

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA CHEMICKEJ A POTRAVINÁRSKEJ TECHNOLOGIE

Evidenčné číslo: FCHPT-5414-58729

**Využitie umelej inteligencie v klasifikačných a predikčných
úlohách**

Diplomová práca

Bratislava 2013

Bc. Martina Švecová

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA CHEMICKEJ A POTRAVINÁRSKEJ TECHNOLOGIE

Evidenčné číslo: FCHPT-5414-58729

**Využitie umelej inteligencie v klasifikačných a predikčných
úlohách**

Diplomová práca

Študijný program : automatizácia a informatizácia v chémii a potravinárstve

Číslo študijného odboru: 2621

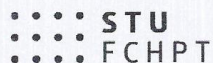
Názov študijného odboru: 5.2.14 automatizácia

Školiace pracovisko: Ústav automatizácie, informatizácie a matematiky

Vedúci záverečnej práce/školiťel': Ing. Marián Gall, PhD.

Bratislava 2013

Bc. Martina Švecová



ZADANIE DIPLOMOVEJ PRÁCE

Študentka: **Bc. Martina Švecová**
ID študenta: 58729
Študijný program: automatizácia a informatizácia v chémii a potravinárstve
Študijný odbor: 5.2.14 automatizácia
Vedúci práce: Ing. Marián Gall, PhD.

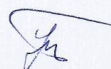
Názov práce: **Využitie umelej inteligencie v klasifikačných a predikčných úlohách**

Špecifikácia zadania:

1. Realizovať prehľad problematiky s dôrazom na dopredné neurónové siete s učiacim algoritmom: Backpropagation of Error.
2. Realizovať teoretickú analýzu neurónových sietí používajúcich algoritmus Backpropagation of Error a jeho modifikáciu s názvom Delta bar Delta. Takisto sa zamerať na Evolučnú Stratégiu, ako možnú náhradu učiaceho algoritmu v dopredných neurónových sieťach.
3. Vytvoriť integrované programové prostredie s implementovanou neurónovou sieťou používajúcou algoritmus Backpropagation of Error, Delta bar Delta modifikáciu a Evolučnú stratégiu.
4. Navrhnuť systém experimentov s dôrazom na sledovanie parametrov vybraných predikčných systémov na testovacích údajoch.
5. Výsledky jednotlivých vybraných metód vyhodnotiť a porovnať navzájom.
6. Vypracovať dokumentáciu podľa pokynov vedúceho diplomovej práce.

Riešenie zadania práce od: 18. 02. 2013

Dátum odovzdania práce: 25. 05. 2013


Bc. Martina Švecová
študentka




prof. Ing. Miroslav Fikar, DrSc.
vedúci pracoviska


prof. Ing. Miroslav Fikar, DrSc.
garant študijného programu

Týmto by som sa chcela poďakovať v prvom rade môjmu školiteľovi diplomovej práce Ing. Mariánovi Gallovi, PhD. Predovšetkým za odborné vedenie, trpezlivosť, poskytnuté konzultácie, cenné informácie, pripomienky pri spracovaní tejto práce a za čas, ktorý mi venoval pri písaní diplomovej práce.

Čestné vyhlásenie

Čestne vyhlasujem, že som diplomovú prácu vypracovala samostatne a uviedla som všetku použitú literatúru.

Dátum:

.....

(Bc. Martina Švecová)

Súhrn

Umelé neurónové siete prežívajú v posledných rokoch explóziu záujmu. Financie, medicína, inžinierstvo, geológia a fyzika patria medzi oblasti v ktorých boli úspešne nasadené a používané. Existencia predikčných a klasifikačných problémov priamo predurčuje použitie umelých neurónových sietí v danej problematike. Ich použitie môžeme zovšeobecniť na všetky situácie, kde vieme nájsť vzťah medzi prognostickými a predpokladanými premennými. V prípade komplexnosti a problematickosti vyjadrenia vzťahu dokážu práve umelé neurónové siete dosahovať veľmi dobré výsledky. V tejto práci sme sa zamerali na porovnanie predikčných schopností troch rôznych dopredných neurónových sietí založených na algoritme so spätným šírením chyby a evolučnej stratégii.

Kľúčové slová: umelé neurónové siete, predikcia časových radov, perceptrón, metóda spätného šírenia chyby, delta-bar-delta pravidlo, evolučná stratégia

Abstract

Within last few years, neural networks have become subject of increased interest. Neural networks were successfully applied in variety of areas, such as finances, medicine, engineering, geology and physics. Artificial neural networks are a natural consequence anywhere, where problems of prediction, classification or control emerge. They can be used in any situation, where a relationship between prognostic variables and assumed variables exists. Artificial neural networks achieve outstanding results even in cases that require high level of complexity and uncertainty of definition in a relationship. In this work we compared prediction abilities of three different feed forward neural networks based on back-propagation of error and evolutionary strategy learning algorithms.

Key words: artificial neural networks, time series prediction, perceptron, back-propagation of error learning algorithm, delta-bar-delta rule, evolutionary strategy

Obsah

1. Úvod	10
2. Umelá inteligencia	11
2.1 História	11
2.2 Teória umelej inteligencie	11
2.3 Ciele umelej inteligencie	12
2.4 Prístupy k riešeniu	12
3. Neurónové siete	13
3.1 Neurón	14
3.1.1 Neurón z biologického hľadiska.....	14
3.1.2 Neurón z hľadiska umelých neurónových sietí	16
3.2 Umelá neurónová sieť	18
3.2.1 Topológia a spôsob šírenia signálu.....	19
3.2.2 Najjednoduchšia neurónová sieť - perceptrón.....	21
3.3 Učenie umelých neurónových sietí	22
3.3.1 Back-propagation of error	23
3.3.2 Delta bar delta pravidlo	27
4. Evolučné stratégie	31
4.1 Mnohočlenné stratégie a kovariancie	32
5. Predikcia	34
5.1 Princípy problematiky predikcie	36
5.1.1 Problematika vstupov a výstupov pre konštrukciu modelu systému a samotný návrh modelu (neurónovej siete).	36
5.1.2 Adaptácia siete.....	36
5.1.3 Extrakcia znalostí z adaptovaného modelu	37
5.2 Časové rady	37
5.2.1 Analýza časových radov.....	38
5.2.2 Dekompozícia časových radov	38
5.2.3 Trend	38
5.2.4 Predpovede v časových radoch	39
5.3 Prediktor na báze umelých neurónových sietí	39
5.3.1 Štruktúra prediktora.....	40
5.3.2 Realizácia prediktora	41
6 Experimenty	43
6.1 Návrh experimentov	43

6.1.1	Štruktúra experimentov	45
6.1.2	Spôsob hodnotenia výsledkov	45
6.2	Vyhodnotenie experimentov	46
6.2.1	Prvá fáza experimentov	47
6.2.2	Druhá fáza experimentov	49
7.	Záver	56
8.	Literatúra.....	57

1. Úvod

Umelé neurónové siete sa v súčasnosti využívajú pri riešení mnohých problémov, ktoré sú vzdialené od tých, kvôli ktorým boli tieto siete vyvinuté. Neurónové siete pôvodne vznikli ako modely mozgu, resp. nervového systému a jeho činnosti. Tieto neurónové siete v teórii aj v aplikáciách prešli od svojho vzniku obrovským vývojom. Môžu fungovať ako univerzálny aproximátor, teda môžeme povedať, že dokážu modelovať s požadovanou presnosťou akúkoľvek spojitú funkciu, ak im dáme správne vzory k ich učeniu.

Aj keď si to vlastne ani neuvedomujeme, predikcia je súčasťou nášho každodenného života. Predikujeme alebo si predstavujeme mnoho vecí, ktoré sa nám môžu udiť v blízkej budúcnosti. Napríklad také situácie ako pri riadení auta. Treba predvídať, čo sa nám môže na ceste udiť. Alebo situácie pri rôznych rozhovoroch alebo diskusiách, kedy predvídame, čo asi tak povie ten druhý človek, aby sme sa vedeli pohotovo zapojiť do diskusie.

Preto rozdeľujeme úlohy pre inteligentný systém na: Identifikáciu ako realizáciu vnútornej reprezentácie pozorovaného systému a predikovanie stavov alebo prejavov = výstup tohto systému a návrh vhodného rozhodnutia, resp. akčný zásah.

Pri klasických neurónových sieťach je identifikácia systému proces učenia siete. Vtedy sa vytvára model systému. Po naučení neurónovej siete, v dobe života, využívame model identifikovaného systému na modelovanie tohto systému napríklad kvôli predikcii správania sa systému.

Neurónové siete vznikli ako snaha napodobniť štruktúru mozgu, v ktorého paralelnej štruktúre spolupracuje obrovské množstvo neurónov, ktoré plnia jednoduché funkcie.

2. Umelá inteligencia

Je to odbor informatiky, ktorý sa zaoberá tvorbou strojov, ktoré vykazujú známky inteligentného správania sa. Definícia tohto pojmu je aj v súčasnosti stále predmetom diskusií. S týmto pojmom prišiel ako prvý John McCarthy v roku 1955 [22].

2.1 História

V rovnakom článku, v ktorom Alan Turing navrhol svoj slávny test, ktorý aj sám nazýval – imitačná hra predpovedal, že za 50 rokov budú mať počítače pamäťovú kapacitu 10^9 bitov a rozhodca imitačnej hry bude mať iba 70% šancu, že správne spozná ľudskú inteligenciu. Zatiaľ čo sa odhad pamäťových schopností ukázal byť celkom presným, dnešné počítače nedokážu pracovať s ľudským jazykom tak dobre ako Turing chcel a dúfal.

Počiatkové nadšenie sa postupom času zmenilo na hlbokú skepsu. Ukázalo sa, že naprogramovanie úloh, ktoré ľuďom pripadajú úplne triviálne, môže byť ťažšie, ako vytvoriť stroje, ktoré riešia klasické úlohy umelej inteligencie ako je napríklad šach.

V tomto zmysle je signifikantný vývoj automatického prekladu, ktorý postupom času prešiel obdobím nekritického nadšenia aj hlbokej dezilúzie. Veľmi podobne sa nádeje vkladajú do umelých neurónových sietí dostali do konfliktu s realitou neúspechu vytvoriť siete, ktoré riešia jednoduché úlohy.

Napríklad:

Neschopnosť jedného perceptronu silmulovať funkciu XOR bola mýlne zovšeobecnená na všetky neurónové siete preto, lebo nepoznali algoritmus pre jeho učenie a tak sa zbrzdil vývoj neurónových sietí na dlhé roky. Pritom aj dvojvrstvová sieť túto funkciu aproximovať dokáže [23].

2.2 Teória umelej inteligencie

Teória umelej inteligencie pokrýva množstvo teórií, ktoré si kladú za cieľ viac či menej napodobňovať schémy ľudského (prípadne vo všeobecnosti biologického) správania sa, vyhodnocovania a analýzy podnetov prostredia, prípadne tvorivej činnosti. Teória umelej inteligencie si stanovuje ciele (čo by sme od inteligentného systému očakávali) a hľadá v rôznych vedných disciplínach dostupné riešenia [22].

Umelá inteligencia sa delí na:

- a) slabú umelú inteligenciu – nedokáže splniť náročnejšie úlohy
- b) silnú umelú inteligenciu – dokonalo premyslená

2.3 Ciele umelej inteligencie

Cieľom systémov umelej inteligencie je vypracovať algoritmy alebo paradigmy, ktoré majú požiadavku, aby stroj vyriešil takú úlohu, ktorú by zvládol iba človek s patričnými znalosťami.

Dnes sa veľa ľudí alebo skupín snaží o vytvorenie algoritmov na riešenie problémov ako napr. hranie hier, kombinátorika, riadenie robotov, riadenie a plánovanie výrobných procesov, navigácie, triedenie vzoriek, rozpoznávanie obrazu, rozpoznávanie reči, reprezentácia zozbieraných dát, analýza a predpovedanie štatistických postupností, zameriavanie a sledovanie cieľov a mnoho ďalších úloh a problémov.

2.4 Prístupy k riešeniu

Je veľmi komplikované vytvoriť umelú inteligenciu, ktorá by bola totožná s ľudskou inteligenciou. Vedci však za posledných 50 rokov vyvinuli sadu postupov, ktoré dosahujú úspechy, čo sa týka rôznych problémov umelej inteligencie.

Neurónové siete

Umelé neurónové siete majú v umelej inteligencii za vzor správanie zodpovedajúcich biologických štruktúr. Skladajú sa z modelov neurónov, ktoré si navzájom predávajú signály a transformujú ich pomocou funkcie na prenos k ďalším neurónom.

Používajú sa pre rozpoznávanie a kompresiu obrazov alebo zvukov, predvídanie vývoja časových radov (napr. burzových indexov), niekedy dokonca k filtrovaniu spamu.

Genetické programovanie

Genetické programovanie, ak ho zobereme doslova, nie je prostriedok na riešenie problémov umelej inteligencie, ale všeobecný programátorský prístup, ktorý namiesto napísania konkrétneho algoritmu pre riešenie úlohy hľadá tento postup evolučnými metódami.

Expertné systémy

Expertný systém je počítačový program, ktorý ma za úlohu poskytovať expertné rady, rozhodnutia alebo odporučiť riešenia v konkrétnej situácii.

Prehľadávanie stavového priestoru

Keď vyvíjame algoritmy na vyriešenie klasických úloh resp. hier ako šach alebo dámy, musíme si zadať množinu stavov, do ktorých sa môžeme v priebehu hry dostať, prípustné ťahy resp. prechody medzi stavmi, počiatočné a koncové pozície. A hľadáme tak cestu od počiatočných stavov ku koncovým, ktoré znamenajú náš úspech. Niektoré stavové priestory môžu byť veľmi rozsiahle, v niektorých prípadoch aj nekonečné, preto musíme zvoliť metódy na orezávanie nevhodných ciest a ohodnotenie pozícií.

Dobývanie znalostí

Veľké súbory dát, ktoré sú väčšinou uložené v databázach, nie sú použiteľné a pochopiteľné priamo, aj keď obsahujú informácie a vzory správania sa sledovaného systému. Tieto metódy dobývania znalostí prevádzajú dáta do kompaktnej a explicitnej formy, ktorá popisuje systém a ktorá je lepšie použiteľná [24].

Fuzzy logika

Fuzzy logika napodobňuje ľudský spôsob uvažovania. Môžeme tu hovoriť o reprezentácii transparentných znalostí, čiže slovne vyjadriteľných. Používa vágne, neurčité resp. približné pojmy. Takisto používa pravidlový prístup.

3. Neurónové siete

Teória neurónových sietí vychádza z neurofyziologických poznatkov. Hovorí nám o správaní, ktoré funguje na princípe spracovania informácií v bunkách nervov. Umelé siete označujeme ako modely mozgu bez mysle. Nezaoberajú sa psychikou, iba sa snažia pochopiť nervový systém. Všetky vedomosti, ktoré získame v oblasti výskumu majú pre neurofyziológiu veľký význam. V dnešnej dobe nemáme k dispozícii experimentálne metodiky, pomocou ktorých dokážeme sledovať aktivitu v prirodzených neurónových sieťach (momentálne poznáme iba metódy na sledovanie aktivity jednej bunky, prípadne niekoľkých nervových buniek súčasne). Kvôli týmto faktom sa preto k niektorým poznatkom v oblasti spracovania informácií nervovým systémom možno dopracovať len myšlienkovým experimentom, napríklad počítačovým modelom [15].

3.1 Neurón

Virtuálny model neurónu sa stal základom všetkých umelých neurónových sietí. Fungovanie umelého neurónu je založené na princípe tak, aby sa čo najviac priblížil k fungovaniu prirodzeného neurónu, teda viac menej k poznatkom o neurónoch [7].

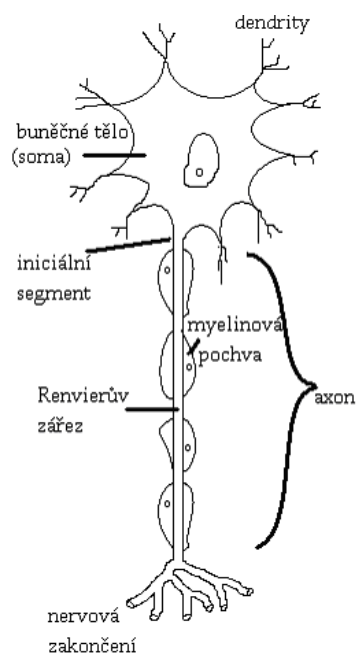
V dnešnej dobe už dokážeme vytvoriť vysoko výkonné neuróny, ktoré prekonávajú schopnosti ľudských neurónov. Sila ľudského mozgu spočíva v obrovskom počte neurónov a medzineurónových spojení. Podľa výskumov má ľudský mozog asi 10^{12} neurónov, pričom počet medzineurónových spojení je približne 10^4 na každý neurón. Takúto masívnu paralelnú neurónovú sieť ešte nedokážeme na dnešných počítačoch nasimulovať [5].

3.1.1 Neurón z biologického hľadiska

Hlavnou funkciou nervového systému je, aby riadil organizmus. Podkladom tejto funkcie je schopnosť nervového systému spracovávať informácie. Tieto informácie sa v nervovom systéme prenesú vo forme zmien membránového potenciálu nervových buniek, čiže neurónov.

Neuróny ohraničuje bunková membrána, ktorú tvorí fosfolipidová dvojvrstva. Ide o membránu semipermeabilnú, cez ktorú môžu prechádzať niektoré molekuly, ktoré majú malú molekulovú hmotnosť. Molekuly bielkovín za normálnych okolností membránou neprechádzajú. Medzi bunkou a jej okolím existuje elektrický gradient. Nie je spôsobený len molekulami bielkovín, ale aj preskupením iónov, ktoré v istom rozsahu dokážu prechádzať bunkovou membránou a vytvárajú rovnovážny stav medzi tokom do bunky a z bunky von.

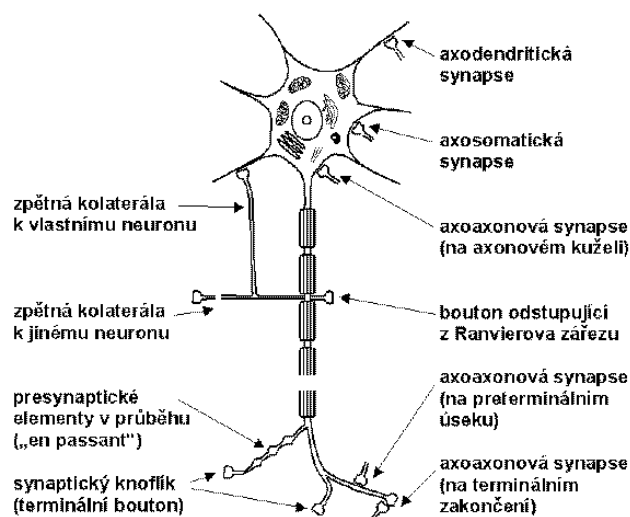
Nervovú bunku tvorí telo a niekoľko výbežkov. Tieto výbežky vieme rozčleniť na dva typy: dendrity, ktoré nám predstavujú vstupnú časť a jeden axón. Po ňom sa vzruch šíri k iným bunkám. V mieste odstupe axónu z tela neurónu sa nachádza iniciálny segment a ten má veľmi významnú funkciu pri šírení vzruchu. Axón je obklopený myelínovou pošvou. Je to bunková membrána, ktorá tesne obmotáva axón. Na hraniciach medzi dvoma susednými gliovými bunkami sa na axóne nachádzajú neobalené úseky, tzv. Ranvierove zárezy.



Obrázok 1: Stavba nervovej bunky [13].

Nervové bunky tvoria funkčné spojenia v mieste tesného kontaktu axónu jednej bunky s membránou inej bunky. Tieto spojenia sa nazývajú synapsie a podľa toho ktoré časti nervových buniek ich tvoria, poznáme axo-dendritické, axo-somatické a výnimočne aj axo-axonálne [14].

Na jednom neuróne sú stovky, tisíceky a na niektorých dokonca mnoho desiatok tisíc synapsií. Synapsia sa skladá z troch častí: z membrány presynaptického terminálu (rozšírené zakončenie axónu), synaptickej štrbiny a postsynaptickej membrány [8].



Obrázok 2 : Synapsia [14].

3.1.2 Neurón z hľadiska umelých neurónových sietí

Základným prvkom a procesnou jednotkou v neurónovej sieti je neurón. Pozostáva z nasledujúcich základných častí:

- vstup do neurónu - dendrit, je funkciou jednotlivých vstupov prichádzajúcich od predsynaptických neurónov. Vo väčšine prípadov ide o súčet týchto vstupov uvažovaných s určitými váhami. Napríklad vstup do i -teho neurónu, ktorý má N predsynaptických neurónov, môže byť vyjadrený v tvare

$$in_i = \sum_{j=1}^N w_{ij} ou_j + \phi_i, \quad (3.1)$$

kde w_{ij} sú synaptické váhy, ou_j sú výstupy z neurónov, s ktorými je prepojený, ϕ_i je prah neurónu i .

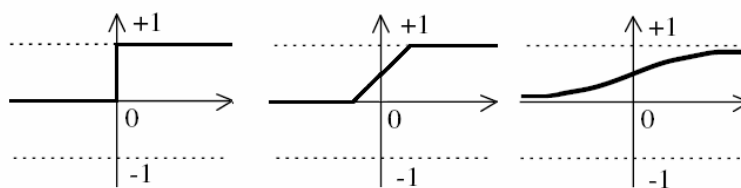
- prah neurónu, je hodnota ϕ_i , ktorá vlastne prispieva ku vstupu z externého sveta.
- aktivačná funkcia neurónu, ktorej výsledkom je x_i . Aj keď existuje veľa aktivačných funkcií, rozlišujeme tri základné typy aktivačných funkcií: prahové, po častiach lineárne a sigmoidálne funkcie. Výber aktivačnej funkcie je podmienený úlohou, ktorú neurónové siete riešia. K najčastejšie používanému druhu aktivačných funkcií patria sigmoidálne funkcie. Ide o ostro rastúce funkcie, ktoré predstavujú akúsi rovnováhu medzi lineárnym a nelineárnym správaním.

Príkladom sigmoidálnej funkcie je logistická funkcia

$$x_i = f(in_i) = \frac{1}{1 + e^{-\alpha in_i}}, \quad (3.2)$$

kde α je parameter strmosti sigmoidy. Táto funkcia je bežná a v mojej práci sa s ňou ešte stretneme.

Hlavnou výhodou týchto funkcií je vlastnosť spojitosti a diferencovateľnosti, ktorá je v teórii neurónových sietí veľmi dôležitá. Navyše sa ich derivácia dá vyjadriť v jednoduchom tvare. Tieto funkcie majú obor hodnôt $[0,1]$. Niekedy však potrebujeme aktivačné funkcie s rozsahom hodnôt $[-1,1]$. To sa dá ľahko dosiahnuť transformáciou predošlých funkcií.



Obrázok 3: prahová funkcia, po častiach lineárna funkcia, sigmoidálna funkcia.

- výstupná funkcia neurónu ou_i je taktiež dôležitou súčasťou neurónu ako procesnej jednotky. Vo všeobecnosti má tvar

$$ou_i = f(x_i) \quad . \quad (3.3)$$

- synaptické váhy, ktoré sú na synaptických spojeniach – synapsiách, ktoré majú svoj smer a spájajú jednotlivé neuróny do NN. Váhy ovplyvňujú celú sieť tým, že ovplyvňujú vstupy do neurónov a tým aj ich stavy. Delíme ich na: kladné (excitačné) a záporné (inhibičné).

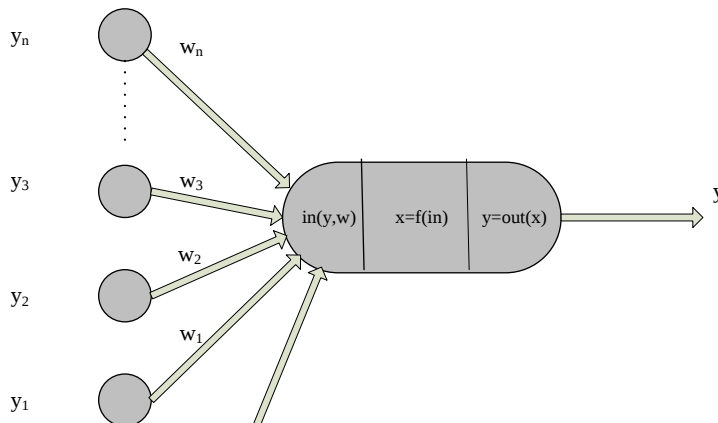
Podľa toku signálu po synapsií poznáme neuróny:

- predsynaptické - zdrojové (pred synapsiou)
- postsynaptické - cieľové (po synapsií)

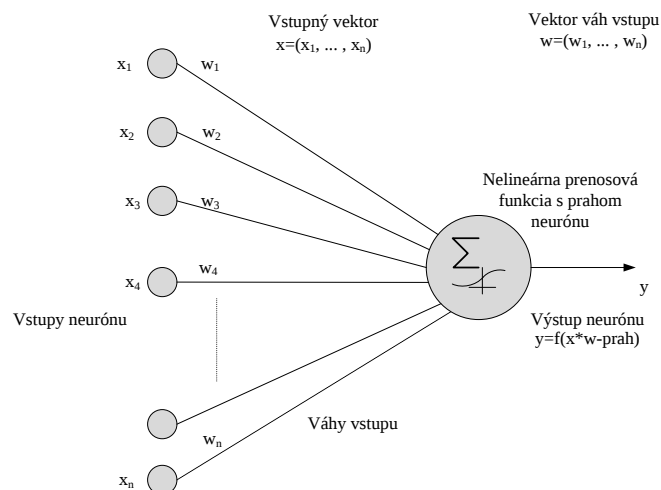


Obrázok 4: Označenie neurónov.

Na označovanie synaptických váh sa používa symbol w_{ij} , kde „i“ označuje postsynaptický neurón a „j“ označuje predsynaptický neurón. Teda ide o synapsiu, ktorá vychádza z neurónu „j“ a cieľi k neurónu „i“. Túto konvenciu pri označovaní je vhodné dodržať.



Obrázok 5: Štruktúra neurónu.



Obrázok 6: Matematický model neurónu.

Takýto jednoduchý neurón pracuje na základe nelineárnej prenosovej funkcie s prahom neurónu:

$$y = f\left(\sum_{i=1}^N w_i x_i - \theta\right), \quad (3.4)$$

kde:

y - výstup neurónu, u - vnútorný potenciál neurónu, w_i - váhy neurónu, x_i - vstupy neurónu, θ - prah neurónu, f - neurónová aktivačná funkcia [6].

3.2 Umelá neurónová sieť

Cieľom systémov umelej inteligencie je vypracovať paradigmy alebo algoritmy, ktoré požadujú od stroja riešiť také úlohy, ktoré by vyriešil len človek, ktorý má znalosti.

Umelú neurónovú sieť definujeme ako masívne paralelný procesor, ktorý má tendenciu k uchovávaní experimentálnych znalostí a ďalšieho využívania týchto znalostí. Dokáže napodobňovať ľudský mozog vo dvoch aspektoch:

- Poznatky zbiera počas učenia.
- Na ukladanie znalostí sú využívané medzineurónové spojenia, nazývané aj synaptické váhy.

Oblasti neurónovej siete:

- Teória neurónovej siete – matematický rozbor činnosti neurónovej siete, problémy neurónovej siete. Je to matematický model, ktorý popisuje správanie sa neurónovej siete a je celkom náročný.

- Simulácia neurónovej siete – simulácia pomocou počítačových systémov. Hlavným problémom zostáva neurónovú sieť na niečo naučiť.
- Implementácia – implementácia naučenej neurónovej siete do hardwarovej formy.

Typy úloh, riešiteľných pomocou neurónových sietí:

- Problémy aproximácie funkcií.
- Klasifikácie do tried, klasifikácia situácií.
- Riešenie predikčných problémov (toto je moja úloha v rámci diplomovej práce).
- Problémy riadenia procesov.
- Transformácia signálov.
- Asociačné problémy, simulácia pamäte.

Činnosť neurónových sietí rozdeľujeme do dvoch fáz:

- Fáza učenia – znalosti sa ukladajú do synaptických váh neurónovej siete. Pojem učenie pri neurónových sieťach vieme nahradiť aj synonymom pojmu adaptácie neurónových sietí. Ide totiž o zbieranie znalostí, resp. o ich uchovanie. Počas učenia sa tieto synaptické váhy menia. Učenie definujeme ako proces, v ktorom sa na základe pravidiel menia parametre neurónovej siete. Pravidlá vyvolávajú zmeny synaptických váh a ich charakter determinuje typ učenia neurónovej siete. Pod pojmom učenie si vieme predstaviť adaptáciu neurónovej siete, ktorá po skončení učenia bude nositeľkou znalostí počas celého učenia. Základnou a podstatnou vlastnosťou neurónových sietí je práve učenie. Avšak pri klasických algoritmoch neexistujú fázy učenia ani fázy života. Pri neurónových sieťach sa navrhuje všeobecný algoritmus učenia, ktorý vo fáze učenia sieť využíva. Vhodnosť tohto algoritmu determinuje kvalitu a rýchlosť učenia na predkladaných reprezentatívnych dátach. Reprezentatívnu vzorku rozdeľujeme na dva typy vzoriek: trénovaciu, ktorá sa používa vo fáze učenia a testovaciu, ktorá sa používa vo fáze života.
- Fáza života – ide o stav, kedy sa synaptické váhy nemenia a získané vedomosti sa využívajú v prospech riešenia rôznych problémov [1, 2].

3.2.1 Topológia a spôsob šírenia signálu

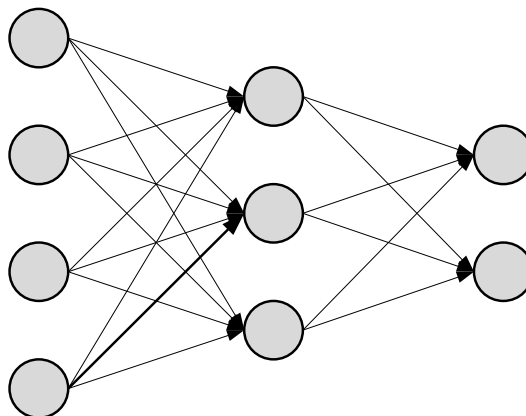
Neurónová sieť môže mať rôznu štruktúru, ktorá je opísateľná hocijakým grafom pomocou vrcholov – neurónov a orientovaných hrán – prepojení.

V neurónových sieťach rozoznávame tieto vrstvy :

- Vstupnú vrstvu, v ktorej neuróny dostávajú vstup len z vonkajšieho sveta a výstup väčšinou pokračuje k ďalším neurónom neurónovej siete.
- Skrytú vrstvu, v ktorej neuróny dostávajú vstup z ostatných neurónov alebo aj z externého sveta cez prahové prepojenia a ich výstupy pokračujú ďalej do neurónovej siete.
- Výstupnú vrstvu, ktorá je veľmi podobná ako skrytá, ale výstup z tejto vrstvy ide do externého sveta.

Topológiu neurónových sietí rozdeľujeme do dvoch základných skupín:

- Dopredné neurónové siete – v nich sa signál šíri po synaptických prepojeniach , ktoré sú orientované len jedným smerom a to dopredu (s takouto sieťou sa stretneme ďalej v mojej práci) (štruktúra vid'. Obrázok 7).
- Rekurentné neurónové siete – pri nich ťažko delíme vrstvy a neuróny na vstupné, výstupné a skryté. Niekedy neuróny predstavujú aj vstupné, niekedy aj výstupné typy neurónov.



Obrázok 7: Štruktúra doprednej neurónovej siete.

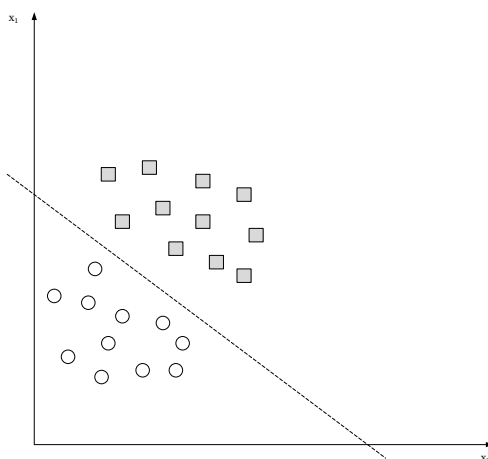
Signál sa môže šíriť veľmi rozmanito. Uvediem pár typov šírenia signálu:

- Synchronne - neuróny menia svoj stav do taktu.
- Sekvenčné – neuróny menia svoj stav postupne pri šírení.
- Blok-sekvenčné – aktivizujú sa len skupiny neurónov, ktoré majú vopred určenú stratégiu.
- Asynchronne – neuróny menia svoje stavy nezávisle jeden od druhého [1, 4].

3.2.2 Najjednoduchšia neurónová sieť - perceptrón

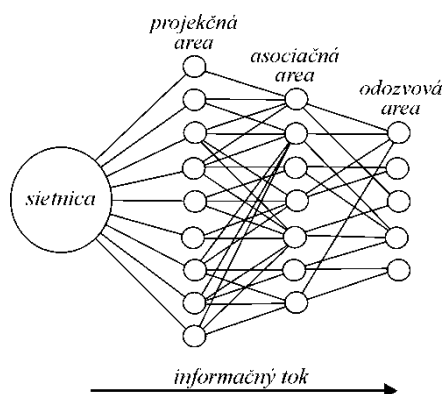
Perceptrón, najjednoduchšia neurónová sieť, je určený na dichotomickú klasifikáciu. To znamená, že je určený na rozdelenie do dvoch tried, pri ktorých predpokladáme, že sú lineárne separovateľné v príkladovom priestore.

Máme danú množinu vektorov a o každom vektore vieme, že určite patrí do triedy CL1 alebo CL2. Pod pojmom lineárna separovateľnosť rozumieme to, keď vieme oddeliť objekty v príkladovom priestore pomocou nadroviny (viď. Obrázok 8) [1, 9].



Obrázok 8: Lineárna separovateľnosť tried v príkladovom priestore.

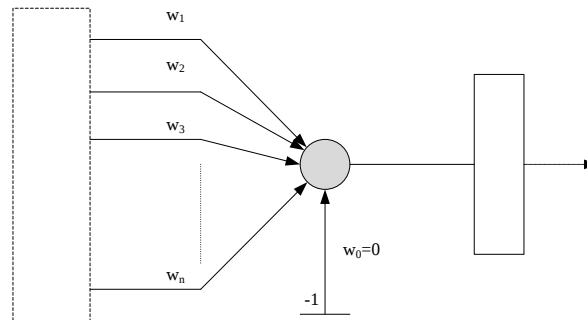
Oblasť, kde sa premieta optickým systémom oka pozorovaný objekt, sa nazýva sietnica. Tá prenáša binárne hodnoty do vrstvy nazývanej projekčná area, kde sa binárne kódovaný obraz numericky predspracováva. Spoje medzi sietnicou a projekčnou oblasťou sú pevné a neadaptabilné. Spoje do druhej vrstvy (asociačnej arey) a tiež aj do tretej vrstvy (odozvová area) sú stochasticky generované. Základným cieľom adaptačného procesu perceptrónu je nastaviť váhové koeficienty spojov tak, aby aktivity neurónov z tretej vrstvy (odozvová oblasť) správne klasifikovali obraz dopadajúci na sietnicu [16,17].



Obrázok 9: Rosenblattova predstava o perceptróne.

Perceptrón tvoria tri vrstvy:

- Senzorovú vrstvu
- Asociatívnu vrstvu
- Výstupný neurón



Obrázok 10: Topológia perceptrónu navrhnutého Rosenblattom.

Asociatívne neuróny boli určené na to, aby sa stanovili príznaky vhodné na klasifikáciu a nazývajú sa aj ϕ funkcie. Podľa charakteru tejto funkcie v podstate môžeme zatriediť jednoduchý perceptrón do rôznych skupín neurónových sietí. Spojenie medzi senzorovou vrstvou a asociatívnou má pevné váhy, teda sa na procese učenia nezúčastňuje. Spojenie medzi asociatívnou vrstvou a výstupným neurónom sa prepája synapsiami s premenlivými synaptickými váhami. Vstup do výstupného neurónu je daný rovnicou [1]:

$$in(t) = \sum_{j=1}^n w_j(t)x_j(t) - \phi \quad . \quad (3.5)$$

3.3 Učenie umelých neurónových sietí

Schopnosť učiť sa a zlepšovať svoj výkon učením je veľmi dôležitou vlastnosťou neurónových sietí. Pojem učenia neurónovej siete je synonymom pojmu adaptácie. Počas učenia sa menia synaptické váhy.

Prístupy k učeniu delíme do dvoch veľkých skupín:

- Kontrolované učenie (učenie s učiteľom) , ktoré sa ďalej delí do podskupín:
 - Štruktúrálné učenie, a to sa delí na dve metódy:
 - autoasociačné – neurónová sieť sa adaptuje tak, aby to, čo bolo na vstupe, bolo aj na výstupe (napr. simulácie pamäte).
 - heteroasociačné – neurónová sieť sa naučí rozpoznávať vstupy a zatried'ovať ich tak, ako to povedal učiteľ.

- Temporálne učenie – na vstup musí prísť za čas $t > 0$ sekvencia vstupov a až tejto sekvencii ako celku je priradený jeden výstup.
- Nekontrolované učenie (učenie bez učiteľa) [1, 10].

V mojej práci sa budeme ďalej venovať metóde kontrolovaného učenia a to metóde spätného šírenia chyby a jej modifikáciami.

3.3.1 Back-propagation of error

Tento algoritmus učenia bol odvodený pre viacvrstvové neurónové siete so spojitou nelineárnou diferencovateľnou aktivačnou funkciou. Pomocou tohto algoritmu sa minimalizuje chybová funkcia a to prostredníctvom adaptácie synaptických váh. K minimalizácii chybovej funkcie sa používajú gradientné metódy. Chybová funkcia sa rovná najčastejšie strednej kvadratickej chybe medzi požadovaným a skutočným výstupom. Na vstup neurónovej siete je privádzaný vektor resp. pre viac vzorov matice vstupných parametrov. Po priechode neurónovou sieťou je výsledok porovnaný s požadovanou hodnotou, je spočítaná chyba, ta sa spätne prepočítava do predchádzajúcich vrstiev a synaptické váhy predstavujúce pamäť sú opravené. Do opravenej siete je znova privedený vstupný vektor resp. matica a proces sa opakuje. Ide o iteratívny proces. Hľadáme minimum chyby medzi skutočnou hodnotou a požadovanou hodnotou pre všetky vzory učenia. Táto metóda je v praxi najčastejšie používaná a je vhodná pre množstvo aplikácií napr. pri predikcii nejakých parametrov alebo pre klasifikáciu. Jej nevýhodou je veľká citlivosť na relevantnosť vstupných dát a na inicializáciu synaptických váh.

Podstatou je minimalizácia učiacej chyby J , ktorú definujeme nasledovne:

$$J(t) = \frac{1}{2} \sum_{i=1}^{N_0} (ev_i(t) - x_i(t))^2 \quad (3.6)$$

kde N_0 označuje počet neurónov vo výstupnej vrstve, ev_i je očakávaná hodnota x_i je vypočítaná hodnota na i -tom výstupe a platí:

$$x_i = f(in_i) = in_i = \sum_{j=1}^M w_{ij}x_j + \theta_i \quad (3.7)$$

kde f je aktivačná funkcia i -tého neurónu, in_i je vstup do i -tého neurónu, w_{ij} je váha medzi neurónmi j a i . Zmena synaptickej váhy je určená nasledovne:

$$\Delta w_{ij}(t) = -\gamma \frac{\partial J(t)}{\partial w_{ij}(t)} = -\gamma \frac{\partial J(t)}{\partial in_i(t)} \frac{\partial in_i(t)}{\partial w_{ij}(t)} = \gamma \delta_i(t) x_j(t) \quad (3.8)$$

kde γ je konštanta, ktorá udáva rýchlosť zmeny váh a δ_i označuje chybový signál, pre ktorý možno napísať:

$$\delta_i(t) = -\frac{\partial J(t)}{\partial in_i(t)} = -\frac{\partial J(t)}{\partial x_i(t)} \frac{\partial x_i(t)}{\partial in_i(t)} = -\frac{\partial J(t)}{\partial x_i(t)} f'(in_i(t)) \quad (3.9)$$

V prípade, ak neurón “i” je výstupným, platí:

$$\delta_i(t) = -\frac{\partial J(t)}{\partial x_i(t)} f'(in_i(t)) = (ev_i(t) - x_i(t)) f'(in_i(t)) \quad (3.10)$$

V prípade, ak neurón “i” nie je výstupným, platí:

$$\begin{aligned} \delta_i(t) &= -f'(in_i(t)) \sum_{h=1}^{N_h} \frac{\partial J(t)}{\partial in_h(t)} \frac{\partial in_h(t)}{\partial x_i(t)} = -f'(in_i(t)) \sum_{h=1}^{N_h} \frac{\partial J(t)}{\partial in_h(t)} \frac{\partial}{\partial x_i(t)} \sum_{l=1}^l w_{hl}(t) x_l(t) = \\ &= -f'(in_i(t)) \sum_{h=1}^{N_h} \frac{\partial J(t)}{\partial in_h(t)} w_{hi}(t) = -f'(in_i(t)) \sum_{h=1}^{N_h} \delta_h(t) w_{hi}(t), \end{aligned} \quad (3.11)$$

kde N_h predstavuje počet synaptických spojení smerujúcich z i -tého neurónu, in_h sú vstupy do postsynaptických neurónov, N_l je počet neurónov, z ktorých smerujú synapsie do neurónov so vstupmi in_h a x_l sú aktivácie príslušných neurónov. Vzťah (3.11) je základný vzťah „back-propagation of error“ vyjadrujúci rekurzívne spätné šírenie chybového signálu δ od výstupných neurónov až po vstup siete. Zmena synaptických váh je potom odvodená z hodnoty práve tohto signálu [1].

Pre lepšie pochopenie vyjadrím najskôr základný algoritmus pre jeden neurón v jednej vrstve a následne algoritmus pre viac vrstiev.

- Algoritmus pre jeden neurón v jednej vrstve

Pre potenciál jedného neurónu teda platí, že ak označíme výstup z neurónu y , požadovanú hodnotu t a ich rozdiel $e = t - y$, potom platí:

$$u = \sum_{i=0}^n w_i x_i = W\bar{x} \quad , \quad (3.12)$$

kde $x_0 = 1$, W je matica váh a x je vstupný vektor.

Ďalej budeme predpokladať, že máme množinu tréningových vzoriek $\{(x(k), t(k)); 1 \leq k \leq K\}$. Vlastný tréning začína privedením všetkých K vstupov do siete a

výpočtom zodpovedajúcich výstupov $\{(y(k); 1 \leq k \leq K)\}$. Suma štvorcov chýb bude vypočítaná podľa nasledujúceho vzťahu:

$$E = \sum_{k=1}^K [e(k)]^2 = \sum_{k=1}^K [t(k) - y(k)]^2 = \sum_{k=1}^K [t(k) - f(Wx(k))]^2 \quad (3.13)$$

Cieľom je úprava matice váh W tak, aby bola dosiahnutá minimálna chyba E . Je to nelineárna optimalizácia metódou najmenších štvorcov. Algoritmus sa dá zapísať maticovým spôsobom pomocou nasledujúceho vzťahu:

$$W(t+1) = W(t) + \Delta W(t) \quad , \quad (3.14)$$

kde $\Delta W(t)$ je odchýlka aktuálnych váh.

„Back-propagation of error“ algoritmus je gradientná metóda. V nasledujúcich vzťahoch popíšem výpočet chyby E v závislosti na jednotlivých synaptických váhach.

$$\frac{\partial E}{\partial w_i} = \sum_{k=1}^K \frac{\partial [e(k)]^2}{\partial w_i} = \sum_{k=1}^K 2[t(k) - y(k)] \left(-\frac{\partial y(k)}{\partial w_i} \right) \quad i = 0, 1, \dots, n \quad (3.15)$$

kde pre deriváciu výstupnej hodnoty $y(k)$ platí:

$$\frac{\partial y(k)}{\partial w_i} = \frac{\partial f(u)}{\partial u} \cdot \frac{\partial u}{\partial w_i} = f'(u) \frac{\partial}{\partial w_i} \left(\sum_{j=0}^n w_j x_j \right) = f'(u) x_i \quad . \quad (3.16)$$

Teda

$$\frac{\partial E}{\partial w_i} = -2 \sum_{k=1}^K [t(k) - y(k)] f'(u(k)) x_i(k) \quad . \quad (3.17)$$

Za predpokladu, že pre chybu medzi požadovanou hodnotou a aktuálnym výstupom platí $\delta(k) = [t(k) - y(k)] f'(u(k))$, má rovnica (3.9) tvar

$$\frac{\partial E}{\partial w_i} = -2 \sum_{k=1}^K \delta(k) x_i(k) \quad . \quad (3.18)$$

Chyba $\delta(k)$ sa vtedy rovná chybe na výstupe z neurónu $e(k) = t(k) - y(k)$ násobenej deriváciou aktivačnej funkcie $f'(u(k))$ a reprezentuje tak korekciu váh w_i prislúchajúcich ku vstupným vzorom $x_i(k)$. Celková zmena Δw_i teda predstavuje sumu takýchto príspevkov cez všetky K tréningové vzory. Potom môžeme napísať vzťah pre modifikáciu váh v tvare:

$$w_i(t+1) = w_i(t) + \alpha \sum_{k=1}^K \delta(k) x_i(k) \quad . \quad (3.19)$$

Konštanta α udáva tzv. rýchlosť učenia (learning rate). Ak je aktivačná funkcia sigmoida, výraz prejde na tvar:

$$\delta(k) = \frac{\partial E}{\partial u} = [t(k) - y(k)]y(k)[1 - y(k)]. \quad (3.20)$$

Počet iterácií, po ktorých dôjde k uloženiu modifikovaných váh, sa nazýva epocha. Zložitejšia situácia nastáva pre viac vrstiev.

- Algoritmus pre viac vrstiev

Zavedieme označenie potenciálu resp. výstupu zodpovedajúcemu k -tej tréningovej vzorke a j -tému neurónu v $L-1$ vrstve v tvare $u_j^{L-1}(k)$ resp. $y_j^{L-1}(k)$. Vstupnú vrstvu označíme ako 0-tou, potom $y_j^0(k) = x_j(k)$. Výstup je privedený do i -teho neurónu v L -tej vrstve prostredníctvom synaptických váh $w_{ij}^L(t)$ alebo pre jednoduchosť označených w_{ij}^L . Pre deriváciu chyby podľa váh platí:

$$\frac{\partial E}{\partial w_{ij}^L} = -2 \sum_{k=1}^K \frac{\partial E}{\partial u_i^L(k)} \cdot \frac{\partial u_i^L(k)}{\partial w_{ij}^L} = -2 \sum_{k=1}^K \left[\delta_i^L(k) \cdot \frac{\partial}{\partial w_{ij}^L} \sum_m w_{im}^L y_{im}^{L-1}(k) \right]. \quad (3.21)$$

Teda:

$$\frac{\partial E}{\partial w_{ij}^L} = -2 \sum_{k=1}^K \delta_i^L(k) y_j^{L-1}(k). \quad (3.22)$$

Predpokladajme M neurónov vo vrstve $L+1$. Do týchto M neurónov vstupuje výstupná hodnota z i -teho neurónu $y_i^L(k)$. Chyba v L -tej vrstve a v i -tom neuróne pre k -tu vzorku je daná rovnicou:

$$\delta_i^L(k) = \frac{\partial E}{\partial u_i^L(k)} = \sum_{m=1}^M \frac{\partial E}{\partial u_m^{L+1}(k)} \cdot \frac{\partial u_m^{L+1}(k)}{\partial u_i^L(k)}, \quad (3.23)$$

$$\delta_i^L(k) = \sum_{m=1}^M \left[\delta_m^{L+1}(k) \frac{\partial}{\partial u_i^L(k)} \sum_{l=1}^L w_{mj}^L f(u_j^L(k)) \right], \quad (3.24)$$

$$\delta_i^L(k) = f'(u_i^L(k)) \sum_{m=1}^M \delta_m^{L+1}(k) w_{mi}^L. \quad (3.25)$$

Táto rovnica teda udáva vzťah pre spätné šírenie chyby z výstupnej vrstvy do vstupnej vrstvy, a to vrstvu po vrstve. Štandardný algoritmus spätného šírenia chyby nezaručuje, že bude

nájdené v priebehu tréningu naozaj globálne minimum. Väčšinou ide o lokálne minimum. Táto situácia vedie často k osciláciám pri zmene váh v priebehu tréningu a následne k jeho predčasnému ukončeniu. K vyriešeniu tohto problému môže niekedy pomôcť zmena topológie neurónovej siete [18].

3.3.2 Delta-bar-delta pravidlo

Toto pravidlo bolo navrhnuté so zámerom urýchlenia a skvalitnenia učebného procesu a je založené na štyroch základných heuristických úvahách:

- Každý parameter chybovej funkcie by mal mať vlastný parameter učenia
- Každý učiaci parameter by mal mať možnosť sa meniť v každej iterácii učenia
- V prípade, ak parciálna derivácia chybovej funkcie podľa konkrétnej synaptickej váhy má rovnaké znamienko po viacerých iteráciách, potom učiaci parameter pre konkrétnu synaptickú váhu by mal byť zvýšený
- V prípade, ak parciálna derivácia chybovej funkcie podľa konkrétnej synaptickej váhy má alternujúce znamienko po viacerých iteráciách, potom učiaci parameter pre konkrétnu synaptickú váhu by mal byť znížený

Tu je nutné si uvedomiť, že už nemáme jeden parameter γ , ale máme γ_{ij} pre každú konkrétnu synapsiu. Teda je tu snaha meniť učiaci parameter pri každej iterácii t . Takýto algoritmus sa nazýva delta-delta pravidlo, resp. jeho ďalšia modifikácia delta-bar-delta pravidlo. Kým predtým sme hovorili, že LSM využíva tzv. prístup postupného zostupu, tu LSM využíva hodnotenie momentálnej situácie na chybovom priestore v záujme urýchlenia konverencie neurónovej siete a zmenšenia chybovej funkcie siete.

Odvodenie modifikácie back-propagation učenia sa urobí v dvoch krokoch:

1. Odvodenie „delta-delta pravidla“. Tu je základom chybová funkcia, ale z prísne matematického hľadiska by sme mali rozlišovať medzi $J(t)$, kde predpokladáme konštantné γ a funkciou $\mathcal{J}(t)$, kde predpokladáme premenlivé $\gamma(t)$ resp. $\mathcal{J}(t) = fcia(\omega, \gamma)$, kým pôvodná $J(t) = fcia(\omega)$. V ďalšom budeme $\mathcal{J}(t)$ vyjadrovať pomocou $J(t)$. Matematické vyjadrenie $\mathcal{J}(t)$ a $J(t)$ je rovnaké, pri identickej výstupnej funkcii, teda

$$J(t) = 0.5 \sum_{j=1}^{N_o} (ev_j(t) - x_j(t))^2 = 0.5 \sum_{j=1}^{N_o} e_j^2(t) \quad (3.26)$$

kde $ev_j(t)$ a $x_j(t)$ sú očakávaná a vypočítaná hodnota výstupného neurónu v i -tej iterácii. Je nutné si pripomenúť, že

$$\Delta w_{ij}(t-1) = -\gamma \frac{\delta J(t-1)}{\delta w_{ij}(t-1)} \quad (3.27)$$

Túto zmenu synaptických váh použijeme vo vzťahu

$$w_{ij}(t) = w_{ij}(t-1) + \Delta w_{ij}(t-1) \quad (3.28)$$

Vzhľadom na to, že γ sa bude viazať ku konkrétnej synaptickej váhe, ale bude sa tiež meniť pre každú iteráciu. Teda ak počítame $\Delta w_{ij}(t)$, musíme uvažovať o $\gamma_{ij}(t)$, a tým dostaneme vzorec pre výpočet synaptickej váhy “ij” v iterácii “t” v tvare

$$w_{ij}(t) = w_{ij}(t-1) - \gamma_{ij}(t) \frac{\delta J(t-1)}{\delta w_{ij}(t-1)} \quad (3.29)$$

Súčasne platí, že

$$\gamma_{ij}(t) = \gamma_{ij}(t-1) + \Delta \gamma_{ij}(t) \quad (3.30)$$

Je potrebné si dobre uvedomiť indexy, ak porovnáваме rovnice (3.28) a (3.30). Súčasne môžeme vstup do neurónu “i” vyjadriť

$$in_i(t) = \sum_{j=0}^M w_{ij}(t)x_j(t) + \theta_i \quad (3.31)$$

Na základe hore uvedených rovníc sme zistili, že:

- $J(t)$ je funkciou $x_i(t)$
- $x_i(t)$ je samozrejme funkciou vstupu $in_i(t)$
- samotný vstup $in_i(t)$ je funkčne závislý cez synaptickú váhu na $\gamma_{ij}(t)$ cez $w_{ij}(t)$

Tieto poznámky nám pomôžu pochopiť nasledujúcu rovnicu. Naším globálnym cieľom je nájsť vzťah zmeny $\gamma_{ij}(t)$ v nejakej iterácii “t” v závislosti od minimalizácie chybovej funkcie $J(t)$ teda môžeme písať:

$$\frac{\partial \mathcal{J}(t)}{\partial \gamma_{ij}(t)} = \underbrace{\frac{\partial \mathcal{J}(t)}{\partial x_i(t)}}_A \underbrace{\frac{\partial x_i(t)}{\partial in_i(t)}}_B \underbrace{\frac{\partial in_i(t)}{\partial \gamma_{ij}(t)}}_C \quad (3.32)$$

Jednotlivé členy A, B, C vo vzťahu (3.32) môžeme vypočítať nasledovne:

- člen A je zrejmý z rovnice (3.26) pre chybovú funkciu

$$\frac{\partial \mathcal{J}(t)}{\partial x_i(t)} = -(ev_i - x_i(t)) = -e_i(t) \quad (3.33)$$

- člen B je zrejmý z definície aktivačnej funkcie, teda

$$\frac{\partial x_i(t)}{\partial in_i(t)} = f'(in_i(t)) \quad (3.34)$$

- člen C ak do rovnice (3.11) dosadíme zrejmu rovnicu

$$w_{ij}(t) = w_{ij}(t-1) + \Delta w_{ij}(t-1) \quad (3.35)$$

dostaneme výraz pre $in_i(t)$

$$in_i(t) = \sum_{j=1}^M \underbrace{(w_{ij}(t) - \gamma_{ij}(t) \frac{\partial \mathcal{J}(t-1)}{\partial w_{ij}(t-1)})}_{w_{ij}(t)} x_j(t) + \theta_i \quad (3.36)$$

Keď predošlú rovnicu zderivujeme podľa $\gamma_{ij}(t)$, dostaneme

$$\frac{\partial in_i(t)}{\partial \gamma_{ij}(t)} = -x_j(t) \frac{\partial \mathcal{J}(t-1)}{\partial w_{ij}(t-1)} \quad (3.37)$$

teda v konečnom dôsledku dostaneme:

$$\frac{\partial \mathcal{J}(t)}{\partial \gamma_{ij}(t)} = \underbrace{-e_i(t) f'(in_i(t))}_{A+B} - \underbrace{x_j(t) \frac{\partial \mathcal{J}(t-1)}{\partial w_{ij}(t-1)}}_C \quad (3.38)$$

Teraz pomocou $\Delta w_{ij}(t) = \gamma \delta_i(t) x_j(t)$ môžeme vyjadriť

$$\frac{\partial \mathcal{J}(t)}{\partial w_{ij}(t)} = \delta_i(t) x_j(t) \quad (3.39)$$

potom za $\delta_i(t)$ môžeme dosadiť z $\frac{\partial \mathcal{J}(t)}{\partial x_i(t)} = -(ev_i(t) - x_i(t))$ a skutočne dostaneme za $\frac{\partial \mathcal{J}(t)}{\partial w_{ij}(t)}$

výrazy A+B a teda

$$\frac{\partial \mathcal{J}(t)}{\partial \gamma_{ij}(t)} = -\frac{\partial \mathcal{J}(t)}{\partial w_{ij}(t)} \frac{\partial \mathcal{J}(t-1)}{\partial w_{ij}(t-1)} \quad (3.40)$$

a v konečnom dôsledku môžeme napísať pre zmenu $\gamma_{ij}(t)$ výraz

$$\Delta\gamma_{ij}(t) = -\varphi \frac{\partial J(t)}{\partial \gamma_{ij}(t)} = \varphi \frac{\partial J(t)}{\partial w_{ij}} \frac{\partial J(t-1)}{\partial w_{ij}(t-1)} \quad (3.41)$$

kde φ predstavuje kladnú konštantu tzv. kontrolný parameter zmeny učiaceho parametra v jednotlivých iteráciách. Hore uvedený vzťah reprezentuje tzv. delta-delta pravidlo učenia neurónovej siete. Po úvahe prichádzame k záveru, že „delta-delta pravidlo“ splňuje tretiu a štvrtú heuristickú úvahu. Problémy však nastávajú s presným určením hodnoty φ , čo predstavuje veľmi citlivý koeficient. Preto sa prišlo k ďalšej modifikácii „delta-delta pravidlo“.

2. Ďalšia modifikácia „delta-delta pravidlo“ vychádza z hore uvedeného a bola navrhnutá v tvare pre zmenu $\gamma_{ij}(t)$ v tvare

$$\Delta\gamma_{ij}(t) = \begin{cases} \kappa & \text{ak } S_{ij}(t-1)D_{ij}(t) > 0 \\ -\beta\gamma_{ij}(t) & \text{ak } S_{ij}(t-1)D_{ij}(t) < 0 \\ 0 & \text{inak} \end{cases} \quad (3.42)$$

kde S a D definujeme ako

$$S_{ij}(t-1) = (1-\zeta)D_{ij}(t-1) + \zeta S_{ij}(t-2) \quad (3.43)$$

a

$$D_{ij}(t) = \frac{\partial J(t)}{\partial w_{ij}(t)} \quad (3.44)$$

kde v rekurzívnom vzťahu pre S uvažujeme $S_{ij}(0) = 0$. Hore uvedené vzťahy považujeme za „delta-bar-delta pravidlo“ učenia. Podobnosť „delta-bar-delta pravidlo“ s „delta-delta rule“ je vyjadrené vo vzťahu (3.42), kde vlastne ovplyvňujeme výpočet $\Delta\gamma_{ij}(t)$ iteráciou t a $t-1$. Súčasne sa naplňujú požiadavky heuristickej úvahy číslo jedna a dva, pretože každý prípad vo vzťahu (3.42) má svoj vlastný koeficient κ a β . Je zrejmé, že celý proces adaptácie samotného γ začne v $t = 2$.

Vzhľadom na dosť komplikovaný výpočet je nutné uvažovať o zmene iteračného prístupu. Kým v obyčajnom „back-propagation of error“ za iteráciu považujeme každý výstup a stále po ňom urobíme zmeny synaptických váh v neurónovej sieti, teraz pod iteráciou budeme rozumieť epochu $B, B > 0$ -vstupov do neurónovej siete a až potom urobíme zmenu synaptických váh. Tomuto prístupu hovoríme dávkové učenie (Batch learning). Potom

$$\frac{\partial J(t)}{\partial w_{ij}(t)} = - \sum_{k=1}^B \delta_i^k(t) x_j^k(t) = D_{ij}(t) \quad (3.45)$$

A teda zmena synaptických váh potom bude mať tvar

$$\Delta w_{ij}(t) = \alpha \Delta w_{ij}(t-1) + \gamma_{ij}(t) \sum_{k=1}^B \delta_i^k(t) x_j^k(t) \quad (3.46)$$

A nová hodnota synaptickej váhy bude vypočítaná podľa obvyklého vzťahu

$$w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t) \quad (3.47)$$

Na záver pripomeniem, že pre praktický výpočet musíme pre každú $w_{ij}(t)$ najprv vypočítať príslušný parameter $\gamma_{ij}(t)$. Je nutné poznamenať, že jednotlivé derivácie na konci vzťahu (3.41) sú v každom prípade spočítavané aj pri klasickom „back-propagation of error“. Teda tu ide iba o výpočet novej hodnoty γ_{ij} z jednotlivých derivácií, ktoré tak či tak by sme v klasickom prístupe „back-propagation of error“ vypočítali. Z toho vyplýva, že z výpočtového hľadiska nejde o výpočtové úkony navyše. Negatívom však je, že pre modifikovaný „back-propagation of error“ založený na delta-bar-delta máme až 5 parametrov a to

- $\gamma_{ij}(0)$ - štartovacie γ
- koeficient α pre výpočet $\Delta w_{ij}(t)$
- koeficienty κ, β, ς pre výpočet $\Delta \gamma_{ij}(t)$

Tento fakt sťažuje použitie tejto metódy, hoci na druhej strane predstavuje systém, ktorý je možné voľnejšie riadiť [1].

4. Evolučné stratégie

Evolučná stratégia patrí historicky medzi prvé úspešné stochastické algoritmy. Bola navrhnutá už počiatkom 60-tych rokov Rechenbergom a Schwefelom. Evolučná stratégia vychádza zo všeobecných predstáv prirodzeného výberu, avšak omnoho vágnejších ako napríklad pri genetickom algoritme. Na rozdiel od väčšiny ostatných stochastických optimalizačných metód nie je evolučná stratégia založená na binárnej reprezentácii premenných, ale manipuluje priamo s reálnou reprezentáciou premenných [21].

Evolučné stratégie patria v súčasnosti medzi dobre rozvinuté stochastické metódy. Prvé evolučné stratégie používali jediný rekombinačný operátor – mutáciu a iba jedno riešenie

v populácií s maximálne dvoma jedincami uchovávanými súčasne – rodič a navrhovaný potomok. Takýto prístup sa nazýva dvojčlenná evolučná stratégia alebo (1+1)ES.

Základom evolučnej stratégie je nasledujúci predpis, ktorý mutuje aktuálne riešenie $\vec{\alpha}$ na nové $\vec{\alpha}^{new}$ nasledujúcim spôsobom $\vec{\alpha}^{new} = \vec{\alpha} + \vec{N}(0, \sigma)$, kde $\vec{N}(0, \sigma)$ je vektor nezávislých náhodných čísel s nulovou strednou hodnotou a štandardnou odchýlkou σ . Jednotlivé čísla sú náhodne rozložené podľa gaussovskej distribúcie, to znamená, že menšie čísla sa objavujú pravdepodobnejšie ako väčšie čísla. Štandardná odchýlka je mierou rozptylu generovaných hodnôt, mierou variácie náhodnej premennej. Určuje šírku dvojrozmerného „kopčeka pravdepodobnosti“, ktorého základňu tvorí os možných hodnôt vygenerovaných čísel a ktorého výška v danom bode odpovedá pravdepodobnosti, s akou bolo číslo vygenerované. Tento „kopček pravdepodobnosti“ sa nazýva funkcia hustoty rozdelenia náhodnej premennej a pre normálnu gaussovskú distribúciu je opísaný ako:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-0.5(x-\mu)^2/\sigma^2} \quad (4.1)$$

kde μ je stredná hodnota premennej x a σ je štandardná odchýlka. Väčšia štandardná odchýlka zväčšuje (v priemere z viacerých pokusov) vzdialenosť medzi rodičom a vygenerovaným potomkom [21].

4.1 Mnohočlenné stratégie a kovariancie

Schwefel a jeho spolupracovníci navrhli ďalšie sofistikovanejšie verzie evolučnej stratégie, takže v súčasnosti už možno hovoriť o celej triede evolučných stratégií. Odpadá tu nutnosť zavádzania pravidiel pre zmeny štandardnej odchýlky, čo mohlo viesť k predčasnému ukončeniu optimalizácie.

Schwefel novšie zaviedol mnohočlenné evolučné stratégie, označované (m + l) alebo (m,l), ktoré imitujú ďalšie základné princípy organickej evolúcie. Pracuje sa tu potom s celým súborom rodičovských vektorov $\mathbf{x}$, ktorých počet je m . V prípade stratégie *PLUS* (m + l) vygenerujeme z týchto rodičov / potomkov. Potom takéto navrhnuté vektory – potomkov a pôvodné „rodičovské“ vektory - zoradíme spolu podľa optimálnosti riešení a prvých m najlepších jedincov – riešení zoberieme ako rodičov do ďalšej populácie. V prípade stratégie *ČIARKA* (m,l) vyberáme budúcich m rodičov iba z práve vygenerovaných/ potomkov a rodičia týchto potomkov vymierajú. Život každého jedinca je obmedzený iba na jednu generáciu. Okrem mutácie sa pri oboch stratégiách používa kríženie, teda čiastočná výmena informácií medzi vektormi reálnych čísel, na rozdiel od kríženia binárnych reťazcov

používaného pri genetických algoritmoch. Kríženie dvoch alebo viacerých jedincov vždy produkuje jedného potomka na rodieľ od genetických algoritmov, kedy dvaja rodičia generujú vždy dvoch potomkov. Rovnako ako pri genetických algoritmoch sa chromozóm môže vybrať ako rodič aj niekoľkokrát [21].

Najväčším „objavom“ evolučnej stratégie je však možnosť nebrať jedinú hodnotu štandardnej smerodajnej odchýlky pre všetky hodnoty optimalizovaného vektora reálnych čísel x , ale uchovávať pre každú premennú z vektora x vlastnú hodnotu smerodajnej odchýlky (rozptyl). Každý jedinec je teda tvorený dvoma vektormi: vektorom x a vektorom smerodajných odchýlok σ . Nesie si tak so sebou aj informáciu o distribúcii pre novú mutáciu. Mutácie sa všeobecne robia rovnako ako pri stratégii (1+1) s pomocou štatistického rozdelenia tak, že drobné zmeny sú pravdepodobnejšie a závažné zmeny nepravdepodobné. Táto tendencia sa časom zvyšuje spolu s približovaním populácie k optimu, teda blízko optima sú mutácie už len malé. Ale ako poznať, že sme už blízko optima? Mnohočlenné evolučné stratégie túto otázku obchádzajú meta-evolučnou samoadaptačnou technikou, kedy sa aj veľkosť mutácií v priebehu algoritmu mení. Evolúcia teda nebude prebiehať iba pri premenných určujúcich riešenie, ale súčasne sa bude vyvíjať informácia o tom, ako každé z riešení bude generovať nového potomka. Táto informácia je uložená v tzv. strategických premenných. Keď zmena aktuálnej hodnoty jednej premennej veľmi nemení zodpovedajúcu hodnotu optimalizovanej funkcie, potom vývoj hodnoty smerodajnej odchýlky priradenej tejto premennej bude smerovať k väčším číslam. Teda zmeny majú inú mierku pri premenných, ktoré majú väčší vplyv na hodnotu funkcie ako iné premenné. Každý nový jedinec je ohodnotený účelovou funkciou. Učenie prebieha na dvoch úrovniach, jedna úroveň hodnotí dosiahnuté výsledky funkcie, druhá úroveň predstavuje schopnosť jedinca učiť sa zakódovanú v smerodajnej odchýlke určujúcej veľkosť mutácie (aj keď táto schopnosť sa zhodnocuje len nepriamo pomocou ohodnotení výsledkov funkcie).

Pri použití smerodajnej odchýlky pre každú premennú poskytuje väčšinou lepšie výsledky „čiarková“ stratégia, kde populácia rodičov celkom vymiera. Aj keď momentálne môže dojsť k zabudnutiu najlepšieho doteraz nájdeného riešenia, v dlhodobom meradle táto stratégia funguje lepšie, lebo sa dostane cez lokálne minimá, kedy nejaké riešenie má síce momentálne vysokú hodnotu optimalizovanej funkcie, ale nastavenie smerodajných odchýlok je zlé. Zabúdnutie rodičovských chromozómov preferuje vybudovanie vnútornej štruktúry – teda smerodajných odchýlok – optimálne nastavených na ďalší vývoj. Takýto prístup potom funguje lepšie pre optimalizáciu zašumených alebo časovo premenných funkcií.

Rodičia pre kríženie sa vyberajú z celej populácie s rovnakou pravdepodobnosťou, nezáleží teda na optimálnosti riešenia (konvergencia je dostatočne zabezpečená selekciou potomkov pre ďalšiu generáciu rodičov). Pre kríženie sa väčšinou používa „diskrétné kríženie“, známe z analógie s genetickými algoritmi, kde potomok postupne postupne preberá hodnoty z jedného alebo druhého rodiča s rovnakou pravdepodobnosťou. Kríženie teraz už zahŕňa nielen výmenu hodnôt vektora $\mathbf{x}$, ale aj hodnôt vektora zodpovedajúcich smerodajných odchýlok σ .

Ďalším z možných spôsobov kríženia je kríženie priemerom, ktoré však často vedie k príliš rýchlej lokálnej konvergencii.

Najnovšie sa objavili ďalšie druhy kríženia, ktoré sa použili tak v genetických algoritmoch, ako aj v evolučnej stratégii. Dva vektory $\mathbf{x}_1$ a $\mathbf{x}_2$ produkujú dvoch potomkov $\mathbf{y}_1$ a $\mathbf{y}_2$, ktoré sú lineárnou kombináciou svojich rodičov

$$\mathbf{y}_1 = a\mathbf{x}_1 + (1 - a)\mathbf{x}_2 \quad (4.2)$$

$$\mathbf{y}_2 = (1 - a)\mathbf{x}_1 + a\mathbf{x}_2 \quad (4.3)$$

Okrem tohto kríženia sa ešte navrhlo diskrétné globálne uniformné kríženie, ktoré si hodnoty vektorov $\mathbf{x}$ a σ nevyberá postupne náhodne z dvoch vektorov, no môže si vyberať zo všetkých rodičovských vektorov. Obdobou „kríženia priemerom“ pre globálne uniformné kríženie je použitie hodnôt jedného rodiča pre všetky premenné, zatiaľ čo druhý rodič sa vyberá pre každú premennú zvlášť [21].

5. Predikcia

Predikcia spočíva v tom, že vieme nájsť systém, ktorý modeluje správanie sa skúmaného systému v referenčnej budúcnosti na základe informácií z referenčnej minulosti, ale dokonca aj prítomnosti. Ak hľadáme hodnoty parametrov vektora $\bar{\mathbf{y}}$ v čase $t > t^*$, na vstup do systému musia prísť vzorky parametrov $\bar{\mathbf{x}}$ v čase $t \leq t^*$.

Môžeme uvažovať takéto možnosti:

1. Na vstupe je $\bar{\mathbf{x}}(t, \dots, t - \tau_i) \Rightarrow$ na výstupe $\bar{\mathbf{x}}(t + 1, \dots, t + \tau_j)$, $\bar{\mathbf{x}}(t) = \bar{\mathbf{y}}(t)$
2. Na vstupe je $\bar{\mathbf{x}}(t, \dots, t - \tau_i) \Rightarrow$ na výstupe $\bar{\mathbf{x}}(t + 1, \dots, t + \tau_j)$, $\bar{\mathbf{x}}(t) \neq \bar{\mathbf{y}}(t)$

3. Na vstupe je $\bar{x}(t, \dots, t - \tau_i) \Rightarrow$ na výstupe $\bar{x}(t + 1, \dots, t + \tau_j)$, ale pre hľadaný vektor $\bar{y}(t)$ a známy vektor $\bar{x}(t)$, $\bar{y} \subset \bar{x}$

To nám hovorí, že buď predikujeme hodnoty $\bar{x}$ na základe $\bar{x}$, alebo hodnoty iného vektora $\bar{y}$ na základe $\bar{x}$, prípadne $\bar{y}$ môže byť podvektor vektora $\bar{x}$.

Systémy chápeme ako :

- otvorené (neplatí Zákon zachovania energie) - Ak je systém otvorený, dokážeme povedať, že sa vyznačuje veľkým množstvom vstupných veličín, ktoré však pri experimentoch veľmi ťažko popisujeme. Dokonca z rôznych možných dôvodov je problematické, dokonca i nemožné namerať a tak vytvoriť databázu historického vývoja veličín, ktorá je nutná pre experimentálnu identifikáciu.
- uzavreté (platí Zákon zachovania energie) - Ak je systém uzavretý, vieme povedať, že existuje aj jeho deterministický model. Uzavretý systém má obrovské množstvo vstupných aj stavových premenných, zúčastňujúcich sa na dejoch, ktoré prebiehajú, ale je ich možné odmerať alebo aj zanedbať pri zachovaní postačujúcej presnosti.

Cieľom predikcie je presne predpovedať krátkodobý vývoj správania sa systému.

Cieľom modelovania je nájdenie takého popisu systému, ktorý úplne vystihuje vlastnosti správania sa systému v dlhodobom horizonte. Predikcia a modelovanie nie sú však vo svojej podstate identické, pretože nájdenie spoľahlivého matematického modelu, ktorý popisuje model z dlhodobého hľadiska, nemusí byť vhodný pre krátkodobú periódu. Na druhej strane model, ktorý je použiteľný pre krátkodobú predikciu, môže mať nevhodné dlhodobé vlastnosti.

Cieľom charakteristiky systému je snaha o determinovanie fundamentálnych vlastností systému takých, ako napríklad stupeň voľnosti systému. Tento cieľ sa prekrýva s predikciou, no je tu taký rozdiel, že zložitosť modelu použiteľného pre predikciu nemusí súvisieť so skutočnou zložitosťou systému. Lineárne modely časových radov majú dve partikulárne vlastnosti: môžu byť pochopiteľné do hlbokých detailov a môžu byť priamo implementované. Negatívum vlastností je však to, že pri zložitých systémoch s nimi nemôžeme uvažovať [3].

5.1 Princípy problematiky predikcie

Problematiku predikcie pri študovaní i aplikovaní možno rozdeliť nasledovne:

5.1.1 Problematika vstupov a výstupov pre konštrukciu modelu systému a samotný návrh modelu (neurónovej siete).

Výstupy sa určia podľa toho, ktoré vlastnosti systému chceme sledovať, resp. ktoré parametre systému sú ako výsledok predikcie pre nás zaujímavé. Vstupy do systému sú však zložitejší pojem.

Prvá možnosť spočíva v tom, že výber vstupov určíme na základe posudku experta a model v takom prípade by mal postihnúť korelácie medzi jednotlivými vstupmi, ktoré môžu byť často aj kvalitatívnej povahy a môžu nadobúdať rôzne rozsahy hodnôt.

Druhá možnosť spočíva v tom, že na sledované hodnoty x sa pozrieme ako na časový rad, resp. ako na diskretnú funkciu. Povieme, že keď na výstupe máme vektor $x(t^*)$, vstup má rovnaký charakter, teda tiež vektor hodnôt x avšak v čase $t > t^*$.

V prvom aj druhom prípade je možné na vstup dať hodnoty z predchádzajúceho kroku ako sú na výstupe, čo je ten najjednoduchší prípad. Ak sieť však nemá schopnosť "vnímať" čas, je účelné vstupy rozšíriť aj o hodnoty zo vzdialenejšej referenčnej minulosti.

Výber metódy, resp. druhu modelu je závislý na definovaných vstupných a výstupných veličinách, na ich počte a charaktere. Je však dôležité, aby sme zohľadnili ich počet a priebeh v čase. Niektoré systémy sa vedia vyznačovať lepšou kvalitou, ak je priebeh ustálený s malými výchylkami a niektoré však vedia vykazovať lepšie vlastnosti, keď sú väčšie zmeny hodnoty. V niektorých prípadoch sme obmedzení počtom vstupov, resp. výstupov, na druhej strane, iné systémy lepšie riešia korelácie medzi vstupmi, resp. výstupmi [3].

5.1.2 Adaptácia siete

Táto časť je závislá od predchádzajúceho bodu. Adaptácia je proces, kedy hľadáme vhodný model systému, ktorý vykazuje čo najlepšie vlastnosti vzhľadom na vstupy a výstupy, ktoré sme mali zadefinované, t.j. znižovanie chyby medzi referenčnými výstupmi a vstupmi siete [3].

5.1.3 Extrakcia znalostí z adaptovaného modelu

Vieme predpokladať, že existuje model systému alebo jeho podsystémov, buď ako matematický model alebo ako databáza pravidiel, prípadne fuzzy pravidiel, ktoré popisujú vzťahy, resp. prebiehajúce deje v systéme. Môžeme povedať, že existuje priradenie $I \rightarrow O$, kde I je množina vstupov a O je množina výstupov. Extrakcia znalostí je akékoľvek konkrétne vyjadrenie tohto vzťahu, no lepšie však je, ak je to vyjadrenie čo najjednoduchšie hlavne kvôli pochopeniu človekom [3].

5.2 Časové rady

Pri predikcii ide o to, aby sme zistili čo najpresnejší odhad správania sa systému v budúcnosti na základe dát jeho správania sa v minulosti. Keďže je prevažná väčšina procesov nelineárna, na ich modelovanie je často potrebné použiť nelineárne parametrické modely, medzi ktoré patria i neurónové siete, ktoré sú svojou možnosťou adaptovania sa a vlastnosťou univerzálneho aproximátora funkcií, na podobné problémy teoreticky pripravené. Skúmané dáta berieme vo všeobecnosti ako stavy skúmaného systému.

Najťažší prípad predikcie je predikcia iba na základe výstupu systému, pretože máme najmenší počet informácií potrebný k modelovaniu tohto systému. Lepší prípad nastane vtedy, keď sú k dispozícii historické údaje viacerých stavov systému a hlavne jeho vstupov, keď ho nemôžeme ani zhruba považovať za autonómny.

Keď máme k dispozícii iba výstup a nemáme žiadny vstup, ide o problém predikcie časových radov alebo to môžeme nazvať *extrapolácia funkcie*. Potrebujeme nájsť takú funkciu f , aby platilo:

$$x(t+1) = f(x(t), x(t-1), \dots, x(t-n)) , \quad (5.1)$$

kde $x(t)$ je skúmaná veličina v čase t a n je vlastný rozmer procesu.

Na to, aby sme riešili predikčné úlohy pomocou neurónových sietí, používame metódy kontrolovaného učenia (supervised learning) a temporal difference learning, ktoré sa hlavne využívajú v reinforcement learning na predikciu hodnotiacej funkcie a ktoré patrí na pomedzie medzi supervised a unsupervised learning. Čo sa týka použitia topológií sietí, možno vo všeobecnosti použiť každú doprednú (feed-forward) sieť, kde na vstupy týchto sietí poskytujeme historické hodnoty procesu a na výstup poskytujeme jeho budúce hodnoty.

Pre predikciu je dôležité, aby si daná metóda počas adaptovania sa na trénovacie dáta zachovala schopnosť čo najväčšej generalizácie a tak sa vyhla problému preučenia

(*overfitting*). Dosiahneme to tak, že často kontrolujeme generalizáciu pomocou *cross-validation*, kde schopnosť zovšeobecňovať kontrolujeme pomocou sietí neznámych dát alebo dynamickým výberom takých učiacich vzoriek, ktoré túto schopnosť nenarušia [11, 19].

5.2.1 Analýza časových radov

Analýza časových radov má tri základné ciele: predikcia, modelovanie a charakterizácia. Súbor dát je chronologicky usporiadaný s časovou periódou obyčajne týždennou, mesačnou, štvrťročnou alebo ročnou. Cieľom analýzy časových dát je konštrukcia vhodného modelu, pomocou ktorého manažment môže na základe získaných dát z minulosti robiť predpovede na určité obdobia do budúcnosti [11, 19].

5.2.2 Dekompozícia časových radov

Časové rady vznikajú ako dôsledok pôsobenia tak podstatných, ako aj nepodstatných činiteľov. Zvyčajne sledujeme vplyvy, ktoré označujeme ako:

- Vývojové (trendové), ktoré pôsobia neustále, dlhodobo a určujú hlavný smer vývoja, trend časového radu
- Periodické, môžu striedavo pôsobiť na rast alebo pokles hodnôt v časovom rade.
Uvažujeme:
 - Sezónnu zložku – ktorá popisuje periodicky sa opakujúce zmeny v časovom rade odohrávajúca sa v priebehu jedného kalendárneho roka alebo sezóny
 - Cyklickú zložku – ktorá popisuje periodické zmeny dlhšie ako jeden rok
- Náhodné, ktoré pôsobia náhodnými fluktuáciami v priebehu časového radu, ale aj chybami, ktorých sa dopustil štatistik pri spracovaní dát (chyby meraní, zaokrúhľovanie, editovanie, drobné nárazy a pod.) [3, 20].

5.2.3 Trend

Určenie trendu spočíva v procese nahradenia časového radu pozorovaním hodnôt radom teoretických hodnôt bez periodického a náhodného kĺzania. Získaný rad teoretických hodnôt nazývame vyrovnaním časového radu. Ak máme graficky zobrazené pozorované hodnoty, vyrovnanie môžeme urobiť „od oka“ tak, aby čiara (ktorou nahradíme pozorované hodnoty) čo najlepšie vystihla pôvodný priebeh. Táto metóda je zrejme informatívna [3, 20].

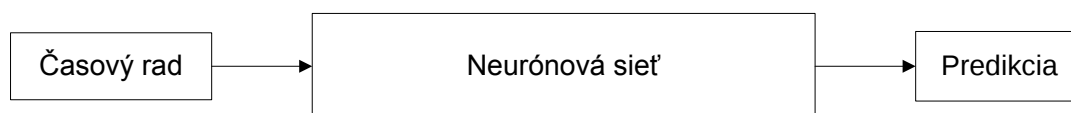
Často používané metódy vyrovnávania časových radov sú:

- mechanická (metóda klzavých priemerov)
- analytická (metóda najmenších štvorcov)

5.2.4 Predpovede v časových radoch

Konštrukcia predpovedí je jednou z najdôležitejších úloh analýzy časových radov.

- Bodová predpoveď je určenie hodnoty, ktorá predstavuje náš odhad budúcej hodnoty uvažovaného časového radu (t.j. hodnoty v konkrétne stanovenom okamžiku). Je samozrejmé, že takto stanovená hodnota môže byť zaťažená určitou chybou.
- Intervalová predpoveď udáva dolnú a hornú hranicu, medzi ktorými bude ležať budúca prognózovaná hodnota s istou pre pravdepodobnosťou. Táto predpoveď je v matematickej štatistike nazývaná intervalom spoľahlivosti [3, 20].

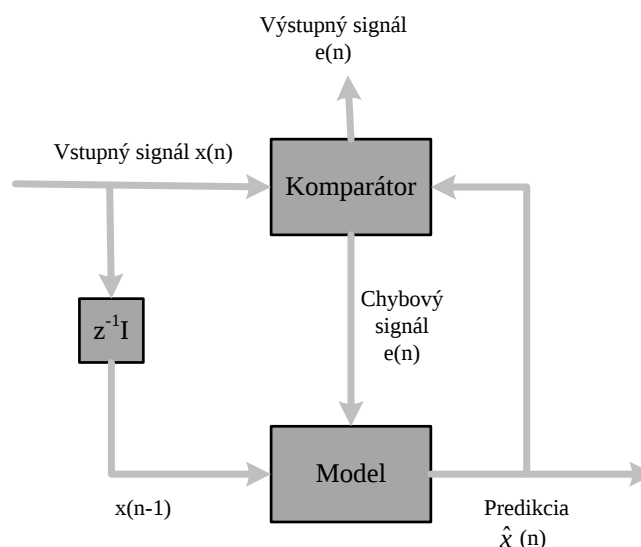


Obrázok 11: Predikcia časového radu neurónovou sieťou.

5.3 Prediktor na báze umelých neurónových sietí

Jednou zo základných výhod použitia systémov postavených na báze neurónových sietí je ich schopnosť učiť sa nové znalosti a tiež ich schopnosť adaptácie. Pred použitím musí systém prejsť procesom učenia, v ktorom je naučený dôležitým štatistickým vlastnostiam prostredia, v ktorom bude pôsobiť [12].

5.3.1 Štruktúra prediktora



Obrázok 12: Blokový diagram predikčného systému.

Obrázok 12 nám ukazuje koncepciu neurónového systému v ktorom blok označený ako *Model* pracuje ako prediktor používajúci predchádzajúce skúsenosti získané v minulosti na to, aby predpovedal, čo sa s najväčšou pravdepodobnosťou v nasledujúcom kroku udeje. Blok označený $z^{-1}I$ sa nazýva časový oneskorovací člen, ktorého úlohou je priviesť na vstup predikčného modulu časovo oneskorený vstupný vektor. Z detailnejšieho pohľadu ide o proces, v ktorom prediktor na základe minulých hodnôt vstupného signálu označených vektorom $x(n-1)$ a parametrov siete vypočítaných v čase $(n-1)$ vykonáva odhad $\hat{x}(n)$ skutočného vektora $x(n)$ v čase n . Vo väčšine prípadov je však predikovaná hodnota (reprezentujúca „skúsenosti“ prediktora) rôzna od skutočnej hodnoty (reprezentujúcej očakávanie), a preto následne blok označený ako *Komparátor* vypočíta odchýlku skutočnej hodnoty $x(n)$ od jej odhadu $\hat{x}(n)$. Vypočítaný chybový signál označovaný vektorom $e(n)$, nazývaný taktiež inovačný signál, reprezentuje novú informáciu obsiahnutú v aktuálnom vstupnom signáli $x(n)$ v čase výpočtu.

V prípade, že je signál $e(n)$ nulový, vstupný signál neobsahuje žiadnu novú informáciu, čo znamená, že model správne odhadol výstupný signál, a preto nie je potrebný žiaden zásah do aktuálneho modelu procesu.

Ak je signál $e(n)$ nenulový, čo je dôsledok výskytu neočakávanej udalosti (novej informácie), model by mal byť upravený tak, aby v ďalšom kroku zahŕňal aj nové znalosti, t.j. systém by sa mal na nové poznatky adaptovať.

Inovačný signál, získaný z bloku Komparátor, môže byť následne využitý dvoma spôsobmi:

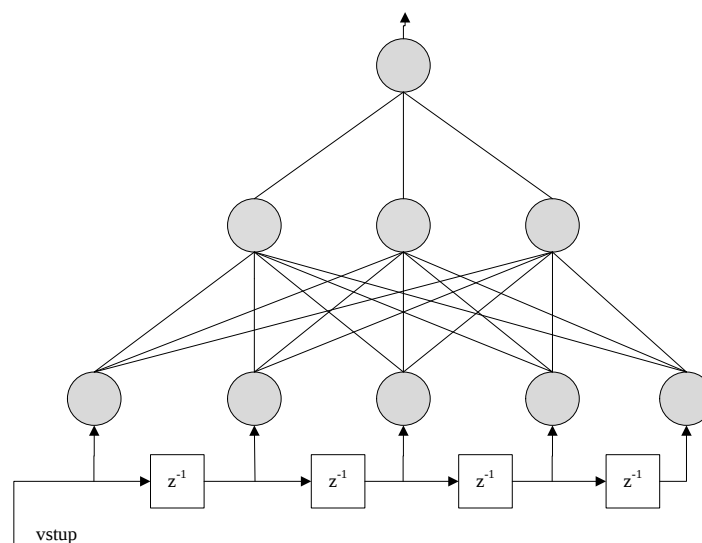
- Signál $e(n)$ plní funkciu korekčného signálu pôsobiaceho na model a spôsobujúceho následnú korekciu stupňov voľnosti siete (t.j. synaptických váh), čím sa model procesu prispôsobí zmenám okolia. Ako vidno z Obrázku 12, táto funkcia je zabezpečená spätnou väzbou od komparátora k modelu.
- Signál $e(n)$ je jedným z výstupov systému a je k dispozícii ďalej (vyššej) úrovni systému na vyhodnotenie a spracovanie. Opakovaním tejto operácie v systéme postavenom na báze viacúrovňového spracovania signálu by mala byť na každej z úrovní spracovávaná informácia, ktorú už nie je možné spracovať na úrovni nižšej. Celková informácia získaná týmto spôsobom by mala byť potom väčšia ako len pri spracovaní na predchádzajúcom stupni [12].

5.3.2 Realizácia prediktora

Najjednoduchší spôsob, akým môžeme realizovať prediktor predikujúci nasledujúci krok, je použiť Back-Propagation sieť s upraveným vstupom a vytvoriť tzv. časové oneskorenie na vstupe doprednej neurónovej siete. Topológia takejto siete je na Obrázku 13.

- Jednokroková predikcia

Ide o vytvorenie časového okna nad vstupmi (údajmi), ktorého šírka predstavuje samotné časové oneskorenie (za predpokladu ekvidistančného vzorkovania vstupných dát). Tým dostávame prediktor mapujúci vstupnú sekvenciu dát na výstup, ktorým je odhad hodnoty pre ďalší krok. Toto bola rozobratá jednokroková predikcia.

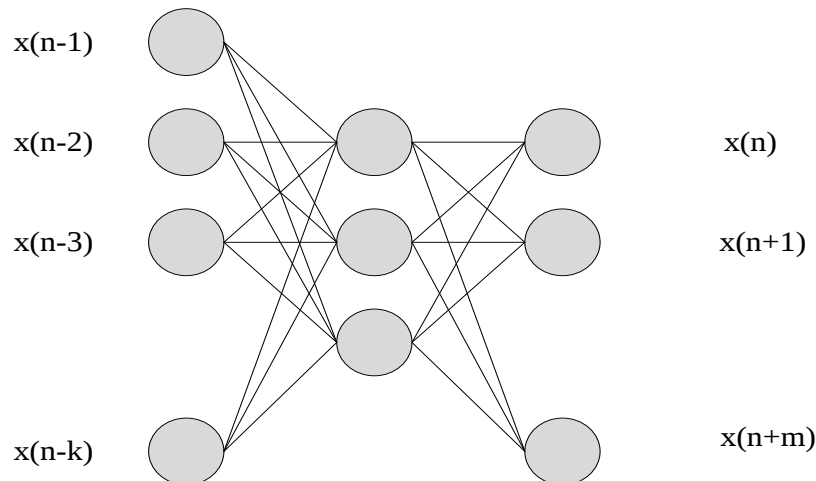


Obrázok 13: Time-delay časového signálu na vstupe BP siete.

- Viackroková predikcia

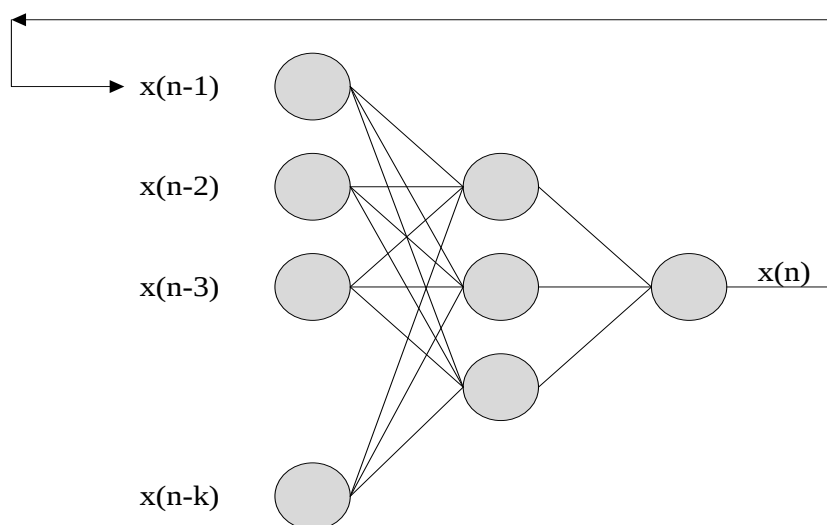
Existujú dva prístupy ako môžeme dosiahnuť viackrokovú predikciu:

Prvým je prediktor (Obrázok 14), u ktorého je viackrokový predikčný mechanizmus priamo implementovaný do topológie siete t.j. zvýši sa počet výstupných neurónov.



Obrázok 14: Prediktor s viackrokovým predikčným mechanizmom implementovaným do topológie siete.

Druhým je využitie krátkodobého prediktora na viackrokovú predikciu (Obrázok 15), u ktorého tzv. iterovanou predikciou vieme dosiahnuť ľubovoľne veľký časový horizont predikcie. Tento spôsob predikcie nazývame taktiež rekurzívna predikcia.



Obrázok 15: Využitie krátkodobého prediktora na viackrokovú predikciu.

Oba z týchto prístupov majú svoje výhody aj nevýhody. Aj keď u prvého prístupu je presnosť predikcie na viac krokov dopredu presnejšia, zväčšenie horizontu predikcie si vyžaduje zmenu topológie celej siete, a teda aj jej pretrénovanie. Naopak pri druhom prístupe je možné horizont predpovedí predlžovať ľubovoľne, ale vo väčšine prípadov za cenu rapidného zníženia presnosti predikcie so zväčšujúcim sa horizontom [12].

6. Experimenty

Experimenty mali za cieľ overiť predikčné vlastnosti systému, ktorý bol postavený na báze dopredných neurónových sietí s učiacim algoritmom so spätným šírením chyby, nazývaným aj back-propagation of error algoritmus, s jeho modifikáciou delta-bar-delta a evolučnou stratégiou. V ďalších častiach budeme vidieť, že dôraz nebol kladený iba na modifikácie v architektúre neurónovej siete, ale práve aj na modifikácie samotného učiaceho algoritmu.

Štandardná neurónová sieť so spätným šírením chyby nedokáže vnímať časový kontext priamo. Práve kvôli tomu, že nedokáže vnímať časový kontext priamo, musíme vytvoriť časové okno na vstupe do neurónovej siete. Viac informácií o časovom okne sme sa mohli dočítať v predchádzajúcich kapitolách.

6.1 Návrh experimentov

Pri návrhu experimentov sme v experimentálnej časti kládli dôraz predovšetkým na tieto oblasti:

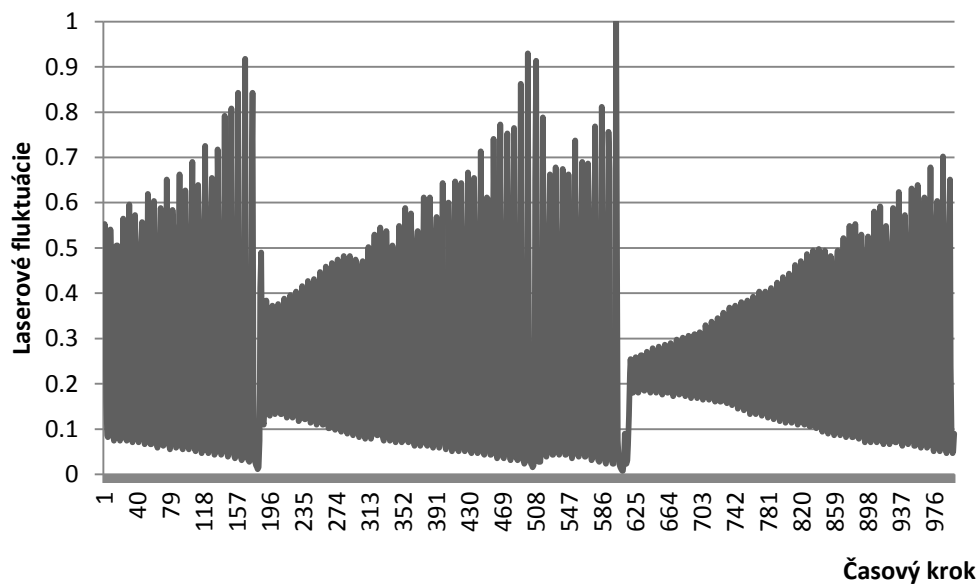
- Identifikácia učiacich parametrov a optimálnej topológie umelej neurónovej siete so spätným šírením chyby.
- Porovnanie dosiahnutých výsledkov medzi jednotlivými modifikáciami či už architektúry alebo učiaceho algoritmu.

Chceli sme overiť kvalitu a na overenie sme použili benchmarkovské dáta, ktoré boli v oblasti predikcie veľmi veľakrát použité a kvalitne overené.

Práve kvôli tomuto dôvodu sme vybrali dáta zo *Santa Fe Institute Time Series Prediction and Analysis Competition* súťaže, ktorá bola vyhlásená v r. 1991 a jej cieľom bolo porovnať rôzne predikčné systémy na údajoch z rôznych oblastí. V tejto súťaži bolo k dispozícii viacero druhov údajov a pre experimenty bola z nich vybratá tzv. sada A, ktorá obsahuje časový

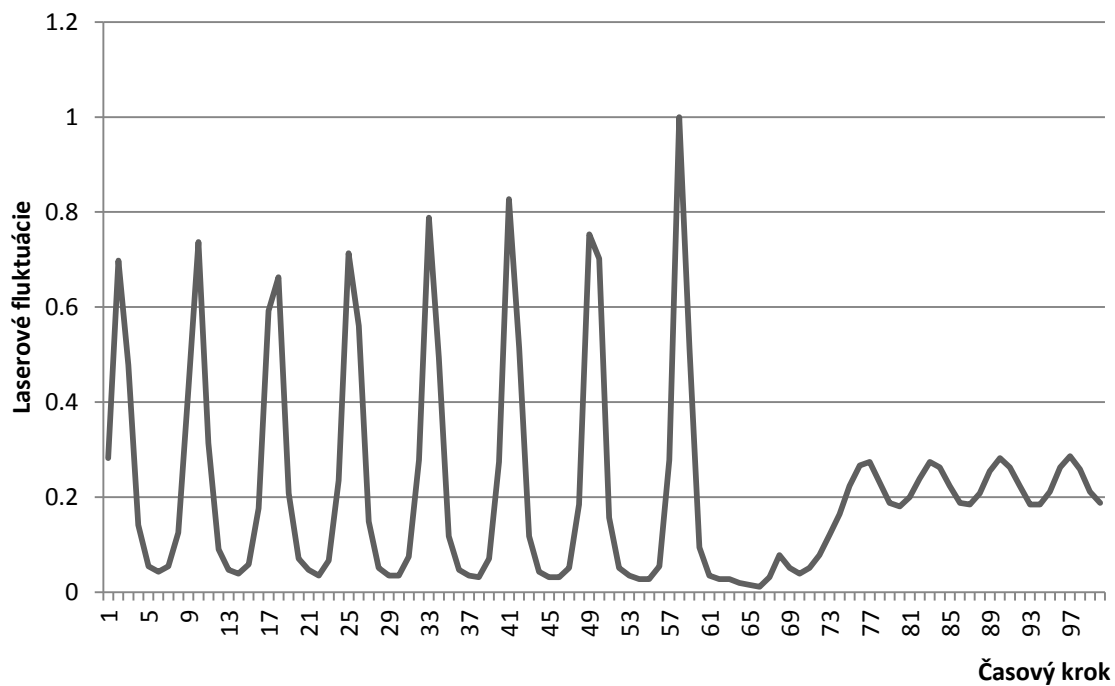
záznam údajov meraných pri fyzikálnom pokuse. Jedná sa o záznam fluktuácií NH_3 Far-Infrared lasera.

Trénovacia množina



Obrázok 16: Trénovacia laserová postupnosť.

Testovacia množina



Obrázok 17: Testovacia laserová postupnosť.

Dané dáta sú veľmi dobrým názorným príkladom správania sa, ktoré vidíme v stacionárnom, nízko rozmernom fyzikálnom systéme, kde sú známe rovnice dynamiky. Údaje sú nezašumené, resp. považujeme ich za nezašumené, keďže sú to veľmi presné laboratórne merania.

Trénovacia množina týchto dát obsahuje 1000 vzoriek a testovaciu množinu predstavuje 100 nasledujúcich vzoriek v časovom rade. Rozpätie údajov sa pohybovalo od 2 do 255. V týchto laserových údajoch sa nenachádzajú žiadne chýbajúce, resp. prázdne hodnoty a ani netypické záznamy. Tak ako trénovacia, tak aj testovacia množina je kompletná, bez potreby dodatočného upravovania dát. Jedinú úpravu predstavovala transformácia údajov do intervalu $<0,1>$ z dôvodu spracovania týchto údajov neurónovou sieťou.

6.1.1 Štruktúra experimentov

. Experimenty v tejto práci sú rozdelené do viacerých skupín podľa rôznych hľadísk.

- Podľa typu algoritmu:
 1. back-propagation of error (algoritmus spätného šírenia chyby)
 2. delta-bar-delta pravidlo
 3. evolučná stratégia
- Podľa spôsobu zmeny synaptických váh:
 1. iteratívny prístup
 2. dávkový prístup
- Podľa typu prediktora:
 1. prediktor implementovaný do topológie neurónovej siete
 2. použitie krátkodobého prediktora na viackrokovú predikciu
- Podľa účelu predikcie:
 1. experimenty na získanie optimálnej topológie
 2. experimenty na získanie najlepších výsledkov

6.1.2 Spôsob hodnotenia výsledkov

Kvalita predikcie sa vyhodnocovala podľa najčastejšie sa vyskytujúcich ukazovateľov.

Zadefinovali sa:

- N – počet jednotlivých vzoriek použitých údajov
- d_i – požadovaný výstup neurónovej siete pre i -tú vzorku

- x_i – skutočný výstup neurónovej siete pre i -tú vzorku

Ďalej bolo potrebné zadať si kritéria hodnotenia chyby predikcie:

- MAPE – Priemerná absolútna percentuálna chyba
- MSE – Stredná kvadratická chyba
- MAO – Maximálna absolútna odchýlka

Chyba MAPE je obzvlášť často používaná kvôli svojej zrozumiteľnosti, jednoduchosti a prehľadnosti, keďže je udávaná v percentách.

$$MAPE = \frac{1}{N} \sum_{i=1}^N |d_i - x_i| * 100 (\%) . \quad (6.1)$$

MSE patrí medzi najčastejšie používané ukazovatele a to nie len v oblasti predikcie časových radov.

$$MSE = \frac{1}{N} \sum_{i=1}^N |d_i - x_i|^2. \quad (6.2)$$

Minimalizácia tejto kvadratickej chyby sa používa najčastejšie ako kritérium na optimalizáciu váh neurónovej siete trénovanej algoritmom Back-Propagation of Error (prípadne jeho modifikácií), vzhľadom na vlastnosti, na ktorých je založená. Tento učiaci algoritmus vychádzajúci z tohto kritéria očividne znevýhodňuje väčšie chyby z dôvodu druhej mocniny vo vzorci. Tým pádom, proces učenia neurónovej siete prebieha za účelom prvotného odstraňovania najväčších chýb a až potom menších chýb. Pri bližšom pohľade na toto učenie zbadáme charakteristický priebeh. V prvotných fázach (vysoká hodnota kvadratickej chyby) nastáva prudký pokles chyby. Následne dochádza k pomalému poklesu chyby, keďže sieť prechádza do jedného z možných minim chybového priestoru.

Teraz si zadefinujeme ďalšiu chybu, ktorú použijeme pri vyhodnotení výsledkov.

$$MAO = \max |d_i - x_i|. \quad (6.3)$$

6.2 Vyhodnotenie experimentov

V tejto podkapitole sú výsledky a zhodnotenia experimentov. Každý experiment je usporiadaný prehľadne v tabuľke aj grafe a je riadne popísaný

Použitý softvér:

Na to, aby sme vykonali experimenty, boli potrebné vytvoriť si softvér, ktorý mal veľmi jednoduché a prehľadné grafické rozhranie. Tento softvér bol vytvorený v programovacom jazyku C# a implementoval všetky použité metódy, ktoré boli potrebné pre naše experimenty.

Experimenty museli byť rozdelené do dvoch častí. Bolo potrebné vykonať prvú časť experimentov na to, aby sme pokračovať v tej druhej časti.

- Prvá časť experimentov – základné experimenty.
- Druhá časť experimentov – hlavné experimenty.

6.2.1 Prvá fáza experimentov

V tejto fáze sme sa zaoberali experimentami, ktorých hlavným cieľom bolo identifikovať optimálnu architektúru a parametre neurónovej siete so spätným šírením chyby.

Cieľom tejto fázy nebolo dokonalé naučenie neurónovej siete, ale získanie informácií o vplyve jednotlivých učiacich parametrov a architektúry siete na kvalitu predikcie na testovacej množine. Počet učiacich cyklov v tejto fáze bol výrazne menší vzhľadom na druhú fázu. Obvykle sa tento počet pohyboval v rozhraní 200 – 300 cyklov. V protiklade k tomu bolo množstvo experimentov extrémne vysoké. V tejto prvej fáze bol kvôli rýchlosti výpočtov použitý iba algoritmus so spätným šírením chyby. Ale výsledky z tejto fázy nám slúžili ako východzí bod v druhej časti tak ako pre algoritmus so spätným šírením chyby, jeho modifikáciu „delta-bar-delta“ a aj „evolučnú stratégiu“.

Jednotlivé parametre a ich rozsahy:

PSV	Počet skrytých vrstiev	<1 – 3>
PNVSV	Počet neurónov v jednotlivých skrytých vrstvách	<10 – 30>
VCO	Veľkosť časového okna na vstupe	<8 – 30>
ALPHA	Strmosť sigmoidy (aktivačnej funkcie)	<0,1 – 1>
GAMA	Učiaci parameter	<0,001 – 0,1>
THETA	Prah	<0,1 – 1>

Tabuľka 1: Reprezentatívne výsledky prvej fázy experimentov na laserových dátach pre algoritmus „back-propagation of error“.

PSV	PNVSV	VCO	ALPHA	GAMA	THETA
2	15 – 12	18	1,0	0,01	0,2
2	18 – 17	15	1,0	0,05	0
2	30 – 25	25	1,0	0,1	0,3
3	20 – 17 – 8	20	1,0	0,01	0,2
1	25	22	1,0	0,03	0

Výsledky v predchádzajúcej tabuľke sú zoradené od najlepšieho po najhorší. Dosiahnuté predikčné výsledky pre jednotlivé experimenty neuvádzame z dôvodu malého počtu učiacich cyklov. Ich výpovedná hodnota by bolo veľmi nízka a zavádzajúca.

Zhodnotenie výsledkov prvej fázy

Hneď v úvode treba poznamenať, že veľký vplyv na jednotlivé výsledky má počiatočná inicializácia synaptických spojení. Hodnoty týchto váh sme volili náhodne z rozsahu $[-1; 1]$.

- Počet skrytých vrstiev a neurónov v jednotlivých vrstvách

Už neurónová sieť s jednou skrytou vrstvou sa dokáže vysporiadať s väčšinou aproximačných problémov. Výsledky dosiahnuté v našich experimentoch nám však potvrdili, že použitie dvoch skrytých vrstiev dokáže priniesť výrazné zlepšenie predikcie v porovnaní s jednou skrytou vrstvou. V ďalších experimentoch sme však už vylúčili neurónové siete obsahujúce iba jednu skrytú vrstvu. Zrejmým faktom je adekvátne zvýšenie časovej náročnosti. Keďže našim primárnym cieľom bolo dosiahnutie čo najlepšieho výsledku pri predikcii, nebrali sme do úvahy toto kritérium. Ako môžeme vidieť z dosiahnutých výsledkov, výrazne zvyšovanie počtu skrytých vrstiev a neurónov v nich nevedie k zlepšeniu kvality predikcie. Dôvod je zrejmý a to preučenie neurónovej siete. Chyba na trénovacej množine výrazne klesne ale s ňou aj schopnosť generalizácie naučených a do siete zakódovaných „poznatkov“. Tým pádom nám začne stúpať chyba na testovacej množine.

- Veľkosť časového okna na vstupe

Jedným z najkritickejších parametrov ovplyvňujúcim presnosť predikcie je určite veľkosť časového okna vytváraného na vstupe neurónovej siete. Toto časové okno predstavuje počet vzoriek z predchádzajúcich časových krokov, ktoré majú vplyv na nasledujúcu vzorku. Ak je toto okno príliš malé, nedokáže neurónová sieť dostatočne aproximovať časový rad. Naopak,

ak je časové okno príliš veľké, dochádza tak ako v predchádzajúcom bode k preučeniu neurónovej siete. V tomto prípade predkladáme sieti vzorky z príliš vzdialených časových krokov, ktoré už nemajú výrazný vplyv na hodnotu časového radu v nasledujúcom kroku a môžu viesť k zhoršeniu kvality predikcie. Najlepšie výsledky sme dosahovali s časovým oknom, ktoré malo hodnotu 18. Na tejto hodnote má výrazný vplyv aj potreba predikovať 100 krokov dopredu.

- Strmost' sigmoidy

Najlepšie výsledky sme dosahovali s hodnotou, ktorá sa rovnala 1. Ale aj v prípade iných hodnôt neboli rozdiely vo výsledkoch nejako výrazné.

- Učiaci parameter

Hodnoty tohto parametra, pri ktorých sme dosahovali najlepšie výsledky, sa pohybovali v rozsahu $<0,01 - 0,02>$. V prípade väčšej hodnoty tohto parametra sme síce dosiahli rýchlejšiu konvergenciu, ale táto bola vykúpená nižšou kvalitou predikcie. V týchto prípadoch väčšinou neurónová sieť nedokázala správne zachytiť globálne minimum stavového priestoru a skončila v niektorom z mnohých lokálnych miním.

- Prah

Úlohou tohto parametra je v prípade potreby posunúť hodnoty časového radu viac do pracovného pásma sigmoidálnej aktivačnej funkcie. Štatistické rozloženie hodnôt časového radu laserovej postupnosti ale použitie prahu nevyžaduje, a ak, tak iba v minimálnej miere.

6.2.2 Druhá fáza experimentov

V prvej fáze sme získali už čiastočnú znalosť správania sa umelých neurónových sietí. Tieto poznatky sme využili v hlavných experimentoch v nasledujúcej druhej fáze, kde sme už vybrali konkrétnu architektúru neurónovej siete práve na základe experimentov z prvej fázy. V tejto druhej časti experimentov dosahovala dĺžka učenia výrazne vyššie hodnoty ako v prvej časti. Naším cieľom bolo identifikovať sieť s najlepšou kvalitou predikcie na testovacej množine a vyhodnotiť jednotlivé modifikácie algoritmov, učiaceho procesu a architektury.

Priebeh experimentov

V nasledujúcich experimentoch sme sa snažili overiť kvalitu predikcie všetkých kombinácií a to či už

- učiaceho algoritmu (back-propagation, delta-bar-delta, evolučná stratégia),

- spôsobu zmeny synaptických váh (iteračný verus dávkový - neaplikuje sa pre evolučnú stratégiu),
- typu prediktora (krátkodobý prediktor v úlohe viackrokového prediktora verus priamo do siete implementovaný prediktor).

Algoritmus „back-propagation of error“ a „delta-bar-delta pravidlo“

Použité skratky:

- ITER – iteračný spôsob zmeny synaptických váh.
- BATCH – dávkový spôsob zmeny synaptických váh.
- KPvUVP – krátkodobý prediktor v úlohe viackrokového prediktora.
- DSIP – do siete implementovaný prediktor.

V nasledujúcich tabuľkách môžeme vidieť najlepšie dosiahnuté výsledky pre jednotlivé kombinácie.

Tabuľka 2: Výsledky najlepšie dosiahnutých hodnôt pre algoritmus „back-propagation of error“.

ID	Zmena synaptických váh	Typ prediktora	MSE	MAO	MAPE [%]
1	ITER	KPvUVP	0,001580	0,243	21,7 %
2	ITER	DSIP	0,000831	0,085	14,9 %
3	BATCH	KPvUVP	0,001842	0,281	24,2 %
4	BATCH	DSIP	0,000880	0,099	15,5 %

Tabuľka 3: Výsledky najlepšie dosiahnutých hodnôt pre algoritmus „delta-bar-delta pravidlo“.

ID	Zmena synaptických váh	Typ prediktora	MSE	MAO	MAPE [%]
1	ITER	KPvUVP	0,001013	0,156	18,9 %
2	ITER	DSIP	0,000728	0,071	12,1 %
3	BATCH	KPvUVP	0,000795	0,075	13,2 %
4	BATCH	DSIP	0,000754	0,073	12,8 %

Evolučná stratégia

V našich experimentoch bola použitá $(m+1)$ evolučná stratégia, pomocou ktorej sme sa snažili nájsť optimálne hodnoty synaptických váh neurónovej siete.

Zavedieme nasledujúci formalizmus. Pod chromozómom (jedincom) $\vec{\alpha}$ budeme rozumieť vektor zložený z reálnych premenných fixnej dĺžky k .

$$\vec{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_k) \in R^k \quad (6.4)$$

Tieto chromozómy sa nachádzajú v populácii P .

$$P = (\vec{\alpha}_1, \vec{\alpha}_2, \dots, \vec{\alpha}_m) \quad (6.5)$$

kde m je kardinalita populácie P , vyjadrujúca počet chromozómov v P . Nech f je účelová funkcia definovaná nad množinou reálnych vektorov dĺžky k

$$f = R^k \rightarrow R. \quad (6.6)$$

Táto funkcia ohodnotí každý reálny vektor $\vec{\alpha} \in R^k$ jedným reálnym číslom. V tomto experimente bola účelovou funkciou MAPE chyba na tréningových údajoch. Chyba bola určovaná zo všetkých vzoriek z tréningovej množiny okrem posledných 10 vzoriek. Týchto posledných 10 vzoriek slúžilo ako validačná množina. Na tejto množine sa po úprave váh overovala kvalita predikcie na vzorkách, ktoré neboli použité v optimalizačnom algoritme pre výpočet hodnoty účelovej funkcie. Týmto spôsobom sme sa snažili zabrániť preučeniu neurónovej siete.

Následne sa pristúpilo k samotnému optimalizačnému algoritmu. Ten bol inicializovaný náhodným vygenerovaním populácie P . Veľkosť populácie mala hodnotu m jedincov a každého jedinca v populácii tvoril reálny vektor dĺžky 450 pre krátkodobý prediktor a 1680 pre priamo do siete implementovaný prediktor (jedinec predstavoval hodnoty synaptických váh neurónovej siete). V ďalšom kroku sa už pristúpilo k zostrojovaniu novej populácie, a to tým spôsobom, že sa kvázi náhodne vybrali 2 chromozómy, na ktoré sa aplikoval reprodukčný operátor.

Tento postup sa opakoval až dovtedy, dokiaľ kardinalita novej populácie nedosiahla hodnotu l . Následne takto navrhnuté vektory – potomkov a pôvodné rodičovské vektory – zoradíme spolu podľa optimálnosti riešení a prvých m najlepších jedincov zoberieme ako rodičov do ďalšej populácie.

Rodičia pre kríženie sa vyberajú z celej populácie s rovnakou pravdepodobnosťou, nezáleží teda na optimálnosti riešenia (konvergencia je dostatočne zabezpečená selekciou potomkov pre ďalšiu generáciu rodičov). Keď už máme vybraných dvoch jedincov $\vec{\alpha}$ a $\vec{\beta}$,

môžeme pristúpiť k reprodukčnému procesu, kde sa pomocou operátora reprodukcie O_{repro} zostrojil jeden nový chromozóm (potomok) $\vec{\lambda}$ čo môžeme zaznačiť formálne takto:

$$\vec{\lambda} = O_{repro}(\vec{\alpha}, \vec{\beta}) \quad (6.7)$$

Reprodukčný operátor O_{repro} tvorila jednak operácia kríženia O_{cross} a jednak operácia mutácie O_{mut} . Operátor kríženia v našom prípade tvorilo diskkrétne kríženie, pomocou ktorého sme dostali jedného nového jedinca $\vec{\lambda}_{half}$.

$$\vec{\lambda}_{half} = O_{cross}(\vec{\alpha}, \vec{\beta}) \quad (6.8)$$

Diskkrétne kríženie znamená, že s rovnakou pravdepodobnosťou sa v novom jedincovi na danom mieste v reťazci vyskytne hodnota z jedného alebo druhého rodiča. Čiže pre každý prvok chromozómu sa vygenerovalo s rovnomernou distribúciou náhodné číslo z intervalu $[0,1]$. Ak dané číslo bolo menšie ako 0.5, tak bola vzatá hodnota z prvého rodičovského vektora a ak bolo číslo väčšie ako 0.5, tak bola vzatá hodnota z druhého rodičovského vektora.

V ďalšom kroku sme už pomocou operácie mutácie O_{mut} vytvorili z jedinca $\vec{\lambda}_{half}$ výsledného jedinca $\vec{\lambda}$, ktorý bol už uložený do novej populácie potomkov.

$$\vec{\lambda} = O_{mut}(\vec{\lambda}_{half}) \quad (6.9)$$

Operátor mutácie bol realizovaný nasledujúcim spôsobom. Pre každý prvok chromozómu sa vygenerovalo číslo s nulovou strednou hodnotou a štandardnou odchýlkou σ . Následne sa to číslo pripočítalo k pôvodnému prvkovi. Takto sa postupovalo pre všetky prvky chromozómu.

Pri každom vyhodnocovaní vhodnosti nového stavu $\vec{\alpha}^{new}$, vypočítavame chybu nie len na trénovacej množine, ale aj na validačnej množine. Práve chyba na validačnej množine nám slúži ako zastavovacia podmienka pre celý algoritmus evolučnej stratégie.

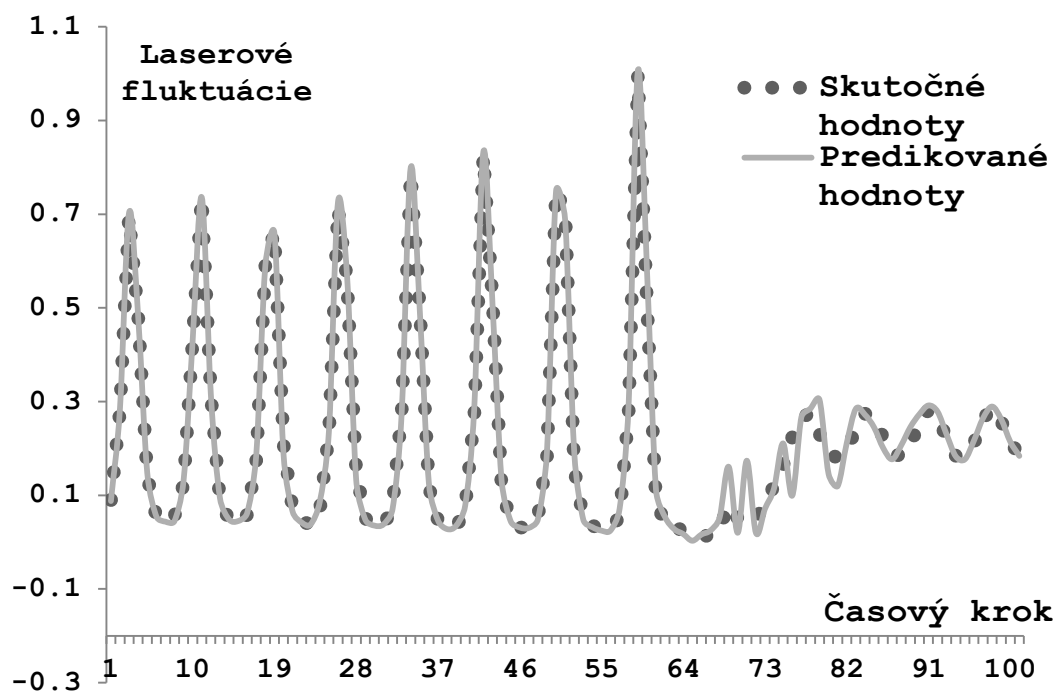
Použité skratky:

- KPVUP – krátkodobý prediktor v úlohe viackrokového prediktora.
- DSIP – do siete implementovaný prediktor.

Tabuľka 4: Výsledky najlepšie dosiahnutých hodnôt pre evolučný algoritmus.

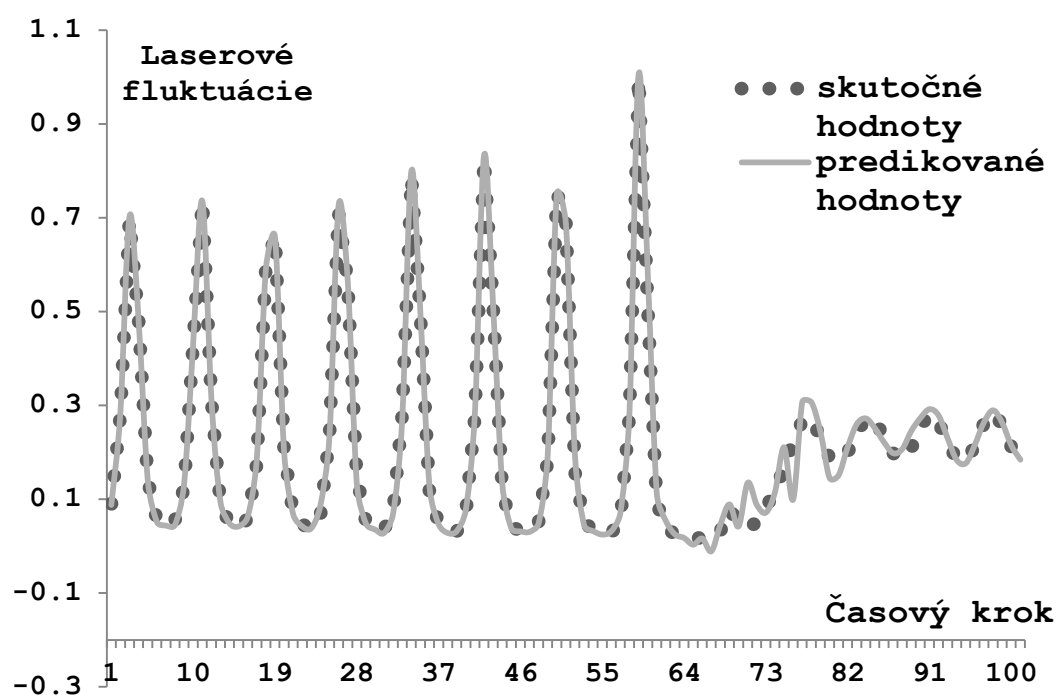
ID	Typ prediktora	Parameter m	Parameter l	Parameter σ	MSE	MAO	MAPE [%]
1	KPvUVP	5000	32000	0,8	0,000825	0,082	14,1 %
2	KPvUVP	7500	52000	1,2	0,000811	0,078	13,5 %
3	KPvUVP	6300	44000	1,5	0,000795	0,074	12,7 %
4	DSIP	7800	28000	1,0	0,000947	0,087	13,6 %
5	DSIP	6450	56700	1,4	0,000876	0,079	12,9 %
6	DSIP	5900	49300	1,3	0,000653	0,068	11,3 %

Algoritmus „back-propagation of error“. MAPE: 14,9 % (experiment č. 2 z Tabuľky č. 2).



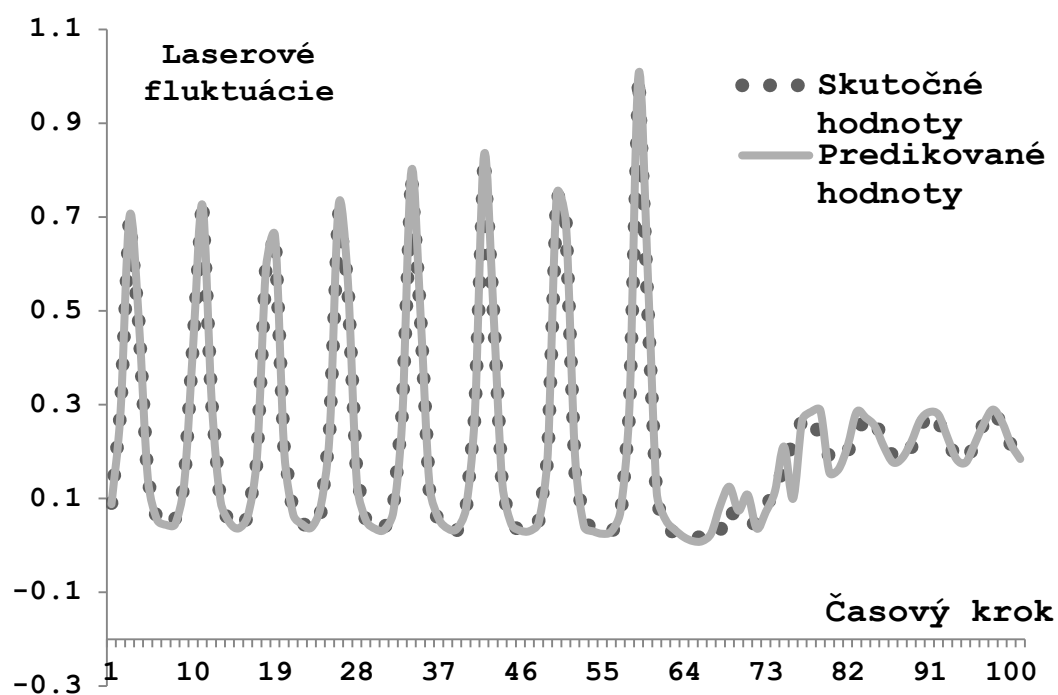
Obrázok 18: 100 testovacích laserových údajov a 100 nami predikovaných hodnôt pomocou algoritmu „back-propagation of error“.

Algoritmus „delta-bar-delta pravidlo“. MAPE: 12, 1 % (experiment č. 2 z Tabuľky č. 3).



Obrázok 19: 100 testovacích laserových údajov a 100 nami predikovaných hodnôt pomocou algoritmu „delta-bar-delta pravidlo“.

„Evolučná stratégia“. MAPE: 11, 3 % (experiment č. 6 z Tabuľky č. 4).



Obrázok 20: 100 testovacích laserových údajov a 100 nami predikovaných hodnôt pomocou „evolučnej stratégie“.

Tabuľka 5: Finálne porovnanie najlepšie dosiahnutých výsledkov pre jednotlivé algoritmy.

ID	Algoritmus	MSE	MAO	MAPE [%]
1	Back-propagation of error	0,000831	0,085	14,9 %
2	Delta-bar-delta pravidlo	0,000728	0,071	12,1 %
3	Evolučná stratégia	0,000653	0,068	11,3 %

Zhodnotenie výsledkov druhej fázy

Umelé neurónové založené na algoritme spätného šírenia chyby nám potvrdili, že v prípade vytvorenia časového okna na vstupe dokážu dosahovať kvalitné predikčné výsledky. Nemalú mieru pri tom zohráva aj správne nastavenie topológie a jednotlivých parametrov učiaceho algoritmu.

- **Algoritmus**

Najlepšie výsledky sme dosiahli pomocou „evolučnej stratégie“. Ba čo viac, okrem lepšej kvality predikcie sme dosiahli aj rýchlejšiu konvergenciu do globálneho minima chybovej funkcie. Čiže neurónová sieť sa dokázala naučiť za kratší čas v porovnaní s klasickým algoritmom „back-propagation of error“. Druhý najlepší výsledok sme dosiahli s „delta-bar-delta pravidlom“, čím sa potvrdila opodstatnenosť tejto modifikácie základného algoritmu „back-propagation of error“. V našom prípade rýchlosť učenia nezohrávala podstatnú rolu, ale v prípade niektorých reálnych aplikácií môže byť práve toto kritérium rozhodujúce. Na tomto mieste treba zdôrazniť aj vyššiu robustnosť a stabilitu algoritmu „evolučnej stratégie“.

- **Spôsob zmeny synaptických váh (neaplikoval sa na „evolučnú stratégiu“)**

Dávkový prístup pri trénovaní umelých neurónových sietí znamená, že synaptické váhy sa nemenia po každej predloženej vzorke, ale až po predložení celej trénovacej množiny. Tento prístup dokáže v niektorých prípadoch zlepšiť kvalitu predikcie ale závisí to hlavne od štruktúry a charakteru dát. V našom prípade sme dosiahli lepšie predikčné výsledky v oboch algoritmoch („back-propagation of error“, „delta-bar-delta pravidlo“) s iteratívnym prístupom a opodstatnenosť dávkového prístupu pri laserových dátach sa nepotvrdila. Dynamika laserovej testovacej množiny nevytvára dostatočný priestor na využitie zmeny synaptických váh až po predložení všetkých trénovacích vzoriek.

- Typ prediktora

V prípade typu prediktora sa potvrdili naše očakávania. Oveľa lepšie výsledky sme dosiahli pomocou prediktora zabudovaného priamo do architektúry neurónovej siete. Jednokrokový prediktor použitý na viackrokovú predikciu sám seba znevýhodňuje už svojou podstatou. Každá hodnota, ktorá je touto sieťou vypredikovaná obsahuje určitú chybu – odchýlku od požadovanej hodnoty. Táto hodnota je potom opäť spätne privádzaná na vstup neurónovej siete. Tento proces pokračuje pre všetky vzorky testovacej množiny, čím sa vznikajúca chyba nabaluje a zvyšuje pre každú ďalšiu testovaciu vzorku. Keďže testovaciu množinu našej laserovej postupnosti tvorilo až 100 vzoriek, bol tento efekt evidentný. Na druhej strane tento typ prediktora vykazuje výrazne nižšie nároky na výpočtovú výkonnosť a tým pádom aj celý učiaci proces neurónovej siete. V prípade prediktora zabudovaného priamo do topológie neurónovej siete, sa s každou ďalšou testovacou vzorkou zväčšuje aj jej architektúra – počet neurónov vo výstupnej vrstve. Od toho sa odvíja aj ďalší nárast množstva synaptických váh a ďalších elementov tejto siete. Aj napriek horšej kvalite predikcie jedнокrokového prediktora si vieme predstaviť reálne aplikácie, kde by tento prediktor bol uprednostnený. V prípade potreby rýchleho naučenia sa neurónovej siete a obmedzeného počtu vzoriek testovacej množiny sa môže javiť tento prediktor ako výhodnejšia voľba.

7. Záver

Cieľom tejto práce bolo ozrejmiť si a naštudovať problematiku predikcie časových radov pomocou dopredných neurónových sietí. Hlavný dôraz sa kladol na algoritmus so spätným šírením chyby, jeho modifikáciu „delta-bar-delta pravidlo“ a „evolučnú stratégiu“ ako možnú náhradu za štandardný algoritmus.

Zhodnotenie zadaných úloh

1. Kapitola č. 2 nám poskytuje ľahký náhľad do oblasti umelej inteligencie.
2. V nasledujúcej kapitole č. 3 sme sa už zamerali na problematiku umelých neurónových sietí ako jedných z hlavných predstaviteľov umelej inteligencie. Nevynechali sme ani biologický pohľad na neurón.
3. Tá istá kapitola nám poskytuje už detailný popis umelých neurónových sietí využívajúcich algoritmus “back-propagation of error“ a jeho modifikáciu pod názvom „delta-bar-delta pravidlo“.

4. V kapitole č. 4 sme sa oboznámili s evolučnou stratégiou ako predstaviteľom optimalizačných metód.
5. Otázku predikcie a predikcie časových radov sme detailne preskúmali v nasledujúcej kapitole č. 5. Tá istá kapitola obsahuje aj popis prediktora založeného na báze umelých neurónových sietí.
6. Programovú realizáciu predikčného systému môžeme nájsť na priloženom CD. Obsahuje všetky použité algoritmy v našich experimentoch.
7. Systém experimentov bol navrhnutý v kapitole č. 6. Reflektuje jednak potrebu sledovania parametrov predikčného systému a jednak použité testovacie dáta.
8. V tej istej kapitole sme sa takisto zamerali na analýzu vlastností predikčného systému, vyhodnotenie a porovnanie jednotlivých dosiahnutých výsledkov.

Úlohy, ktoré boli vymedzené v zadaní tejto práce môžeme záverom skonštatovať za splnené. Túto prácu sme postavili na porovnaní predikčných schopností viacerých modifikácií dopredných neurónových sietí vychádzajúcich z algoritmu so spätným šírením chyby. Pod modifikáciami treba chápať nie len zmeny algoritmu, ale aj typu prediktora a v neposlednom rade celkovej architektúry systému. Výsledky dosiahnuté porovnaním predikčných schopností jednotlivých modifikácií nám dokazujú fakt, že aj nové prístupy môžu smelo konkurovať existujúcim a v praxi osvedčeným algoritmom.

Modulárnosť a adaptačné schopnosti predstavujú jednu z hlavných výhod umelých neurónových sietí v problematike predikcie časových radov. Na základe dosiahnutých výsledkov môžeme konštatovať opodstatnenie rozvoja spomínanej problematiky nie len v oblasti teórie, ale aj v realizácii jednotlivých predikčných

8. Literatúra

- [1] Sinčák, P. Andrejková, G.: *Neurónové siete Inžinierský prístup (1.diel)*, Elfa press, Košice, 1996.
- [2] Marček, D.: *Neurónové sítě a fuzzy časové řady*, Ediční středisko FPF SU, Opava, 2002.
- [3] Kaleta, S.: *Výpočtová inteligencia pre predikciu časových radov*, Technická univerzita, Košice, 1999.

- [4] Šíma, J.: *Neuronové sítě jako modely analogových výpočtu*, Praha, 2006.
- [5] Orsagiová, J.: *Predpovedanie finančných časových radov pomocou neurónových Sietí*, (Diplomová práca), Univerzita Komenského v Bratislave, 2009.
- [6] Kelíšek, A.: *Analýza časových radov pomocou neurónových sietí*, In: Sborník příspěvků z konference mladých vědeckých pracovníků Věda a krizové situace, Ostrava, pp. 68 – 73, 2007.
- [7] Kolesár, J.: *Analýza infračerveného spektra organických zlučenin použitím umělých neurónových sietí*, (Diplomová práca), Univerzita Komenského v Bratislava, 2006.
- [8] Kvasnička, V. Beňušková, L. Pospíchal, J. Farkaš, I. Tiňo, P. Král' A.: *Úvod do teórie neurónových sietí*, Vydavateľstvo IRIS, Bratislava, 1997.
- [9] Beňušková, L.: *Umelé neurónové siete*, Umelá inteligencia, Bratislava, 2002.
- [10] Šíma, J.: *An Introduction to Neural Networks*, Technical Report, Institute of Computer Science, Academy of Sciences of the Czech Republic, Praha, 1998.
- [11] Robel, A.: *The dynamic pattern selection algorithm: Effective training and controlled generalization of propagation neural networks*, Technical report, Technische Universitat Berlin, 1994.
- [12] Haykin, S.: *Neural Networks – A Comprehensive Foundation*, Macmilian Publishing, New York, 1994.
- [13] <http://absolventi.gymcheb.cz/2008/kakutov/nerv.html>
- [14] <http://www1.lf1.cuni.cz/~zfisar/bp/2.2.htm>
- [15] Clark, J.W. Rafelski, J. Winston, J.V.: *Brain without mind: computer simulations of neural networks with modifiable neuronal interactions*, Physics Report, University of Cape Town, 1985.
- [16] Rosenblatt, F.: *The perceptron: A probabilistic model for information storage and organization in the brain*, Psychological Review, Vol. 65, pp. 386 – 408, 1958.
- [17] Rosenblatt, F.: *Principles of Neurodynamic*, Spartan Books, Washington D.C., 1962.
- [18] Tučková, J. Bártů, M. Zetocha, P.: *Aplikace umělých neuronových sítí při zpracování signálů*, Skripta FEL ČVUT, Praha, 2009.
- [19] Hamilton, J.D.: *Time series analysis*, Princeton University Press, Princeton, 1994.
- [20] Harvey, A.C.: *Forecasting, structural time series models and the Kalman filter*, Cambridge University Press, Cambridge, 1989.
- [21] Kvasnička, V., Pospíchal, J., Tiňo, P.: *Evolutionary algorithms* (in Slovak), STU Publishing: Bratislava 2000.

- [22] Zelinka Ivan: Umělá inteligence - hrozba nebo naděje? [BEN - technická literatura](#), 2003, [ISBN 80-7300-068-7](#)
- [23] Zelinka Ivan: Umělá inteligence v problémech globální optimalizace [BEN - technická literatura](#), 2002, [ISBN 80-7300-069-5](#)
- [24] Vladimír Mařík, Olga Štěpánková, Jiří Lažanský: Umělá inteligence 1-5, [Academia](#)