri vrstvy sú na danú úlohu nevhodné – dosahujú neprípustne veľké chyby. Podobne zle si viedli aj modely, ktoré mali hyperbolický tangens na výstupnej vrstve. Ďalším poznatkom je, že modely s iba dvoma neurónmi na prvej skrytej vrstve dosahovali taktiež pomerne zlú úspešnosť. Naopak dobre si viedli dve skupiny modelov: dvojvrstvové, s aktivačnými funkciami *sig-sig* alebo *tanh-sig*, a trojvrstvové s aktivačnými funkciami *sig-sig-sig* alebo *tanh-sig-sig*. Pre tieto modely sme sa preto rozhodli urobiť ďalší test. **Pre tanh() na výstupe treba kódovať kategórie ako {-1, +1}, nie {0, 1}.**

Tretí test z tejto sekcie pozostával zo spomínaných dvoch skupín modelov s pevne určenými možnosťami pre aktivačné funkcie. Počet skrytých neurónov sa vyberal z rozsahu {10,20,35,50} a počet epoch tréningu sa zvýšil na 250. Ostatné parametre ostali opäť nezmenené. Výsledky ukazuje Tabuľka 4.

Počet vrstiev	Počet neurónov		Aktivačná funkcia na 1. skrytej vrstve									
	1. skrytá vrstva	2. skrytá vrstva	Aktivačná funkcia na 2. skrytej vrstve									
			Aktivačná funkcia na výstupnej vrstve									
			sig	sig	tanh	tanh	∇	sig	sig	tanh	tanh	∇
			sig	tanh	sig	tanh	∇	sig	tanh	sig	tanh	∇
			sig	sig	sig	sig	sig	tanh	tanh	tanh	tanh	tanh
1	-	-	44,1%					45,0%				
2	2	-	32,7%	32,8%				35,1%	38,4%			
	5	-	21,4%	21,9%				26,5%	34,5%			
	10	-	13,1%	15,8%				22,3%	36,0%			
	20	-	11,0%	11,8%				24,8%	38,1%			
	50	-	13,9%	13,5%				28,5%	43,4%			
3	2	2	34,5%	32,5%	31,9%	33,0%		37,4%	37,5%	34,8%	39,4%	
	2	5	32,4%	32,6%	31,1%	32,9%		37,0%	36,1%	35,0%	38,5%	
	2	10	32,5%	32,1%	29,9%	31,6%		35,9%	36,4%	34,0%	37,0%	
	2	20	31,0%	30,8%	32,4%	33,6%		36,1%	38,5%	35,4%	38,5%	
	2	50	40,5%	32,0%	65,6%	44,4%		35,6%	51,6%	63,5%	64,3%	
	5	2	23,6%	21,3%	25,9%	30,0%		25,3%	33,8%	31,2%	36,1%	
	5	5	17,6%	20,4%	18,7%	22,4%		26,6%	29,3%	25,4%	32,3%	
	5	10	19,2%	19,4%	18,6%	21,0%		22,6%	24,1%	27,1%	32,3%	
	5	20	19,1%	19,6%	21,4%	23,1%		21,6%	26,5%	26,1%	39,6%	
	5	50	47,9%	22,3%	64,1%	54,5%		21,6%	47,5%	43,5%	66,0%	
	10	2	18,5%	20,1%	22,1%	26,5%		24,1%	30,0%	32,1%	34,3%	
	10	5	19,5%	17,0%	17,0%	18,4%		19,7%	28,6%	21,0%	27,9%	
	10	10	15,0%	17,4%	17,6%	17,3%		18,1%	25,0%	23,4%	32,4%	
	10	20	15,8%	15,1%	13,6%	20,1%		20,2%	26,9%	19,4%	45,2%	
	10	50	52,8%	34,3%	67,5%	59,3%		19,3%	35,2%	20,2%	65,8%	
	20	2	39,7%	24,9%	20,5%	26,6%		24,8%	30,5%	32,0%	32,9%	
	20	5	15,8%	19,8%	12,9%	14,0%		22,5%	26,8%	20,6%	29,1%	
	20	10	14,4%	15,4%	11,6%	15,8%		20,0%	25,4%	21,9%	29,0%	
	20	20	12,6%	17,8%	12,4%	36,1%		17,9%	24,9%	20,9%	43,1%	
	20	50	59,8%	43,6%	66,1%	67,9%		20,4%	38,3%	18,9%	65,5%	
	50	2	45,9%	44,5%	29,0%	33,7%		54,3%	44,5%	29,5%	47,3%	
	50	5	23,2%	36,6%	18,1%	27,9%		29,6%	43,4%	23,2%	51,5%	
	50	10	19,6%	20,8%	18,9%	20,6%		28,4%	36,3%	20,3%	53,9%	
	50	20	15,5%	22,5%	25,8%	56,0%		20,4%	25,6%	19,6%	47,6%	
	50	50	66,0%	56,8%	70,1%	65,6%		18,4%	50,8%	23,4%	64,8%	
4	∇	∇	47,6%	36,4%	42,7%	40,6%	44,4%	42,5%	43,3%	38,8%	50,9%	45,0%

Tabuľka 3 - architektúra siete - počet vrstiev a skrytých neurónov

Skrytých neurónov		Aktivačné funkcie			
		sig-sig		tanh-sig	
1.		EE	VE	EE	VE
10		8,4%	9,5%	17,5%	19,0%
20		6,0%	6,9%	15,6%	18,7%
35		5,7%	6,6%	19,0%	23,6%
50		6,2%	7,0%	22,8%	25,1%

Skrytých neurónov		Aktivačné funkcie			
		sig-sig-sig		tanh-sig-sig	
1.	2.	EE	VE	EE	VE
10	10	9,8%	10,5%	9,7%	13,1%
10	20	8,9%	10,5%	10,0%	12,4%
10	35	9,1%	10,9%	10,3%	11,8%
10	50	18,3%	21,5%	10,2%	12,3%
20	10	7,6%	10,6%	8,8%	12,5%
20	20	7,5%	10,0%	8,1%	10,4%
20	35	7,9%	10,3%	9,3%	13,4%
20	50	49,6%	49,5%	15,2%	18,3%
35	10	6,6%	8,1%	9,4%	11,0%
35	20	6,6%	8,6%	8,4%	12,3%
35	35	7,4%	10,4%	15,8%	17,3%
35	50	64,1%	64,3%	23,8%	25,4%
50	10	8,3%	11,0%	12,2%	13,3%
50	20	7,4%	12,0%	10,8%	12,8%
50	35	8,5%	12,5%	15,9%	18,9%
50	50	57,5%	58,9%	38,6%	41,1%

Tabuľka 4 - architektúra siete - posledné testy

Z posledných výsledkov vidíme víťaza pre ideálnu architektúru siete: dvojvrstvový MLP s 35 neurónmi na skrytej vrstve, a na oboch vrstvách použitý sigmoid ako aktivačná funkcia. S týmto modelom budeme teda pokračovať v ďalšom testovaní.

**Výber parametrov siete:** Efektivitu vybranej architektúry stále môžeme ovplyvniť piatimi dôležitými parametrami:  $\alpha$ ,  $\mu$ ,  $\epsilon$ ,  $\Delta\epsilon$  a počet epoch tréovania. Pri poslednom spomínanom platí, že čím viac epoch, tým lepší výsledok (až do konvergenencie), tu teda nie je dôvod na vykonávanie nejakých testov. Pri ostatných štyroch parametroch však situácia nie je natoľko jednoznačná.

**Toto platí ( $\pm$  oscilácie) iba pre estimačnú chybu. Validačná a testovacia sa môže aj zhoršovať.**

Prvý test tejto série bol opäť pomerne rozsiahly. Testovalo sa na zvolenej architektúre pri 250 epochách, s parametrami v nasledujúcich rozsahoch:

- $\alpha \in \{0.005, 0.01, 0.05, 0.1, 0.2\}$
- $\mu \in \{0, 0.01, 0.1, 0.2, 0.4, 0.6\}$
- $\epsilon \in \{0, 0.0001, 0.001, 0.01, 0.1, 0.2\}$
- $\Delta\epsilon \in \{1, 0.99, 0.98, 0.97, 0.96, 0.95, 0.94, 0.93, 0.92, 0.91\}$

Kľúčové slová sú Overfitting a Early stopping, skús si pozrieť e.g. wiki.

Pre kompaktnosť sú v Tabuľka 5 uvedené iba priemerné validačné chyby modelu a vybrané hodnoty parametrov. Pri  $\epsilon = 0$  nezáleží na hodnote  $\Delta\epsilon$ , preto je uvedený iba jeden riadok.

		$\alpha$																			
		$\mu$																			
		0,005				0,01				0,05				0,1				0,2			
$\epsilon$	$\Delta\epsilon$	0	0,2	0,4	0,6	0	0,2	0,4	0,6	0	0,2	0,4	0,6	0	0,2	0,4	0,6	0	0,2	0,4	0,6
0	✓	42,1%	40,6%	32,3%	28,6%	18,8%	15,2%	11,5%	9,7%	6,2%	6,3%	6,3%	7,7%	7,2%	7,6%	8,6%	9,6%	10,3%	12,4%	12,5%	15,6%
0,001	0,92	37,6%	38,5%	36,8%	34,8%	17,1%	11,9%	15,0%	8,6%	6,1%	6,6%	6,0%	5,9%	8,9%	7,9%	9,4%	10,0%	9,8%	12,1%	15,3%	15,4%
	0,94	43,5%	40,6%	32,1%	19,1%	23,0%	21,5%	9,9%	8,1%	7,3%	7,6%	6,1%	6,1%	7,4%	8,3%	10,6%	9,0%	8,8%	12,9%	12,9%	15,6%
	0,96	43,4%	41,5%	34,4%	32,5%	26,9%	14,0%	13,8%	7,4%	4,8%	7,1%	6,9%	5,9%	7,1%	7,4%	9,4%	7,9%	10,8%	11,9%	14,4%	16,1%
	0,98	43,3%	44,0%	37,5%	26,1%	24,1%	18,0%	13,9%	7,6%	5,9%	5,6%	7,4%	7,0%	7,4%	7,3%	7,9%	6,8%	10,5%	12,2%	13,3%	15,3%
0,1	1	59,0%	61,1%	57,4%	55,0%	57,4%	60,0%	57,1%	59,1%	54,9%	53,9%	52,8%	52,8%	50,3%	50,6%	49,1%	37,5%	31,8%	30,9%	29,8%	30,9%
	0,92	38,3%	37,8%	32,3%	30,9%	11,4%	16,3%	8,5%	7,8%	7,4%	6,4%	6,3%	6,3%	7,0%	9,4%	9,0%	8,9%	10,3%	13,4%	15,5%	13,6%
	0,94	38,8%	23,5%	28,3%	23,2%	22,5%	12,9%	12,6%	11,4%	6,4%	6,0%	6,1%	7,9%	7,1%	8,1%	8,6%	8,1%	9,6%	12,8%	14,1%	14,0%
	0,96	35,6%	30,0%	35,3%	20,1%	19,6%	13,4%	10,3%	7,9%	6,3%	6,5%	5,8%	5,8%	6,4%	6,8%	8,1%	8,9%	9,4%	10,8%	11,6%	19,0%
	0,98	32,5%	38,9%	29,3%	31,6%	16,9%	16,0%	9,1%	8,4%	6,3%	5,9%	7,6%	6,3%	7,5%	8,3%	8,6%	7,4%	9,8%	10,4%	12,4%	13,1%
0,2	1	69,0%	69,0%	69,0%	69,0%	69,0%	69,0%	69,0%	69,0%	69,0%	69,0%	69,0%	68,3%	69,0%	69,0%	69,0%	65,9%	67,1%	64,1%	64,9%	63,4%
	0,92	37,4%	34,0%	28,6%	25,4%	14,8%	11,5%	9,0%	6,8%	5,5%	5,9%	4,9%	8,1%	7,8%	7,6%	7,1%	8,3%	8,4%	11,1%	13,9%	15,0%
	0,94	39,2%	36,3%	31,3%	25,5%	20,3%	11,8%	9,0%	10,6%	5,9%	7,0%	5,9%	7,5%	7,3%	6,5%	7,9%	9,9%	12,4%	9,3%	13,1%	14,3%
	0,96	36,1%	27,4%	23,2%	26,0%	20,6%	11,1%	12,8%	10,9%	6,6%	6,1%	4,9%	6,4%	6,2%	8,0%	7,6%	8,5%	9,4%	12,1%	14,3%	14,8%
	0,98	39,6%	34,0%	31,6%	27,9%	18,1%	17,8%	11,5%	7,5%	4,6%	6,4%	7,5%	7,8%	6,4%	7,3%	6,1%	8,9%	9,6%	10,1%	14,1%	13,1%
	1	69,0%	69,0%	69,0%	69,0%	69,0%	69,0%	69,0%	69,0%	69,0%	69,0%	69,0%	69,0%	69,0%	69,0%	69,0%	69,0%	63,4%	67,3%	68,6%	68,6%

Tabuľka 5 - parametre modelu

Z výsledkov sú jasne viditeľné dva fakty: ideálne nastavenie  $\alpha$  je okolo 0,05, a že modely s  $\Delta\epsilon = 1$  dosahujú veľmi zlé výsledky. To potvrdzuje, že zavedenie tohto utlmujúceho faktora pre parameter  $\epsilon$  zlepšilo efektivitu siete. Viac informácií z týchto výsledkov však už nevidíme, preto bolo vykonané ešte jedno testovanie.

Pri poslednom testovaní tejto série sme ponechali iba úspešné hodnoty jednotlivých parametrov, a navyše sme v prípade  $\alpha$  pridali ďalšie vhodné voľby. Intervaly parametrov boli:

- $\alpha \in \{0.03, 0.04, 0.05, 0.07\}$
- $\mu \in \{0, 0.01, 0.1\}$
- $\epsilon \in \{0.01, 0.1, 0.2\}$
- $\Delta\epsilon \in \{0.98, 0.97, 0.96, 0.95, 0.94, 0.93\}$



		$\alpha$											
		$\mu$											
		0,03			0,04			0,05			0,07		
$\epsilon$	$\Delta\epsilon$	0	0,01	0,1	0	0,01	0,1	0	0,01	0,1	0	0,01	0,1
0,01	0,93	8,00%	5,88%	7,00%	6,00%	4,50%	5,13%	6,88%	7,38%	6,75%	8,00%	6,50%	6,25%
	0,94	6,75%	6,00%	6,88%	5,38%	6,88%	5,13%	6,50%	6,13%	5,88%	5,75%	6,13%	6,88%
	0,95	5,00%	8,00%	6,00%	5,25%	5,38%	6,75%	5,25%	5,50%	6,87%	7,38%	7,63%	6,38%
	0,96	6,13%	7,50%	8,00%	6,75%	6,38%	6,25%	4,25%	5,38%	5,38%	7,00%	6,25%	8,13%
	0,97	6,13%	5,25%	5,88%	5,13%	6,38%	5,88%	5,38%	4,63%	5,00%	6,88%	5,63%	6,75%
	0,98	5,38%	5,25%	5,00%	4,13%	4,88%	4,50%	5,63%	6,50%	6,13%	6,50%	5,00%	4,87%
0,1	0,93	6,25%	4,75%	6,50%	6,13%	5,75%	5,75%	5,88%	6,25%	6,50%	5,13%	6,75%	6,25%
	0,94	5,88%	5,13%	6,00%	6,63%	4,88%	5,25%	6,00%	5,50%	6,50%	6,13%	5,88%	7,00%
	0,95	7,38%	6,00%	6,75%	5,50%	6,25%	5,50%	6,25%	4,38%	5,88%	5,63%	7,25%	6,62%
	0,96	7,00%	5,75%	5,75%	4,25%	6,50%	6,00%	6,50%	5,75%	5,75%	5,75%	6,13%	6,63%
	0,97	6,00%	6,00%	6,25%	5,88%	5,13%	4,75%	5,13%	6,13%	6,50%	6,13%	6,75%	6,38%
	0,98	5,50%	5,88%	6,25%	6,38%	7,25%	5,38%	5,50%	5,00%	7,00%	5,13%	6,38%	4,75%
0,2	∅	6,23%	6,35%	6,00%	5,71%	5,60%	6,35%	6,31%	5,52%	5,67%	6,65%	6,42%	6,46%

Tabuľka 6 - parametre modelu - posledné testy

Výsledky pre  $\epsilon = 0,2$  boli všeobecne horšie ako ostatné výsledky v tabuľke, preto je opäť uvádzaný iba ich priemer. V tabuľke vidíme niekoľko modelov, ktoré sú úspešnejšie od ostatných. Najúspešnejšie nastavenie parametrov je  $\alpha = 0,04$ ,  $\mu = 0$  (t.j. bez použitia momenta),  $\epsilon = 0,01$ ,  $\Delta\epsilon = 0,98$  s priemernou validačnou chybou iba 4,125% (v tabuľke zaokrúhlené). Tieto parametre teda použijeme v našom výslednom MLP.

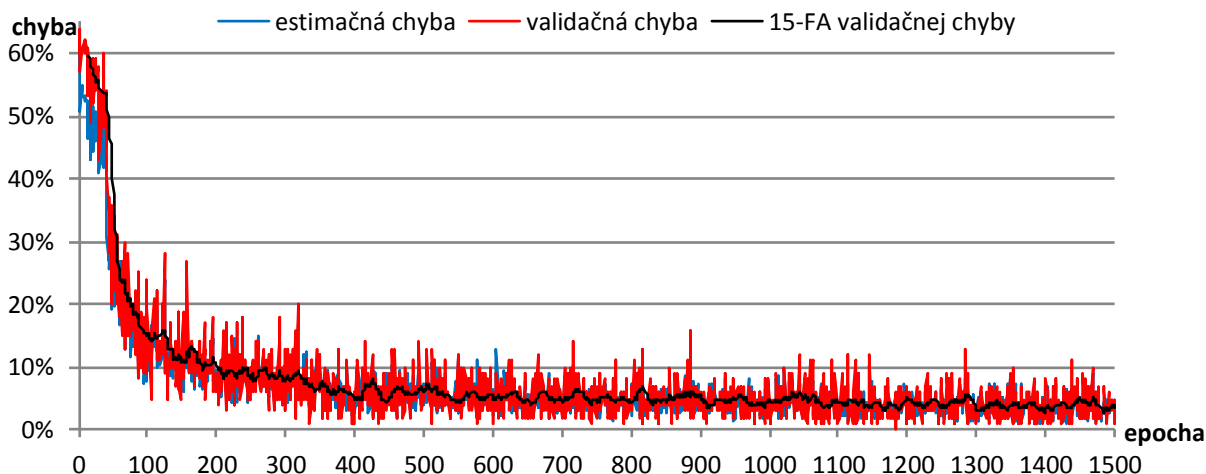
## VÝSLEDKY VYBRANÉHO MODELU

Po rozsiahлом testovaní optimálnej architektúry a parametrov modelu sme dospeli k nasledovnému nastaveniu:

- dvojvrstvová sieť, 35 neurónov na skrytej vrstve
- sigmoid ako aktivačná funkcia na oboch vrstvách
- $\alpha = 0,04$ ,  $\mu = 0$ ,  $\epsilon = 0,01$ ,  $\Delta\epsilon = 0,98$

Takto nastavenému modelu sme podložili trénovacie dáta, pričom sme pozorovali jeho správanie počas učenia. Následne sme ho otestovali na testovacích dátach a vyhodnotili jeho úspešnosť.

### Trénovanie:



Graf 1 - vývoj estimačnej a validačnej chyby počas trénovania

Počas tréovania vybraného modelu trvajúceho trvalo 1500 epoch sme si všímali vývoj estimačnej a validačnej chyby. Výsledky pre najlepšiu inštanciu (s najmenšou záverečnou validačnou chybou) zobrazuje Graf 1. Čiernou čiarou je na ňom zobrazený plávajúci priemer validačnej chyby s oknom veľkosti 15. Na grafe vidíme očakávané správanie: chyba spočiatku prudko klesá zhruba prvých 100 epoch, následne sa postupne ustáľuje ďalších cca 300-400 epoch, a od určitého bodu skomverguje k finálnej hodnote, okolo ktorej osciluje až do konca učenia.

**Testovanie:** Všetkých osem inštancií modelu, ktoré vznikli využitím *k-fold cross validation* metódy sme otestovali na testovacích dátach. Vyhodnocovali sme percento zle klasifikovaných príkladov.

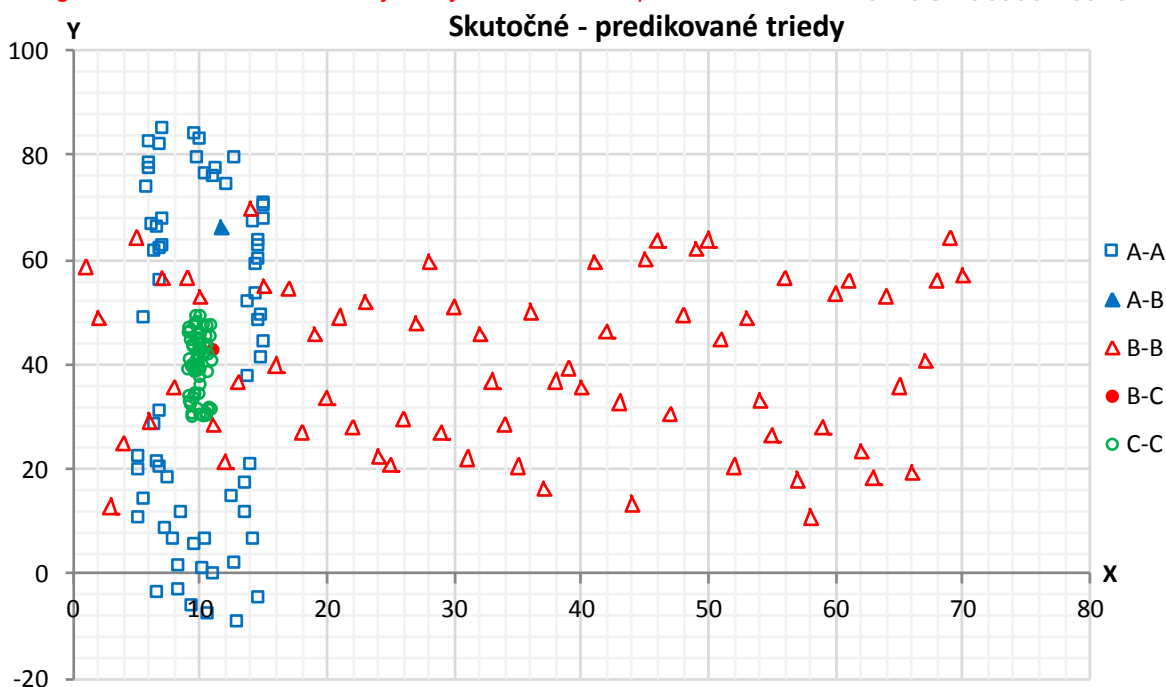
Inštancia	1	2	3	4	5	6	7	8	Priemer	Št. odchýlka
Chyba	1,5%	2,5%	1,0%	1,5%	2,0%	2,0%	1,5%	1,0%	1,63%	0,52%

Tabuľka 7 - klasifikačné chyby inštancií modelu

V tabuľke vidíme, že klasifikačná chyba všetkých inštancií modelu je pomerne stabilná, a pohybuje sa okolo hodnoty 1,5%. Zvýraznená tretia inštancia je vybraná najlepšia inštancia modelu, teda tá, ktorá mala najnižšiu validačnú chybu. Pri použití v praxi by sme teda do nasadenia vybrali práve túto inštanciu. Očakávaným pozitívnym výsledkom je aj fakt, že táto vybraná inštancia mala tiež najnižšiu testovaciu chybu.

Na grafe 2 si môžeme pozrieť graficky znázornené výsledky klasifikácie nášho najlepšieho modelu. Farby v grafe predstavujú skutočné hodnoty, tvary bodov zas predpovedané hodnoty.

S týmto grafom sa čosi stalo, mal by to byť obdĺžnik v elipse v obdĺžniku. Možno si zabudol "odnormalizovať"?



Graf 2 - porovnanie skutočných a predpovedaných hodnôt

	A		B		C	
A	66	98,5%	1	1,5%	0	0,0%
B	0	0,0%	70	98,6%	1	1,4%
C	0	0,0%	0	0,0%	62	100,0%

Tabuľka 8 - confusion matrix



## **ZÁVER**

Počas tohto projektu sme ukázali, že viacvrstvový perceptrón je vhodný na úlohu klasifikácie dát do niekoľkých tried. Bolo vytvorené a implementované prostredie pre testovanie a porovnávanie rôznych modelov MLP, a postupným vyhodnocovaním testov sme dospeli k voľbe najlepšieho modelu pre našu úlohu. Počas tohto procesu sme tiež vyskúšali niekoľko viac aj menej štandardných metód pre zlepšenie výsledkov siete a otestovali ich účinnosť. Výsledkom bola neurónová sieť s čo možno najlepšou schopnosťou generalizácie. Tú sme nakoniec otestovali na nových dátach, pričom sme získali veľmi uspokojivý výsledok 1% chyby pri použití najlepšieho modelu.