

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE



Fakulta životního prostředí

Katedra vodního hospodářství a environmentálního modelování

Bakalářská práce

Neuronové sítě ve vodním hospodářství

Vedoucí bakalářské práce: doc. Ing. Petr Máca, Ph.D.

Vypracovala: Anna Morozova

PRAHA

2015

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Katedra vodního hospodářství a environmentálního modelování

Fakulta životního prostředí

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Anna Morozova

Vodní hospodářství

Název práce

Neuronové sítě ve vodním hospodářství

Název anglicky

Neural networks in water management

Cíle práce

Cílem práce je rešerše poznatků souvisejících s modely neuronových sítí a jejich aplikacemi ve vodním hospodářství.

Metodika

Provedte rešerši na téma neuronových sítí a jejich aplikací ve vodním hospodářství.

Zaměřte se na

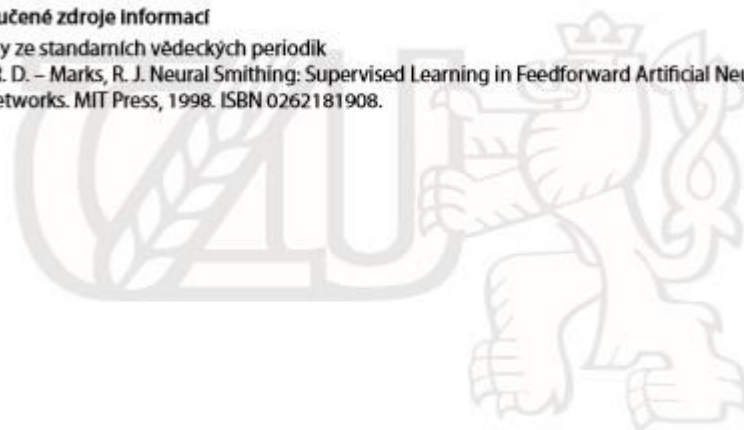
1. Obecné modely neuronových sítí
2. Vrstevnaté perceptrony
3. Případové studie, které dokumentují aplikaci modelů neuronových sítí v různých oblastech vodního hospodářství

Doporučený rozsah práce
standardní

Doporučené zdroje informací

fulltexty ze standardních vědeckých periodik

Reed, R. D. – Marks, R. J. Neural Smoothing: Supervised Learning in Feedforward Artificial Neural Networks. MIT Press, 1998. ISBN 0262181908.



Předběžný termín obhajoby
2015/06 (červen)

Vedoucí práce
Ing. Petr Máca, Ph.D.

Elektronicky schváleno dne 8. 4. 2015

prof. Ing. Pavel Pech, CSc.

Vedoucí katedry

Elektronicky schváleno dne 8. 4. 2015

prof. Ing. Petr Sklenička, CSc.

Děkan

V Praze dne 13. 04. 2015

Prohlášení

Prohlašuji, že jsem celou bakalářskou práci na téma „Neuronové sítě ve vodním hospodářství“ vypracovala samostatně pod vedením doc. Ing. Petra Máci, Ph.D., a že jsem uvedla všechny literární prameny, ze kterých jsem čerpala.

V Praze, dne 15.04.2015

.....

Anna Morozova

Poděkování

Ráda bych poděkovala vedoucímu mé bakalářské práce, doc. Ing. Petru Máčovi, Ph.D. za odborné vedení práce, dobré rady a trpělivost. Dále bych chtěla poděkovat mamince, rodině a přátelům za neobyčejnou podporu a motivaci.

ABSTRAKT

Cílem této bakalářské práce „Neuronové sítě ve vodním hospodářství“ je vytvořit souhrn informací, které jsou doposud známé o aplikaci umělých neuronových sítí ve vodním hospodářství. V poslední době jsou modely neuronových sítí úspěšně používají jako predikční modely v různých oblastí vodního hospodářství. Text bakalářské práce je rozdělen na několik kapitol. První kapitola poskytuje obecné informace o neuronových sítích. V další části práce je popsána historie a rozvoj, základní jednotky neuronové sítě. Největší část práce pojednává o využití neuronových sítí ve vodním hospodářství, zabývá se architekturou sítí a popisuje procesy, které probíhají uvnitř sítě. Velká pozornost věnovaná vrstevnatému perceptronu a nejčastěji používanému algoritmu učení neuronové sítí. V závěru je rozebrána případová studie, ve které uveden příklad použití vícevrstevného perceptronu v rámci modelování odtoku v Novém Zélandu na řece Kapakapanui.

Klíčová slova: vrstevnatý perceptron, algoritmus zpětné propagace chyb, hydrologické modelování

ABSTRACT

Presented Bachelor thesis „Neural networks in water management“ is aimed on a review of so far discovered information about application artificial neural network (ANNs) in water management. In recent years, it have become extremely popular to use ANNs for prediction in a number of areas. The text of this thesis is divided on a few chapters. The first chapter named Preliminary contains general facts about ANNs. Next part provides review of the history of ANNs, single processing element and it's models. The largest part of this thesis describes the application ANNs in water management, architectures of nets and processes inside the network. In this part the special attention is given to multi-layer perceptron (MLP) and the most commonly used method for training MLP. The application of MLP is described using the case study of runoff forecast in New Zealand in the end of presented bachelor thesis.

Key words: Multi-layer perceptron, Back-propagation algorithm, hydrological modelling

Obsah

1. Úvod a cíl práce.....	10
2. Historie neuronových sítí	12
3. Umělé Neuronové Sítě.....	16
3.1 Inspirace.....	16
3.2 Neuron a jeho modely.....	16
3.2.1 Model biologického neuronu.....	17
3.2.2 První model neuronu	18
3.2.3 Formální model neuronu.....	19
3.2.4 Aktivační funkce.....	19
3.3 Využití neuronových sítí.....	21
3.4 Přehled architektur Umělých Neuronových sítí	22
3.4.1 Jednovrstevný perceptron (Single-layer perceptron).....	23
3.4.2 Více vrstevnatý perceptron (Multi-layer perceptron).....	23
3.4.3 Radiální báze funkce (Radial Basis Function Networks)	24
3.4.4 Rekurentní Neuronové Sítě (Recurrent Neural Network).....	25
3.4.5 Obecné regresní sítě (Generalized Regression Neural Network).....	26
4. Metodický postup aplikace modelu umělých neuronových sítí	27
4.1 Výkonnost UNS	27
4.2 Příprava dat.....	27
4.2.1 Rozdělení dat	28
4.2.2 Standardizace	28
4.3 Volba vstupů do neuronových sítí	28
4.4 Volba architektury sítě.	29
4.4.1 Topologie.....	29
4.4.2 Prořezávací algoritmus (Pruning algorithms)	30
4.4.3 Konstruktivní algoritmus (Constructive algoritmus)	30
4.5 Trénování sítě.....	30
4.5.1 Lokální metody I. a II. řádu – algoritmus zpětné propagace chyb	31
4.5.2 Globální metody	38
4.5.3 Zastavovací kritéria	41
4.6 Validace.....	41
5. Aplikace umělých neuronových sítí ve vodním hospodářství.....	42
5. 1. Případová studie.....	42
5.1.1. Charakteristika povodí.....	43
5.1.2. Charakteristika dat.....	44

5.1.3	Trénování UNS.....	46
5.1.4	Výsledky.....	47
5.1.5	Diskuse.....	48
6.	Závěr.....	49
7.	Zdroje.....	50
7.1	Seznam literatury.....	50
7.2	Internetové zdroje.....	56
8.	Seznam obrázku.....	57
9.	Příloha.....	59
9.1	Hydrogramy.....	59

1. Úvod a cíl práce

Co to jsou Umělé Neuronové Sítě a jak fungují? Jednotná definice neexistuje, ale můžeme si představit, že neuronové sítě jsou tvořené několika jednoduchými procesními jednotkami (neurony), z nichž každý má několik vstupů a jeden výstup. Neurony mezi sebou vyměňují informace pomocí elektrického potenciálu. Myšlenka, ze které vychází vývoj UNS, bylo přání modelovat a kopírovat hlavní funkce mozku a vytvořit model umělé inteligence, který by mohl plnit intelektuální funkce, podobné těm, co plní náš mozek. Ve dvou směrech se neuronové sítě podobají lidskému mozku: UNS získají znalosti v procesu učení a chrání informace uvnitř neuronu (Smith, 1997; Tučková, 2003; Reed & Marks, II, 1998).

Proč Umělé Neuronové Sítě (UNS)? Protože umělé neuronové sítě se staly součástí našeho života a jsou nyní široce používány pro řešení různých problémů. Aktivně se používají tam, kde obecné algoritmické řešení nejsou efektivní. Mezi úkoly, které řeší UNS, patří: rozpoznávání textů, kontextová reklama na internetu, filtrování spamů, bezpečnostní a kamerové systémy a predikce odtoku, slanosti, odtoku a srážek (Tučková, 2003). Nejpoužívanějším typem neuronové sítě pro predikce v oblastech vodního hospodářství jsou vícevrstevný perceptron (MLP) a radiální bázové funkce (RBF) (Dawson & Wilby, 2001). V mé práci je popsáno použití neuronových sítí, zvláštní pozornost je věnována vrstevnatému perceptronu pro predikci odtoku.

Cílem mé bakalářské práce poskytnout přehled základních znalostí o poměrně mladém oboru neuronových sítí a jejich aplikací v rámci vodního hospodářství. Součástí práce je rozbor případové studie, která se zaměřuje na modelování odtoku neuronovými sítěmi.

Hlavními zdroji informací pro vytvoření této bakalářské práce byly práce Maiera & Dandyho (2000; 1998), v teoretické části kniha Šima & Nerudy (1996), Šnorka & Jiřiny (1996) a Nováka (1992). V části pojednávající o vrstevnatém perceptronu je kniha Reeda & Marksa (1998). Klíčovým zdrojem v případové studii byl článek Fernando & Kerryho (2003).

Značení

V dalších kapitolách budeme používat následující značení:

x	Vstupy do neuronové sítě
y	Výstupy z neuronové sítě
w	Synaptické váhy
S	Nelineární aktivační funkce
θ	Práh neuronu
$f(a)$	Aktivační funkce
γ_n	Velikost kroku
α	Spočítaný výstup
d	Vektor směru klesání chybové funkce
n	Počet opakování
$F(\varphi)$	Sigmoida
E	Energie, Chybová funkce
η	Rychlost učící konstanty
α	Momentům
H	Hessova matice
t	Čas

Zkratky

UNS – Umělé Neuronové Sítě

ANN – Umělé Neuronové Sítě

MLP – Vrstevnatý Perceptron

RBF – Radiální bázová funkce

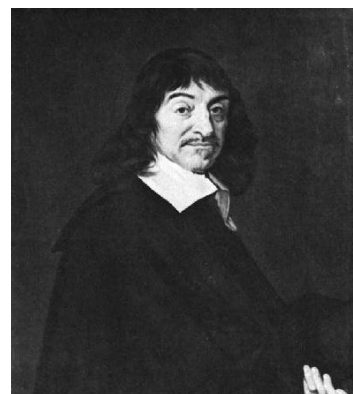
RNS – Rekurentní Neuronové Sítě

ORS – Obecné Regresní Sítě

2. Historie neuronových sítí

Historický vývoj modelů neuronových sítí je datován do doby, když lidi začali přemýšlet o tom, jakým způsobem myslíme, jak si sebe i okolní svět uvědomujeme a jak si získané poznatky i vědomosti pamatujeme. Nosným tématem je sestavení efektivního modelu učení. Staré latinské úsloví říká: „*Repetito est mater studiorum*“ – což znamená: základem učení je opakování (Kopecká, 2011; Novák, 1992).

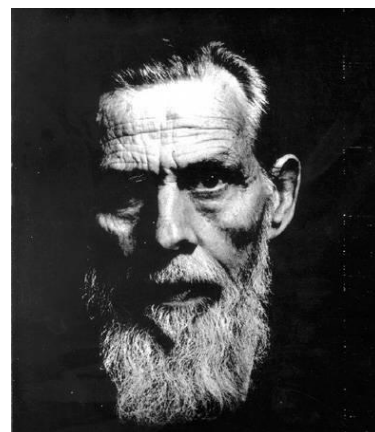
Tajemství myšlení se lidé snažili pochopit již ve středověku. Základy teorie myšlení se ve všeobecnější podobě pokusil formulovat již René Descartes (Obr. 1) před 370 léty. Descartes definoval myšlení jako vnitřní hybné síly jednání, které souvisejí s reflexivním charakterem nervové činnosti (Novák, 1992).



Obr. 1: René Descartes (URL 1)

V současné době je pojem myšlení spojen s poznávacím procesem, kterým si ověřujeme realitu a také řešíme nejrůznější problémy v teorii a v praxi. Intelekt je pak možno chápat jako dispozice k myšlení, umělá inteligence je kvalitní a kvantitativní vyjádření této dispozice (Cejthamr & Dědina, 2010).

Mnoho let po Descartovi byly pak objeveny neurony jako základní stavební jednotka všech nervových systémů vyšších živých organismů. Před více než 70 lety se americký fyziolog Warren Sturgis McCulloch (Obr. 2) v Chicagu se zabýval myšlenkou na vytvoření umělé inteligence. V roce 1943 se svým mladým 17 letým studentem W. Pittsem, publikoval zásadní práci, které se týkali modelování základních funkcí neuronů. Ta výrazně ovlivnila celý další vývoj oboru (Novák, 1992; Šnorek & Jiřina, 1996).



Obr. 2: Warren Sturgis McCulloch (URL 2)

Frank Rosenblatt (Obr. 3) v roce 1956 na základě McCullochova a Pittsova modelu neuronu, vytvořil jednoduché zařízení, které bylo schopno technickými, elektronickými a dalšími prostředky napodobit činnost celé skupiny neuronů. Nazval je *perceptron*. Pro tento model taky vytvořil učicí algoritmus a ukázal, že zařízení se může naučit rozřadit množinu jednoduchých útvarů do dílčích skupin navzájem si podobných jedinců. Tento výsledek vzbudil velký zájem o UNS v té době (Šnorek & Jiřina, 1996).



Obr. 3: Frank Rosenblatt (URL 3)

V současné době, velká část paradigmat neuronových sítí dodnes pracuje na základě McCullochova a Pittsova modelu (Šíma & Neruda, 1996).

Na přelomu padesátých a šedesátých let došlo k velkému projevu zajmu a úspěšnému rozvoji UNS (Šíma & Neruda, 1996). Další rozvoj UNS je spojen s vývojem informativní technologií. V polovině padesátých let Newell a Simon založily základy problematiky tzv. umělé inteligence nebo umělý myslící systém (Novák, 1992). Widrow a jeho kolegové vytvořili první umělé neuronové sítě, které už byly schopny řešit praktické úlohy (Šnorek & Jiřina, 1996).

Analytický poměr vedl od McCullochových a Pittsových prací ze čtyřicátých let, vyústivších ve vytvoření teorie formálních neuronů, chápaných jako modely funkce neuronu biologických přes jednoduché neuronové sítě, jako byly Rosenblattovy perceptrony na začátku šedesátých let (Rosenblatt, 1962).

V roce 1959 předpokládali, že na bázi perceptronu bude možné vytvořit informační systémy, které budou pracovat na základě spíše statistických principů než na logických (Novák, 1992).

Počátkem šedesátých let skupina dalších talentovaných odborníků a především Bernard Widrow ze Stanfordovy univerzity v USA, učinili další velmi úspěšný progres v oboru UNS. Začali konstruovat, z hlediska dnešního dne, velmi jednoduchý, ale přesto poměrně velmi výkonný model UNS. Vytvořili modely ADELIN (ADAPtive LINear Elements) a MADELINE (Multiple ADAPtive LINear Elements). Model ADELIN byl vybaven výkonným učícím pravidlem. Tato neuronová síť se využívá dosud. V ČR v této době navazovali na tyto práce pracovníci

České Vysoké Učení Technické (ČVUT) v Praze (Novák, 1992; Šíma & Neruda, 1996; Nilsson, 1996).

V roce 1969 byla publikovaná kniha M Minského a S. Paperta „Perceptrony“, ve které bylo dokázáno zásadní omezení perceptronů (Minsky & Papert, 1969). Kvůli této kritice, je v historii oboru takové „tiché“ období, kdy došlo k výraznému poklesu zájmu o UNS (Novák, 1992; Šnorek & Jiřina, 1996). Oba tito autoři ve své práci došli k názoru, že modely neuronových sítí typu perceptron s prahovou aktivační funkcí nelze řešit jiný než lineární separabilní problém (Novák, 1992).

V počátcích tohoto „tichého“ období se umělými neuronovými sítěmi zabývalo jen několik badatelů. Do značné míry o nich můžeme dnes mluvit jako o hrdinech (Novák, 1992; Šíma & Neruda, 1996). V USA se objevilo několik prací těchto badatelů, které ukázaly úplně nový přístup v oblasti UNS: Jeden z nich byl John Hopfield. To je světový uznávaný fyzik, měl zásluhu na renesanci oboru neuronových sítí. Hopfield ukázal jak překonat dosavadní omezení neurosítě první generace a propracoval teorii neuronových sítí, která se zabývá modelováním asociativní paměti (Šíma & Neruda, 1996; Vasenkov, 2007).

V roce 1986 ve sborníku editovaném Davidem Rumelhartem a Jamesem McClellandem publikovali své výsledky badatele „PDP skupiny“ (Šíma & Neruda, 1996). V tomto sborníku byl nalezen článek, ve kterém popsali učící algoritmus „zpětné propagace chyby“ (Back-propagation) pro vícevrstevnou neuronovou síť (Multi-layer Perceptron). Tento článek napsali Rumelhart, Hinton a Williams Ronald. Vyřešili ten problém, který se Minskému a Papertovi v šedesátých letech jevil jako nepřekonatelný problém pro využití a další rozvoj neuronových sítí. Tento algoritmus je dosud nejpoužívanější učící metodou neuronových sítí. Tato renesance byla prudká a výrazná a v podstatě trvá až do teď. Nic na tom nemění fakt, že uvedený algoritmus byl znovu objeven, protože byl již publikován některými vědci v „tichém“ období, např., Bryson & Yu-Chi Ho (1969), Werbos (1974) (Novák, 1992; Šíma & Neruda, 1996).

21. – 24. června v roce 1987 v San Diegu poprvé konala velká konference IEEE (International Conference of Neural Networks) specializovaná na neuronové sítě (Butler, 1987). Na konferenci bylo kolem 1700 účastníků a byla založená mezinárodní společnost pro výzkum neuronových sítí INNS (International Neural Network Society). V roce 1988 INNS začala vydávat svůj časopis *Neural Networks*. (Novák, 1992; Šíma & Neruda, 1996). Ve stejném roce Sejnomskim a Rosenbergem byl vyvinut systém NETtalk, který úspěšně konvertoval anglické psaný text na mluvení. Ten

systém byl vytvořen v krátké době učení neuronové sítě z příkladů (Šíma & Neruda, 1996).

V roce 1989 použil Jan Lekun algoritmus zpětné propagace chyby (ZPC) pro trénování neuronové sítě pro řešení úkolu rozeznávání ručně psaných ZIP kódů. Nevýhodou této metody je její trénování trvalo až 3 dny. Kvůli tomu využití této metody bylo značně omezeno. Nízká rychlost učení souvisí s mnoha faktory, včetně problematiky snižujícího se gradientu. Kvůli těmto problémům, už nebylo tak populární využití neuronových sítí, jako v 80. letech (Starý, 2004).

V roce 1992 bylo vyvinuto první hardwarové řešení neuronových sítí – neuronové karty (LORAL – USA), které se vkládají do počítače. Do tohoto roku byly všechny neuronové sítě řešeny softwarovou simulací. Proces učení velmi urychlilo vyvinutí hardwarového řešení několik tisíc krát. V následujícím roce byly vyvinuté komerční dostupné hardwarové karty (CNAPS), které se dalo použít jako akcelerátor při trénování UNS (Starý, 2004; Airportal, 2014).

V roce 2012 odborníci v Googlu vytvořili dosud největší model neuronové sítě. Tento model, který obsahoval kolem 1 mld. spojení, byl překonán hned následujícím rokem. 18. června roku 2013, firma NVIDIA, která ve spolupráci s týmem vědců ze Stanfordské Univerzity, vytvořila největší umělou neuronovou síť na světě, která je určena pro studování procesů, které probíhají v lidském mozku. Tato síť je 6,5 krát větší než síť, kterou vytvořil Google (Paramonov, 2012).

Odborníci předpovídají ještě více udivujících úspěchů ve vývoji umělých neuronových sítí ve všech oblastech lidské činnosti a co je nejdůležitější - v intelektuální činnosti. Předpovídají, že budou sestaveny inteligentní domácí spotřebiče, které se budou moci přizpůsobit majitelům. Bezpečnostní systém bude schopen rozpoznat nejen hlas majitele, ale rozliší i jeho vzhled. Řízení domácností inteligentními řídicími systémy se již nebude zdát tak vzdálené pohádce jako tomu bylo dříve (Biručková, 2014).

3. Umělé Neuronové Sítě

V této kapitole se budeme věnovat rozborů základních poznatků v teorii umělých neuronových sítí. Popíšeme model neuronu – biologický, formální a matematický, architektury UNS a popíšeme učení neuronových sítí a tzv. algoritmus zpětné propagace chyby. Také vysvětlíme parametry, které mohou ovlivnit výkon sítě.

3.1 Inspirace



Obr. 4: Inspirace (URL 4)

Rozvoj neuronových sítí je inspirován principem lidského nervového systému (Obr. 4). Mnoho odborníků a laiků již mnoho let musí neustále rozšiřovat své znalosti v moderní biologii, aby našli nové biologické struktury, které jsou schopny popsat mozek (Šnorek & Jiřina, 1996). V mnoha případech, vytváření nových struktur umělých neuronových sítí vede k odmítnutí biologické podoby. Mozek se stává jenom metaforou a tvoří se takové sítě, které nejsou možné v živě podobě. Ale nehledě na to, že spojení s biologii je příliš slabé, UNS se pořád srovnává s lidským mozkem, protože fungování UNS často připomíná lidské poznání (Wosserman, 1992).

3.2 Neuron a jeho modely

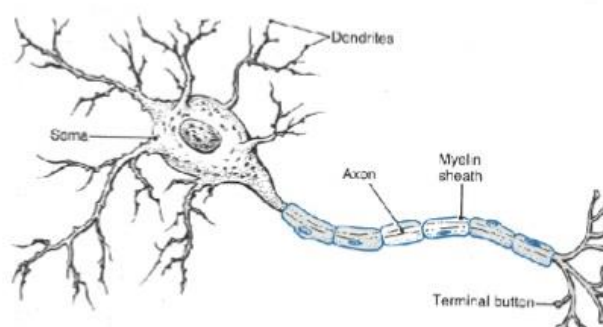
Neuron je jednotka v neuronové síti, která zpracovává informace (Obr. 5). Existuje celá řada různých druhů neuronu. Ve všech případech model neuronu se skládá ze tří částí: vstupní, výstupní a funkční. Ve vstupní části jsou *dendrity*, kterých je kolem deseti tisíc. Ve funkční části, jsou: *váhy*, *tělo* a *bias*. Tato část obsahuje informaci, která má být neuronem zpracována. Výstupní část je *axon*, který je bohatě rozvětven na svém konci (Šnorek & Jiřina, 1996; StatSoft, 2014).



Obr. 5: Biologická neuronová síť. (URL 5)

3.2.1 Model biologického neuronu.

Biologický neuron je základní funkční a stavební jednotkou nervové soustavy člověka převážně mozku a ostatních živočichů. Funkce neuronů jsou uchovávání, přenos informací pomocí iontů a jejich zpracování (Šnorek & Jiřina, 1996). Jednoduchý příklad biologického neuronu je uveden na schematickém obrázku 6:



Obr. 6: Lidský neuron (URL 6)

Mozek člověka obsahuje přibližně 100 mld. neuronových buněk (Novák, 1993). Během života nahrazuje organismus odumřelé buňky novými. U neuronových buněk to ale neplatí, přesto se za 80 let sníží počet neuronů jen o 0,2 až 0,5 % (Šnorek & Jiřina, 1996).

Hlavní výjimečnou vlastností nervové soustavy člověka je schopnost se učit, pamatovat si informace a v případě potřeby si ji vybavit. Smyslové orgány (mechanoreceptory, fotoreceptory a další receptory) využíváme k příjmu informací.

Míša (2006) dělí neuron na:

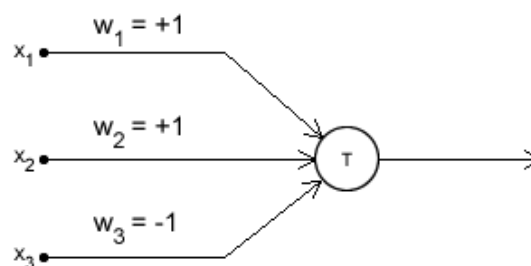
- Tělo (soma);
- Dendrity, jsou spojené s tělem buňky;
- Iniciální segment;
- Axon je to jedná dlouhá tkán;
- Nervové zakončení;

3.2.2 První model neuronu

Za počátek vzniku oboru neuronových sítí je považována práce neurofyziologa Warrena McCullocha a matematika-kybernetika Waltera Pittse z roku 1943 (Šíma & Neruda, 1996), kteří vytvořili velmi jednoduchý model umělého neuronu (Utkarash, 2010).

Jejich model založen na jednoduché prahové aktivační funkci. Byl nazýván *binárním prahovým neuronem* (Ilachinsky, 2001). Tento model má pevný počet vstupů a neměnnou prahovou hodnotu. Takový model dokáže provádět jenom jednoduché logické funkce (Novák, 1993).

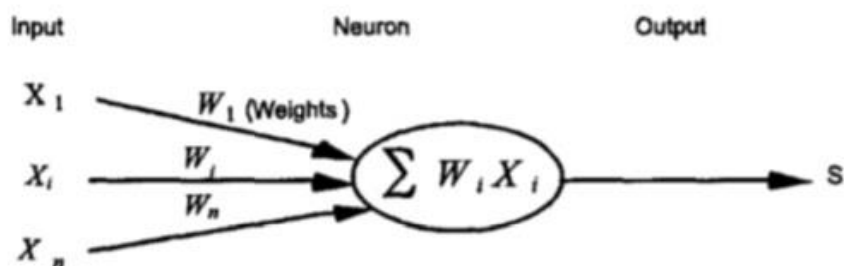
Model McCullocha a Pittsa (Obr. 7) se ale příliš vzdaluje od skutečného biologického modelu.



Obr. 7: Grafická representace McCulloch-Pitts modelu (URL 7)

Kde w_i jsou váhy a x_i jsou vstupy. Takže v tomto modelu, je-li váha se rovná 1, to je excitační vstup, pokud se rovná -1, to je inhibiční vstup (Utkarash, 2010).

3.2.3 Formální model neuronu



Obr. 8: Formální neuron (Riad & kol.)

Výkonným prvkem umělé neuronové sítě je *formální neuron* (Obr. 8), který je zároveň základem matematického modelu. Neuron získáme přeformulováním zjednodušené funkce neurofyziologického neuronů do matematické řeči (Šíma & Neruda, 1996).

Zpravidla vstupní údaje zpracovává formální neuron podle vztahu:

$$y = S\left(\sum_{i=1}^N w_i x_i + \theta\right) \quad (1)$$

kde x_i jsou vstupy do neuronové sítě,

w_i jsou synaptické váhy,

S je obvykle nelineární aktivační funkce neuronu,

θ je práh neuronu.

Výraz v závorce někdy označujeme jako *vnitřní potenciál aktivace neuronu* (Šnorek & Jiřina, 1996). Vnitřní potenciál *aktivační funkce* převádí do výstupu, který je definován v oboru výstupních hodnot aktivační funkce.

3.2.4 Aktivační funkce

Při vstupu signálu do neuronů dochází k aktivaci neuronů. Jak biologický tak i umělý neuron zpracovává vstupní informace. Aktivace je dále transformována tzv. aktivační funkcí $f(a)$. Nejčastější používanější aktivační funkce jsou tyto čtyři: lineární, skoková, omezená a sigmoida (Šnorek & Jiřina, 1996). Biologický neuron zpracovává informace pomocí skokové funkce (2). Je to nejjednodušší typ aktivační funkce a má tvar:

$$f(a) = \begin{cases} 1 & \text{jestliže } a \geq \theta \\ 0 & \text{jestliže } a < \theta \end{cases} \quad (2)$$

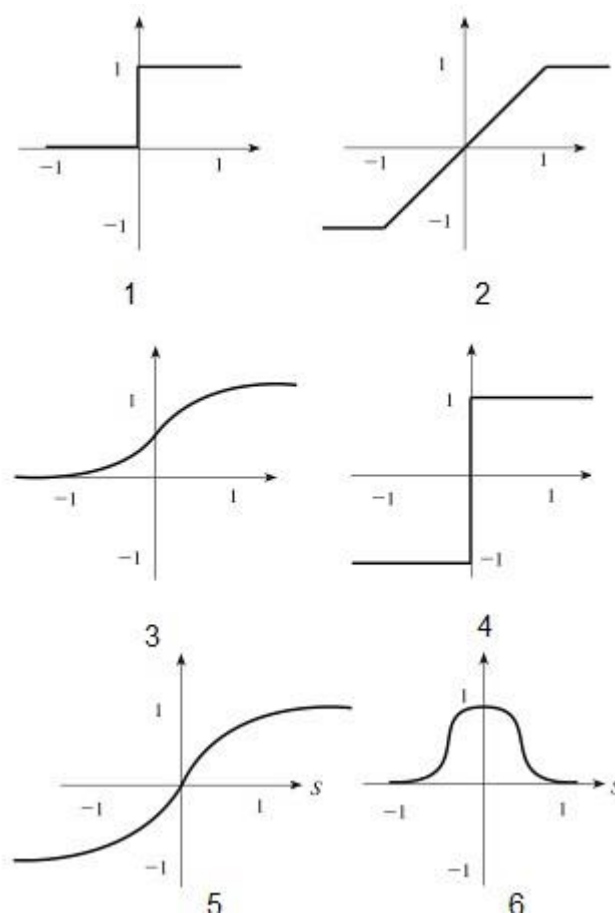
Pokud hodnota vstupu dosáhne prahu θ , vznikne aktivační potenciál (Hristev, 1998).

Nejpoužívanější funkce je sigmoidální funkce ve tvaru:

$$f(a) = \frac{1}{1+e^{-a}} \quad (3)$$

kde $f(a) \in (0;1)$ (Kurkova & kol., 2001; StatSoftCR.,2014).

Mezi další varianty aktivačních funkcí patří lineární a hyperbolická tangenta. Této případy nám ukazuje Lapygin (2008) na obrázku 9:



Obr. 9: Možné varianty aktivačních funkcí podle Lapygina (1 a 4 jsou skokové, 2 je lineární, 3 je sigmoidální funkce, 5 je hyperbolická tangenta, 6 je radiální básová funkce)

Aktivační funkce má vliv na rychlost učení sítě. Maier & Dandy ve své práci od roku 2000 srovnávali rychlost učení sítě s použitím lineární funkce, logistické sigmoidy a hyperbolické tangenty. Sigmoida jasně poskytovala nejlepší výsledky. Ale Moody a Yarvin (1992) srovnávali výkonnost logistické, polynomiální a racionální aktivační

funkce na souboru dat, ve kterých jsou různé stupně šumu a nelinearity. A autoři se shodli že nesigmoidální funkce má nejlepší výkonnost, když data jsou řízeny nelineárními vztahy (Maier & Dandy, 2000). V mé práci jako aktivační funkce byla použita hyperbolická tangenta (Fernando & Kerr, 2003).

3.3 Využití neuronových sítí

V poslední době lidé se snažily spojit neuronové sítě s expertními systémy. V takovém systému neuronová síť reaguje na většinu základních úkolů a všechny ostatní úkoly kontroluje expertní systém (Bogoslavsky, 2007).

Podle Bogoslavského (2007) a StatSoft ČR (2014) neuronové sítě se nejčastěji používají například v:

- Rozpoznávání textů a řeči;
- Predikce změny cen akcií;
- Bezpečnostní systémy;
- Kompresi dat;
- Dopravní signalizace;
- Detekce úvěrového rizika;
- Odhad kvality ropných produktů;

Takže neuronové sítě se používají v mnoha dalších oblastech a k řešení různých typů inženýrských úloh. Využití modelů UNS pro predikce našly v takových oborech jako medicína, ekonomie, fyziologie, biologie a ve vodní hospodářství (Moreno & kol., 2011; Maier & Dandy, 2000).

Svou sílu neuronové sítě berou od paralelizace informací a od schopnosti samoučení, tj. schopnost získat výsledek na základě dat, které se nepotkávali během učení. Taková schopnost dělá neuronové sítě ještě silnější (Braspenning & kol., 1995).

Takže neuronové sítě mají několik specifických výhod:

- automatické učení na příkladech (hlavní výhoda UNS);
- není třeba formulovat matematický model vztahů mezi vstupy a výstupy;
- rychlé učení (pokud neuronová síť má méně než 50 neuronů);
- přizpůsobivost ke změnám ve vstupních hodnotách;
- zevšeobecňování;
- modularita;

Specifické nevýhody neuronových sítí jsou:

- UNS jsou vhodnější pro situace, kde měření se provádí v reálném čase na nezávislých a závislých proměnných (Fernando & Kerr, 2003);
- neuronová síť se chová jako „černá skříňka“(black box);

Model „černé skříňky“ je velmi složitý model, který popisují chování systému pouze na základě znalosti vstupních dat (Reiss, 1964).

- Již natrénovanou síť se nedá použít na jiné povodí, každý problém potřebuje velký soubor dat pro svojí vlastní vstupní a výstupní proměnné (Fernando & Kerr, 2003);

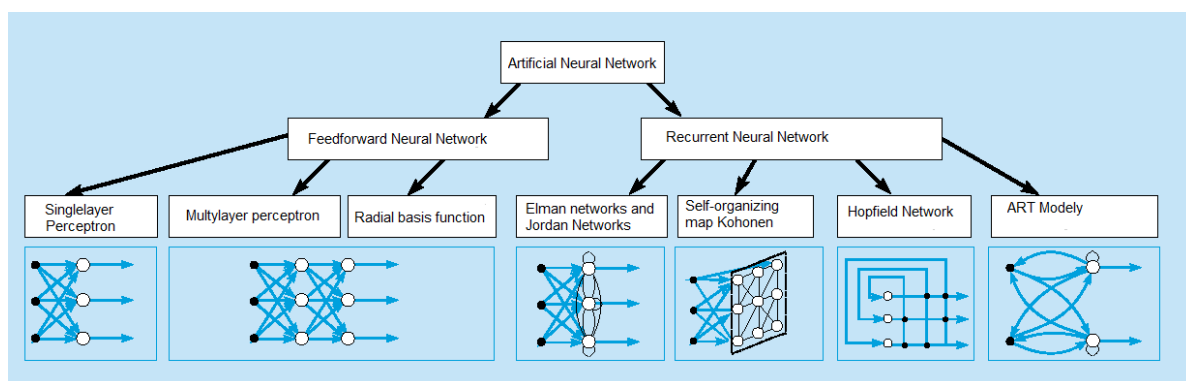
Ale nehledě na nevýhody, umělé neuronové sítě dnes mohou řešit složité úkoly, které v dnešní době považované za těžko řešitelné (Braspenning & kol., 1995; Farid, 1996).

3.4 Přehled architektur Umělých Neuronových sítí

Během doby došlo k velkému rozvoji architektur a algoritmů učení, souhrnně nazývaných paradigmaty. V současné době je široce využíváno okolo 200 architektur neuronových sítí. Z této množiny rozlišujeme 2 základní typy architektur (Šíma & Neruda, 1996):

- **cyklická** (resp. Rekurentní síť);
- **acyklická** (resp. Síť s dopředným šířením signálu);

Rozlišení je dobře vidět na obrázku 10:

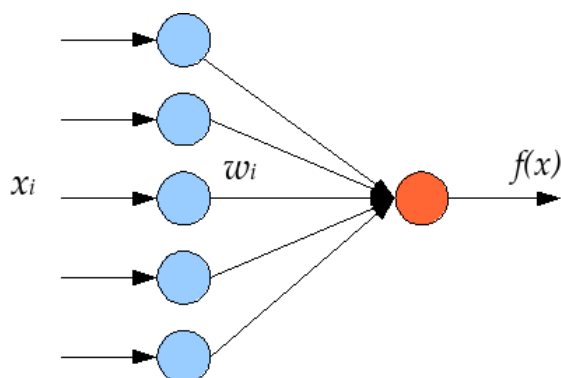


Obr. 10: Typické sítě každé architektury (URL 8)

Rekurentní sítě jsou dynamické, protože pomocí zpětných vazeb modifikují vstupy neuronů, což vede ke změně stavu sítě. Sítě typu s dopředným šířením signálu jsou statistické v tom smyslu, že z daného vstupu vzniká jeden souhrn výstupu, který nezávisí na předchozím stavu sítě (Baykov, 2009; Šíma & Neruda, 1996).

3.4.1 Jednovrstevný perceptron (Single-layer perceptron)

Jednovrstevný perceptron (Obr. 11) byl široce prostudován v roce 1959 Rosenblattem a popsán Widrowem a Lehrem v roce 1990. Perceptron založen na lineární kombinaci vstupních proměnných a vah neuronů, které se transformují nelineárními aktivačními funkcemi (Bishop, 1995).



Obr.11: Single layer perceptron (URL 10)

Jak je vidět na obrázku 11 jednovrstevný perceptron nemá skrytých vrstev. Jedna vrstva vah spojuje vstupy přímo s výstupy. Výstupy jsou nezávislé, takže síť se chová jako tři samostatné sítě (Reed & Marks, 1998).

Jednovrstevná síť se může být trénována pomocí různých metod, například Newtonovou, Levenberg-Marquardtovou metodou a včetně algoritmem se zpětnou propagací chyb. Ale je to v tom případě, když nelinearita je hladká a chybová funkce je diferencovaná (Reed & Marks, 1998).

3.4.2 Více vrstevnatý perceptron (Multi-layer perceptron)

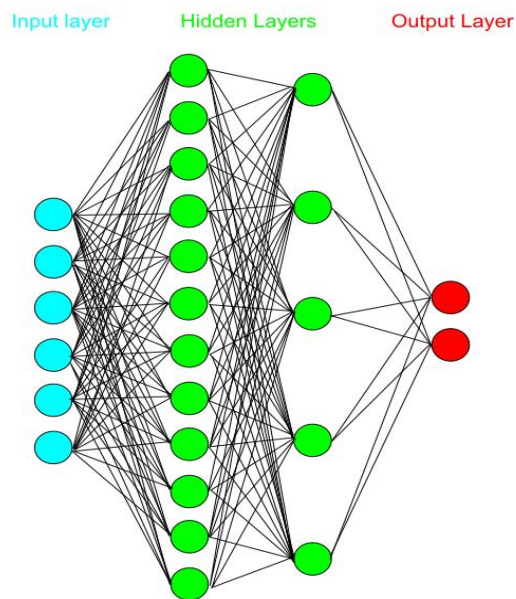
Více vrstevnatý perceptron je nejpopulárnější architektura modelů neuronových sítí, která nejčastěji se používá jako predikční model v oblasti hydrologického modelování (Dawson & Wilby, 2001). Síť 'se většinou učí algoritmem zpětné propagace chyby. Tato architektura byla poprvé navržena Rumelhartem a McClellandem v roce 1986 a podrobně popsána v téměř všech učebnicích o neuronových sítích. (Šíma & Neruda, 1996).

Více vrstevnatý perceptron je model neuronových sítí s dopředním šířením signálu. Všechny vrstvy perceptronu jsou propojeny s neurony následující vrstvy, co je dobře vidět na obrázku 12.

Reed & Marks (1998) se dělí tvoření vrstevnatého Perceptronu:

- Množstvím vstupních uzlů (= neuronů), které tvoří vstupní vrstvu;

- Jedním nebo dvěma skrytými vrstvami;
- Jednou výstupní vrstvou;



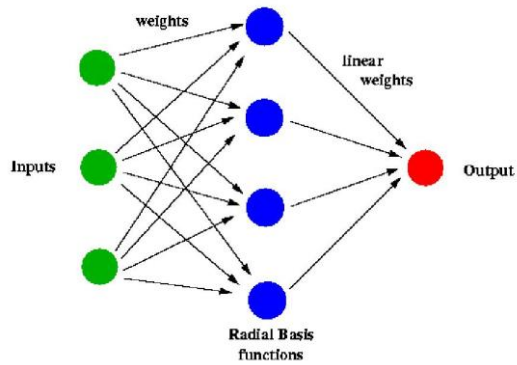
Obr. 12: Multi-layer perceptron (URL 11)

Každý vrstevnatý perceptron se úspěšně používá pro řešení různých složitých úkolů a mají 3 charakteristické vlastnosti (AirPortal, 2014):

- 1) Každý neuron má nelineární aktivační funkce;
- 2) Perceptron má několik skrytých vrstev;
- 3) Neurony ve dvou sousedních vrstvách jsou vzájemně plně propojeny;

3.4.3 Radiální bázové funkce (Radial Basis Function Networks)

Modely UNS - založené na Radiální bázové funkce (RBF) - byly vyvinutý Powellem (1987) a Broomheadem a Lowem (1988). Spolu s vrstevnatým Perceptronem se jedná o modely, které jsou často aplikovaný pro predikci časových řad vodohospodářských veličin. RBF modely vykazují menší chyby a vyšší rychlost konvergenčí při učení (Lendasse & kol., 2003). Většinou má taková síť tři vrstvy: vstupní, skrytou a výstupní. Schéma této sítě je zobrazena na obrázku 13:



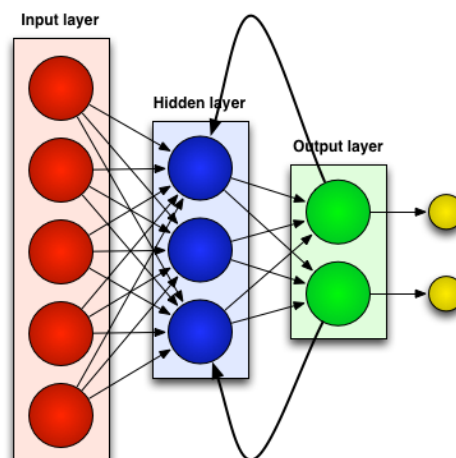
Obr. 13: Radial Basis Function Neural Network (URL12)

V roce 1990 Girosi a Poggio ukázali že RBFNN poskytují lepší aproximace. Jedna se o existence vztahu mezi několika vstupních proměny a jedné výstupní proměny. Tento vztah není známý a síť snaží vytvořit aproximátor (model „černé skřínky“) mezi vstupy a výstupy. Aproximátor má být kalibrován tak, aby nejlépe reprezentovat vstupně–výstupní závislost. Síť likviduje soubor vstupních a výstupních dat, které považované za trénovací soubor dat k aproximátoru (Bishop, 1995; Lendasse & kol., 2003).

3.4.4 Rekurentní Neuronové Síť (Reccurent Neural Network)

Rekurentní neuronová síť (RNS) je propojená tak, že neurony v první vrstvě jsou propojeny s neurony následující vrstvy, s neurony předchozí vrstvy a s neurony stejné vrstvy. To znamená, že RNS obsahují zpětné vazby. Díky těmto vazbám má RNS výhodu v tom, že síť může „nahlízet“ zpět (Maier & Dandy, 2000).

Příklad propojení neuronů se zpětnými vazbami znázorňuje obrázek 14:

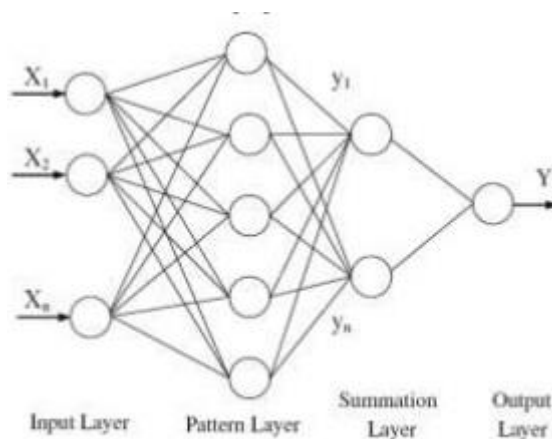


Obr. 14: Reccurent Neural Network (URL 13)

V současné době je velký zájem o to, jak najít efektivní způsoby trénování Rekurentních Neuronových Sítí, protože mají velký potenciál na trénování a mají schopnost zapamatovat si informace v jejich skryté vrstvě na dlouhou dobu. Takže jsou docela biologicky realistické. RNS se používají při řešení predikčních problémů. Sítě RBF jsou ideální pro časově závislá data, protože má v sobě spojení, které slouží jako vstupy k predikci dalších časových kroků v budoucnosti (Aqil & kol., 2007; Hinton & kol., 2013, in litt.).

3.4.5 Obecné regresní sítě (Generalized Regression Neural Network)

Obecné regresní sítě (ORS) nevyžaduje opakovaný trénovací proces jako sítě se zpětným propagací chyby. Hlavní výhodou modelu ORS je sítě je kvalitnější učení při existenci rozsáhlých vstupních dat. Architektura ORS (Obr. 15) je sloužena ze 4 vrstvy: vstupní, vzorovou, sumační a výstupní. Počet neuronů ve vstupní vrstvě závisí na celkovém počtu pozorovaných parametrů. První vrstva je spojena se vzorovou vrstvou a v této vrstvě každý neuron představuje trénovací vzor. Vzorová vrstva je spojena se sumační vrstvou. Sumační vrstva obsahuje jenom dva neurony. Výstupní vrstva zpracovává informace od neuronů ze sumační vrstvy. Hannan & kol. (2010) použili ORS pro predikci průtoku. V jejich modelu ORS je výstup stanoven jako podíl výstupní hodnoty prvního a druhého neuronu sumační vrstvy.



Obr.15: Generalized Regression Neural Network (Hannan & kol., 2010)

4. Metodický postup aplikace modelu umělých neuronových sítí

Aplikace modelů UNS ve vodním hospodářství je popsána podle článků Maiera a Dandy (2000). Autoři nám uvádějí přehled základních kroků, které je nutno učinit v případě, že bude problém řešen metodami neuronových sítí.

4.1 Výkonnost UNS

Před tím než začneme vytvářet modely neuronové sítě je potřeba zvážit kritéria, která ovlivňují její výkonnost. Kritériem výkonnosti je kritérium, které vyhodnocuje simulační schopnost neuronové sítě. Ve většině aplikací kritérium výkonnosti ovlivňuje: rychlost trénování a přesnost predikce.

Rychlost trénování sítě je závislá na optimalizační metodě, velikosti trénovacího souboru a na velikosti sítě. Optimalizační metoda a její parametry mají velký vliv na konvergenční rychlost. Čím menší počet trénovacích dat, tím větší rychlost trénování. Rychlost učení se snižují s rostoucím souborem trénovacích dat (Rojas, 1996).

Dalším kritériem je funkce rychlosti zpracování. To je čas mezi předložením vstupů a příjmem výstupů pro trénovanou síť. Tato funkce je závislá na počtu vah (w_i) a na typu spojení mezi nimi. Aby se maximalizovala rychlost, je potřeba nechávat počet vah co nejmenší a spojení mezi nimi co nejjednodušší.

Dalším kritériem je přesnost predikce. Tento kritérium byl použit například v případě Hecht-Nielsen (1990), Lachtermacher & Fuller (1994) a Shukla & kol. (1996). Kritérium výkonnosti je ovlivněno optimalizačním algoritmem. Algoritmus má schopnost se vyhnout lokálnímu minimum v oblasti chyb. Přesnost predikce taky závisí na schopnosti generalizace. Tato schopnost je definována jako schopnost modelu dobře vykonávat údaje, které nebyly použity ke kalibraci a je to funkce poměru počtů trénovaných vzorů a počtu vah (Cheng & Titterington, 1994). Pokud je tento poměr příliš malý, tak může dojít k přeučení sítě.

4.2 Příprava dat

Důležitým krokem před začátkem vytvoření modelu UNS je příprava dat. Obvykle přeprava dat má dva kroky:

- 1) Rozdělení dat
- 2) Standardizace dat

4.2.1 Rozdělení dat

Nejčastěji v praxi se data rozdělují do třech podsouborů: trénovací soubor dat (*training set*), testovací soubor dat (*testing set*) a nezávislý validační soubor dat (*an independent validation set*). Poslední se používá k validaci, trénovací soubor dat se používá pro učení sítí a testovací soubor dat slouží pro hodnocení výkonnosti učení v jeho různých fázích (Minns & Hall, 1996). Rozdělení dat do třech souborů použili například (Coulibaly & kol. 2000) a rozdělení dat jenom do dvou souborů použili (Riad & kol., 2004).

Špatnou predikce lze očekávat, když validační data obsahují hodnoty mimo rozsah dat používaných při trénování. Při omezení dostupných dat, může být obtížné sestavit reprezentativní validační soubor dat. Jeden ze způsobů, který maximalizuje využití dostupných dat je metoda *holdout* (Masters, 1993). Základní myšlenkou této metody validace je vymezení malého validačního souboru dat a zbývající data se určují k trénování. Celý proces se opakuje, jestli všechna dostupná data nepoužijí v rámci trénování. Způsob rozdělení dat závisí na použité metodě validace. Metoda křížové validace (*cross-validation*), která se používá pro určení ukončení trénování a porovnání schopnosti generalizace různých modelů (Burden & a kol., 1997). V metodě křížové validace nezávislý testovací soubor se používá pro hodnocení výkonnosti modelu v různých fázích učení.

4.2.2 Standardizace

Pod pojmem standardizace rozumíme transformaci dat, která dostupná data rozdělí do požadovaného rozsahu. Standardizaci dat je nutno provést před začátkem trénování neuronové sítě, aby všechny proměnné měly stejný rozsah. Rozsah vstupní a výstupní dat musí být zmenšen na interval, který je shodný s oborem hodnot aktivační funkce (Minns & Hall, 1996).

Například pro Vícevrstevný perceptron se sigmoidální aktivační funkcí, jejíž data musí být šlakována na interval (0;1). Transformace výstupních dat je nutností a plyne z definice této funkce. Po natrénování neuronové sítě a provedení výpočtu pro nová vstupní data jsou pak výstupy neuronové sítě inverzní transformací předvedeny na interval skutečných hodnot výstupních dat (Starý, 2004).

4.3 Volba vstupů do neuronových sítí

V poslední době se použití modelů umělých neuronových sítí pro aplikace ve vodních zdrojích rozrostla. Výběr vstupů v jakémkoliv modelu je velmi důležitá věc (Faraway & Chatfield, 1998; Kaastra & Boyd, 1995). Ale ve většině aplikací dané problematice věnovaná malá pozornost. Příčinou malé pozornosti je to, že postup pro

UNS se řídí daty, kde statistické metody se řídí pomocí modelu. UNS mají schopnost stanovit, které vstupy jsou kritické (Lachtermacher & Fuller, 1994). Čím síť má více vstupů, tím má větší rozměr. A čím je síť větší, tím má nižší rychlost zpracování a vyšší množství dat, které jsou potřebné k efektivnímu odhadu vah.

Maier & kol. (2005) nám ukazují dvě metodologie pro výběr vstupu do neuronové sítě. První metoda je bezmodelový (*model-free*) postup, který využívá odhad vzájemné informace (*Mutual information*), která charakterizuje závislost mezi potenciální vstupní a výstupní proměnnou modelu. K usnadnění výpočtu vzájemné informace v případě více vstupů, se používá částečná vzájemná informace (*Partial mutual information*).

Druhý způsob stanovení souboru vstupních dat je založený na metodě (SOM) - samoorganizující mapa. Chceme-li zjistit, jaký vstup určuje významný vztah s výstupem (závislé) proměnné, použijeme hybridní genetický algoritmus a obecnou regresní neuronovou síť (General Regression Neural Network), (Bowden & kol., 2005).

4.4 Volba architektury sítě.

Dalším důležitým krokem je volba architektury neuronové sítě. Architektura neuronové sítě určuje počet vah, a šířením signálu neuronové sítě. Odhad vhodné architektury sítě je jeden z nejdůležitějších a také jeden z nejtěžších úkolů v procesu výstavby modelu (Maier & Dandy, 2000).

Zvolenou architekturou v mé práci pro predikce odtoku je vícevrstevný perceptron. Vícevrstevný perceptron je model neuronových sítí s dopředním šířením signálu. Všechny neurony jedné vrstvy jsou propojeny s neurony následující vrstvy. Teď je potřeba určit počet vah a jejich uspořádání. Toto stanovení může řešit topologie.

4.4.1 Topologie

Pomocí topologie určujeme počet vah a uspořádání neuronů. To je dáno počtem skrytých vrstev a počtem neuronů v každé vrstvě. Kumar (1993) uvádí, že je lepší určovat počet vah počtem skrytých vrstev než počtem neuronů ve vrstvách.

Menší sítě mají lepší generalizační schopnost (Castellano & kol., 1997), vyžadují méně fyzických zdrojů, mají vyšší rychlost zpracování (během trénování a testování), mají menší náročnost na hardware (Bedis & Georgiopoulos, 1994) a provádějí extrakci informace zpravidla jednoduše (Towel & kol., 1991). Na druhou

stranu, menší neuronové sítě se trénují složitěji, protože chybové funkce obsahují více lokálních minim (Bedis & Georgiopoulos, 1994).

Výhody velkých neuronových sítí jsou v tom, že se učí rychleji (Plaut & Hinton, 1987) a mají moc dobrou schopnost ignorovat lokální minima (Rumelhart & kol., 1986). Ale nevýhody velkých umělých neuronových sítí také existují. Velké sítě vyžadují větší počet trénovacích vzorů a jsou implementačně nákladné (Bedis & Georgiopoulos, 1994).

4.4.2 Prořezávací algoritmus (Pruning algorithms)

Pomocí prořezávacího algoritmu zjistíme optimální počet neuronů. Základní myšlenka prořezávacího algoritmu je ta, že začneme velmi komplexní neuronovou sítí a postupně odebíráme neurony. A pokud neklesne výkonnost sítě, tak pokračujeme do té doby, dokud nám klesne pod nějaký, předem zvolený, práh. Jakmile nám klesne pod nějaký práh, tak ta předchozí architektura neuronové sítě má nejmenší složitost. Jedná se tedy o optimální architekturu UNS (Hecht-Nielsen, 1990), Několik autorů ve své literatuře využili tento algoritmus, například (Sietsma & Dow, 1991; Karnin, 1990; Chung & Lee, 1992).

4.4.3 Konstruktivní algoritmus (Constructive algorithm)

Konstruktivní algoritmus postupuje úplně opačným způsobem než prořezávací. Začíná odspoda. Na začátku učení vytvoří nejjednodušší neuronovou sítí, postupně přidává neurony, sítí zvyšují se tím složitost UNS a současně s tím roste schopnost trénování UNS. To platí pouze do určité míry, protože pokud přidáme příliš hodně neuronů, tak se učení už nezlepší od určitého počtu neuronů. Konstruktivní algoritmus používal ve svých příkladech (Hirose & kol., 1991; Setiono & Hui, 1995; Chen & kol., 1997).

4.5 Trénování sítě.

Trénování modelů UNS je optimalizační proces. Výsledkem optimalizace je nalezení hodnot vah modelu UNS. (Maier & Dandy, 2000). Při trénování, sítí učí vztahy mezi dostupnými daty, a tím vzniká jedinečná sítí pro daná data (Reed & Marks, 1998). V průběhu trénování do neuronové sítě vstupují postupně trénovací vzory. V závislosti na velikostech vah a prahů pak vystupují z výstupních neuronů hodnoty výstupních trénovacích vzorů. Výstupy ze sítě se porovnají s očekávanými výstupy z trénovacího souboru. A dokud chyba nebude minimální, optimalizační proces se opakuje (Starý, 2004).

Optimalizační proces proveden jenom za použití lokálních nebo globálních metod. Lokální metody lze rozdělit do dvou hlavních kategorií: prvního a druhého řádu. Metoda prvního řádu je založená na lineárním modelu (gradient (tečný vektor) klesání), zatímco metoda druhého řádu je založená na kvadratickém modelu (Battiti, 1992). V obou případech je účelem procesu adaptace vah, pro kterou by byla chyba sítě nejmenší.

Rovnice pro aktivaci hodnot váhy má následující obecný tvar (Parisi & kol., 1996):

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \gamma_n \mathbf{d}_n \quad (4)$$

kde w_n je vektor vah, γ_n je velikost kroku, d_n je vektor, definující směr klesání a index n označuje iteraci optimálního procesu. Hlavní rozdíl mezi většinou algoritmu je volba hodnoty d_n , která určuje rychlost konvergence a výpočetní složitost optimalizaci.

4.5.1 Lokální metody I. a II. řádu – algoritmus zpětné propagace chyb

Lokální metody I. řádu – algoritmus zpětné propagace chyb

Lokální metody I. řádu jsou reprezentované metodou největšího spadu gradientu. Hlavním pravidlem metody největšího spadu gradientu je v stanovení gradientu – chybové funkce a sestupu po něm dolů (Parisi & kol., 1996).

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \gamma_n \nabla_{\mathbf{w}_n} E. \quad (5)$$

kde $\nabla_{\mathbf{w}_n} E$ je gradient chybové funkce.

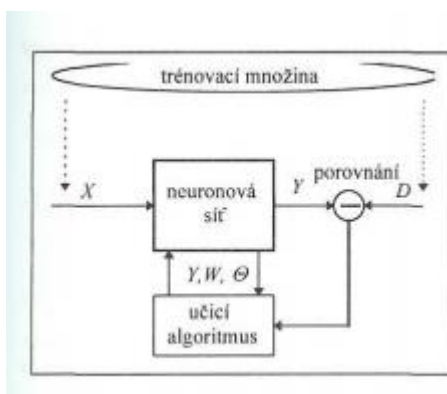
Algoritmus zpětné propagace chyby (ZPC) pro optimalizace vrstevnatého perceptronu je nyní používán nejčastěji (Rumelhart & kol., 1986). Poprvé byla tato metoda popsána Galushkinem (Galushkin, 1974) a současně a nezávisle také Polem Werbosem (Werbos, 1974). Dále byl rozvinut v roce 1986 Davidem I. Rumelhartem, Hintonem a Williamsem (Rumelhart & kol., 1986) a současně a nezávisle S.I. Barcevm a V.A. Ohoninym (Barcev & kol., 1986).

Princip algoritmu je stejný, jako ve škole. Pokud se při zkoušení zjistí, že žák neumí nějakou část látky, tak musí se k ní vrátit a zpětně se jí doučit. Podobným

způsobem funguje algoritmus ZPC. Neuronovou síť považujeme za žáka, zkoušení za mechanismus, ve kterém testujeme, zda-li neuronová síť odpovídá na vstupní vektory přesně podle trénovací množiny. Jí považujeme za látku, kterou se musí žák naučit. Pokud zjistíme, že síť nereaguje tak, jak je potřeba, musíme jí měnit váhy tak dlouho, dokud nezačne reagovat správně (Šnorek & Jiřina, 1996).

Základní operace (Obr. 16) podle autorů Reeda & Marksa (1998) algoritmu zpětné propagace chyby jsou:

- 1) Inicializace
- 2) Předložení trénovacích vzorů a jejich šíření pro získání výstupu.
- 3) Srovnání výstupů s požadovanými hodnoty a spočítávání chyb.
- 4) Zpětná propagace chyby a modifikace vah
- 5) Opakování kroků 2 a 3 dokud chyba neklesne pod zvolený práh, nebo není dosažen počet maximální iteraci.



Obr. 16: Algoritmus zpětné propagace chyby (URL 14)

Chybová funkce posuzují schopnost UNS modelovat vzorová data. Chybová funkce je sumou parciálních chyb pro jednotlivé vzory, nebo také sumou součtu druhých mocnin odchylek reálných a očekávaných výstupu tréninkového vzorů vynásobenou koeficientem $1/2$ (Šíma & Neruda, 1996).

Globální chybová funkce nejčastěji vypadá takto:

$$E(t) = \frac{1}{2} \sum (d_j(t) - y_j(t))^2. \tag{6}$$

kde $E(t)$ je globální chybová funkce v diskrétním čase t , $y(t)$ je predikční výstup v diskrétním čase t , $d_j(t)$ je počítaný výstup v diskrétním čase t (Maier & Dandy, 1998). Koeficient $1/2$ byl dodefinován kvůli jednoduššímu derivování. Derivace je využita při algoritmu zpětné propagace chyb (Šíma & Neruda, 1996).

Jednotlivé kroky při použití algoritmu se zpětnou propagací chyby.

Jednotlivé kroky při použití algoritmu zpětné propagace chyby nám uvádí autoři Šnorek & Jiřina (1996):

Krok 1. Inicializace

Všechny váhy v síti nastavíme náhodně na hodnoty v doručovaném rozsahu (-0.3; 0.3)

Krok 2. Předložení trénovacích vzorů a jejich šíření pro získání výstupů

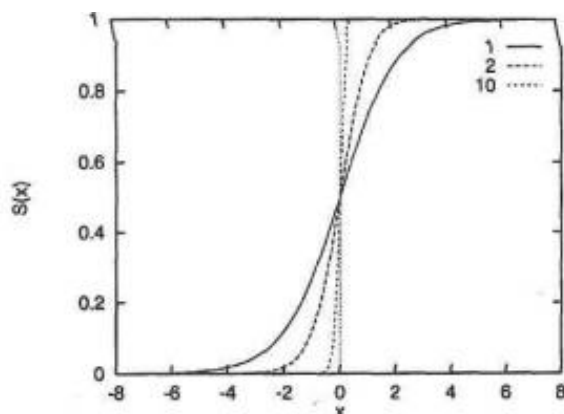
Vybereme vstupní vzor z trénovací množiny a šíříme jeho signál neuronovou sítí. Po vrstvách směrem od vstupu k výstupu síť počítáme výstupy jednotlivých neuronů dle vztahů:

$$y = F(\sum x_i w_i + \theta). \quad (7)$$

Výstupní nelineární funkce F je sigmoida definovaná vztahem:

$$F(\varphi) = \frac{1}{1+e^{-\gamma\varphi}}. \quad (8)$$

kde γ je koeficient, který určuje strmost křivky v počátku souřadné soustavy. Příklad je na obrázku 17.



Obr. 17: Vstupní a výstupní hodnoty uvažované v rozsahu (0,1) (Šnorek & Jiřina, 1996)

Krok 3. Srovnání s výstupy s požadovanými hodnoty a spočítávání chyb

V kroku srovnání nejprve vypočítáme chybu:

$$E = \frac{1}{2} \sum (y_i - d_i)^2. \quad (9)$$

kde y_i je výstup sítě, d_i je předpokládaný výstup. Čím bližší si tedy budou hodnoty d_i a y_i , tím bude chyba sítě nižší.

Krok 4. Zpětné propagace chyby a modifikace vah

Vypočítáme chybu pro výstupní vrstvu:

$$\delta_i^o = (d_i - y_i^o) \gamma y_i^o (1 - y_i^o) \quad (10)$$

$$\frac{\partial y_i^o}{\partial \varphi_i} = \gamma y_i^o (1 - y_i^o) \quad (11)$$

$\frac{\partial y_i^o}{\partial \varphi_i}$ je derivace sigmoidy.

$d_k - y_k^o$ je RMSE (root mean square error) chyba výstupních neuronů.

Tímto jsme získali gradient chybové funkce, ze kterého vypočítáme hodnoty, kterými budeme váhy upravovat. Vektor přírůstku vah Δw vypočteme ze vztahu:

$$\Delta w = -\eta \nabla E \quad (12)$$

Váhy upravujeme vždy η násobkem záporné hodnoty gradientu. Změna znaménka je nezbytná, jelikož gradient je orientován směrem k vrcholu.

Pro výstupní vrstvu platí:

$$\Delta w_{ij}^o(t) = \eta \delta_i^o(t) y_j^h(t) + \alpha \Delta w_{ij}^o(t-1) \quad (13)$$

$$\Delta \theta_i^o(t) = \eta \delta_i^o(t) + \alpha \Delta \theta_i^o(t-1) \quad (14)$$

kde koeficient η je nazýván učící konstanta. Koeficient α je to tzv. momentum, která nám říká, jak dlouho se budeme držet určitého směru, než zabočíme podle nové změny gradientu. Člen s momentem byl přidán pro překonávání lokálních minim chybové funkce.

Váhy a prahy upravíme dle vztahů:

$$\mathbf{w}_{ij}^o(t-1) = \mathbf{w}_{ij}^o(t) + \Delta \mathbf{w}_{ij}^o(t) \quad (15)$$

$$\boldsymbol{\theta}_i^o(t+1) = \boldsymbol{\theta}_i^o(t) + \Delta \boldsymbol{\theta}_i^o(t) \quad (16)$$

Výpočet vah ve skryté vrstvě.

Váhy a prahy se ve skrytých vrstvách se počítají na základě stejných úvah jako váhy a prahy vrstvy výstupní. Nyní se pro jednoduchost zaměříme na skrytou vrstvu, která těsně předchází vrstvu výstupní.

Pro každou váhu skryté vrstvy spočteme $\frac{\partial E}{\partial w_{ij}^h}$ dle vztahu:

$$\frac{\partial E}{\partial w_{ij}^h} = \frac{\partial E}{\partial y_k^o} \frac{\partial y_k^o}{\partial \varphi_k^o} \frac{\partial \varphi_k^o}{\partial y_i^h} \frac{\partial y_i^h}{\partial \varphi_i^h} \frac{\partial \varphi_i^h}{\partial w_{ij}^o} \quad (17)$$

$$w_{ki}^o = \frac{\partial \varphi_k^o}{\partial y_i^h} \quad (18)$$

$$\delta_k^o = \frac{\partial E}{\partial y_k^o} \frac{\partial y_k^o}{\partial \varphi_k^o} \quad (19)$$

$$\delta_i^h = \frac{\partial E}{\partial y_k^o} \frac{\partial y_k^o}{\partial \varphi_k^o} \frac{\partial \varphi_k^o}{\partial y_i^h} \frac{\partial y_i^h}{\partial \varphi_i^h} \frac{\partial \varphi_i^h}{\partial w_{ij}^o} \quad (20)$$

Ze vztahu (20) vidíme, že δ_i^h vypočteme tak, že vynásobíme chybu δ_k^o (19) výstupní vrstvy hodnotou příslušného váhového koeficientu. Tím přeneseme hodnotu chyby na skrytou vrstvu.

Protože ve výstupní vrstvě bude více neuronů a ne jen jeden, jak jsme doposud uvažovali, bude se nutně výstup skrytých neuronů větvit. Z tohoto důvodu je hodnota δ_i^h dána součtem příspěvků $w_{ki}^o \delta_k^o$ od všech neuronů výstupní vrstvy (17).

Obecně toto zapíšeme jako:

$$\delta_i^h = y_i^h(1 - y_i^h) \sum_{k=1}^n w_{ki}^o \delta_k^o \quad (21)$$

Hodnoty Δw_{ij}^o , $\Delta \theta_i^o$, θ_i^h , w_{ij}^h spočteme dle vztahů:

$$\Delta w_{ij}^l(t) = \eta \delta_i^l(t) y_j^{l-1}(t) + \alpha \Delta w_{ij}^l(t-1) \quad (22)$$

$$\Delta \theta_i^l(t) = \eta \delta_i^l(t) + \alpha \Delta \theta_i^l(t-1) \quad (23)$$

$$w_{ij}^o(t+1) = w_{ij}^o(t) + \Delta w_{ij}^o(t) \quad (24)$$

$$\theta_i^o(t+1) = \theta_i^o(t) + \Delta \theta_i^o(t) \quad (25)$$

$x_j(t)$ je j-tý vstup neuronové sítě (pouze pokud se jedná o skrytou vrstvu bezprostředně následující vstupy sítě) nebo j-tý výstup předcházející skryté vrstvy (ostatní skryté vrstvy)

Výraz pro vypočet δ lze zobecnit pro celou síť následovně:

$$\delta_i^{h-1} = y_i^{h-1}(1 - y_i^{h-1}) \sum_{k=1}^n w_{ki}^h \delta_k^h \quad (26)$$

Kde skryté vrstvy jsou číslovány vzestupně směrem od vstupů k výstupům. Pokud tento vztah aplikujeme postupně od výstupní vrstvy ke vstupům, vypočteme

hodnoty δ pro celou síť. Odtud získal algoritmus zpětné propagace svůj název, protože chybu šíříme zpětně od výstupů ke vstupům.

Krok 5. Ukončení výběru vzorů z trénovací množiny.

Jestliže jsme předložili síť všechny vzory z trénovací množiny, pokračujeme krokem 6, jinak se vracíme na krok 2.

Krok 6. Ukončení procesu učení.

Jestliže byla chyba neuronové sítě za poslední epochu menší, než námi zvolené kritérium, nebo jsme vyčerpali počet epoch, pak učení ukončíme. Jinak pokračujeme krokem 2 (Šnorek & Jiřina, 1996).

Lokální metody II. řádu.

Zatímco metoda prvního řádu je založená na lineárním modelu (gradient (tečný vektor) klesání), metoda druhého řádu je založená na kvadratickém modelu (Battiti, 1992). Optimalizační metody druhého řádu jsou reprezentovány následujícími algoritmy: Newtonův algoritmus a Levenberg-Marquardtův algoritmus (Reed & Marks, 1998).

Newton algoritmus

Newtonova metoda je základní algoritmus, podle kterého se hodnotí optimalizační metody. Váhy jsou Newtonovou metodou stanoveny (Parisi & kol., 1996):

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \mathbf{H}^{-1} \nabla_{\mathbf{w}_n} E \quad (27)$$

Kde \mathbf{w}_{n+1} je vektor vah získaný při iteraci $n+1$, \mathbf{H}^{-1} inverzní matice k Hesseve matici, $\nabla_{\mathbf{w}_n} E$ gradient chybové funkce (Maier & Dandy, 2000).

Hessova matice (28) se používá při optimalizaci modelu II. řádu. Hessova matice je čtvercová matice druhých parciálních derivací skalární chybové funkce (Reed & Marks, 1998).

$$H = \begin{bmatrix} \frac{\partial^2 E}{\partial x_1^2} & \frac{\partial^2 E}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 E}{\partial x_1 \partial x_n} \\ \frac{\partial^2 E}{\partial x_2 \partial x_1} & \frac{\partial^2 E}{\partial x_2^2} & \cdots & \frac{\partial^2 E}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 E}{\partial x_n \partial x_1} & \frac{\partial^2 E}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 E}{\partial x_n^2} \end{bmatrix} \quad (28)$$

Determinant Hessove matice se nazývá Hessian H . To je chybové funkce podle vah je matice druhé derivace s prvky:

$$h_{ij} = \frac{\partial^2 E}{\partial w_i \partial w_j} \quad (29)$$

Znalost Hessovy matice je důležitá z mnoha důvodů. Konvergence většiny optimalizačních algoritmů se řídí vlastnostmi Hessove matice. V metodách druhého řádu, se matice používá přímo pro výpočet směrů hledání. Některé prořezávací algoritmy používají Hessove informace pro řešení odebrání vah. A Hessova matice je velice užitečná při hledání architektury sítě (Reed & Marks, 1998). Použití Hessove matice v UNS popsali (Mizutani & kol., 2008).

Algoritmus Levenberg-Marquardt.

Nevýhodou Newtonova algoritmu je zaměření jenom na pozitivně definitivní H , ale ten může být nejasný v případě obecně nelineární funkce, hlavně když je v oblasti vzdálené od minima. Algoritmus Levenberg-Marquardt považuje se za kompromis mezi klasickým Newtonovým algoritmem, který rychle konverguje minima a metodou největšího spadu, která konverguje všude, ale tato metoda je pomalá (Reed & Marks, 1998).

4.5.2 Globální metody

Globální optimalizační algoritmy slouží k nalezení vhodného (optimálního) řešení problému v případech, kdy není znám matematický popis řešení problému. Globální metody hledají řešení pro globální minimum bez rizika nalezení lokálního minima. Výhoda globálních metod je ta, že jsou v principu velmi jednoduché, jsou založeny na různých způsobech prohledávání prostoru řešení a nevyžadují aktivační funkci. Nevýhoda je v tom že, prakticky je nelze využít bez počítače (Štroner, 2012; Šíma & Neruda, 1996).

Štroner (2012) rozdělil globální optimalizační metody podle vlastnosti:

- Pravděpodobnostní metody (Monte-Carlo);
- Evoluční metody;
- Metody kolektivní inteligence (Swarm Intelligence);
- Přímé vyhledávací metody;

Simulované žíhání

Simulované žíhání (Pravděpodobnostní metody) je jiná optimalizační metoda pro nalezení globálního minima, která je daná fyzikální analogií se žíháním slitin (Maier & Dandy, 2000).

Autoři Reed & Marks (1998) nám ukázali jako ilustrace simulovaného žíhání, ztuhnutí kapaliny do pevné látky. Kapalina to je neuspořádaný systém, který má vysokou energii. Jakmile energie klesá, systém chladne a tuhne. V případě, že kapalina se ochladí příliš rychle, má tendenci ztuhnout do množství malých krystalů s mnoha nedokonalostmi. Tyto nedokonalosti jsou stránky vnitřního stresu, které mají vysokou energii. Jestli systém tuhne pomalu, tak má tendenci formování velkého množství dobře uspořádaných krystalů s nízkou energií.

Takže při velkých teplotách síť se chová jako nahodilý model, a schopná skákat z jednoho lokálního minima na jiný. Při nízkých teplotách síť se chová jako deterministický model. Ale pokud systém ochlazován pomalu, může dojít ke globálnímu minimu (Maier & Dandy, 2000; Reed & Marks 1998).

Genetické algoritmy

Genetické algoritmy (Evoluční metody) jsou základní optimalizační metody, které se používají v trénování sítí a v řadě dalších problémů. Při použití genetických algoritmů pro trénování, váhy jsou zvolené náhodně od začátku trénování a zakódované v binární řadě (Maier & Dandy, 2000).

Z názvu vyplývá, že genetické algoritmy jsou založené na analogii k biologickým evolučním mechanismům. Základní idea tohoto algoritmu je ta, že je to konkurence mezi alternativními řešeními a „přežitím nejschopnějších“. Přes několik generací, populace adaptuje k jejímu životnímu prostředí. Zřídka se stávají nevýhodné znaky populace a průměrná způsobilost má tendenci se zvyšovat v průběhu času (Reed & Marks, 1998).

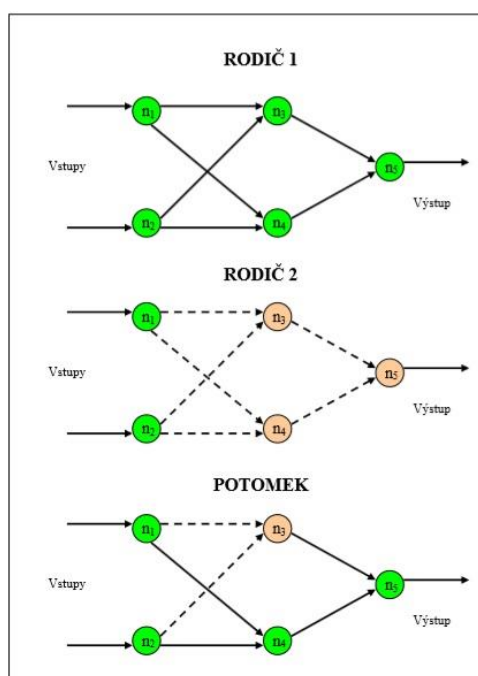
Výhoda je v tom, že algoritmus potřebuje málo problém-specifické informace. Aby použit tento metod na specifickou problému je potřeba jenom vhodná funkce. Tato funkce vyhodnocuje individuální řešení a vrátí kvalitu (Reed & Marks, 1998).

Základní operace jsou:

1) Výběr na základě způsobilosti

Genetický algoritmus vybírá parametry pro reprodukce s úměrnou pravděpodobnosti k jejich způsobilosti. Parametry s větší způsobilostí budou vybrány, ale parametry s nižší způsobilostí taky mají šance byt vybrané (Šíma & Neruda, 1996).

2) Křížení

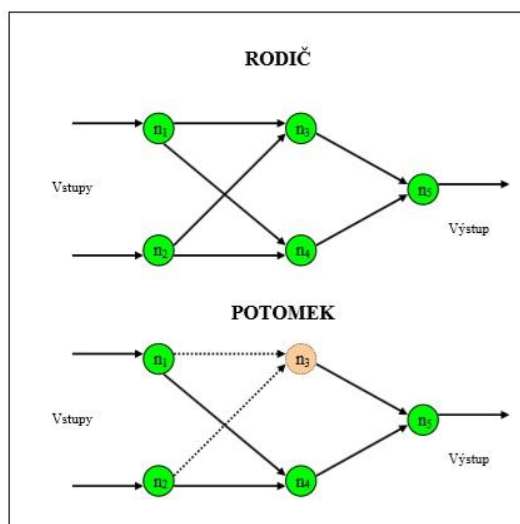


Obr. 18: Křížení (převzato z Starý, 2004)

Křížení vyžaduje dva rodiče v neuronové síti je možno jej implantovat tak, že ze dvou NS – rodičů – vytváříme třetí – potomka. Na obrázku 18 je znázorněno křížení dvou neuronových sítí. Je to nejčastější typ křížení, které se jmenuje *jednobodové*. Potomek obdržel skrytý neuron n_3 od druhého rodiče a první skrytý neuron od prvního rodiče. Někdy se používá i *vícebodové* křížení, které se jmenuje *uniformní* křížení. Potomek náhodně dodrží skrytý neuron (v závislosti na způsobu zakódování). Výhodou vícebodového křížení je, že u něj nezáleží na pořadí, v jakém rodičů jsou informace zakódovány (Starý, 2004; Šíma & Neruda, 1996).

3) Mutace

Mutace vyžaduje jednoho rodiče. Během ní je náhodné procento neuronů změněno modifikací vah ve hranách, které je spojují. Frekvence změn a velikost jejich změny je zadaná uživatelem. Na obrázku 19 jsou dvě neuronové sítě: jedna je neuronová síť rodiče a druhá potomka. NA tomto příklade skrytý neuron n_3 mutoval z rodiče na potomka. Výhodou mutace je v tom že, dokáže vytvářet zcela nové hodnoty parametrů a zabraňuje tak uvíznutí celého procesu v oblasti lokálního extrému optimalizované funkce (Starý, 2004; Šíma & Neruda, 1996).



Obr. 19: Mutace (převzato z Starý, 2004)

Hlavní nevýhodou genetických algoritmu je množství vypočtu potřebné k vyhodnocení a skladování velké populace na více generací (Reed & Marks, 1998). Jiná nevýhoda genetického algoritmu je rychlost zpracování. Ta je menší ve srovnávání s lokálními metody druhého řadu (Maier & Dandy, 2000). Také nastavování parametrů (velikost populace, metoda křížení, hodnocení mutace) zabírá hodně času a má vliv na výkonnost sítě (Reed & Marks, 1998).

4.5.3 Zastavovací kritéria

Kritéria se používají pro rozhodnutí, kdy tréninkový proces ukončit, a jsou životně důležité. Rozhodují determinovat, jestli model byl optimálně nebo sub-optimálně naučen. Špatně nastavené kritéria mohou způsobit předčasnému přeučení (*overfitting*) modelu UNS (Maier & Dandy, 2000). Amari & kol. (1997) ukazují na některé případy, kde trénování bylo zastaveno příliš brzy. Když takové případy nestávají je zřejmě použít metodu křížové validaci (*cross-validation*).

4.6 Validace

Jakmile bude trénovací fáze ukončená, tak výkon sítě musí být ověřen na nezávislém souboru dat pomocí vybraných kritérií. Při přípravě dat je velmi důležité,

aby validační soubor byl nezávislý na souboru trénovacích dat, který byl oddělen při přípravě dat (Maier & Dandy, 2000). Lachtermacher & Fuller (1994) ukazují 3 nevýhody rozdělení dat na tři části. První nevýhoda je rozsah dat. Pro malý soubor dat je rozdělení do 3 souborů obtíží proveditelně. Druhou nevýhodou je nemožnost rozdělení dat do nezávislých souborů. Třetí nevýhoda je ovlivnění souborů náhodným šumem (Dawson & Wilby, 2001).

5. Aplikace umělých neuronových sítí ve vodním hospodářství

UNS byly již úspěšně použity pro ohodnocení vlivu klimatických změn na hydrologické a ekologické parametry toku (Poff, 1996), pro predikci slanosti (DeSilets, 1992), pro popis výskytu modrozelených řas (Maier & Dandy, 1998), koncentrace ozónu (Roadnight, 1997), koncentrace řas (Whitehead, 1997), pro predikci srážek (Goswami & Srividya, 1996), pro predikci odtoku (Smith & Eli, 1995) a odhad hustoty kladení jiker hnědých pstruhu (Maier & Dandy, 2000). Následující studie pracuje s MLP pro simulace odtoku v Nové Zélande.

5. 1. Případová studie.

Autoři této studii Fernando & Kerr (2003) jedné z prvních se zabývali analýzou možnosti použití umělých neuronových sítí pro predikci odtoku a srážek v Nové Zélande. Modely UNS byly srovnané s konvenčními hydrologickými modely. Hlavní výhodou použití modelu UNS je oproti konvenčním hydrologickým modelům, jejich schopnost se učit ze vzorců.

V této studii byla použita pro predikce odtoku v povodí Kapakapanu nejpopulárnější architektura – vícevrstevný perceptron s jednou skrytou vrstvou a učícím algoritmem se zpětnou propagací chyby. Pro vytvoření modelu byly použity dříve pozorované hodnoty srážek a průtoku. Trénovací soubor dat byl sestaven taky z pozorovaných hodnot. Pomocí trénování se model UNS naučil modelovat nelineární vztah mezi predikovaným odtokem a vstupními veličinami. Validací soubor dat sloužil pro ohodnocení výkonnosti sítě během trénování. V testovacích fázích natrénovaná síť sloužila pro predikce odtoku.

Vytváření neuronové sítě začíná tím, že nahrajeme vstupní vektory do vstupní vrstvy. Pro zlepšení výkonnosti sítě byla použita metoda nelineární standardizace. Všechny vstupní a výstupní data, z důvodu použití hyperbolické tangenty jako aktivační funkce, standardizuje do intervalu (-1;1) Neurony ve vstupní vrstvy nepočítají nic, jenom posílají kopii vstupních proměnných do neuronu ve skryté

vrstvě. Teto kopii dosahují neuronů skryté vrstvy přes váhy mezi vstupní a skrytou vrstvou.

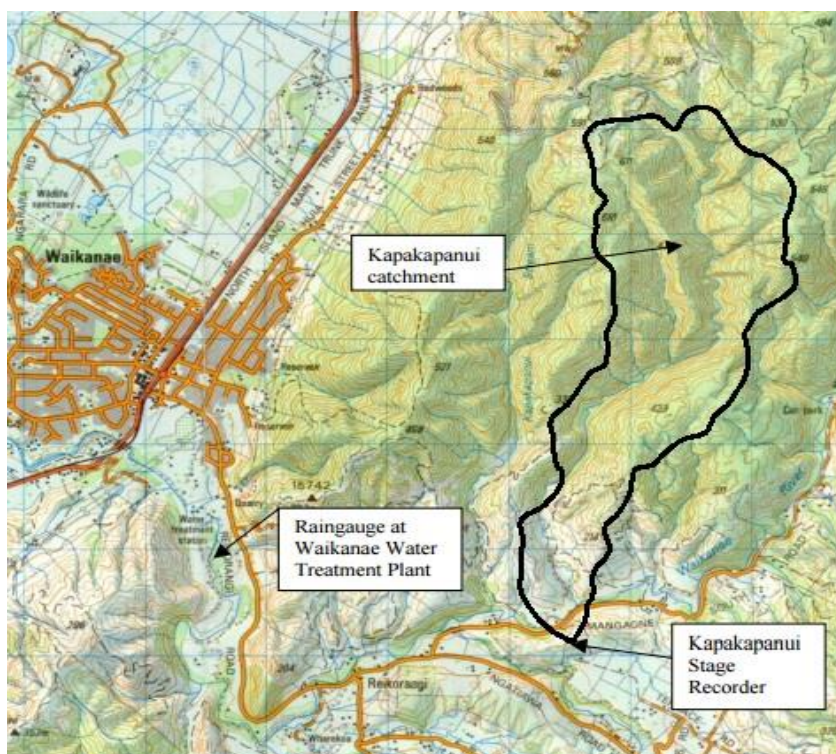
Ve skryté vrstvě se vypočítává suma všech vstupních signálů. Použitou aktivační funkci pro skrytou vrstvu byla hyperbolická tangenta. Aktivační funkce standardizuje vstupní proměnné do určitého intervalu. Výstupní signály ze skryté vrstvy se stávají vstupními signály do výstupní vrstvy (nebo se stávají vstupními signály následující skryté vrstvy, pokud síť má dvě skryté vrstvy).

Ve výstupní vrstvě všechny neurony se vypočítávají. Suma je transformovaná k výstupní vrstvě. Těto výstupní signály právě tvoří celou neuronovou síť.

V této studii jako metod optimalizace byl použit algoritmus zpětné propagace chyby. Je to nejčastější trénovací metod pro vrstevnatou neuronovou síť.

5.1.1. Charakteristika povodí.

Povodí se nachází v severním ostrově v podhůří Tararua Ranges v blízkosti městečka Waikanae. Jižní část povodí je odvodňována do potoka Kapakapanui, který je přítokem řeky Waikanae. Přítok se nachází přibližně 5,5 kilometrů výše na řece od úpravny vody Waikanae (Obr. 20).



Obr. 20: Lokační mapa povodí Kapakapanui

5.1.2. Charakteristika dat

Pozorované data srážek a odtoku byly dostupné pro období 21.07.1998 – 17.10.1999 s 10 minutovým časovým krokem. V této studii data byly zpracované v 30 minutovém časovém kroku.

Data byly odhadnuté z naměřených výšek hladin z hladinoměru. Záznamníkem sloužil plovákový limnigraf (Obr. 21). 21 června 1998 záznamník vodní hladiny byl nainstalován v potoce řeky Kapakapanui přibližně 300 m nad její soutokem s řekou Waikanae.



Obr. 21: Záznamník

Všech 19 zaznamenávání průtoku bylo provedeno v období od června 1998 roku do prosince 1999 roku. Měření bylo provedeno pomocí standardních metod a chyby byly v rozsahu $\pm 8 \%$ od aktuálního průtoku. Změřený průtok byl v rozsahu 1,050 L/s – 51 L/s.

V roce 1995 byl nainstalován automatický srážkoměr venku povodí (Obr. 20) v Úpravně vody Waikanae.

Trénovací data.

V tabulce 1 jsou sumarizované data analyzovaného povodí. Většinou, se používá nějaký soubor dat pro trénování, protože je to těžký predikovat „neviditelné situace“. Pod pojmem standardizace rozumíme rozdělení dat do požadovaného rozsahu. Některé podsoubory obsahují pohybující v různých intervalech proměnné, proto standardizací je nutno provést před začátkem učení. Zvolený rozsah je závislý na aktivační funkci. V této studii aktivační funkce pro vícevrstevný perceptron byl hyperbolický tangens. Všechna dostupná data byly šlakována na interval (-1, 1).

V tabulce 1 trénovací soubor dat neobsahuje ani maximální průtok, ani maximální intenzitu srážek. Testovací soubor dat naopak obsahuje největší kulminační průtok, který je o 20 % vyš než v trénovacím souboru dat.

Fáze	Čas začátku	Čas konce	Doba trvání (dny)	Maximální kulminační průtok (L/s)	Minimální kulminační průtok (L/s)	Maximální intenzita srážek (mm/.5hr)	Srážky celkem (mm)
Trénovací fáze	27/10/98 13:15	16/11/98 22:15	20.38	11830	123	9.84	138.09
Validační fáze	05/10/98 16:45	19/10/98 10:45	13.75	5466	126	7.26	167.34
Testovací fáze č. 1	19/10/98 11:15	27/10/98 12:45	8.06	14301	234	10.87	125.06
Testovací fáze č. 2	21/07/98 12:45	23/08/98 11:45	32.96	5102	131	17.37	166.99
Testovací fáze č. 3	30/08/98 11:15	03/10/98 08:15	33.88	5222	110	8.55	139.91
Testovací fáze č. 4	11/05/99 17:45	11/06/99 02:45	30.88	9925	57	10.57	173.19
Testovací fáze č. 5	16/08/99 10:15	05/09/99 16:15	20.25	8076	100	7.97	89.43

Tab. 1: Použité daty

Validační data

Validační soubor dat je součástí pozorovaných dat. Používá se pro hodnocení a zjištění toho - jak dobře natrénovaná síť. Jak je vidět v tabulce 1 byl použit jenom jeden validační soubor dat. Neexistuje žádný kritérium pro výběr validačního souboru dat, ale schválně byly vyloučené soubory povodí s největšími srážkami a odtokem.

Testovací data.

V tabulce 1 jsou ukázané pět souborů dat, které byly použité jako testovací.

Vstupní a výstupní proměnné.

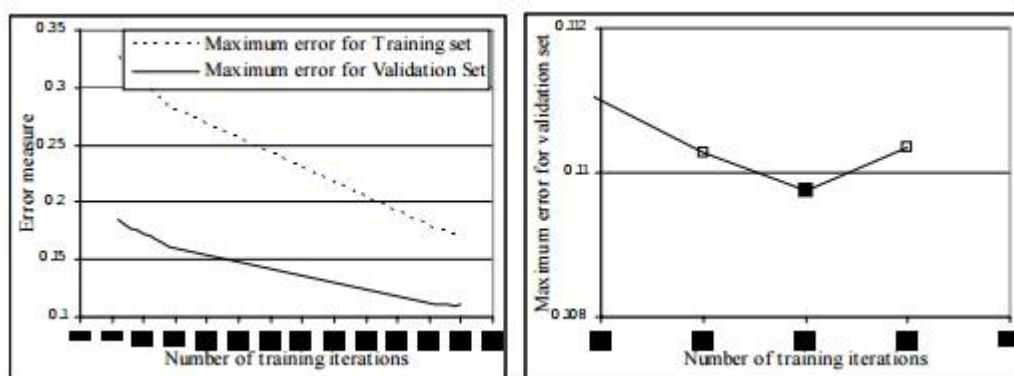
V této studii byl zvolen vícevrstevný perceptron s jednou skrytou vrstvou a jedním výstupním neuronem. Pro zlepšení výkonnosti UNS mezi vstupním a výstupními časovými řadami byly vybrány té nejdůležitější. Jako vstupní proměnné měli by být dříve pozorované srážky a odtok, jako výstupní proměnná měl by být

predikovaný odtok. Ale pro predikce odtoku Q nebylo jasné, jaké data jsou nejvýhodnější. Fernando & Kerr (2003) to vyjádřili pomocí seriální a křížové korelace.

5.1.3 Trénování UNS

V této studii byl použit algoritmus zpětné propagace chyb. To je nejpopulárnější metod při optimalizaci vícevrstevného perceptronů. Algoritmus zpětné propagace chyb funguje tak, že v prvním kroku všechny váhy v síti inicializujeme náhodně. Potom pro každý trénovací vzor se šíří a získávají výstupní hodnoty sítě a rozdíl hodnot oproti požadovanému výstupu se propaguje zpět do vstupní vrstvy. Zároveň váhy se upravují tak, aby se byla minimalizovaná chyba. Tento postup dopředního a zpětného chodu se opakuje do té doby, než nebude dosaženo nějaké minimální hodnoty chyby. Algoritmus zpětné propagace chyb je to iterační proces, takže má být zastaven zastavovacím kritériím. Teto kritérium slouží, aby zastavit trénovací proces v případech kdy: 1) chyba je nejmenší; 2) počet vah je malý; 3) předem stanovený počet trénovacích vzorků je celkem kompletní; 4) začíná růst chyba při křížové validace. V této studii, trénování bylo zastaveno při kombinace případu (1), (3) a (4).

Pro vytvoření požadovaného modelu UNS bylo potřeba 15 vstupních vrstev, 8 skrytých a 1 výstupní vrstva. Na obrázku 22 je vidět změnu chyby během trénování v trénovacím a validačním souboru dat. Když chyba začala zvyšovat- opakování bylo zastaveno. Po 59 opakování chyba byla minimální a tato výsledná síť byla použita pro predikce v testovacích fázích.



Obr. 22: Změna chyby během trénovacího procesu.

Výkonnost v testovacích souborech dat.

Aby získat 30 minutovou predikci, vstupní vektory pro testovací fáze byly převáděné k trénování neuronové sítě. V příloze 9.1 jsou hydrogramy č. (1), (2) a (3), které ukazují pozorovanou predikce na jeden časový krok dopředu (30 minut) pro

trénovací fázi, validační fázi a první testovací fázi. Hydrogram č. 4 je ukazuje srovnávání hydrogramů pro ostatní testovací fázi (2-5). Na hydrogramech je vidět že ve většině případech predikční a pozorované hydrogramy jsou velmi blízko.

5.1.4 Výsledky

Tento model UNS byl úspěšně použit pro predikce odtoku, protože nevyžaduje žádnou jinou informace oproti konverzním modelům.

Celkově, tato studie uvádí několik výsledků. Nejprve jsou reprezentovány výsledky 30 minutové predikce a sítě s architekturou 50-8-1 (50 vstupů, 8 skrytých neuronů a jeden výstupní neuron) a 128 vah. Vstupem do sítě sloužili dříve pozorované data. Trénovací fáze obsahovala 964 vzorku.

Tabulka č. 2 ukazuje statistiku predikce pro všechny fázi.

Fáze	Pozorovaný průměrný průtok	Predikovaný průměrný průtok	RMSE chyba %	Pozorovaný maximální kulminační průtok (L/s)	Predikovaný kulminační průtok (L/s)	Kulminační rozdíl %
Trénovací fáze	820	764	22.3	11830	11833	0.03
Validační fáze	739	659	22.3	5467	5466	-0.17
Testovací fáze č. 1	2306	2129	13.6	14302	13413	-6.2
Testovací fáze č. 2	471	420	27.8	5103	5414	6.1
Testovací fáze č. 3	316	292	33.2	5523	5312	1.7
Testovací fáze č. 4	346	326	77.3	9926	11397	14.8
Testovací fáze č. 5	384	373	45.0	8076	8466	4.8

Tab. 2: Statistika predikce pomocí UNS

RMSE chyba to je procentní podíl pozorovaných dat a používá se pro vyhodnocení výkonnosti modelu. Ve většině fázích RMSE chyba je vyhovující, jenom kromě testovací fázi č. 4. Taková velká chyba byla dosáhnuta kvůli přírodním faktorům. Během této fázi průtok (57 L/s) byl mnohem menší než během trénovací fázi (123 L/s), neuronová síť nebyla schopná predikovat a byla dosáhnuta tak velká RMSE chyba. Taky během této fázi byl špatně predikován velký kulminační průtok, protože při podrobném průzkumu pozorované hodnoty neodpovídají predikovaným.

Výkonnost UNS je stejně dobrá jako při trénování. Model UNS nadhodnotí průtok, protože podle naučeného vztahu vzniká, že v povodí větší odtok pro dané srážky.

5.1.5 Diskuse.

Autoři Fernando & Kerr (2003) se ve své práci zabývali použitím UNS pro modelování srážko – odtokového procesu, a zjišťovali jejich vhodnosti. Detailněji popisovali stav povodí. Při použití srážko- odtokových dat byly výsledky docela úspěšné při použití MLP jako modelu UNS. Ve srovnání s uvedenou studií, autoři Riad & kol. (2004) ve své práci rozdělili dostupná data jenom na dva soubory dat: trénovací a testovací. Použili lineární korelace pro zlepšení výkonnosti sítě a vstupní proměnné byly rozděleny do intervalu (0;1). Ve studii těchto autorů byly modely UNS taky úspěšně použity pro predikce odtoku, ukázaly výbornou výkonnost a vysokou rychlost konvergence. Ve srovnání s ostatními metodami predikce odtoku a srážek, metoda UNS ukazuje nejlepší výsledky. Ale autoři různých rešerší pořád pokračují ve vylepšování predikce pomocí modelů UNS ve vodním hospodářství. Hlavní výhodou UNS je schopnost se učit ze vzorků a jejich vysoká rychlost učení a predikce. Modely UNS jsou také vhodné pro predikce povodňových situací, kde online systémy zapisují proměnné odtoku a srážek.

6. Závěr

Cílem této práce byla rešerše poznatků souvisejících s modely neuronových sítí a jejich aplikacemi ve vodním hospodářství. Předložená práce mi poskytla znalosti pro vytvoření modelu, jeho trénování a hodnocení výkonnosti sítí. V dnešní době modely UNS se používají jako predikční modely a jsou uvedené ve mnoha zdrojích a většinou modely UNS ukazují dobré výsledky.

V první části práce byla zmapovaná historie neuronových sítí, definované základní pojmy, které se tykají oblasti neuronových sítí. Byl popsán základní neuron – formální, biologicky a matematický. Zjistili jsme, jaká byla inspirace rozvoje UNS, jejich využití v současné době a nevýhody. Taky první část práce se zaměřovala na architektury UNS, jaké se používají nejčastěji pro predikce v různých oblastech současného světa.

Druhá část práce se zabývala popisem postupu vytváření modelu pro použití UNS. To zahrnuje kritéria výkonnosti, přípravu a rozdělení dostupných dat, výběr vstupů, popis volby vhodné architektury, optimalizace a validace modelu. Taky v této části, byl popsán nejpoužívanější algoritmus pro učení MLP– algoritmus zpětné propagace chyb. Tyto poznatky vymezují základní teoretické znalosti pro použití UNS a vytvoření modelu pro predikci odtoku.

Poslední část se zaměřovala na aplikace umělých neuronových sítí ve vodním hospodářství, včetně popsání postupu vytvoření modelu, charakteristiky povodí a dat. V případové studii jsem popsala práce autorů Fernando & Kerr (2003). Ve studii byla použita síť MLP s jednou skrytou vrstvou a jedním výstupním neuronem. Cílem případové studii bylo vytvoření modelu jako nástroje pro predikci odtoku na povodí Kapakapanui v Novém Zélandu.

Díky této práci jsem získala základní znalosti o neuronových sítí a jejich aplikacích ve vodním hospodářství. Pokud budu mít tu možnost, tato práce mi poslouží jako základ při vypracování diplomové práce, kde se pravděpodobně popíše umělá neuronová síť určité architektury a pokusí se na základě nyní získaných znalostí a informací vytvořit model UNS pro predikci odtoku určitého povodí.

7. Zdroje

7.1 Seznam literatury

AMARI S., MURATA N., MULLER K.R., FINKE M. & YANG H.H., 1997: Asymptotic statistical theory of overtraining and cross-validation. IEEE Transaction on neural Networks 8 (5): 985-996.

AQIL M., KITA I., YANO A. & NISHIYAMA S., 2007: Neural Networks for Real Time Catchment Flow Modeling and prediction. Water Resour Manage 21: 1781 – 1796.

BARCEV C. & OHONIN V., 1986: Adaptivní sítě zpracování Informace. Krasnojarsk, PrePrint, 200 s.

BATTITI R., 2002: First and second-order methods for learning: Between steepest descent and Newton's method. Neural Computation 4, 141-166

BAYKOV E., 2009: Horizonty vědy Baškirská. BGU, 447s.

BEDIS G. & GEORGIOPOULOS M., 1994: Feed forward neural networks. IEEE Potentials October/November, 27-31.

BISHOP C. M., 1995: Neural Networks for Pattern Recognition. Clarendon Press. 482 s.

BOGOSLAVSKY S., 2007: Scope of Artificial Neural Networks and Prospect of their development. KubGAU: UDC 007.681.5

BOWDEN G. J., DANDY G.C. & MAIER H. R., 2005: Input determination for neural networks models in water resources applications. Journal of Hydrology 301: 75 – 92

BOWDEN G. J., DANDY G.C. & MAIER H. R., 2005: Optimal division of the data for neural models in water resources. Water Resources Research 38 (2): 1-11.

BRASPENNING P.J., THUIJSMAN F. & WEJTERS A.J.M.M., 1995: Artificial Neural Networks. An introduction to ANN. Theory and practice. Germany: Springer-Verlag Berlin Heidelberg. 312 s.

BRYNISON A.E. & HO Y.C., 1969: Applied Optimal Control. New York, Blaisdell. 218 str.

- BURDEN F.R., BRERETON R.G. & WALSH P.T., 1997: Cross-validators selection of test and validation sets in multivariate calibration and neural networks as applied to spectroscopy. *Analyst* 122 (10): 1015-1022.
- BUTLER C., 1987: First International Conference on Neural Networks, SOS Print, 109 s.
- CASTELLANO G., FANELLI A.M. & PELILO M., 1997: An iterative pruning algorithm for feedforward networks. *IEEE Transaction on Neural Networks* 8 (3): 519-970.
- CEJTHAMR V. & DEDINA J., 2010: Management a organizační chování. Grada Publishing a.s., 239 s.
- COULIBALY P., ANCTIL F. & BOBÉE B., 2000: Daily reservoir inflow forecasting using artificial neural networks with stopped training approach. *Journal of hydrology* 230: 244-257.
- DAWSON C.W. & WILBY R.L., 2001: Hydrological modelling using artificial neural networks. *Progress in Physical Geography* 25: 80-108
- DESILETS L., GOLDEN B., WANG Q. & KUMAR R., 1992: Predicting salinity in the Chesapeake Bay using backpropagation. *Computer and Operation Research* 19 (3/4), 227–285.
- FARAWAY J. & CHATFIELD C., 1998: Time series forecasting with neural networks: a comparative study using the airline data. *Applied Statistics* 47: 231-250.
- FARID U.D., 1996: Solving Problems in Environmental Engineering and Geosciences with Artificial Neural Networks. MIT Press, 250 s.
- FERNANDO A.K. & KERR T.G., 2003: Runoff forecasting with artificial neural network model. The 3rd Pacific Conference on storm water and aquatic resource protection. New Zealand Water and Wastes Association publication of conference proceeding in CD-ROM, Auckland. Montgomery Watson Harza (New Zealand) Ltd.
- GALUSHKIN A., 1974: Syntéza vícevrstevných systém rozpoznávání obrazů. *Energie*, 366 s.
- GOSWAMI P., SRIVIDYA., 1996: A novel neural network design for long range prediction of rainfall pattern. *Current Science* 70 (6), 447-457.
- HANNAN S.A., MANZA R.P. & RAMTEKE R.J., 2010: Generalized Regression Neural Network and Radial Basis Function for Heart disease Diagnosis. *International Journal of Computer Application*, 0975-8887.

- HECHT – NIELSEN R., 1987: Kolmogorov's mapping neural network existence theorem. Proceedings of the First IEEE International Joint Conference on Neural Networks. San-Diego, 11-14.
- HIROSE Y., YAMASHITA K. & HIJIYA S., 1991: Back-propagation algorithm which varies the number of hidden units. Neural Networks 4(1): 61-66.
- HRISTEV R.M., 1998: The ANN Book. GNU Public Licence, 392 s.
- CHEN K., YANG L.P., YU X. & CHI H.S., 1997: A self-generating modular neural network architecture for supervised learning. Neurocomputing 16 (1): 33-48.
- CHUNG F.L. & LEE T., 1992: a NODE PRUNING ALGORITHM FOR BACKPROPAGATION NETWORKS. International Journal of Neural Systems 3 (3): 301-314.
- IIACHINSKI A., 2001: Cellular Automata: A Discrete Universe. World Scientific Pub., 840 s.
- KAASTRA I. & BOYD M.S., 1995: Forecasting futures trading volume using neural networks. The journal of Futures Markets 15: 953-970.
- KARNIN E.D., 1990: A simple procedure for pruning backpropagation trained neural networks. IEEE Transaction on neural Networks 1: 239-242.
- KOPECKÁ II., 2011: Psychologie. Grada Publishing, a.s., 200 s.
- KUMAR K.K., 1993: Optimization of the neural net connectivity pattern using a backpropagation algorithm. Neurocomputing 5: 273-286.
- KURKOVA V., N.C.STEELE & R. NERUDA, 2001: Artificial Neural Nets and Genetic Algorithms: Proceeding of the International Conference in Prague. Springer, Praha, 524 s.
- LACHTERMACHER G. & FULLER J.D., 1994: Backpropagation in hydrological time series forecasting. In K.W. Hipel & kol., (Eds). Stochastic and Statistical Methods in hydrology and Environmental Engineering, Vol 3, 229-242.
- LAPYGIN U., KRYLOV V. & CHERNJAVSKYY A., 2008: Ekonomické prognózování. Exmo, 256s.
- LENDASSE A., J. LEE., DE BODT E., WERTZ V. & VERLEYSEN M., 2003: Approximation by radial basis function networks. Connection approaches in Economics and Management Sciences. 203-214.

- MAIER H. R. & DANDY G. C., 1998: The effect of internal parameters and geometry on the performance of back-propagation neural networks: an empirical study. *Environmental Modelling & Software* 13: 193–209.
- MAIER H. R. & DANDY G. C., 2000: Neural Networks for the prediction and forecasting of water resources variables: a review of modelling issues and application. *Environmental Modelling & Software* 15: 101–124.
- MASTERS T., 1993: *Practical Neural Network Recipes in C ++*. Academic Press, San Diego, 493 s.
- MINNS A.W. & HALL M.J., 1996: Artificial neural network as rainfall-runoff models. *Hydrological Sciences Journal*, 399-417.
- MINSKY M. & PAPER S., 1969: *Perceptrons*. Cambridge, MIT Press, Massachusetts, 258 s.
- MIZUTANI E., DREYFUS S.E. & DEMMEL J. W., 2008: Second-order stagewise backpropagation for Hessian matrix analyses and investigation of negative curvature. *Neural Networks* 21: 193-203.
- MOODY J. & YARVIN N., 1992: *Networks with learned unit response function*. R.P, San Mateo, 1049 s.
- MORENO J.J.M., POL A.P. & GRACIA P.M., 2011: Artificial Neural Network applied to forecasting time series. *Psicothema* 322–329
- NILSSON N., 1996: *Introduction to Machine Learning*. Unpublished draft, 179 s.
- NOVÁK M., 1992: *Neuronové Síťe a neuropočítače*. Praha, Senzo, 192 s.
- PARISI R., DI CLAUDIO E.D., ORLANDI G. & RAO B.D., 1996: A generalized learning paradigm exploiting the structure of feedforward MLP. *IEEE Transaction on neural Networks* 7 (6): 1451-1460.
- PLAUT D.C. & HINTON G.E., 1987: Learning sets of filters using backpropagation. *Comput. Speech Language* 2: 35-61.
- POFF N.L., TOKAR S. & JOHNSON P., 1996: Stream hydrological and ecological responses to climate change assessed with an artificial neural network. *Limnology and Oceanography* 41(5), 857–863.
- REISS R.F., 1964: *Neural Theory and Modeling*. Stanford University Press, 427 s.

- RIAD S., MANIA J., BOUCHAOU L. & NAJJAR Y., 2004: Rainfall-Runoff Model Using an Artificial Neural Network Approach, *Mathematical and Computer Modelling*, 839-846.
- ROADKNIGHT C.M., BALLS G.R., MILLS G. E. & PALMER-BROWN D., 1997: Modeling complex environmental data. *IEEE Transaction on Neural Pnetworks* 8(4), 852-862.
- ROJAS R., 1996: *Neural networks: A systematic introduction*. Springer-Varlag, Berlin, 502 s.
- ROSENBLATT F., 1962: *A Comparison Of Several Perceptron Models*, Spartan Books, Washington, 495 s.
- RUMELHART D. E., HINTON G.E. & WILLIAMS R.J., 1986: *Learning Internal Representations by Error Propagation*. Cambridge, MIT press, 346 s.
- REED D. R. & MARKS R. J., 1998: *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*. MIT Press, 346 s.
- SETINO R. & HUI L.C.K., 1995: Use of a quasi –Newton method in a feedforward neural network construction algorithm. *IEEE Transactionon neural Networks* 6: 273-277.
- SHUKLA M.B., KOK R., PRACHER S.O., CLARK G. & LACROIX R., 1996: Use of artificial neural networks in transient drainage design. *Transactions of the ASAE* 39 (1): 119-124.
- SIETSMA J. & DOW R.J.F., 1991: Creating artificial neural networks that generalize. *Neural Networks* 4: 67-79.
- SMITH J., ELI R.N., 1995: Neural network models of rainfall-runoff processes. *Journal of Water Resources Planning and Management* 121 (6)., 499-508.
- ŠNOREK. M. & JIŘINA M., 1996: *Neuronové Sítě a Neuropočítače*. Praha: Vydavatelství ČVUT. 124 str.
- STARÝ M., 2004: *Užití Umělých Neuronových Sítí v Aplikované Hydrologii*. VUTIUM, Brno, 40 s.
- SMITH S. W., 1997: *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Pub., 1st edition, 626 s.

ŠÍMA J. & Neruda R., 1996: Teoretické otázky neuronových sítí. Matfyzpress, Praha, 390 s.

TOWEL GG., CRAVEN M.K. & SHAVLIK J.W., 1991: Constructive induction in knowledge-based neural networks. In; Proceedings of the 8th International Workshop on Machine Learning, 289 s.

TUČKOVÁ J., 2003: Úvod do teorie a aplikací umělých neuronových sítí. Vydavatelství ČVUT, Praha, 103 s.

VASENKOV D. V., 2007: Metody Učení Umělých Neuronových Sítí, Saint-Petersburg, 29 s.

WERBOS P.J., 1974: Beyond Regression: New Tools for Prediction and Analysis in the behavioural sciences. Cambridge, MIT Press, 906 s.

WHITEHEAD PG., HOWARD A., ARULMANI C., 1997: Modeling algal growth and transport in rivers – a comparison of time series analysis, dynamic mass balance and neural network techniques. Hydrologia 349, 39-46.

WOSSERMAN F., 1992: Neuropočítačová technika. 184 s.

7.2 Internetové zdroje

AIRPORTAL, 2014: Historie Neuronových Sítí, online: <http://www.aiportal.ru/articles/neural-networks/history.html>

BIRUČKOVÁ N., 2014: Umělé Neuronové Sítě: budoucnost, online: <http://1url.cz/70Wo>, 21.01.2014.

MÍŠA J., 2006: Umělé Neuronové sítě: Neuron stavba a funkce; C2 – 01, online: <http://statnice.e-misa.info/C2-01.pdf>

PARAMONOV V., 2012: Neuronová Síť GOOGLE se naučila hlídat koček na obrázkách, online: <http://compulenta.computerra.ru/archive/ai/689629/>, cit. 27.06:2015.

STATSOFT CR s.r.o., 2014: Úvod do neuronových sítí, online: http://www.statsoft.cz/file1/PDF/newsletter/2013_02_05_StatSoft_Neuronove_site_li nky.pdf

ŠTRONER M., 2012: Rozbory Přesnosti. Globální optimalizační algoritmy. IngGeo, online: <http://1url.cz/R0WB>

UTKARASH S., 2010: First artificial neurons: The McCulloch-Pitts model, online: <http://www.aishack.in/tutorials/first-artificial-neurons-the-mccullochpitts-model/>

8. Seznam obrázku

URL 1: Rene Descartes (online), dostupné z: <<http://oregonstate.edu/instruct/phl302/philosophers/descartes.html>>

URL 2: McCulloch (online), [cit. 24.01.2008], dostupné z: <<https://materialdidacticonm.wordpress.com/2008/01/24/1141-mcculloch/>>

URL 3: Alan Turing (online), dostupné z: <<http://www.rutherfordjournal.org/article040101.html>>

URL 4: Google and Stanford create a digital brain that, like an infant, learns to identify a human face from scratch (online), [cit. 26.06.2012], dostupné z: <<http://www.extremetech.com/extreme/131717-google-and-stanford-create-a-digital-brain-that-like-an-infant-learns-to-identify-a-human-face-from-scratch>>

URL 5: Blog mladého vědce (online), [cit. 22.11.2014], dostupné z: <http://phdblog.org.ua/m/lskusstvennyj_nejron.html>

URL 6: MindCreators (online), dostupné z: <<http://www.mindcreators.com/neuronbasics.html>>

URL 7: First artificial neurons: The McCulloch Pitts model (online), [cit. 2014], dostupné z: <<http://www.aishack.in/tutorials/first-artificial-neurons-the-mccullochpitts-model/>>

URL 8: Umělé Neuronové Síť (online), [cit. 17.04.1997], dostupné z: <<http://scorcher.ru/neuro/science/neurocomp/mem52.php>>

URL 10: Single Layer Perceptron, (online), [cit. 19.08.2007], dostupné z: <https://ru.wikipedia.org/wiki/%D0%A4%D0%B0%D0%B9%D0%BB:Single_layer_perceptron.png>

URL 11: Multi-Layer Perceptron Tutorial (online), [cit. 28.05.2009], dostupné z: <<http://blog.refu.co/?p=931>>

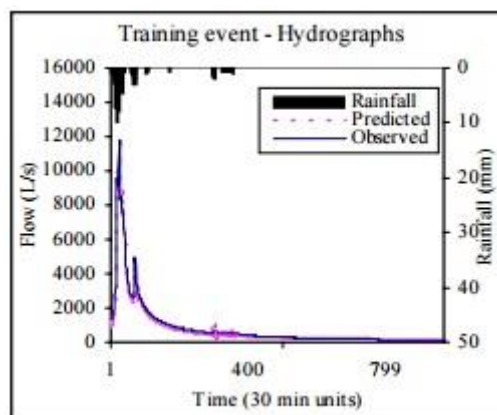
URL 12: Artificial neural Network (online), [cit. 2012], dostupné z: <<http://www.cenaero.be/Page.asp?docid=27097>>

URL 13: Reccurent Neural Networks in Ruby (online), [cit. 29.10.2012], dostupné z: <<http://blog.josephwilk.net/ruby/recurrent-neural-networks-in-ruby.html>>

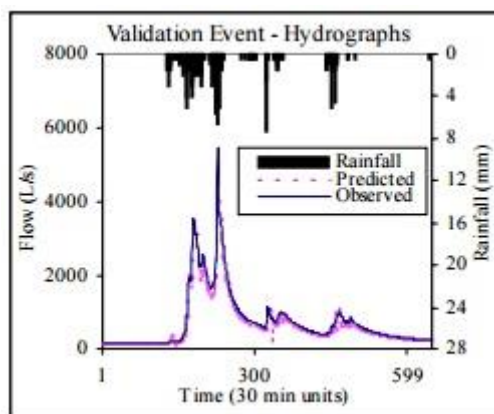
URL 14: Vícevrstvé síť (online), dostupné z:
<[http://www.eamos.cz/amos/kat_inf/externi/kat_inf_76600/6_\[Rezim_kompatibility\].pdf](http://www.eamos.cz/amos/kat_inf/externi/kat_inf_76600/6_[Rezim_kompatibility].pdf)>

9. Příloha

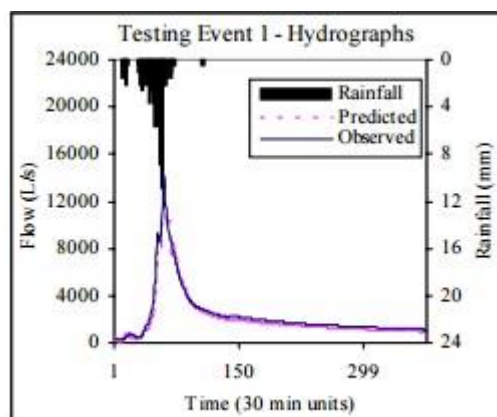
9.1 Hydrogramy



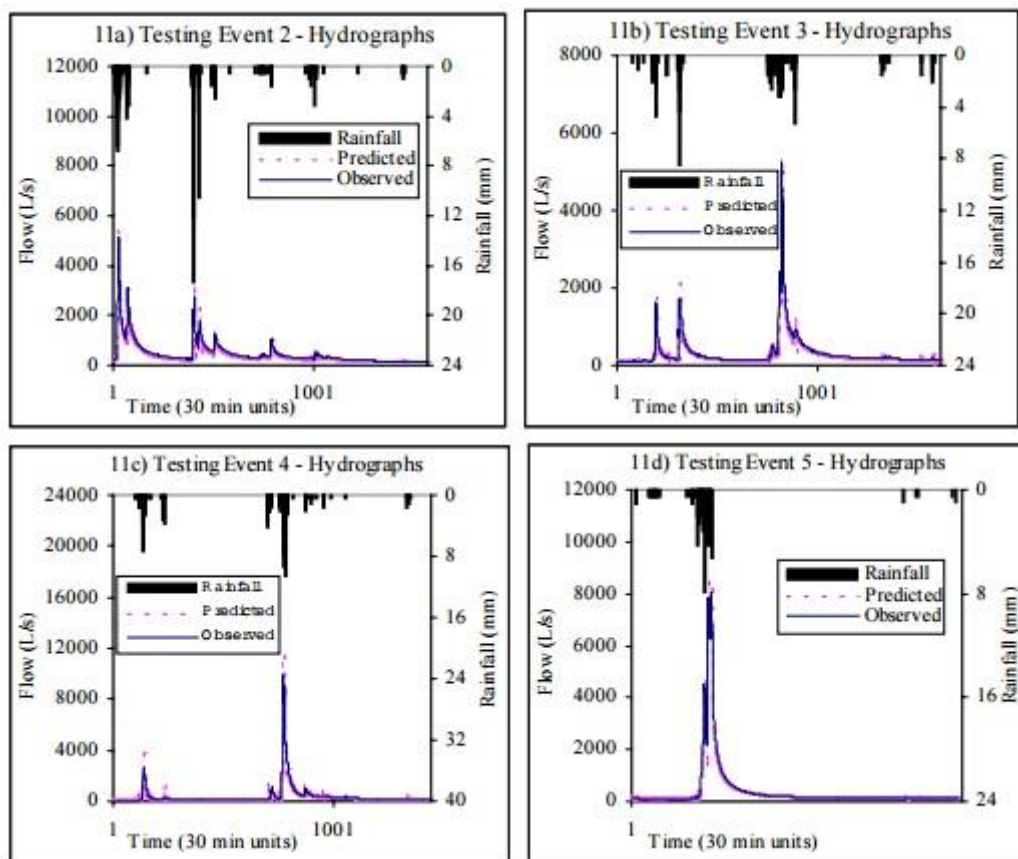
Hydrogram 1: Pozorovaný a predikovaný průtok během trénovací fázi
(Fernando & Kerr, 2003)



Hydrogram 2: Pozorovaný a predikovaný průtok během validační fázi
(Fernando & Kerr, 2003)



Hydrogram 3: Pozorovaný a predikovaný průtok během Testovací fázi č. 1
(Fernando & Kerr, 2003)



Hydrogram 4: Pozorovaný a predikovaný průtok během testovacích fází (2-5)
(Fernando & Ker