FAKULTA
APLIKOVANÝCH VĚD
ZÁPADOČESKÉ
UNIVERZITY
V PLZNI

Semestrální práce

KKY/NSES

Filip Majer

Neuronové sítě - Zadání semestrální práce

1. Natrénujte

   a) jednovrstvou neuronovou síť
   b) dvouvrstvou neuronovou síť

   pro klasifikaci předmětů popsaných dvěma příznaky do 5 tříd. Trénovací množiny jsou v souborech tren_data1.txt a tren_data2.txt. Každou síť natrénujte pro každou trénovací množinu zvlášť.

   Příklad výpisu trénovací množiny:

6.9161	-5.7103	1
7.0601	-5.6065	1
7.2404	-5.3848	2
-7.0375	-5.4146	3
…	…	…

   Každá řádka odpovídá jedné trénovací dvojici. Na každém řádku jsou 3 položky, které jsou od sebe odděleny mezerou. První dvě položky reprezentují vstup sítě, třetí položka je informace o zařazení klasifikovaného předmětu do konkrétní třídy.

2. Při trénování zobrazujte:
   - průběh chyby v závislosti na trénovacích cyklech
   - polohu přímek tvořených jednotlivými neurony ve vstupní rovině

   Tento bod je volitelný

3. Funkční program předveďte vyučujícímu.

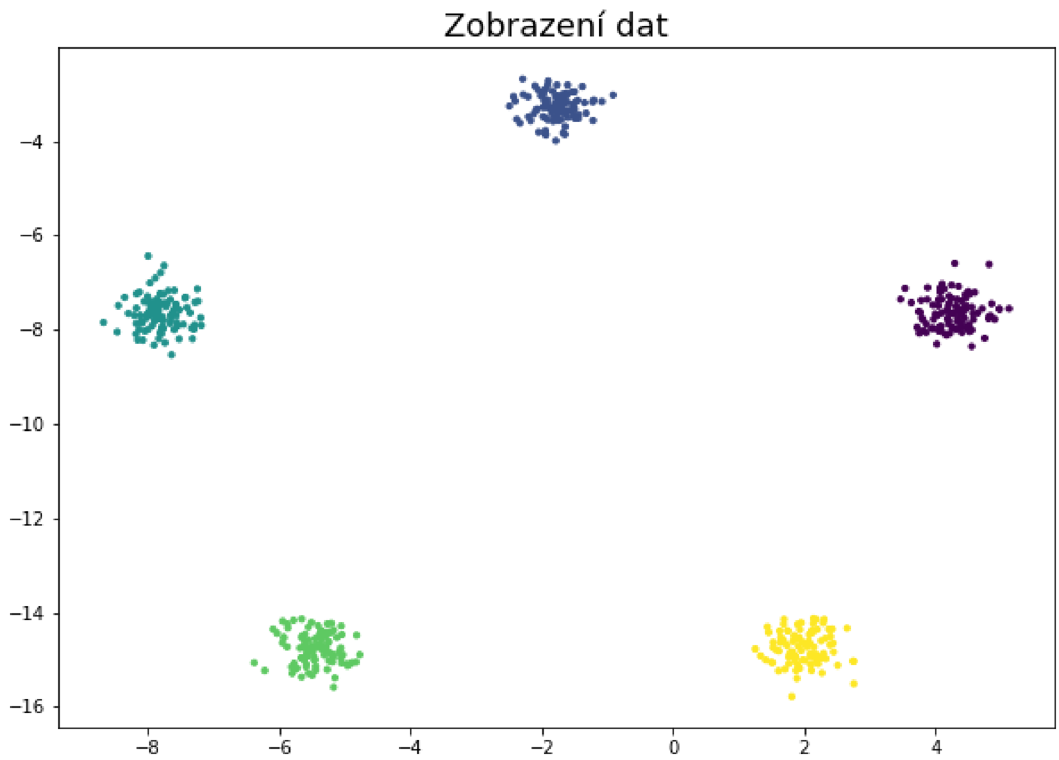
4. Vypracujte referát, který bude obsahovat:
   - toto zadání,
   - jednoznačný popis (např. grafické znázornění) použité sítě,
   - počáteční hodnoty parametrů sítě a hodnoty volitelných konstant algoritmu trénování,
   - hodnoty parametrů sítě po jejím natrénování a počet trénovacích cyklů, po kterých byla síť natrénována,
   - grafické znázornění průběhu chyby v závislosti na trénovacích cyklech,
   - znázornění oblastí ve vstupním prostoru, jak výsledná síť klasifikuje vstupní body,
   - zhodnocení dosažených výsledků.

21.10.2022, Vlasta Radová

1 Zobrazení trénovací množiny

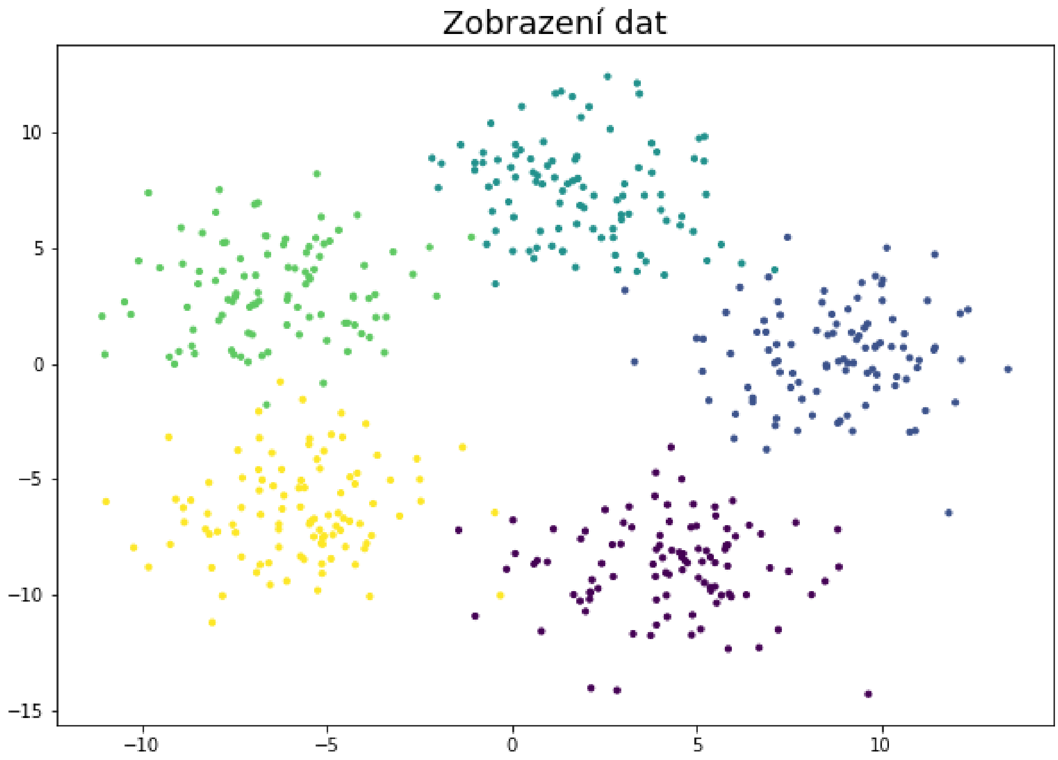
Před samotnou tvorbou modelů neuronových sítí bude vhodné podívat se lépe na trénovací množiny, se kterými budeme pracovat. Jedná se o dvoudimenzionální data, které lze vidět na následujících dvou obrázcích. V obou případech jsou data rozdělena do 5 tříd.

V první trénovací množině jsou jednotlivé třídy lineárně separabilní. Taková data tedy dokáže jak jednovrstvá, tak dvouvrstvá síť od sebe oddělit, v obou případech s nulovou chybou.



Obrázek 1: První trénovací množina

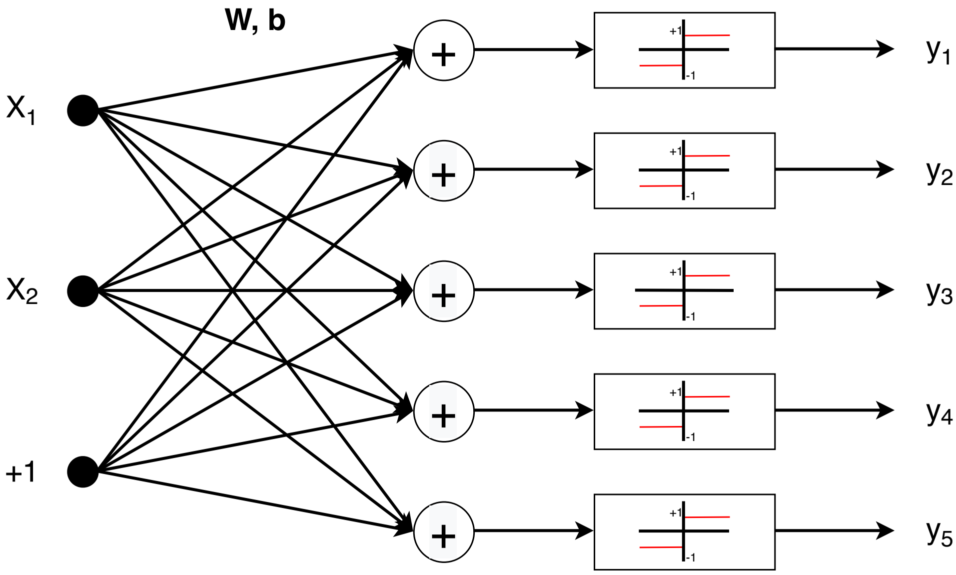
V druhé trénovací množině jsou jednotlivé třídy promíchané, nejsou lineárně separabilní. Taková data již jednovrstvá síť s nulovou chybou oddělit nedokáže. S nulovou chybou to nedokáže ani dvouvrstvá síť, protože neumí oddělit nekonvexní oblasti, které v datech vznikají.



Obrázek 2: Druhá trénovací množina

2 Jednovrstvá neuronová síť

Pro naše trénovací množiny bude mít jednovrstvá síť 2 neurony na vstupu (odpovídá dimenzi vstupních vektorů) a 5 neuronů na výstupu (odpovídá požadovanému počtu tříd). Jako aktivační funkci použijeme bipolární binární aktivační funkci. Síť je zobrazena na následujícím obrázku, **W** značí váhovou matici, **b** značí prahový vektor.



Obrázek 3: Model jednovrstvé neuronové sítě

Příslušnost bodů k jednotlivým třídám převedeme z čísel 1,2,3,4,5 z dodaného souboru na vektory o 5 prvcích, kde 1 bude vždy na pozici, která odpovídá dané třídě. Na ostatních pozicích bude -1. Bod, který by měl náležet první třídě, bude mít tedy vektor $[1, -1, -1, -1, -1]$. Takový vektor budeme požadovat na výstupu neuronové sítě po natrénování.

2.1 Trénování na první trénovací množině

Pro trénování sítě byla vygenerovaná náhodná váhová matice a práhový vektor samých nul odpovídající dimenze. Váhová matice nulová být nesmí, jinak by se síť nenatrénovala. Prahový vektor nulový být může, ale mohla by se zvolit i jiná libovolná čísla.

$$W = \begin{bmatrix} -0.00405043 & 0.01262627 \\ 0.01618187 & -0.00684171 \\ -0.01794442 & 0.01845338 \\ -0.00593382 & 0.00702208 \\ -0.00218642 & -0.00091378 \end{bmatrix}$$

$$b = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Konstanta učení c byla nastavena na $c = 0.01$, maximální přípustná chyba E_{max} na 0 a maximální počet iterací (epoch) na 1000. Při trénování byla použita metoda dávkového učení.

Síť byla úspěšně natrénována po 701 iteracích. Konečné hodnoty váhové matice a prahového vektoru jsou následující:

$$W = \begin{bmatrix} 0.11643037 & 0.05216694 \\ -0.00480868 & 0.01702954 \\ -0.06785205 & 0.0569537 \\ -0.01931541 & -0.0168978 \\ 0.07522727 & -0.04249802 \end{bmatrix}$$

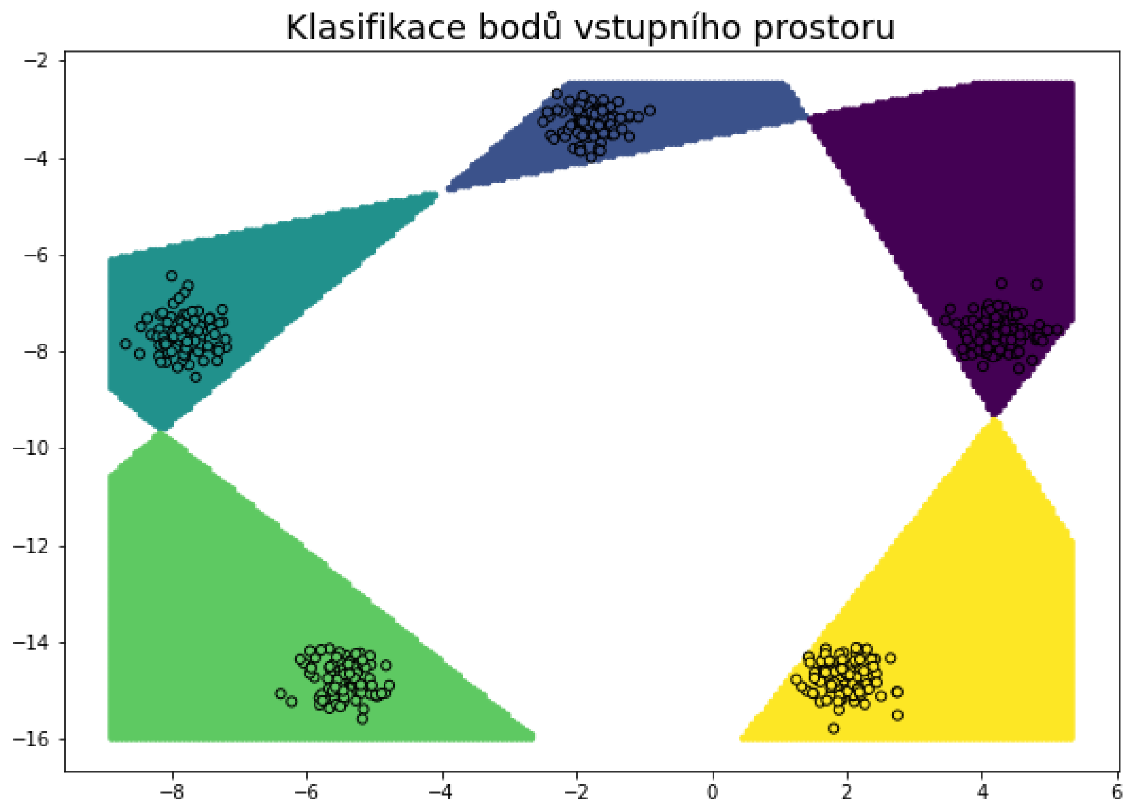
$$b = \begin{bmatrix} 0. \\ 0.06084 \\ -0.00272 \\ -0.32044 \\ -0.71336 \end{bmatrix}$$

Síť dosáhla požadové maximální přípustné chyby. Vývoj této chyby lze vidět na následujícím obrázku, který zachycuje každou desátou chybu, čímž je průběh křivky hladší. Kdybychom vykreslovali každou spočtenou chybu, graf by byl "zašuměný". Chyba totiž neklesá v každém kroku.

Z grafu je také dobře vidět, že neuronová síť se relativně rychle naučí jednotlivé třídy dobře oddělit, ale pak se na dlouhou dobu zastaví kolem hodnoty 250. To je pravděpodobně dáno rozložením tříd, které jsou sice lineárně separabilní, ale neuronová síť se musí docela přesně trefit, aby dané shluky oddělila opravdu s nulovou chybou.

Obrázek 4: Průběh chyby z závislosti na počtu iterací

Dle zadání byl ještě vygenerován rastr bodů, který byl neuronovou sítí oklasifikován. Na obrázku je dobře vidět veliká část, kterou nedokázala neuronová síť oklasifikovat.



Obrázek 5: Oklasifikovaný rastr bodů

2.2 Trénování na druhé trénovací množině

Pro trénování na druhé trénovací množině byla opět vygenerována jiná náhodná váhová matice a prahový vektor samých nul.

$$W = \begin{bmatrix} 0.00660079 & -0.00577473 \\ 0.00945252 & -0.01149609 \\ -0.01230288 & -0.01670482 \\ 0.00675001 & 0.0087402 \\ -0.00873515 & -0.01830313 \end{bmatrix}$$

$$b = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

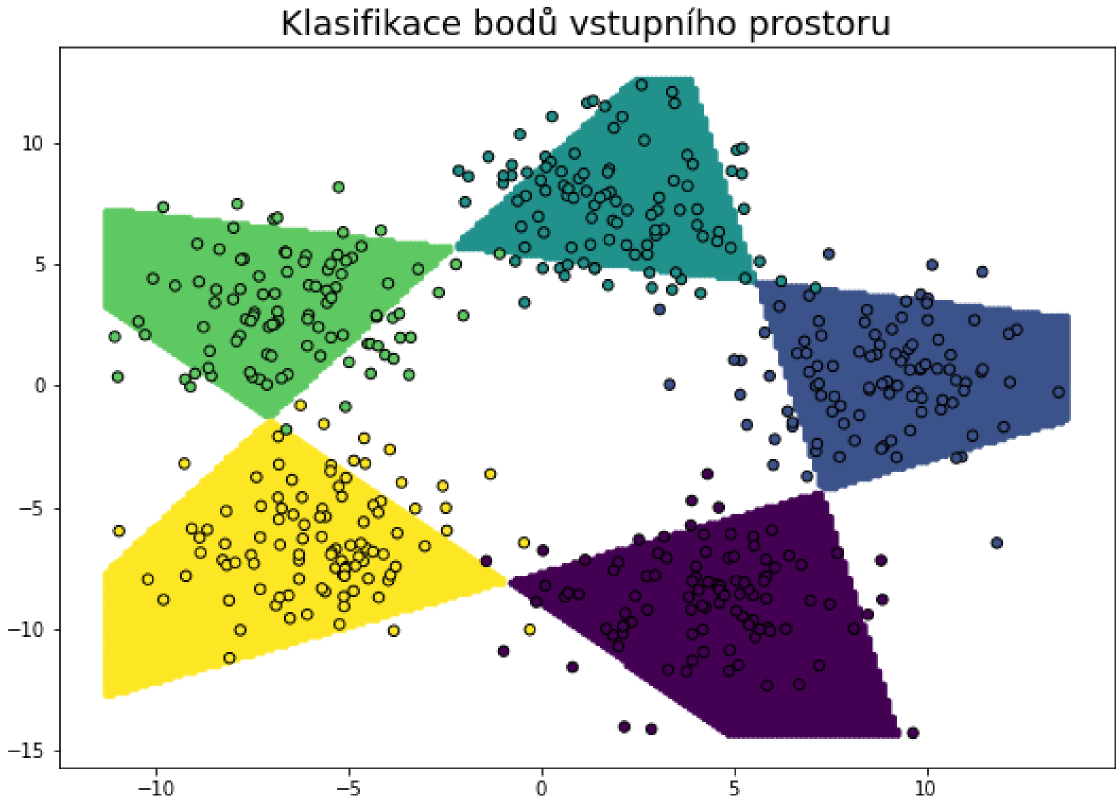
Konstanta učení byla opět nastavena na $c = 0.01$, maximální přípustná chyba E_{max} na 240 a maximální počet iterací (epoch) na 200. Při trénování byla opět použita metoda dávkového učení. Hodnota E_{max} je nastavena na danou hodnotu, protože naše jednovrstvá síť nikdy nedokáže lepší výsledek než přibližně 230-240. Shluky totiž nejsou lineárně separabilní (viz. Obrázek 7). Přesnost neuronové sítě (tedy poměr správně oklasifikovaných bodů z trénovací množiny vůči všem bodům trénovací množiny) je 75%.

Síť byla úspěšně natrénována po 58 iteracích a dosáhla chyby 234. Konečné hodnoty váhové matice a prahové vektoru jsou následující:

$$W = \begin{bmatrix} 0.00885156 & -0.01937964 \\ 0.02268954 & 0.00450539 \\ 0.0037901 & 0.02128033 \\ -0.01884557 & 0.01273133 \\ -0.01579815 & -0.01447229 \end{bmatrix}$$

$$b = \begin{bmatrix} -0.14892 \\ -0.146 \\ -0.11272 \\ -0.11564 \\ -0.1312 \end{bmatrix}$$

Obrázek 6: Průběh chyby z závislosti na počtu iterací



Obrázek 7: Oklasifikovaný rastr bodů

3 Dvouvrstvá neuronová síť

Navržená dvouvrstvá neuronová síť bude mít 2 neurony ve vstupní vrstvě, 20 neuronů ve skryté vrstvě a 5 neuronů ve výstupní vrstvě. Všechny neurony budou používat bipolární spojitou aktivační funkci parametrizovanou jejím sklonem λ.

Počet neuronů ve skryté vrstvě není náhodně vybrané číslo. Provedl jsem řadu experimentů a 20 neuronů ve skryté vrstvě vycházelo nakonec nejlépe. Pro 5 neuronů se mi nedařilo síť natrénovat. Ani po 10000 iteracích se chyba nedostala pod 300. Pro síť bylo asi těžké naučit se přímo mapovat skrytou vrstvu na vrstvu výstupní. 10 neuronů vycházelo, co se týkalo výsledků, stejně dobře jako 20 neuronů, akorát trénování trvalo přibližně o třetinu déle. 40 neuronů se mi podařilo natrénovat ještě rychleji než síť s 20 neurony, ale už se mi taková síť zdála neadekvátně složitá. Více neuronů než 60 už mělo problémy se vůbec natrénovat.

3.1 Trénování na první trénovací množině

Pro trénování byly vygenerovány náhodné váhové matice **W1** a **W2** (prahové vektory **b1** a **b2** byly opět nastaveny na vektory nul odpovídajících dimenzí). Pro účely zachování úhledné dokumentace zde mám na ukázku jen váhovou matici **W1**.

$$W1 = \begin{bmatrix} -0.00062684 & 0.00204371 \\ 0.00967507 & -0.05454992 \\ 0.00942817 & 0.11996725 \\ 0.04115558 & 0.04298223 \\ -0.03419058 & 0.02848809 \\ 0.03377992 & 0.11175493 \\ -0.02983139 & 0.11779228 \\ -0.20330673 & 0.00208681 \\ -0.11477921 & 0.08866643 \\ 0.05870032 & 0.27941992 \\ 0.15447049 & 0.19123232 \\ 0.01156107 & 0.10593097 \\ -0.01918093 & -0.0806266 \\ 0.13807789 & 0.07848413 \\ 0.04901136 & 0.0287882 \\ -0.09590006 & -0.12602575 \\ -0.15587565 & 0.13910919 \\ 0.04551581 & 0.07014361 \\ 0.11942625 & 0.18156623 \\ -0.13179549 & 0.10420798 \end{bmatrix}$$

Konstanta učení byla zvolena $c = 1$, maximální přípustná chyba $E_{max} = 1$ a maximální počet iterací byl nastaven na 1000. Maximální chyba by šla nastavit i na nižší hodnotu, například na 0.1. Na výsledky trénování sítě to ovšem nemá žádný vliv. Jelikož je použitá bipolární spojitá funkce, síť by se nikdy nenaučila na výstupu dávat přesně 1 nebo -1. Výstupy sítě zaokrouhluji, abych je mohl porovnávat s požadovanými výstupy.

Síť byla natrenována po 551 iteracích a dosáhla chyby 0.99. Konečné hodnoty váhové matice **W1** a prahového vektoru **b1** jsou následující (matici **W2** s vektorem **b2** opět neuvadím):

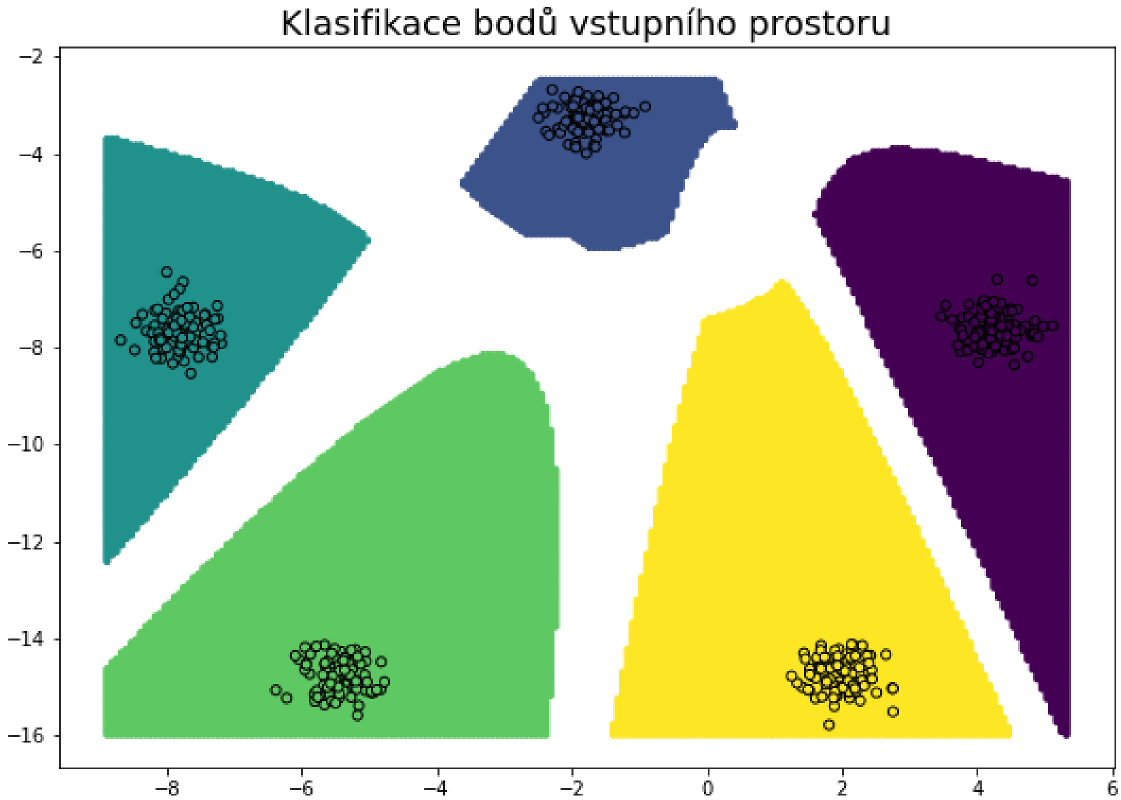
$$W1 = \begin{bmatrix} 1.43639017 & 0.46436347 \\ -0.23254065 & -0.28772915 \\ -0.76582398 & 0.50475989 \\ 1.74230999 & 0.55720861 \\ -0.03989565 & 0.65280907 \\ 0.78146683 & -0.0285049 \\ 1.11449941 & -0.73383926 \\ -1.83663 & 0.15728289 \\ -1.75374344 & -0.54026991 \\ 0.41714128 & 1.11912058 \\ 1.54367731 & -1.09854534 \\ -0.96652841 & 0.63917072 \\ 0.13042578 & -1.08549218 \\ 0.07649038 & -0.47352349 \\ -0.02922815 & 0.57947342 \\ -0.16379723 & -0.26538972 \\ -1.58958018 & -0.47127366 \\ -0.44457695 & -0.96873538 \\ -0.01371434 & 0.38162152 \\ -1.36864661 & 0.76216398 \end{bmatrix}$$

$$b1 = \begin{bmatrix} 0.20867237 \\ -2.10549435 \\ 0.6727876 \\ 0.31994795 \\ -0.04159986 \\ 1.40692762 \\ -0.09781511 \\ 0.23402081 \\ -0.01206008 \\ -0.25875141 \\ 0.9615287 \\ 0.03249124 \\ 0.15667218 \\ -0.02188049 \\ 0.01865041 \\ -1.50018101 \\ -0.10591984 \\ -0.47334102 \\ 0.56440072 \\ 1.02338089 \end{bmatrix}$$

Z vývoje chyby na následujícím obrázku lze dobře vidět, že síť se nejvíce natrénuje v prvních 100 iteracích a pak už se jen velmi pozvolna zdokonaluje, i když tohle zdokonalování již nemá na výsledné natrénované hodnoty téměř žádný vliv.

Obrázek 8: Průběh chyby z závislosti na počtu iterací

Oklasifikovaný rastr se značně liší od rastru jednovrstvé sítě. Nevznikají tak velké neoklasifikované plochy. Na obrázku jsou stále vidět některé přímky, které jsou tvořeny jednotlivými neurony.



Obrázek 9: Oklasifikovaný rastr bodů

3.2 Trénování na druhé trénovací množině

Opět byly vygenerovány váhové matice a prahové vektory. Uvádím matici **W1** pro představu:

$$W1 = \begin{bmatrix} 0.0155882 & -0.11474158 \\ 0.04811727 & 0.0498877 \\ -0.05195941 & 0.1685059 \\ 0.04203419 & 0.18757649 \\ 0.06636633 & -0.15040948 \\ -0.09491413 & -0.04939571 \\ -0.07431092 & -0.13336884 \\ -0.09244228 & -0.06671646 \\ -0.07428366 & 0.06629696 \\ -0.13073403 & 0.10655846 \\ -0.03616412 & 0.02072864 \\ -0.05796914 & -0.00879459 \\ 0.02238074 & -0.19590328 \\ -0.0965202 & 0.07165282 \\ 0.0057184 & -0.02316133 \\ 0.07893662 & 0.0303206 \\ 0.02734672 & 0.0795843 \\ -0.22533092 & 0.05108415 \\ 0.03879463 & 0.04195925 \\ 0.00620358 & 0.02074824 \end{bmatrix}$$

Konstanta učení je stále 1 a maximální počet iterací je 1000. Oproti první trénovací množině se zde změnilo nastavení E_{max}, které je nyní 25. Opět se jedná o hodnotu nastavenou na základě delšího trénování a pozorování výsledků, kdy nemá smysl snažit se o nižší hodnotu chyby. V trénovací množině je totiž hodně tzv. outlierů, které pokud by chtěla neuronová síť správně klasifikovat, klasifikovala by zase jiné body špatně.

Síť byla natrénována po 695 iteracích a dosáhla chyby 24.99. Síť dosáhla úspěšnosti klasifikace trénovací množiny 97%. Konečné hodnoty váhové matice **W1** a prahového vektoru **b1** jsou následující (matici **W2** s vektorem **b2** opět neuvadím):

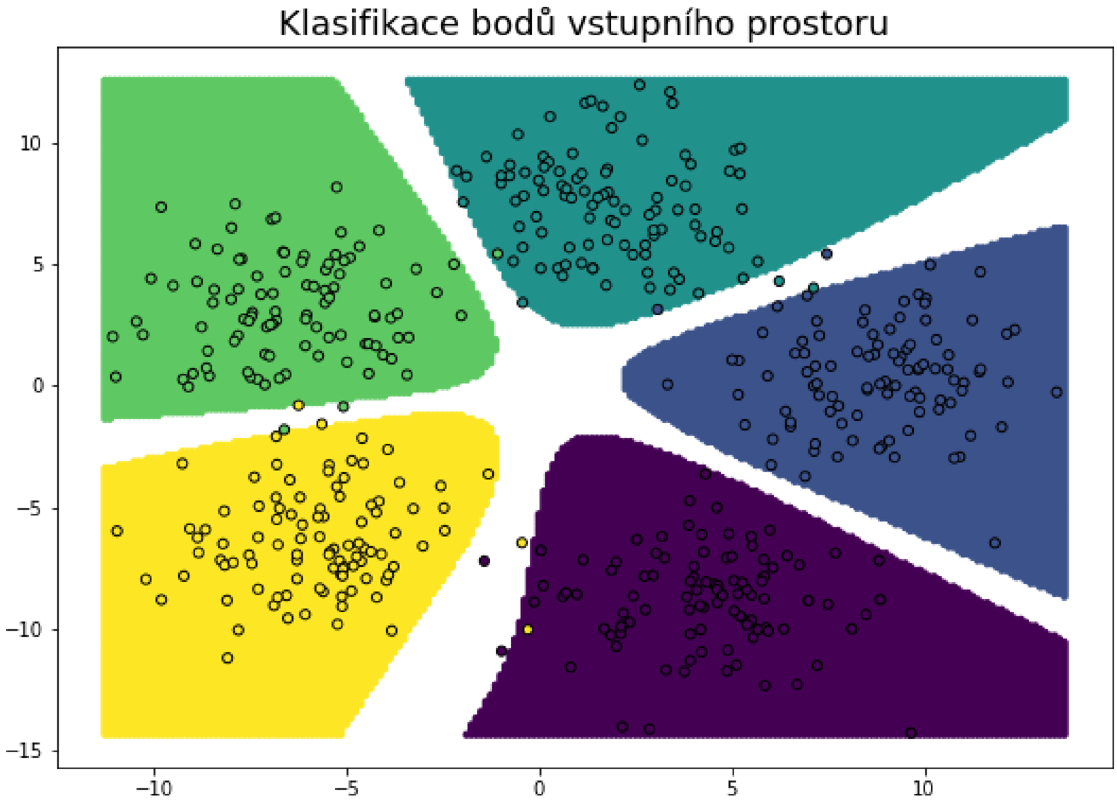
$$W1 = \begin{bmatrix} -0.00397169 & 0.00788239 \\ 0.04311366 & -0.00791982 \\ -0.43107668 & 1.23726512 \\ 0.8518371 & 1.04402946 \\ -0.26098701 & -0.67523044 \\ -1.41546401 & -0.46285451 \\ -0.9767496 & -0.52243236 \\ -0.85093659 & 0.26192573 \\ -0.80426558 & -0.929099 \\ -1.387276 & 0.55591152 \\ 0.42818369 & -0.21085377 \\ -0.03914871 & -1.10374684 \\ 0.25955841 & -0.64277911 \\ -1.72018348 & -0.27811803 \\ 0.43585741 & -0.30978221 \\ 0.01890378 & -0.06422221 \\ 0.14011381 & 0.87926783 \\ -0.72503426 & 0.91082322 \\ 0.80681688 & 1.20564945 \\ 0.56209537 & -1.36478543 \end{bmatrix}$$

$$b1 = \begin{bmatrix} -0.00397169 & 0.00788239 \\ 0.04311366 & -0.00791982 \\ -0.43107668 & 1.23726512 \\ 0.8518371 & 1.04402946 \\ -0.26098701 & -0.67523044 \\ -1.41546401 & -0.46285451 \\ -0.9767496 & -0.52243236 \\ -0.85093659 & 0.26192573 \\ -0.80426558 & -0.929099 \\ -1.387276 & 0.55591152 \\ 0.42818369 & -0.21085377 \\ -0.03914871 & -1.10374684 \\ 0.25955841 & -0.64277911 \\ -1.72018348 & -0.27811803 \\ 0.43585741 & -0.30978221 \\ 0.01890378 & -0.06422221 \\ 0.14011381 & 0.87926783 \\ -0.72503426 & 0.91082322 \\ 0.80681688 & 1.20564945 \\ 0.56209537 & -1.36478543 \end{bmatrix}$$

Vývoj chyby se značně podobá průběhu u první trénovací množiny, kdy největší změna proběhla tentokráte v prvních 50 iteracích.

Obrázek 10: Průběh chyby z závislosti na počtu iterací

Oklasifikovaný rastr se opět značně liší od rastru jednovrstvé sítě. Dvouvrstvá síť si daleko lépe poradila s lineárně neseparabilními daty a vzniklo jen málo oblastí, kde nebyla síť schopna body oklasifikovat.



Obrázek 11: Oklasifikovaný rastr bodů

4 Závěr

Semestrální práce nám přišla adekvátní rozsahu a zaměření předmětu KKY/SM. S jejím vypracováním jsme měli pouze minoritní problémy v terminologii, které jsme museli diskutovat s lektorem na cvičení. Potěšilo nás množství souvislostí s předmětem KKY/LS1, většina témat v semestrální práci nám díky absolvování látky lineárních systému byla již známá. Zaměření práce na mechanickou soustavu nám přišlo opodstatněné a názorné.