

Použitie genetického algoritmu na adaptáciu neurónovej siete

Jozef Kriška

Záverečná práca z predmetu Evolučné algoritmy, Jún 2006

Fakulta informatiky a informačných technológií
Ilkovičova 3, 842 16 Bratislava 4
E-mail: jozef.kriska@pobox.sk

Abstrakt. V súčasnosti predstavujú neurónové siete jeden z dôležitých prístupov subsymbolického spracovania informácií. Jednou z veľmi významných vlastností neurónovej siete je, že svojím spôsobom predstavuje tzv. univerzálny aproximátor funkcií a preto sa používa vo viacerých problémových oblastiach, ako napr. aproximácia funkcií, klasifikácia do tried, riešenie predikčných problémov atď. Hlavným problémom v oblasti neurónových sietí je adaptácia váh spojení jednotlivých neurónov. Pre riešenie tohto problému existujú rôzne optimalizačné metódy. Jednu kategóriu týchto metód tvoria stochastické optimalizačné algoritmy s evolučnými črtami, medzi ktoré patria aj genetické algoritmy. Cieľom tejto práce je použitie genetického algoritmu na adaptáciu váh neurónovej siete, ktorá má interpretovať danú boolovskú funkciu.

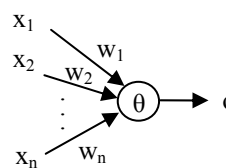
Kľúčové slová: dopredná neurónová sieť, genetický algoritmus, optimalizácia váh

1 Neurónová sieť

Neurónová sieť je masívne paralelný procesor, ktorý má sklon k uchovávaniu experimentálnych znalostí a ich ďalšieho využívania [3]. Napodobňuje ľudský mozog v dvoch aspektoch:

- Poznatky sú zbierané v neurónovej sieti počas učenia
- Medzineurónové spojenia (synaptické váhy) sú využívané na ukladanie znalostí

Vo všeobecnosti je neurónová sieť zložená z formálnych neurónov, ktoré sú vzájomne prepojené tak, že výstup jedného neurónu je vstupom ďalších neurónov. Stav jednotlivých neurónov určuje celkový stav neurónovej siete a synaptické váhy jednotlivých prepojení medzi neurónmi určujú konfiguráciu neurónovej siete. Počet neurónov a ich vzájomné prepojenie v sieti definuje architektúra danej siete. Z hľadiska významu, resp. využitia jednotlivých neurónov rozdelujeme neuróny na vstupné, skryté (tzv. pracovné) a výstupné. Základný model neurónu je znázornený na obrázku 1.1. Neurón prijíma vstupné signály x_i cez synaptické váhy w_i . Výstup neurónu je definovaný nasledujúcim vzťahom:



Obrázok 1.1 Model neurónu.

$$o = f(p) = f\left(\sum_{i=1}^n w_i x_i + \theta\right) \quad (1.1)$$

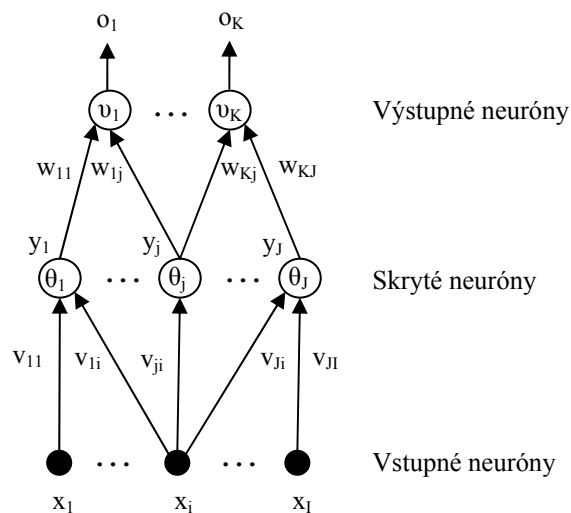
kde (x_1, x_2, \dots, x_n) je vektor vstupov, (y_1, y_2, \dots, y_J) je vektor váh, θ predstavuje prah excitácie neurónu. Funkcia f je aktivačná funkcia neurónu. V prípade optimalizácie váh prostredníctvom genetického algoritmu môže byť táto funkcia ľubovoľná. Najčastejšie sa však používa a aj v tomto prípade bude použitá unipolárna sigmoida, ktorá je definovaná vzťahom:

$$f(p) = \frac{1}{1 + e^{-\lambda p}} \quad (1.2)$$

kde konštanta $\lambda > 0$ sa nazýva strmota sigmoidy. Použitá hodnota konštanty λ bude 1.

Ako už bolo spomenuté, počet neurónov v sieti a ich vzájomné prepojenie je definované architektúrou neurónovej siete. Z hľadiska architektúry rozlišujeme dva typy neurónových sietí: acyklické (dopredné) a cyklické (rekurentné) neurónové siete. Keďže pre interpretáciu boolovských funkcií nepotrebujeme rekurentnosť v neurónovej sieti, ďalej sa budeme zaoberať len doprednými neurónovými sieťami.

Pri acyklických neurónových sieťach môžeme neuróny vždy (disjunktne) rozdeliť do tzv. vrstiev, ktoré sú usporiadané tak, že spoje medzi neurónami vychádzajú z jednej vrstvy do druhej - z nižších vrstiev do vyšších vrstiev, pričom vo všeobecnosti môžu preskočiť jednu alebo viac vrstiev. Špeciálnym prípadom takejto architektúry je tzv. viacvrstvová dopredná neurónová sieť (ďalej len dvojvrstvová). Takéto siete sa vyznačujú tým, že v nich existujú iba dopredné spojenia medzi neurónmi. Každý neurón nižšej vrstvy vysiela signály na každý neurón vyššej (nasledujúcej) vrstvy. Avšak spojenia z vyššej do nižšej (predchádzajúcej) alebo v rámci jednej vrstvy neexistujú, viď. obrázok 1.2.



Obrázok 1.2 Dvojvrstvová dopredná neurónová sieť.

Uvedenú architektúru z obrázku 1.2 môžeme zapísať ako I-J-K, kde I je počet vstupných neurónov, J je počet skrytých neurónov a K predstavuje počet výstupných neurónov. Pre interpretáciu boolovských funkcií nám bude postačovať architektúra I-J-I.

Výpočet aktivít neurónov pre dané váhové a prahové sa realizuje jednoduchým postupom, použitím vzťahov (1.3a) a (1.3b). Vstupné aktivity x_1, \dots, x_I sú známe. Pomocou (1.3a) zostrojíme aktivity skrytých neurónov y_1, y_2, \dots, y_J . Následne, pomocou (1.3b) zostrojíme aktivity výstupných neurónov o_1, o_2, \dots, o_K . Uvedený spôsob výpočtu postupuje teda zdola nahor neurónovou sieťou (viď. obrázok 1.2).

$$y_j = f\left(\sum_{i=1}^I v_{ji} x_i + \theta_j\right) \quad (\text{pre } j = 1, 2, \dots, J) \quad (1.3a)$$

$$o_k = f\left(\sum_{j=1}^J w_{kj} y_j + v_k\right) \quad (\text{pre } k = 1, 2, \dots, K) \quad (1.3b)$$

kde f je už spomínaná aktivačná funkcia.

Pre experimentovanie bude teda použitá neurónová sieť s architektúrou I-J-1. Aktivačná funkcia neurónov bude unipolárna sigmoida (1.2) so strmou $\lambda = 1$. Váhy neurónovej siete budú optimalizované genetickým algoritmom, ktorý je opísaný v nasledujúcej kapitole.

2 Genetický algoritmus

Moderná informatika dnes hľadá inšpiráciu v živej prírode, snaží sa formalizovať a implementovať paradigmy živej prírody pre návrh nových algoritmov, postupov a metód. Formalizáciou týchto paradigiem, presnejšie formalizáciou Darwinovej evolučnej teórie vstupujú do modernej informatiky metódy nazvané evolučné algoritmy. Evolučné algoritmy sa delia do troch hlavných skupín. Jednu skupinu tvoria genetické algoritmy, čo vyplýva aj z tzv. „evolučnej rovnice“ $EA = GA + ES + EP$.

Genetický algoritmus je v súčasnosti najpoužívanejší evolučný optimalizačný algoritmus. Tento algoritmus sa používa hlavne v problémových oblastiach, v ktorých klasické metódy zlyhávajú. Preto je vhodný aj pre riešenie problému optimalizácie váh neurónovej siete.

Vo všeobecnej rovine môžeme evolúciu populácie chápať ako zmenu populácie, ktorá sa realizuje v čase po jednotlivých krokoch (generáciách). Všeobecne môžeme túto zmenu zapísať vzťahom [4]:

$$P_{t+1} = R(P_t) \quad (2.1)$$

kde R reprezentuje „reprodukčný“ operátor, P je populácia jedincov. Reprezentácia jedincov v populácii závisí od konkrétneho problému. Prístup založený na vzťahu (2.1) je príliš všeobecný, preto je potrebná jeho podrobnejšia špecifikácia, ktorá je určená nasledujúcim algoritmom [4]:

```

procedure Genetic_Algorithm(input  $t_{\max}$ , output  $\alpha_{\text{opt}}$ )
begin
 $t := 0$ ;  $\text{stop\_crit} := \text{false}$ ;
 $P = \{\text{náhodne vybraná popul.}\}$ 
while ( $t < t_{\max}$ ) and (not  $\text{stop\_crit}$ ) do
  begin  $t := t + 1$ ;  $Q := \{\}$ ;
    while  $|Q| < |P|$  do
      begin vyber ruletou 2 chromozómy  $\alpha_1, \alpha_2 \in P$ ;
        if random  $< P_{\text{repro}}$  then  $\text{Reproduction}(\alpha_1, \alpha_2, \alpha'_1, \alpha'_2)$ 
        else begin  $\alpha'_1 = \alpha_1$ ;  $\alpha'_2 = \alpha_2$ ; end;  $Q = Q \cup \{\alpha'_1, \alpha'_2\}$ 
      end;  $P := Q$ ;
      if je splnené konv. Kritérium then  $\text{crit\_stop} := \text{true}$ ;
    end;  $\alpha_{\text{opt}} = \text{najlepší jedinec v populácii } P$ ;
  end;

```

Algoritmus 2.1 Pseudopascalovská implementácia genetického algoritmu [4].

Ako môžeme vidieť z algoritmu 2.1, kľúčovú časť tvorí reprodukčný operátor. Tento operátor sa väčšinou skladá z kríženia jedincov a následnej mutácie vzniknutých potomkov. Kríženie býva zväčša realizované ako jednobodové kríženie, aj keď existujú rôzne iné spôsoby. Mutácia závisí od reprezentácie jedinca pre daný problém. Do reprodukčného procesu sú jedinci vyberaní kvázi náhodne, teda lepší jedinci majú väčšiu šancu – ide o tzv. selekčný tlak.

Pre použitie genetického algoritmu na hľadanie riešenia pre daný problém je teda potrebné dodefinovať nasledujúce časti:

- Genetické kódovanie jedinca
- Ohodnotenie jedinca
- Operátor kríženia
- Operátor mutácie

Avšak, jedným z hlavných problémov genetického algoritmu 2.1 je udržiavanie diverzity v populácii jedincov. Vplyvom selekčného tlaku genetický algoritmus smeruje k jednému riešeniu, ktoré však nemusí predstavovať globálne optimálne riešenie a tak algoritmus skončí pri lokálnom riešení, keďže je v populácii malá diverzita. Z tejto situácie nám čiastočne môže pomôcť operátor mutácie. Existujú však aj sofistikovanejšie prístupy, preto pre optimalizáciu váh neurónovej siete bude použitá modifikácia klasického genetického algoritmu, ktorá je priblížená v nasledujúcej podkapitole.

2.1 Modifikácia genetického algoritmu

Jednou kategóriou metód používaných pre udržiavanie diverzity v populácii sú aj tzv. angl. niching metódy [5]. Cieľom týchto metód je prevencia, alebo zabezpečenie pred predčasnou konvergenciou populácie k jednému (mnohokrát lokálnemu) riešeniu. Tieto metódy umožňujú genetickému algoritmu paralelne preskúmať mnoho vrcholov problémového priestoru. Zvolenou technikou z týchto metód je technika zdieľania ohodnotenia (angl. explicit fitness sharing, EFS) medzi genotypovo podobnými jedincami [5, 6], ktorá bude nasledovne priblížená.

Prirodzený systém namiesto toho, aby evolúciou vytváral jednoduchú populáciu tvorenú podobnými jedincami, vytvára populáciu tvorenú rôznymi subpopuláciami jedincov – druhmi jedincov. V prírode teda majú odlišné štruktúry tendencie združovať sa do druhov, kde si navzájom konkurujú. Druh v populácii predstavuje skupina jedincov s podobnými vlastnosťami, ktorý sú schopný kríženia medzi sebou. Pre každý druh existuje konečný počet zdrojov v ich prostredí, ktoré musia byť zdieľané v rámci druhu.

Myšlienkou je teda rozdeliť celú populáciu do druhov jedincov, pričom genotypovo podobný jedinci sa vyvíjajú v rámci jedného druhu. Rozdelenie populácie je zvyčajne realizované na základe definovaného kritéria – miery podobnosti jedincov. Každý jedinec v populácii musí teda patriť do nejakého druhu, ktorý má reprezentatívneho jedinca. Na základe podobnosti k reprezentatívnemu jedincovi sa určuje príslušnosť ostatných jedincov k danému druhu. Ak jedinec nie je dostatočne podobný ani k jednému z reprezentantov druhov, tak vznikne nový druh, ktorého reprezentantom sa stáva tento jedinec. Vždy, keď vytvárame novú generáciu je vhodné najprv zistiť príslušnosť nových jedincov k druhom z predchádzajúcej generácie. Takto môžeme sledovať stagnáciu, resp. postupné zlepšovanie sa jednotlivých druhov počas evolúcie.

Avšak samotné rozdelenie populácie do jednotlivých druhov ešte nezaručuje, že celá populácia sa raz nepohltí jedným druhom. Preto je vhodné ako reprodukčný mechanizmus použiť metódu EFS, ktorá vychádza z obmedzenosti zdrojov a ich zdieľania v priestore. EFS modifikuje priestor jedincov druhu v zmysle redukcie ich produkcie v rámci husto osídleného priestoru. Čo znamená redukciu ohodnotenia jedincov v pomere k počtu jedincov nachádzajúcich sa v danom spoločnom priestore.

Zvyčajne sa zdieľané ohodnotenie, f_i' jedinca určí vzťahom [6, 7]:

$$f_i' = \frac{f_i}{\sum_{j=1}^n sh(\delta(i, j))} = \frac{f_i}{N_i} \quad (2.2)$$

kde

$$sh(\delta(i, j)) = \begin{cases} 0 & \text{pre } \delta(i, j) > \delta_t \\ 1 & \text{pre inak} \end{cases} \quad (2.3)$$

predstavuje tzv. sharing funkciu, N_i je teda počet jedincov v druhu (zjednodušenie, keďže sumácia by mala bežať cez celú populáciu, čo je však časovo náročné, keďže to treba urobiť pre každého jedinca po každom evolučnom kroku), $\delta(i, j)$ predstavuje mieru podobnosti jedinca k inému jedincovi, δ_t je prah kompatibility jedincov, f_i je pôvodné ohodnotenie jedinca. Po rozdelení jedincov do druhov a následnom určení ich f_i' hodnôt, môžeme pre každý druh určiť podiel potomkov na celkovej produkcii v rámci celej populácie podľa nasledujúceho vzťahu [7]:

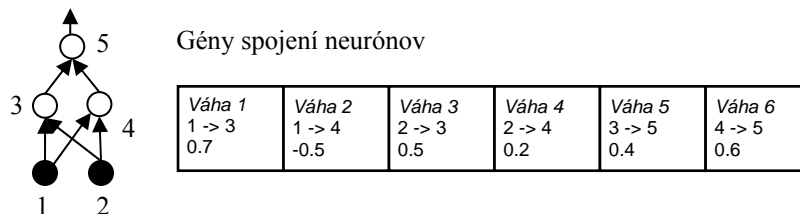
$$p_k = \frac{\bar{F}_k}{\bar{F}_{tot}}, n_k = p_k |P_n| \quad (2.4)$$

kde \bar{F}_k je priemerné ohodnotenie druhu k , $\bar{F}_{tot} = \sum_k \bar{F}_k$, $|P_n|$ je počet jedincov v druhu. Každý druh teda môže vyprodukovať n_k jedincov.

Predchádzajúcimi myšlienkami by mala byť počas evolúcie udržiavaná diverzita v populácii a preto môžeme použiť aj „agresívnejšiu“ formu genetického algoritmu a tak vytvárať novú populáciu len z najlepších jedincov. Pred reprodukčným mechanizmom vyberieme daný počet najlepších jedincov, ktorý budú pozostávať z elitárov jednotlivých druhov doplnených najlepšími jedincami (na základe zdieľaného ohodnotenia). Následne sa v jednotlivých druhoch ponechajú len tí jedinci, ktorý sa nachádzajú v spomínanom výbere. Z jednotlivých druhov sa potom vyprodukuje daný počet potomkov. Do nasledujúcej generácie sa tiež prenášajú jedinci z výberu ako aj ich kópie, ktoré môžu zmutovať.

2.2 Genetické kódovanie a miera podobnosti jedincov

Keďže počas evolúcie neurónových sietí bude ich štruktúra nemenná – ide len o optimalizáciu váh, neurónovú sieť bude kódovaná reťazcom váh, ako je to znázornené na obrázku 2.1. Váhy budú reprezentované reálnym číslom.



Obrázok 2.1 Genetické kódovanie neurónovej siete.

Na základe kódovania môžeme určiť vzťah pre výpočet miery kompatibility dvoch jedincov, ktorý je potrebný pre zaradenie jedinca do druhu, na základe podobnosti

k reprezentatívneému jedincovi daného druhu. Pri určovaní tohto vzťahu sa vychádzalo zo vzťahu uvedeného v publikácii [7]. Jeho zjednodušením dostávame vzťah pre výpočet miery kompatibility dvoch jedincov A a B:

$$\delta(A, B) = \overline{W} = \frac{\sum_{i=1}^{Pocet_vah} abs(A_vaha_i - B_vaha_i)}{Pocet_vah} \quad (2.5)$$

2.3 Ohodnotenie jedinca

Pre výpočet ohodnotenia jedinca sa najprv spočíta kumulovaná chyba podľa vzťahu (2.6) [2] a následne podľa vzťahu (2.7) ohodnotenie na základe tejto chyby.

$$E_p = \frac{1}{2} \sum_{k=1}^K (d_k - o_k)^2, E_{total} = \sum_p E_p \quad (2.6)$$

kde K je počet výstupných neurónov (v našom prípade K = 1), d_k je požadovaný výstup k-teho výstupného neurónu, o_k je skutočný výstup k-teho výstupného neurónu, p je počet vzorov.

$$Fitness = \frac{1}{1 + E_{total}} \quad (2.7)$$

2.4 Operátor kríženia a mutácie

Na rozdiel od klasického jednobodového kríženia, budem používať tzv. diskrétnu kríženie [4], ktoré sa používa hlavne v oblasti evolučných stratégií. Výsledkom diskrétného kríženia je jeden potomok, ktorý preberá hodnoty váh na zodpovedajúcich miestach náhodným výberom z jedného alebo z druhého rodiča s pravdepodobnosťou 0.5.

Váha 1 1 -> 3 0.7	Váha 2 1 -> 4 -0.5	Váha 3 2 -> 3 0.5	Váha 4 2 -> 4 0.2	Váha 5 3 -> 5 0.4	Váha 6 4 -> 5 0.6	Rodič 1
↓			↓	↓		
Váha 1 1 -> 3 0.7	Váha 2 1 -> 4 3.4	Váha 3 2 -> 3 -0.1	Váha 4 2 -> 4 0.2	Váha 5 3 -> 5 0.4	Váha 6 4 -> 5 -3.6	Potomok
	↑	↑			↑	
Váha 1 1 -> 3 1.6	Váha 2 1 -> 4 3.4	Váha 3 2 -> 3 -0.1	Váha 4 2 -> 4 0.7	Váha 5 3 -> 5 -4.4	Váha 6 4 -> 5 -3.6	Rodič 2

Obrázok 2.2 Diskrétna kríženie jedincov.

Keďže sú váhy reprezentované reálnym číslom operátor mutácie predstavuje gaussovskú mutáciu. Daná váha sa s určitou pravdepodobnosťou zmutuje podľa nasledovného vzťahu:

$$váha_i = váha_i + N(0, \Theta) \quad (2.8)$$

kde $N(0, \Theta)$ náhodné číslo z gaussovskej distribúcie s nulovou strednou hodnotou a štandardnou odchýlkou Θ . Hodnoty váh sa môžu pohybovať v danom intervale (napr. (-500.0, 500.0)). Ak je po mutácii presiahnutá jedna z hraníc, váha sa nastaví na

presiahnutú hraničnú hodnotu. Váhy sú na začiatku inicializované na náhodné hodnoty z intervalu (-1, 1) prostredníctvom generátora náhodných čísel s rovnomerným rozdelením.

3 Experimenty

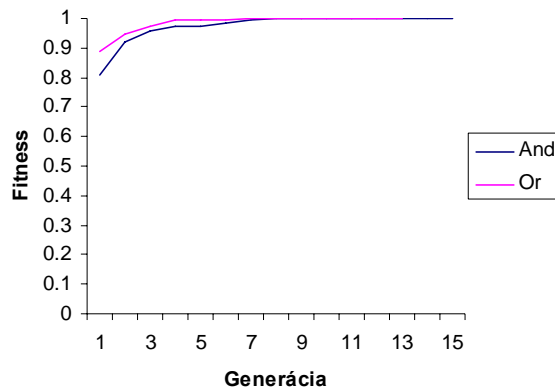
Experimenty sú zamerané na adaptáciu neurónovej siete, ktorá má interpretovať zvolenú boolovskú funkciu. Vstupom algoritmu je teda boolovská funkcia zadaná tabuľkou a nasledovné parametre simulácie. V tabuľke sú uvedené aj niektoré hodnoty atribútov, ktoré boli použité počas simulácií. Počet skrytých neurónov je rovný počtu vstupných neurónov, pri problémoch pri ktorých sú potrebné.

Veľkosť populácie	150
Veľkosť výberu do reprodukčného procesu	0.2
Pravdepodobnosť mutácie váhy	0.72
Veľkosť štandardnej odchýlky pre mutáciu	1.5
Počet skrytých neurónov	x
Maximálna hodnota váhy	500.0
Minimálna hodnota váhy	-500.0
Maximálny počet generácií a behov	2500, 10
Cieľové ohodnotenie	0.9999
Prah kompatibility jedincov	2.5

Tabuľka 3.1 Parametre simulácie.

3.1 AND, OR, XOR

Riešenie pre prvé dve funkcie AND a OR bolo nájdené v priebehu približne 15 generácií. Počet skrytých neurónov bol nulový. Grafy priebehov pre ohodnotenie najlepšieho jedinca sú na obrázku 3.1.

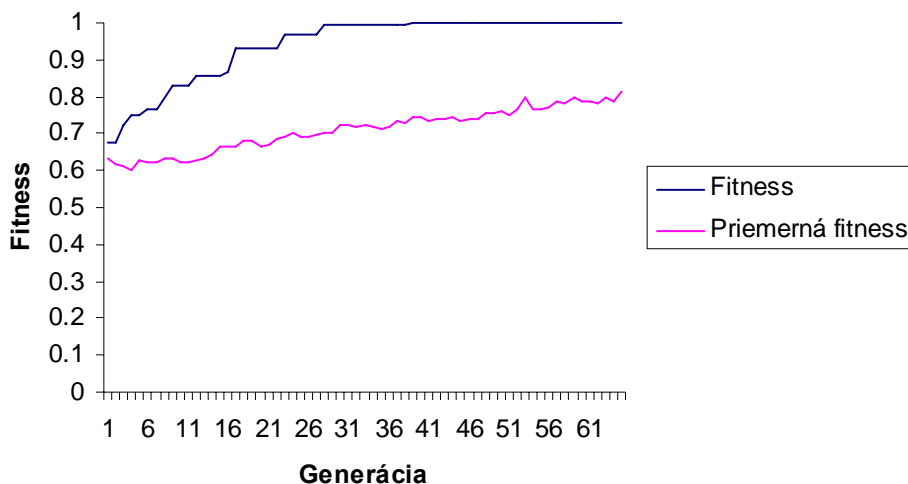


Obrázok 3.1 Priebeh fitness pre funkcie AND a OR.

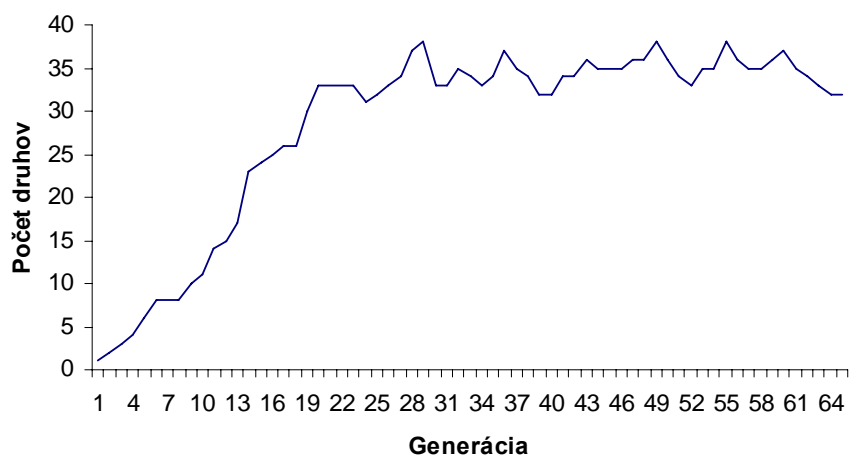
Pre 2-bitový problém XOR s dvoma skrytými neurónmi bolo riešenie nájdené v priemere v priebehu 36 generácií. Nasledujúca tabuľka 3.2 zobrazuje výstupy najlepšieho jedinca. Graf priebehu fitness a priebehu počtu druhov v populácii je znázornený na obrázku 3.2 a 3.3.

Vstup 1	Vstup 2	Požadovaný výstup	Skutočný výstup
0	0	0	0.0071
0	1	1	0.9913
1	0	1	0.9983
1	1	0	0.0066

Tabuľka 3.2 Výstupy najlepšieho jedinca pre XOR problém.



Obrázok 3.2 Priebek fitness pre evolúciu neurónovej siete pre XOR problém.



Obrázok 3.3 Priebek počtu druhov počas evolúcie siete pre XOR problém.

Z priebehu na obrázku 3.2 si môžeme všimnúť, že už v prvej generácii dosahuje najlepší jedinec ohodnotenie dosť vysoké ohodnotenie, čo vyplýva aj z exponenciálneho charakteru fitness funkcie (2.7).

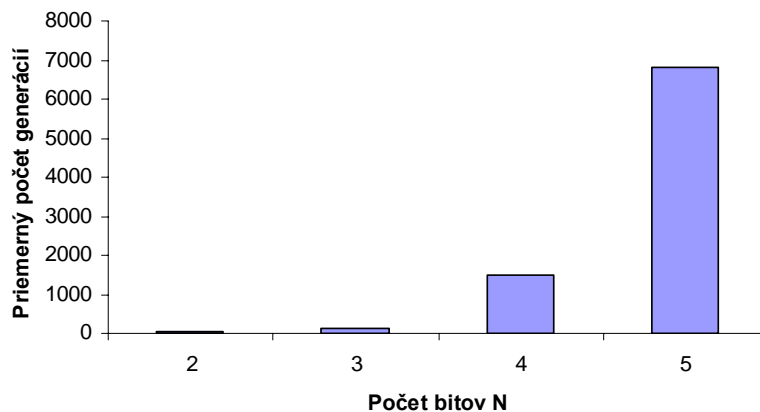
Na obrázku 3.3 je znázornený priebeh zmeny počtu druhov. Tu môžeme vidieť, že na začiatku evolúcie existuje jeden druh, keďže váhy všetkých sietí boli na začiatku

inicializované z intervalu $(-1, 1)$, teda jedinci si boli veľmi podobný, čo samozrejme závisí aj od hodnoty prahu kompatibility jedincov. Jedinci sa postupne vplyvom reprodukčného procesu od seba vzdiaľujú a tak vytvárajú nové druhy. Počet druhov v populácii v ďalších generáciách osciluje okolo hodnoty 35.

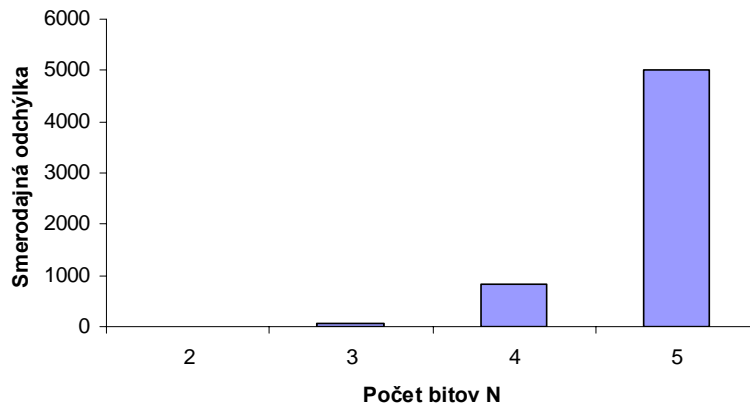
Keďže boli tieto úlohy pomerne jednoduché, ďalší experiment sa zameriava na N-bitový XOR.

3.2 N-bitový XOR

Cieľom tohto experimentu bolo zistiť, ako závisí dĺžka evolučného procesu od veľkosti XOR problému. Z časových dôvodov boli experimenty vykonané pre $N = 2, \dots, 5$. Počet skrytých neurónov bol rovný počtu vstupných neurónov (teda bol rovný N). Veľkosť populácie ako aj ostatné parametre boli počas experimentov nastavené podľa tabuľky 3.1. Výsledky – priemerný počet generácií a smerodajná odchýlka boli počítané z 50 behov simulácie pre každé N a sú znázornené na obrázkoch 3.4 a 3.5, z ktorých môžeme vidieť, že počet generácií so zväčšujúcim sa počtom bitov pomerne rýchlo rastie.



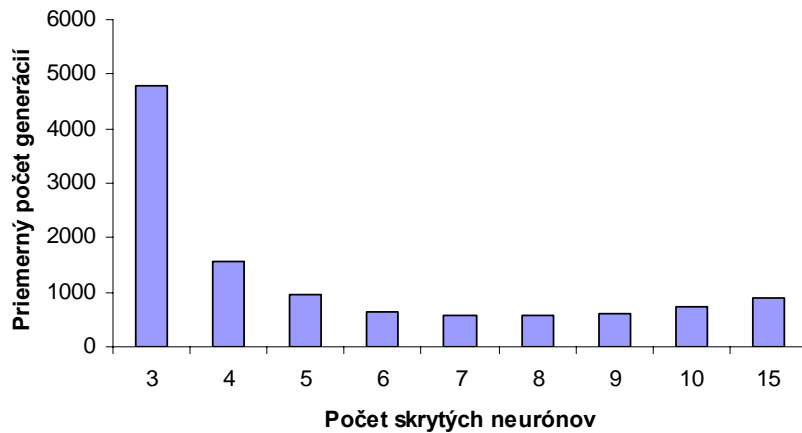
Obrázok 3.4 Závislosť počtu generácií od počtu bitov XOR problému.



Obrázok 3.5 Smerodajná odchýlka pre počty generácií pre rôzne počty bitov XOR problému.

Výsledné reakcie najlepších sietí (z jedného z behov) pre počet bitov 2, 3, 4, 5 sú zachytené v tabuľke 4.1 v prílohe.

Posledným časťou tohto experimentu bolo zistenie závislosti počtu generácií od počtu skrytých neurónov v sieti. Pre tento experiment bol vybraný 4 bitový XOR. Pre každý počet skrytých neurónov bolo vykonaných 10 behov (z časových dôvodov len 10), pričom každý beh bežal dovtedy kým nebolo nájdené žiadané riešenie. Výsledky experimentu sú zobrazené na obrázku 3.6.



Obrázok 3.6 Znáznornenie závislosti počtu generácií od počtu skrytých neurónov v sieti.

Z priebehu na obrázku 3.6 je vidieť, že pre 3 skryté neuróny evolúcia trvá pomerne dlho oproti evolúcii s väčším počtom skrytých neurónov. So zväčšujúcim sa počtom skrytých neurónov sa riešenie nájde po menšom počte generácií. Optimálny počet skrytých neurónov je v tomto prípade 7, resp. 8. Priemerný počet generácií pri 7 skrytých neurónoch bol 565, pri 8 skrytých neurónoch 567. Ďalej, s rastúcim počtom skrytých neurónov počet generácií potrebných na nájdenie riešenia mierne rastie, čo vyplýva z toho, že pre riešenie spomínaného problému postačuje menší počet skrytých neurónov, pri väčšom počte sa už len zrejme zbytočne zväčšuje prehľadávaný priestor váh, keďže už aj pri menších počtoch skrytých neurónov boli dosiahnuté požadované výsledky.

4 Príloha

Vstup	N = 2	N = 3	N = 4	N = 5
0 0 0 0	3.5723E-08	0.00158	0.0021	0.0018
0 0 0 1	0.9915	0.9990	0.9996	0.9999
0 0 1 0	0.9961	0.9925	0.9980	0.9999
0 0 1 1	0.0014	0.0005	0.0012	3.3959E-05
0 0 1 0 0	x	0.9998	0.9995	0.9999
0 0 1 0 1	x	0.0014	0.0038	1.9310E-05
0 0 1 1 0	x	0.0003	0.0015	0.0010
0 0 1 1 1	x	0.9966	0.9994	0.9915
0 1 0 0 0	x	x	0.9995	0.9999
0 1 0 0 1	x	x	0.0011	0.0018
0 1 0 1 0	x	x	0.0012	0.0018
0 1 0 1 1	x	x	0.9879	0.9999
0 1 1 0 0	x	x	9.3741E-5	0.0018
0 1 1 0 1	x	x	0.9993	0.9999

0 1 1 1 0	x	x	0.9969	0.9999
0 1 1 1 1	x	x	1.1149E-5	1.9791E-05
1 0 0 0 0	x	x	x	0.9983
1 0 0 0 1	x	x	x	0.0018
1 0 0 1 0	x	x	x	0.0018
1 0 0 1 1	x	x	x	0.9999
1 0 1 0 0	x	x	x	0.0018
1 0 1 0 1	x	x	x	0.9999
1 0 1 1 0	x	x	x	0.9999
1 0 1 1 1	x	x	x	2.0673E-05
1 1 0 0 0	x	x	x	0.0016
1 1 0 0 1	x	x	x	0.9999
1 1 0 1 0	x	x	x	0.9999
1 1 0 1 1	x	x	x	0.0018
1 1 1 0 0	x	x	x	0.9999
1 1 1 0 1	x	x	x	0.0018
1 1 1 1 0	x	x	x	0.0026
1 1 1 1 1	x	x	x	0.9999

Tabuľka 4.1 Tabuľka výstupov (zaokrúhlených) najlepších sietí pre 2, 3, 4, 5 bitový XOR. Počet vstupov do danej siete je N, pričom vstupy sú brané zľava v bitovom reťazci.

Použitá literatúra

- [1] Kvasnička, V. et al.: Úvod do teórie neurónových sietí: IRIS Bratislava, 1997.
- [2] Návrat, P. et al.: Umelá inteligencia, 2002.
- [3] Sinčák, P., Andrejková, G.: Neurónové siete – Inžiniersky prístup (1. diel), 1996.
- [4] Kvasnička, V., Pospíchal, J., Tiňo, P.: Evolučné algoritmy, 2000.
- [5] Mahfoud, S. W.: Niching Methods for Genetic Algorithms, 1995.
- [6] Sareni, B., Krähenbuhl, L.: Fitness Sharing and Niching Methods Revisited. IEEE Transactions on Evolutionary Computation, vol. 2, No. 3, September 1998.
- [7] Stanley, K. O., Miikkulainen, R.: Evolving Neural Networks through Augmenting Topologies, 2002. Evolutionary Computation, 10 (2):99-127 (2002).