

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ
ÚSTAV MATEMATIKY
FACULTY OF MECHANICAL ENGINEERING
INSTITUTE OF MATHEMATICS

MODELOVÁNÍ A SIMULACE V OBLASTI ODPADOVÉHO HOSPODÁŘSTVÍ

MODELLING AND SIMULATION IN THE FIELD OF WASTE MANAGEMENT

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

IVA PAŘÍZKOVÁ

VEDOUcí PRÁCE
SUPERVISOR

Ing. MICHAL TOUŠ, Ph.D.

BRNO 2014

Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav matematiky

Akademický rok: 2013/2014

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

student(ka): Iva Pařízková

který/která studuje v **bakalářském studijním programu**

obor: **Matematické inženýrství (3901R021)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma bakalářské práce:

Modelování a simulace v oblasti odpadového hospodářství

v anglickém jazyce:

Modelling and simulation in the field of waste management

Stručná charakteristika problematiky úkolu:

Práce se bude zabývat aplikací regresních metod se zaměřením na odpadové hospodářství a výrobu energie. Bude provedena rešerše regresních metod. Vybrané metody budou následně použity na datech z průmyslové sféry. Bude provedena diskuse výsledků. Předpokládá se aplikace softwaru Statistica.

Cíle bakalářské práce:

Student se seznámí s teoretickými poznatky v dané oblasti a konkrétním průmyslovým problémem. Osvojí si znalost použití statistického softwaru. S využitím dat naměřených v provozu sestaví regresní model popisující vybraný blok provozu. Model pak implementuje do simulačního nástroje.

Seznam odborné literatury:

Generalized Linear Models: with Applications in Engineering and the Sciences, 2nd Edition,
Raymond H. Myers, Douglas C. Montgomery, G. Geoffrey Vining, Timothy J. Robinson
ISBN: 978-0-470-45463-3, 496 pages, April 2010

Neural networks: A comprehensive foundation, Simon Haykin, ISBN: 81-7898-300-0, 2005

An introduction to Neural Networks, Ben Krose, Patrick van der Smagt, 1996, University of
Amsterdam

Vedoucí bakalářské práce: Ing. Michal Touš, Ph.D.

Termín odevzdání bakalářské práce je stanoven časovým plánem akademického roku 2013/2014.

V Brně, dne 22.11.2013

L.S.

prof. RNDr. Josef Šlapal, CSc.
Ředitel ústavu

prof. RNDr. Miroslav Doupovec, CSc., dr. h. c.
Děkan fakulty

Abstrakt

Bakalářská práce se zabývá aplikací vícevrstevné perceptronové sítě v modelech technologie zařízení pro energetické využití odpadu (ZEVO). Jedná se konkrétně o modely pro množství páry vyrobené v kotlích a množství páry odebírané externím subjektem. Nejprve je uvedena základní teorie k neuronovým sítím, následována popisem k modulu Neuronové sítě v programu STATISTICA, pomocí kterého byly tvořeny modely v praktické části. Další část práce je věnována rozboru vzniklých modelů. Jedná se zejména o analýzu vlivu vstupních faktorů, porovnání neuronové sítě a lineární regrese, důvody zvolení složitějšího modelu neuronové sítě a samozřejmě popisu výsledného modelu neuronových sítí. V neposlední řadě byl v rámci této práce vytvořen program pro implementaci neuronové sítě do prostředí Visual Basic of Application (VBA). Po úpravě syntaxe je však použitelný pro implementaci do libovolného programovacího jazyka.

Summary

This bachelor thesis is focused on the application of multilayer perceptron net for modeling the technological units of waste-to-energy facility ZEVO Malesice. It was specifically created to model the amount of steam generated in steam-boilers and to quantify the consumption of steam by an external subject. Firstly, the basics of neuron theory are presented. In the following, a Statistica artificial neural network module is described. This module was used to develop the neural network models. The models appearing in practical section were created with the use of STATISTICA software. Last chapter deals with detailed description of the developed models and their comparison with simple linear and nonlinear regression models. Last but not least, the description of a software providing easy implementation of neural network models into Visual Basic for Application programming language is presented.

Klíčová slova

vícevrstevná perceptronová síť, umělá neuronová síť, predikce, STATISTICA, modelování, VBA, zařízení na energetické využití odpadu

Keywords

multilayer neural network, artificial neural network, prediction, STATISTICA, modeling, VBA, waste-to-energy facility

PAŘÍZKOVÁ, I. *Modelování a simulace v oblasti odpadového hospodářství*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2014. 38 s. Vedoucí Ing. Michal Touš, Ph.D.

Prohlašuji, že jsem bakalářskou práci *Modelování a simulace v oblasti odpadového hospodářství* vypracovala samostatně pod vedením Ing. Michala Touše, Ph.D. s použitím materiálů uvedených v seznamu literatury.

Iva Pařízková

Ráda bych na tomto místě poděkovala svému vedoucímu Ing. Michalu Toušovi PhD. za vedení, cenné rady a připomínky při zpracování této bakalářské práce.

Iva Pařízková

Obsah

1	Úvod	3
2	Základní teorie neuronových sítí	5
2.1	Neuron v biologii člověka	5
2.2	Matematický model neuronu	6
2.3	Neuronové sítě	7
2.3.1	Architektura neuronové sítě	7
2.3.2	Dynamika neuronové sítě	8
2.4	Učení neuronové sítě	9
2.4.1	Hebbův zákon učení	9
2.4.2	Windrow-Hoffovo pravidlo	9
2.4.3	Druhy učení	9
2.5	Vícevrstevná síť perceptronů a Back-Propagation	9
2.5.1	Algoritmus Back-Propagation	10
2.5.2	Problémy algoritmu Back-Propagation	11
2.6	Použití neuronových sítí v oblasti energetiky	12
3	Neuronové sítě v programu STATISTICA	14
3.1	Výběr analýzy	14
3.2	Regrese pomocí neuronových sítí	15
3.2.1	Výběr proměnných a strategie pro vytvoření predikčních modelů	15
3.2.2	Vzorkování dat	16
3.2.3	Automatická tvorba sítí – volba parametrů sítě	17
3.2.4	Výsledkové dialogy	19
3.3	Algoritmus výpočtu neuronové sítě pro regresi v systému STATISTICA	21
4	Predikční modely pro plánování provozu zařízení na energetické využití odpadu	22
4.1	ZEVO Malešice	22
4.2	Hodnocení kvality modelů	22
4.2.1	Korelační koeficient	23
4.2.2	Průměrná absolutní chyba	23
4.3	Model kotlů	23
4.3.1	Analýza vlivů faktorů	24
4.3.2	Neuronové sítě zvolené pro modely jednotlivých kotlů	26
4.3.3	Porovnání neuronové sítě a lineární regrese	26
4.4	Model odběru páry do Laktos	27
4.4.1	Analýza vlivu faktorů	28
4.4.2	Výsledný model	28

4.4.3	Porovnání neuronové sítě a nelineární regrese	29
5	Implementace sítí do prostředí VBA	31
5.1	Funkce aplikace	31
6	Závěr	33
	Literatura	35
A	Výsledné neuronové sítě - model kotlů	37
B	Výsledné neuronové sítě - model Laktos	38

Kapitola 1

Úvod

Tato práce se zabývá aplikací neuronových sítí v modelu na plánování provozu zařízení na energetické využití odpadu (ZEVO).

Jak již název „neuronové sítě“ napovídá, jedná se o velmi zjednodušený model nervové soustavy živých tvorů. Pouze k tomuto účelu původně neuronové sítě vznikly – měly modelovat lidskou nervovou soustavu a posloužit tak k jejímu lepšímu pochopení. Později však bylo zjištěno, že s velmi dobrými výsledky dokáží modelovat rozmanité oblasti lidské činnosti. Již jediný neuron dokáže velmi dobře rozhodovat, o který ze dvou různých prvků se jedná. Obvykle však potřebujeme modelovat složitější problémy, a tak se častěji používá spojení více formálních neuronů do tzv. neuronové sítě. Ty byly aplikovány například v oblasti rozpoznávání obrazu, k vytvoření různých predikčních modelů v technické praxi či v lékařství k vyhodnocování signálů EEG a EMG [2].

Oproti elementárním metodám používaným k predikcím má neuronová síť obrovskou výhodu v tom, že dokáže napodobovat učení a inteligentní chování člověka. Ze zadané tréninkové množiny se síť v učícím procesu naučí pravidla, která si navíc zobecní, a to jí pak umožní řešit podobné příklady, se kterými se ještě nesetkala. Oproti tomu elementární algoritmy pouze uloží do paměti tréninková data. Oproti elementárním algoritmům je však výpočetně náročnější, což může být v mnoha ohledech její velká nevýhoda (např. optimalizační úlohy) [2].

Názorným příkladem rozdílu mezi neuronovou sítí a elementárním algoritmem může být příprava studenta na zkoušku z matematiky. Pokud se student naučí nazpaměť postup všech vzorových příkladů ze sbírky, u zkoušky, kde dostane příklad podobný, nikoli stejný, neuspěje. K úspěšnému složení zkoušky je potřeba zobecnit si zákonitosti řešení vzorových příkladů. Prvním postupem, tedy učením zpaměti, by postupoval klasický algoritmus. Druhým, podstatně efektivnějším, neuronová síť [2].

Praktická část práce se věnuje dvěma modelům, které byly vytvořeny pomocí neuronových sítí. Je to model pro páru vyrobenou v kotlích a páru odebírající externí firmou. Neuronové sítě byly zvoleny v případě modelu kotlů z důvodu lepší predikce a v případně dodávky do externí firmy navíc ještě z důvodu kategoriálních proměnných.

Na samém začátku první kapitoly je několik poznatků o nervové soustavě z biologie člověka. Tato část je zařazena z důvodu lepšího pochopení pozdějších pasáží kapitoly, které se týkají základní teorie neuronových sítí. Následuje kapitola, která by měla být návodem k modulu Neuronové sítě v softwaru STATISTICA. Poslední část textu se věnuje praktickému užití neuronových sítí pro model páry vyrobené v kotlích v pražské spalovně ZEVO Malešice a modelu pro předpověď nepravidelného odběru do firmy Laktos (používáno pro předeřhřev technologie na výrobu mléčných výrobků). Tyto modely jsou součástí projektu, v rámci kterého autorka strávila 14 dní na pracovní stáži primo v provozu ZEVO

Malešice, pro predikci předávaného elektrického výkonu do přenosové soustavy.

Tato práce přispívá k řešení aktivit v rámci centra kompetence Waste-to-Energy (projekt č. TE02000236) financovaného Technologickou agenturou České republiky.

Kapitola 2

Základní teorie neuronových sítí

Tato kapitola je zpracována převážně za použití textů [1] a [2].

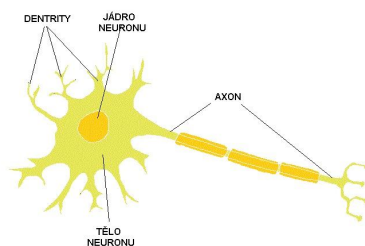
2.1 Neuron v biologii člověka

Nervová soustava člověka slouží k zachycení a zpracování podnětů působících na organismus a zajištění odpovídající reakce na ně. Přijímat mechanické, tepelné, chemické a další podněty umožňují tzv. *receptory*. Vzruchy se dále šíří k dalším nervovým buňkám, které tyto informace zpracovávají a přivádí k výkonným orgánům, tzv. *efektorům*. Dále se vzruchy po projekčních drahách dostávají do mozkové kůry, která je nejvyšším řídicím centrem nervového systému.

Základním stavebním prvkem nervové soustavy je nervová buňka, tzv. *neuron*. Neuron jsou vysoce specializované buňky, které přenášejí a zpracovávají informace z vnějšího i vnitřního prostředí organismu a tím podmiňují jeho schopnost na ně adekvátně reagovat. Struktura neuronu je schématicky znázorněna na obrázku 2.1.

Neuron má kromě vlastního těla, tzv. *somatu*, i vstupní a výstupní kanály, tzv. *dendrity* a *axon*. Z axonu obvykle vychází řada větví – *terminálů* zakončených blánou, která se stýká s výběžky dendritů jiných neuronů. K přenosu informace pak dochází na mezineuronovém rozhraní, tzv. *synapse*. Synapse dělíme podle funkce na tzv. *excitační*, které rozšiřují vzruch v nervové soustavě, a na tzv. *inhibiční*, které vzruch tlumí. Šíření vzruchu probíhá díky tomu, že soma i axon jsou obaleny membránou, která je schopna tvořit elektrické pulzy. Ty se přenášejí na další neurony, kde je podle synaptických propustností určena míra podráždění dalších neuronů. Po každém průchodu signálu se synaptická propustnost mění, což je předpokladem paměťové schopnosti neuronů. Propojení neuronů se během života vlivem zkušeností, učení, zapomínání a dalšího mění.

Na podobném principu je založen i matematický model neuronu.



Obrázek 2.1: Neuron [3]

2.2 Matematický model neuronu

Základním prvkem matematického modelu neuronové sítě je tzv. *formální neuron*, jehož struktura je podobná biologickému neuronu a je znázorněna na obrázku 2.2. Do formálního neuronu vstupuje n obecně reálných *vstupů* x_1, \dots, x_n (modelují dendrity), které jsou ohodnoceny synaptickými *váhami* w_1, \dots, w_n . Výraz

$$\xi = \sum_{i=1}^n x_i w_i \quad (2.1)$$

udává tzv. *vnitřní potenciál* neuronu, který po dosažení prahové hodnoty h udává *výstup* (elektický impuls axonu) neuronu. Obvykle přidáváme další formální vstup $x_0 = 1$, jehož váha $w_0 = -h$ se nazývá *bias*. Touto formální úpravou dosáhneme toho, že práh má nulovou hodnotu a vnitřní potenciál je pak dán funkcí:

$$\xi = w_0 + \sum_{i=1}^n x_i w_i.$$

Výstupní hodnotu $y = \sigma(\xi)$ dostaneme pomocí tzv. *aktivační funkce* σ . Nejjednodušším typem aktivační funkce, která je inspirována funkcí biologického neuronu, je *ostrá nelinearita*, kterou definujeme vztahem:

$$y = \sigma(\xi) = \begin{cases} 1 & \text{jestliže } \xi \geq 0 \\ 0 & \text{jestliže } \xi < 0. \end{cases} \quad (2.2)$$

Další typy aktivačních funkcí jsou například saturovaná lineární funkce:

$$y = \sigma(\xi) = \begin{cases} 1 & \text{jestliže } \xi > 1 \\ \xi & \text{jestliže } 0 \leq \xi \leq 1 \\ 0 & \text{jestliže } \xi < 0, \end{cases} \quad (2.3)$$

standardní (logistická) sigmoida:

$$y = \sigma(\xi) = \frac{1}{1 + e^{-\xi}} \quad (2.4)$$

či hyperbolický tangens

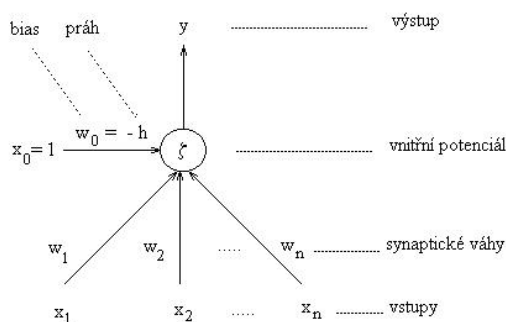
$$y = \sigma(\xi) = \frac{1 - e^{-\xi}}{1 + e^{-\xi}}. \quad (2.5)$$

Formální neuron dokáže řešit jednoduchou klasifikaci. Na obrázku 2.3 si vysvětlíme princip této úlohy. Vstupy neuronu chápeme jako souřadnice bodů v rovině (obecně v n -rozměrném Euklidovském prostoru). Uvažujeme-li aktivační funkci podle rovnice 2.2, pak zavedením biasu získáme nadrovinu (v případě E_2 přímka) o rovnici

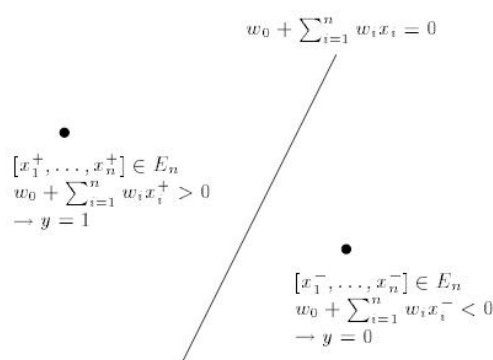
$$\xi = w_0 + \sum_{i=1}^n x_i w_i,$$

kteřá dělí prostor na dvě poloroviny. Je zřejmé, že neuron klasifikuje, ve které polovině daný bod leží (rozlišuje zda vnitřní potenciál neuronu nabývá kladné nebo záporné hodnoty). Neuron provádí tzv. *dichotomii* vstupního prostoru.

Funkce jednoho neuronu je však velmi omezena. Uvažujme funkci XOR pro dva binární vstupy 0 a 1 definovanou jako: $\text{XOR}(0,0) = 0$, $\text{XOR}(0,1) = 1$, $\text{XOR}(1,0) = 1$, $\text{XOR}(1,1) = 0$. Z obrázku 2.4, který graficky znázorňuje funkci XOR, je patrné, že nelze oddělit hodnoty nula a jedna jedinou přímkou. Proto se v praxi využívá spojení více neuronů do neuronové sítě.



Obrázek 2.2: Formální neuron [2, str. 24]



Obrázek 2.3: Klasifikační funkce neuronu [2, str. 25]

2.3 Neuronové sítě

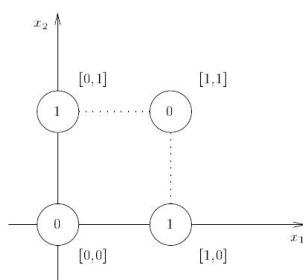
Neuronová síť se skládá z formálních neuronů, které jsou vzájemně propojené tak, že výstup neuronu je vstupem obecně více neuronů (opět podobně jako v biologickém modelu). Počet neuronů a jejich vzájemné propojení určuje tzv. *architekturu* neuronové sítě. Podle funkce rozlišujeme *vstupní*, *skryté* a *výstupní neurony*.

2.3.1 Architektura neuronové sítě

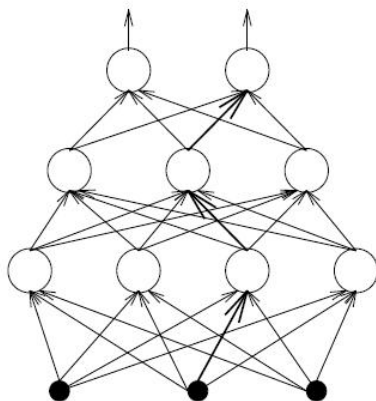
Rozlišujeme 2 typy architektury neuronové sítě.

Dopředná (resp. acyklická) síť je charakteristická tím, že spoje mezi neurony vedou pouze z nižších vrstev do vyšších, nikoli naopak. Speciálním případem je tzv. *vícevrstevná síť*, kde do každého neuronu v příslušné vrstvě vede spoj z každého neuronu předcházející vrstvy. Architekturu této sítě lze zadat jen počty neuronů v jednotlivých vrstvách. Avšak řekneme-li, že síť je třívrstevná, myslíme tím síť, která má ve skutečnosti čtyři vrstvy. Platí, že vstupní vrstvu do počtu vrstev sítě nepočítáme. Na obrázku 2.5 lze vidět dvojvrstevnou síť s архитектурou 4-3-2.

V případě *rekurentní (resp. cyklické) sítě* existuje skupina neuronů, která je zapojena v kruhu (cyklu). Nejjednodušším typem je tzv. *zpětná vazba*, kdy výstup neuronu je zároveň jeho vstupem.



Obrázek 2.4: Funkce XOR [2, str. 28]



Obrázek 2.5: Vícevrstevná síť [2, str. 32]

2.3.2 Dynamika neuronové sítě

Stavy všech neuronů v síti určují tzv. *stav* neuronové sítě a synaptické váhy všech spojení představují tzv. *konfiguraci* neuronové sítě. Neuronová síť se v čase vyvíjí, mění se propojení a stav neuronů a adaptují se váhy. Celkovou dynamiku sítě dělíme do tří dynamik.

Organizační, která popisuje architekturu neuronové sítě. Změna topologie se většinou uplatňuje v rámci adaptivního režimu tak, že síť je případně rozšířena o další neurony a příslušné spoje.

Aktivní, která specifikuje *počáteční stav* sítě a způsob jeho změny v čase při pevné topologii a konfiguraci. Na počátku (diskrétní čas 0) jsou stavy vstupních neuronů nastaveny na tzv. *vstup sítě*, zbylé neurony jsou v uvedeném počátečním stavu. Poté (v diskrétních časech 1, 2, 3, ...) je podle daného pravidla aktivní dynamiky vybrán jeden nebo více neuronů, které změni svůj stav na základě svých vstupů. Stav výstupních neuronů, který se obecně mění v čase, je tzv. *výstupem* neuronové sítě. Aktivní dynamika sítě je vlastně výpočet výstupních hodnot pro dané vstupní údaje, který je realizován na síti s pevnou topologií a konfigurací pomocí aktivační funkce neuronu. Aktivační funkce neuronu je obvykle stejná pro všechny neurony v síti.

Adaptivní, která specifikuje počáteční konfiguraci a to, jakým způsobem se mění váhy dané sítě v čase. Všechny možné konfigurace sítě tvoří tzv. *váhový prostor*. Váhy se adaptují tzv. *učením* neuronové sítě.

2.4 Učení neuronové sítě

Cílem učícího procesu je nalézt takovou konfiguraci vah, aby síť realizovala zadanou funkci co nejpřesněji (tedy aby vzniklá rezidua mezi skutečnými a výstupními daty ze sítě byla co nejmenší). Učení neuronové sítě je velmi složitý nelineární optimalizační proces [2]. Existuje velké množství učících algoritmů, jejichž základem jsou dvě pravidla – Hebbův zákon učení a Windrow-Hoffovo pravidlo.

2.4.1 Hebbův zákon učení

Základní myšlenkou Hebbova zákona učení je posilování vazeb u neuronů, které jsou ve stejnou chvíli aktivní. Pokud není aktivní ani jeden z dvojice propojených neuronů, vazba slábne. Je-li aktivní pouze jedna vazba, zůstává nezměněna. Je-li y_j vstup do neuronu k , pak Hebbův zákon učení pro změnu vah mezi neurony j a k můžeme zapsat takto:

$$\Delta w_{jk} = \gamma y_k y_j,$$

kde $0 < \gamma < 1$ udává *rychlost učení*. V analogii s lidským učením se obvykle na začátku volí malá rychlost učení, která postupně během adaptace roste [4].

2.4.2 Windrow-Hoffovo pravidlo

Windrow-Hoffovo pravidlo je založeno na rozdílu výstupu z neuronové sítě y_k a požadované hodnoty d_k . Je-li y_j vstup do neuronu k , pak se váhy adaptují podle vzorce:

$$\Delta w_{jk} = \gamma y_j (d_k - y_k),$$

kde $0 < \gamma < 1$ je opět *rychlost učení*.

2.4.3 Druhy učení

Dle tréninkové množiny dělíme učení na dva základní druhy.

Učení s učitelem je založeno na použití *tréninkové množiny*, která obsahuje sadu vstupů a jejich výstupy. Pomocí těchto dvojic jsou upravovány váhy spojení tak, aby bylo dosaženo co největší shody mezi výstupy z neuronové sítě a vzorky trénovací množiny. Na této metodě je založena např. vícevrstevná síť perceptronů a její učící algoritmus Back-Propagation.

V případě, že tréninková množina obsahuje jen vstupy sítě je užito *učení bez učitele*. Síť si vzory organizuje například do shluků a odhaluje jejich statisticky významné vlastnosti. Tohoto typu učení využívá například Kohonenova síť.

2.5 Vícevrstevná síť perceptronů a Back-Propagation

Vícevrstevná perceptronová síť s učícím algoritmem zpětného šíření chyby (Back-Propagation) je nejběžnější z neuronových sítí. Je užita asi v 80 % případů všech aplikací neuronových sítí. Vzhledem k rozšířenosti tohoto modelu a jeho určitým nedostatkům existuje mnoho modelů algoritmu Back-Propagation.

Nejprve bude uveden základní algoritmus Back-Propagation, za kterým následuje stručná zmínka o některých jeho modifikacích.

Právě vícevrstevná síť perceptronů je užita v praktické části této práce.

2.5.1 Algoritmus Back-Propagation

Uvažujme vícevrstevnou síť perceptronů s aktivační funkcí logistická sigmoida

$$y = \sigma(\xi) = \frac{1}{1 + e^{-\lambda\xi}}, \quad (2.6)$$

kde parametr λ je reálný parametr *strmosti*, jenž určuje nelineární nárůst standardní sigmoidy v okolí nuly. Výstup neuronu i je vstupem do neuronu j .

Celková chyba všech vzorů se určí podle vztahu:

$$E(\mathbf{w}) = \sum_{k=1}^p E_k(\mathbf{w}), \quad (2.7)$$

kde $E_k(\mathbf{w})$ je chyba k-tého tréninkového vzoru daná vztahem:

$$E_k = \frac{1}{2} \sum_j (y_j(\mathbf{x}_k, \mathbf{w}) - d_{kj})^2. \quad (2.8)$$

K minimalizaci chybové funkce se používá metoda nazývaná „Metoda největšího spádu“. Změna vah se provádí dle vztahu:

$$\Delta w_{ji} = -\varepsilon \frac{\delta E}{\delta w_{ji}}, \quad (2.9)$$

kde ε udává rychlost učení. Nová konfigurace vah se pak spočítá jednoduše dle vztahu:

$$w_{ji} = w_{ji} + \Delta w_{ji}. \quad (2.10)$$

Derivace ze vzorce 2.9 se počítá dle vzorce pro derivování složených funkcí, tedy:

$$\frac{\delta E_k}{\delta w_{ji}} = \frac{\delta E_k}{\delta y_j} \frac{\delta y_j}{\delta \xi_j} \frac{\delta \xi_j}{\delta w_{ji}}. \quad (2.11)$$

Pro derivaci $\frac{\delta \xi_j}{\delta w_{ji}}$ dle 2.1 platí:

$$\frac{\delta \xi_j}{\delta w_{ji}} = y_i^1. \quad (2.12)$$

Derivaci členu $\frac{\delta y_j}{\delta \xi_j}$ získáme úpravou derivace aktivační funkce (2.6). Tedy

$$\frac{\delta y_j}{\delta \xi_j} = \frac{\lambda_j e^{-\lambda_j \xi_j}}{(1 + e^{-\lambda_j \xi_j})^2} = \lambda_j y_j (1 - y_j). \quad (2.13)$$

Po dosazení 2.12 a 2.13 do 2.11 dostáváme:

$$\frac{\delta E_k}{\delta w_{ji}} = \frac{\delta E_k}{\delta y_j} y_i \lambda_j y_j (1 - y_j). \quad (2.14)$$

Výpočet parciální derivace $\frac{\delta E_k}{\delta y_j}$ je už poněkud složitější. Právě zde je užito zpětné šíření, podle kterého je algoritmus pojmenován. Pokud je neuron výstupní, pak derivaci umíme snadno spočítat derivováním 2.8:

$$\frac{\delta E_k}{\delta y_j} = y_j - d_{kj}. \quad (2.15)$$

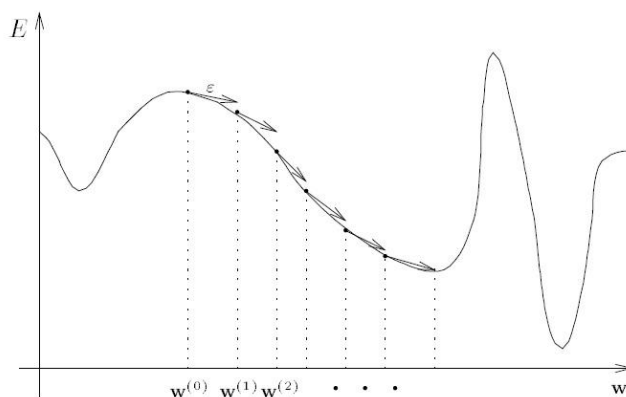
¹vstupem do neuronu j je výstup z neuronu i

Pro skryté neurony využijeme následující vzorec:

$$\frac{\delta E_k}{\delta y_j} = \sum_{r \in j \rightarrow} \frac{\delta E_k}{\delta y_r} \frac{\delta y_r}{\delta \xi_r} \frac{\delta \xi_r}{\delta w_j}. \quad (2.16)$$

Ve vzorci převedeme parciální derivaci $\frac{\delta E_k}{\delta y_j}$ na derivaci $\frac{\delta E_k}{\delta y_r}$, kde neuron r je takový, že neuron j je jeho vstup. V praxi tedy při výpočtu vypočteme derivaci $\frac{\delta E_k}{\delta y_j}$ pro výstupní neurony podle vzorce 2.15. Poté počítáme derivace pro neurony v poslední skryté vrstvě, pro které již známe členy $\frac{\delta E_k}{\delta y_r}$. Stejným postupem pokračujeme postupně po vrstvách až ke vstupním neuronům.

Na obr. 2.6 je znázorněna chybová funkce $E(\mathbf{w})$, kde \mathbf{w} je mnohazměrný vektor vah, který představuje konfiguraci v daném čase. Na počátku, kdy jsou váhy nastaveny náhodně, se dá předpokládat, že chyba bude velká (konfigurace v čase $\mathbf{w}^{(0)}$). Při adaptaci dle 2.9 sestrojíme v tomto bodě ke křivce tečný vektor a posuneme se o ε dolů ve směru gradientu. Z obrázku je patrné, že chyba při nové konfiguraci $\mathbf{w}^{(1)}$ je určitě menší než předchozí. Toto platí zřejmě i pro každou další chybu. Je tedy jasné, že v konečném počtu kroků bude dosaženo lokálního minima, kde je gradient nulový a adaptace se tedy zastaví. Problémem je, že výpočetní čas může být velmi dlouhý a jak je vidět na obr. 2.6, dosažené minimum nemusí být globální.



Obrázek 2.6: Gradientní metoda [2, str. 55]

2.5.2 Problémy algoritmu Back-Propagation

Kromě výše zmíněných problémů nás může první problém potkat již na začátku, kdy volíme hodnotu rychlosti učení ε . Jestliže bude hodnota ε příliš malá, bude celý proces trvat nekonečně dlouho. Naopak při příliš velké rychlosti učení ε bude chybová funkce divergovat a nikdy tedy nedosáhneme ani lokálního minima. Zvolit vhodnou hodnotu ε je ale nesmírně obtížné a je nutné již mít určitou zkušenost. V průběhu adaptace je vhodné ε měnit (obvykle na počátku volíme nižší rychlost učení ε). Naštěstí, v dnešní době má již mnoho algoritmů vycházejících ze základní varianty Back-Propagation optimalizační metody pro adaptaci rychlosti učení ε .

Poměrně častá modifikace využívá tzv. *moment*, který vyjadřuje předchozí změnu vah:

$$\Delta w_{ji}^{(t)} = -\varepsilon \frac{\delta E}{\delta w_{ji}(\mathbf{w}^{(t-1)})} + \alpha \Delta w_{ji}^{(t-1)}, \quad (2.17)$$

kde $0 < \alpha < 1$ je parametr momentu, který určuje míru vlivu předchozí změny. Obvykle se volí. Díky této úpravě algoritmus lépe kopíruje tvar chybové funkce $E(\mathbf{w})$. Touto úpravou lze alespoň částečně vyřešit problém s volbou ε .

Řešením problému s lokálními minimy je například náhodné vnášení šumu do konfigurace neuronové sítě, jestliže se síť dostala do lokálního minima. Tím docílíme toho, že se síť dostane mimo tuto oblast a může dále konvergovat k hlubšímu globálnímu minimu. Dalším možným řešením je užití pravděpodobnostní metody, která pomáhá vyhnout se lokálnímu minimu. Nevýhodou je ovšem její rychlost.

Všechny výše zmíněné problémy lze také vyřešit užitím propracovanějších metod nelineární optimalizace jako je *Newtonova metoda* či *Metoda sdružených gradientů* [2].

Software STATISTICA, ve kterém byl vytvořen model v praktické části této práce, používá metodu Quasi-Newton, která vychází z Newtonovy metody.

2.6 Použití neuronových sítí v oblasti energetiky

Neuronové sítě mají uplatnění v nejrůznějších oblastech, ať už se jedná o rozpoznávání obrazu, analýzu signálu či nejrůznější předpovědi v průmyslové praxi. Pomáhají například i s vyhodnocováním signálů EEG a EMG v lékařství. V neposlední řadě se uplatňují i v tzv. *expertních systémech*. [2]

Protože se tato práce zabývá problematikou z oboru energetiky, budou v následující části zmíněny aplikace z této oblasti.

Kalagirou[5] uvádí, že vícevrstevná síť perceptronů má rozmanité užití v oblasti energetiky. Konkrétními aplikacemi je například modelování solárního parního generátoru pro předpověď průměrné měsíční produkce páry nebo pro výpočet faktoru, který určuje podíl energie absorbované a užitečné. Dále se autor zmiňuje o aplikacích v oblasti solárního vytápění či predikce spotřeby energie v pasivních domech. V textu je uvedeno mnoho aplikací nejen z oblasti spotřeby elektrické energie a techniky prostředí, ale například i ohledně chlazení.

O aplikaci v oblasti solárních systémů pojednává Kalagirou a kol.[6] kolektivu autorů. Neuronová síť zde byla použita pro výpočet denního množství energie ze solárního článku nebo pro výpočet teploty v zásobníku na konci denního cyklu. Neuronová síť zde dosáhla velmi dobrých výsledků a to i v případě, že data nebyla úplná. Pro stejný úkol byla dokonce použita vícenásobná lineární regrese, avšak její výsledky nebyly zdaleka tak dobré. Jako velkou přednost neuronové sítě zmiňují autoři zejména její jednoduchost či možnost dále zlepšovat její výkon přidáním dalších provozních dat.

Aplikací neuronových sítí v problematice chlazení, klimatizací a tepelných čerpadel se v článku[7] zabýval Mohanraj at al. Autoři uvádí, že existuje více než devadesát různých úspěšných použití neuronových sítí ve zmíněné oblasti. Úspěšná byla zejména aplikace v RACHP systémech, kde bylo dosaženo vysoké přesnosti.

Problematice předpovědi spotřeby obnovitelných zdrojů energie se věnoval Azadeh a kol[8]. Autoři tvrdí, že vícevrstevná perceptronová síť dosahuje přesnosti 99,9 %. Srovnání s běžnou a fuzzy regresí ukazuje výhody a převahu neuronové sítě. Přesné odhady jsou velmi důležité nejen pro vlády a osoby rozhodující v energetickém sektoru, ale například i ve venkovských oblastech, kde neexistuje možnost měřit spotřebu energie z obnovitelných zdrojů.

Pánove Ekici a Aksoy [9] se zabývali aplikací neuronových sítí v problematice energetické náročnosti budov a uspokojujícími výsledky (odchylka 3,43 %, predikční úroveň 94,8 až 98,5 %).

Jak je vidět výše v tomto odstavci, neuronové sítě mají velmi široké uplatnění, mimo jiné i v oblasti energetiky, kde pomocí nich bylo provedeno již mnoho rozsáhlých výzkumů a modelů. Ve výše uvedeném výčtu je uvedeno sice jen velice málo problémů, kde byly sítě užity, avšak už i zde je vidět, že síť se dá využít ve velmi širokém spektru problémů.

Kapitola 3

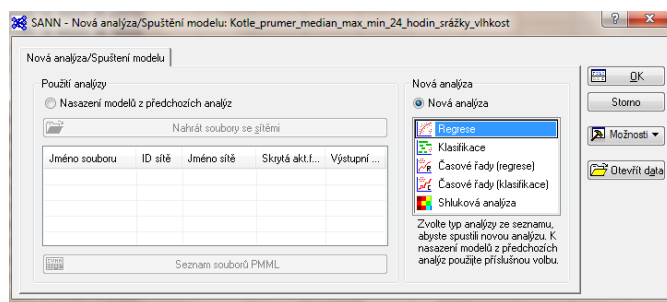
Neuronové sítě v programu STATISTICA

SANN je modul programu STATISTICA a je to zkratka slov STATISTICA Automated Neural Network. Tento modul lze snadno použít i bez předchozích znalostí se základním programem STATISTICA. Práce s tímto modulem bude představena v následující kapitole.

Všechny dále zmíněné postupy odpovídají verzi STATISTICA 10. Následující text je zpracován zejména s použitím nápovědy softwaru STATISTICA.

3.1 Výběr analýzy

Po kliknutí na „Neuronové sítě“ v záložce „Data Mining“ se otevře dialogové okno s názvem SANN - Nová analýza/Spouštění modelu: název souboru (viz. obrázek 3.1).



Obrázek 3.1: STATISTICA – Nová analýza/Spouštění modulu: název souboru

Vybereme-li v této nabídce možnost „Nasazení modelů z předchozích analýz“, budeme pracovat se sítí, kterou jsme již předtím vytvořili. Tyto sítě musí být uloženy ve formátu PMML a musí být vytvořeny v modulu SANN. Je nutné vybrat síť, která slouží ke stejnému účelu (regrese, klasifikace, ...) a má stejný počet vstupů a výstupů. PMML soubor nahrajeme stisknutím tlačítka „Nahrát soubory se sítěmi“. Položka „Seznam souborů PMML“ vygeneruje tabulku PMML souborů, které jsme se pokusili úspěšně či neúspěšně nahrát. Možnost „Nová analýza“ vede k vytvoření nové neuronové sítě pomocí aktuálních dat. V rámečku pod touto volbou můžeme volit mezi několika typy [10]:

Regrese umožňuje předpovědět spojitý výstup na základě spojitých či kategoriálních predikátorů.

Klasifikace třídí vzory do odpovídajících tříd. Používá se například v bankovníctví pro posouzení schopnosti splácení daného klienta nebo v medicíně při problémech, zda poslat pacienta na dané vyšetření.

Časové řady (regrese) slouží k modelování spojitých proměnných, které prochází vývojem v čase. Model může záviset buď jen zpětně na hodnotách výstupu, nebo můžeme přidat další spojitě či kategoriální vstupy.

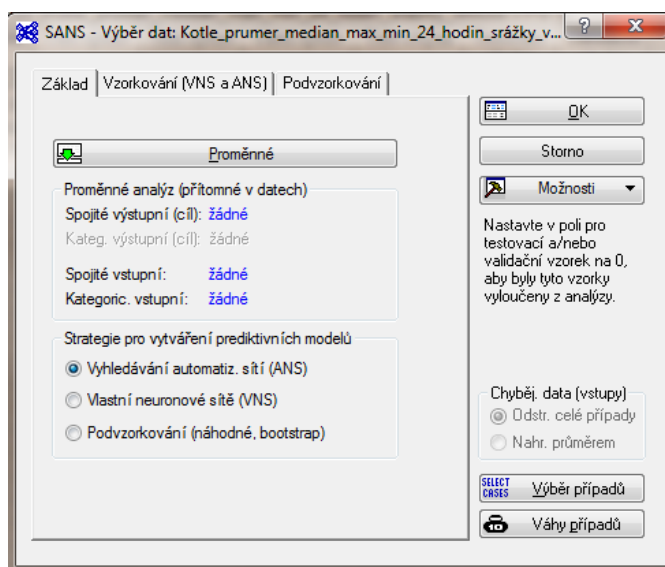
Časové řady (klasifikace) použijeme v případě, že výstupní proměnná je kategoriální. Stejně jako v předchozím případě můžeme modelovat výstup v závislosti na jeho historických stavech, nebo můžeme přidat další kategoriální i spojitě vstupy.

Shluková analýza využívá učení bez učitele. Síti předložíme pouze vstupní data a cílem je detekovat netriviální shluky v datech. Jde o tzv. Kohonenovu síť.

Protože se v dalším textu budeme zabývat regresí pomocí neuronové sítě, vybereme možnost „Nová analýza“ a typ „Regrese“ (defaultní volba) a pro další pokračování klikneme na tlačítko „OK“.

3.2 Regrese pomocí neuronových sítí

3.2.1 Výběr proměnných a strategie pro vytvoření predikčních modelů



Obrázek 3.2: STATISTICA – SANS – Výběr dat: Název souboru

V dialogovém okně pro regresi, která je na obrázku 3.2, v kartě „Základ“ výběrem možnosti „Proměnné“ vybereme spojitou cílovou proměnnou, spojitě predikátory a kategoriální predikátory (např. dny v týdnu, pohlaví atd.). Samozřejmě jestliže to nevyžaduje úloha, není nutné použít oba typy predikátorů. Ve spodní části stejné karty lze zvolit typ strategie pro vytváření predikčních modelů. Jedná se o:

Vyhledávání automatizovaných sítí (ANS) umožňuje vytvořit neuronovou síť pro zadaná data. Je navržena architektura a konfigurace sítě, která reprezentuje vztah

mezi vstupními a výstupními proměnnými co nejlépe.

Vlastní neuronové sítě (VNS) je nástroj, kde uživatel vybere požadovanou typologii a tréninkový algoritmus.

Podvzorkování (náhodné, bootstrap) umožňuje vytvořit výběr dat z originálního souboru. Stejnou možnost nabízí i karta „Podvzorkování“.

3.2.2 Vzorkování dat

Karta „Vzorkování (ANS a VNS)“ je rozdělena na dvě části podle metody vzorkování. Vzorkování rozdělí data do tří skupin – trénovací, testovací a validační. Toto rozdělení je důležité z důvodu, že síť se může perfektně naučit trénovací data, ale výsledek pro nová data již nemusí být uspokojivý. Tomuto stavu říkáme přeučení sítě. Trénovací množina je náhodně vybraná část dat, která slouží k učení sítě. Testovací množina je další část dat, která slouží k zastavení učení, aby nedošlo k přeučení. Validací množinu tvoří data, která dosud neměla síť k dispozici a ověříme na nich konečnou kvalitu modelu. Validací vzorek není pro tvorbu sítě nutný, avšak při dostatku dat je dobré si na validací množinu vyhradit alespoň 15 % dat. Obvykle se doporučuje dělení dat v poměru 50-25-25 nebo 70-15-15. [10]

V části „Náhodné vzorkování“ lze určit procentuální rozsah trénovací, testovací a validační množiny tak, aby součet nebyl přirozeně větší než 100 %. Poznamenejme, že součet může být dokonce menší než 100 %. V tomto případě existují data, která nejsou zahrnuta ani v jedné z množin, což je výhodné například při velkém rozsahu souboru. Zahrnutí pouze části dat vede ke snížení časové náročnosti učícího procesu. Pokud nechceme data dělit do těchto tří podmnožin, nastavíme rozsah testovací a validační množiny na nulu. Semínko pro náhodný výběr musí být celé kladné číslo a je užito pro náhodné generování množiny ze vstupního souboru dat. Při použití stejného semínka na začátku procesu dostaneme totožnou množinu jako při předchozím průchodu. Chceme-li dostat rozdílnou množinu, musíme změnit hodnotu semínka.

Druhou možností je určit tzv. vzorkovací proměnnou, podle níž se data rozdělí do skupin. Po zakliknutí možnosti „Vzorkovací proměnná“ můžeme kliknutím na tlačítko „Trénovací vzorek“ a „Výběr vzorkovací proměnné“ vybrat vzorkovací proměnnou. Po specifikaci této hodnoty bude její název a kód zobrazen vedle tlačítka „Trénovací vzorek“. Stejně postupujeme i u testovacích a validačních vzorků.

Tlačítko „Výběr případů“ slouží ke specifikaci případů, které budou do analýzy zahrnuty.

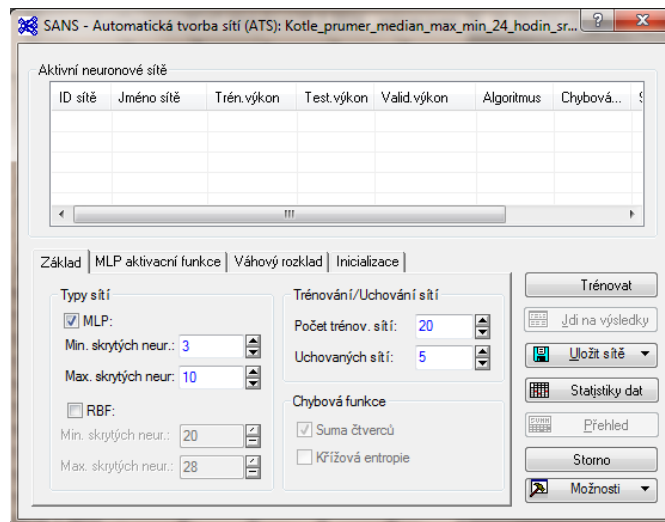
Na počátku adaptace jsou váhy všech spojení nastaveny na 1. Tlačítkem „Váhy případů“ můžeme toto defaultní nastavení změnit. Je to užitečné zejména v případě, kdy některým případům chceme přiřadit menší důležitost než jiným. To je vhodné například v nevyvážených datových souborech nebo v situacích, kdy je pro nás důležitá přesnější predikce některých případů. Prioritním případům přiřadíme váhu větší jak jedna. Chyba způsobená těmito případy je tedy větší. Naopak pro méně důležité případy navolíme váhu menší než 1 a jejich chyba tedy klesá. Případy s váhou nula budou z dat vyřazeny. Váhy mohou být celá či desetinná čísla.

Pokud nemáme speciální požadavky a do analýzy chceme zařadit všechny případy se stejnou váhou, nebudeme výše uvedené volby používat.

Jestliže máme vybrané proměnné a případně vše další nastavené podle svých požadavků, klikneme na tlačítko „OK“ a dostaneme se do dialogu s názvem „SANS – Automatická tvorba sítí (ATS): Název souboru“.

3.2.3 Automatická tvorba sítí – volba parametrů sítě

Na obrázku 3.3 je vidět právě otevřené dialogové okno. V sekci „Základ“ lze zvolit, zda chceme síť MLP (Multiple Layer Perceptron – vícevrstevná perceptronová síť), nebo PBF (Radial Basis Function).



Obrázek 3.3: STATISTICA – SANS – Automatická tvorba sítí: Název souboru

MLP je nejpoužívanější neuronová síť, která ale při tréninku využívá iterace. To vede při větším počtu skrytých neuronů nebo dat k větší časové náročnosti. Naopak PBF je velmi rychlá při trénování, ale její předpověď je obvykle horší než u sítě MLP (zejména máme-li větší počet vstupů).

U každé ze sítí lze zvolit minimum a maximum skrytých neuronů. Zvolit počet skrytých buněk není jednoduché, a proto dle je dle mého názoru nejlepší, ponechat volbu, kterou program nabídne. Obecně větší počet skrytých neuronů vede ke zvýšení přesnosti sítě, ale síť je rozsáhlejší, pomalejší a je obtížnější ji vytrénovat. Hlavně je však více náchylná k přetrénování.

Jestliže máme data, která jsou relativně podobná, obsahují velké množství nestandardních dat (přetrénování) nebo počet případů je malý, je lepší použít menší počet skrytých buněk. Jestliže zkusíme různé konfigurace a najdeme konfiguraci, kde je dobrý testovací výkon, avšak ve skutečnosti není předpověď ideální, došlo pravděpodobně k přetrénování sítě.

Počet trénovaných sítí je vhodné volit co nejvyšší s ohledem na výkonnost našeho počítače. Počet uložených sítí volíme dle našeho uvážení. Software vybere zadaný počet nejlepších sítí a ty uloží k další práci.

Sekce „MLP aktivační funkce“ umožňuje vybrat různé aktivační funkce. STATISTICA rozlišuje aktivační funkci pro skryté a výstupní neurony. Není tedy problém vytvořit síť, která má jistou aktivační funkci pro skryté neurony a jinou pro výstupní. Přehled aktivačních funkcí pro MLP v systému STATISTICA lze vidět v tabulce 3.1.

Záložka „Váhový rozklad“ přesahuje běžné využití neuronových sítí, a proto je lepší nezabývat se touto záložkou. Váhový rozklad má význam spíše ve chvíli, kdy se snažíme co nejlépe nastavit víceméně již nakoncipovanou neuronovou síť. Užitím váhového rozkladu můžeme přispět k tomu, aby u sítě nedošlo k přeučení. V případě této metody jsou penalizovány váhy, které mají velký vliv, a naopak jsou podporovány menší váhy, což vede k tomu, že v síti nejsou velké rozdíly v hodnotách vah.

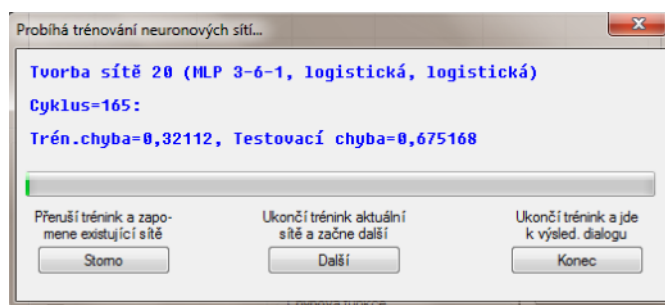
Funkce	Předpis	Obor hodnot
Identita	ξ	$(-\infty, \infty)$
Logistická sigmoida	$\frac{1}{1+e^{-\xi}}$	$\langle 0, 1 \rangle$
Hyperbolický tangens	$\frac{e^{\xi} - e^{-\xi}}{e^{\xi} + e^{-\xi}}$	$\langle -1, 1 \rangle$
Exponenciální	$e^{-\xi}$	$\langle 0, \infty \rangle$
Sinus	$\sin(\xi)$	$[0, 1]$

Tabulka 3.1: Aktivační funkce v systému STATISTICA

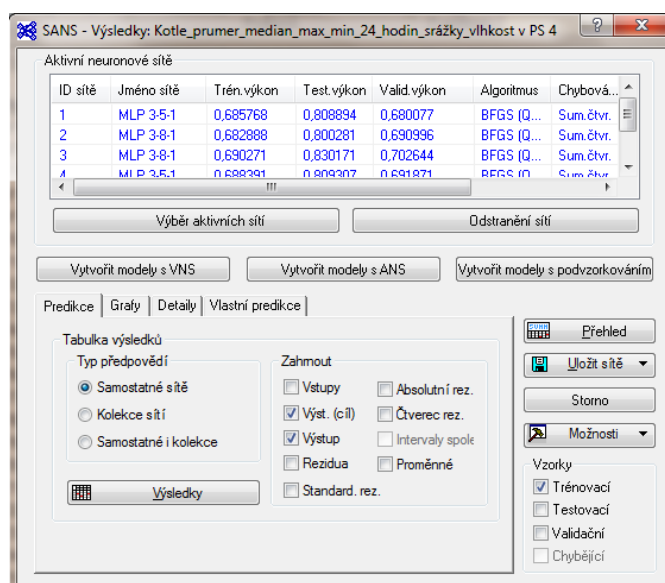
V tabulce zvolíme, kde chceme váhový rozklad použít a následně specifikujeme minimální a maximální hodnotu váhového rozkladu pro příslušnou vrstvu. [11]

Před stisknutím příkazu „Trénovat“ ještě můžeme zvolit chybovou funkci. Trénování trvá různě dlouhou dobu dle výkonnosti počítače, náročnosti sítě atd. Okno, které uvidíme na obrazovce v průběhu trénování lze vidět na obr. 3.4.

Po skončení trénování se zobrazí dialogové okno (obrázek 3.5). Detailnější popis tohoto dialogu následuje v další kapitole.



Obrázek 3.4: STATISTICA – Trénování



Obrázek 3.5: STATISTICA – Výsledky

3.2.4 Výsledkové dialogy

Tabulka „Aktivní neuronové sítě“, která je zobrazena na obrázku 3.5, zobrazuje uchované sítě (počet závisí na naší předchozí volbě). V prvním sloupci vidíme typ sítě a její konfiguraci, v dalších následují trénovací, testovací a validační výkony, trénovací algoritmus, použitá chybová funkce a aktivační funkce pro skryté i výstupní neurony. Kliknutím na tlačítko „Přehled“ v pravé části okna dostaneme stejný výpis, který se ale uloží do sešitu pod současnou analýzu.

Kliknutím na tlačítko „Výběr aktivních sítí“ můžeme vybrat užší výběr z uchovaných sítí. V tabulce „Aktivní neuronové sítě“, pak budou zobrazeny jen námi vybrané sítě. Chceme-li naopak nějaké sítě úplně vymazat, provedeme to kliknutím na tlačítko „Odstranění sítí“.

Další tři tlačítka slouží k vytvoření dalších modelů.

Ve spodní části dialogu se nachází čtyři záložky: Predikce, Grafy, Detaily a Vlastní predikce.

Záložka „Predikce“

V záložce Predikce můžeme kliknutím na tlačítko „Výsledky“ uložit do jednoho sešitu výstupní hodnoty ze sítě, cílový výstup, rezidua atd. Co se uloží závisí na naší volbě. Asi nejčastěji budeme chtít uložit cílový výstup a výstupy z neuronové sítě. Toho docílíme, jestliže v levé spodní části dialogu zvolíme „Samostatné sítě“ a v sekci „Zahrnout“ vybereme možnosti „Výstup (cíl)“ a „Výstup“. Dále můžeme v této sekci zvolit libovolné další položky dle našich požadavků.

Další možností v levé části okna jsou „Kolekce“. Dle [12] jsou kolekce soubor neuronových sítí kooperujících za účelem lepší kvalitnější předpovědi. Do kolekce může být zařazeno libovolné množství sítí, které obecně ani nemusí mít stejné vstupy a výstupy. V systému STATISTICA mají ovšem všechny uložené „Aktivní neuronové sítě“ stejné vstupy i výstupy¹, a proto se dále budu zabývat jen touto možností. V tomto případě odhaduje síť výstup kolekce jako kombinaci výstupů jednotlivých sítí. Spojité výstupy jsou průměrovány. Obecně mohou být jednotlivým výstupům přiřazovány různé váhy, avšak software STATISTICA volí všechny tyto váhy rovny jedné. V případě kategoriálních výstupů² jsou výstupy kombinovány v módu výstupu sítí. Stejně jako v případě samostatné sítě můžeme dle našeho uvážení volit výpis různých možností v sekci „Zahrnout“.

Posledním typ předpovědi s názvem „Samostatné i kolekce“ vygeneruje kombinaci obou předchozích typů.

V neposlední řadě lze v pravém dolním rohu dialogu zvolit, pro jaké vzorky budou výsledky vypisovány.

Záložka „Grafy“

Následující záložka, jak již název napovídá, umožňuje vytvořit nejrůznější grafy, které často poslouží k lepší interpretaci výstupů než zobrazení ve formě tabulek.

Nejprve vybereme proměnné, které budou na jednotlivých osách. Zde bych ráda upozornila na mnohdy asi ne zcela jasný detail v názvech proměnných Výstup (cíl) a Výstupní proměnná. Výstup (cíl) představuje pozorovanou hodnotu a Výstupní proměnná jsou hodnoty výstupu z neuronové sítě. Chceme-li vykreslit histogram stačí zvolit proměnnou pouze na Ose X. Přirozeně pro 2D X a Y graf je nutné zvolit proměnnou na ose X a Y a pro

¹modelují stejný problém

²použití např. pro klasifikaci

3D graf i třetí proměnnou. Software vždy vykreslí požadované grafy pro všechny sítě, kterou jsou zvoleny v tabulce „Aktivní neuronové sítě“. Při velkém počtu uchovaných sítí je tedy lepší zvolit kvůli přehlednosti grafu jen některé a grafů raději vytvořit více pro různé výběry sítí.

Grafy je možné vykreslit standardně pomocí karty „Grafy“, musíme však dbát na to, abychom měli označenou správnou tabulku v sešitě výsledků jako vstup.

Graf trénovací chyby zobrazíme kliknutím na tlačítko „Trénovací chyba“.

Záložka „Details“

Jedná se o záložku, která nám poskytne pravděpodobně nejvíce informací o vzniklých sítích.

Tlačítkem „Přehled“ získáme výpis do sešitu, který obsahuje stejné informace jako tabulka v horní části tohoto dialogu „Aktivní neuronové sítě“. Tlačítko „Váhy“ vygeneruje výpis vah. Výpis je ve tvaru jako na obrázku 3.6. Skrytá a vstupní strannost je označení STATISTICY pro bias.

Váhy sítí (Kotle_prumer_median_max_min_24_hodin_srážky_vlhkost v PS 2)			
ID vah	Spojení 1.MLP 3-3-1	Hodnoty vah 1.MLP 3-3-1	Spojení 2.MLP 3-5-1
1	dva -> skrytý neuron 1	0,63183	dva -> skrytý neuron 1
2	tři -> skrytý neuron 1	-0,86145	tři -> skrytý neuron 1
3	čtyři -> skrytý neuron 1	0,79453	čtyři -> skrytý neuron 1
4	dva -> skrytý neuron 2	0,19516	dva -> skrytý neuron 2
5	tři -> skrytý neuron 2	-2,63669	tři -> skrytý neuron 2
6	čtyři -> skrytý neuron 2	0,18083	čtyři -> skrytý neuron 2
7	dva -> skrytý neuron 3	2,00924	dva -> skrytý neuron 3
8	tři -> skrytý neuron 3	2,56287	tři -> skrytý neuron 3
9	čtyři -> skrytý neuron 3	-0,98859	čtyři -> skrytý neuron 3
10	vstup. strannost -> skrytý neuron 1	-0,24422	dva -> skrytý neuron 4
11	vstup. strannost -> skrytý neuron 2	0,64811	tři -> skrytý neuron 4
12	vstup. strannost -> skrytý neuron 3	-1,99351	čtyři -> skrytý neuron 4
13	skrytý neuron 1 -> průměr K1	0,86623	dva -> skrytý neuron 5
14	skrytý neuron 2 -> průměr K1	0,42310	tři -> skrytý neuron 5
15	skrytý neuron 3 -> průměr K1	0,73478	čtyři -> skrytý neuron 5
16	skrytá strannost -> průměr K1	0,77999	vstup. strannost -> skrytý neuron 1
17			vstup. strannost -> skrytý neuron 2
18			vstup. strannost -> skrytý neuron 3
19			vstup. strannost -> skrytý neuron 4
20			vstup. strannost -> skrytý neuron 5
21			skrytý neuron 1 -> průměr K1
22			skrytý neuron 2 -> průměr K1
23			skrytý neuron 3 -> průměr K1
24			skrytý neuron 4 -> průměr K1
25			skrytý neuron 5 -> průměr K1
26			skrytá strannost -> průměr K1
27			

Obrázek 3.6: STATISTICA – Výpis vah

Stisknutím tlačítka „Korelační koeficienty“ dostaneme výpis korelačních koeficientů pro trénovací, testovací a validační vzorky jednotlivých sítí. Tlačítko „Statistiky pro předpovědi“ vygeneruje tabulku s přehledem maximálních a minimálních predikcí pro trénovací, testovací a validační data a dále rezidua a standardní rezidua pro stejné tři množiny dat. Tlačítko „Statistiky dat“ vygenerují statistiky (průměr, směrodatná odchylka, minimum a maximum) pro vstupní a výstupní proměnné.

Globální analýza citlivosti slouží k posouzení, jak velký vliv mají jednotlivé vstupní proměnné na výstup. Obecně platí, že čím větší je číslo u dané proměnné, tím větší je její vliv. Naopak má-li nějaká proměnná hodnotu menší než jedna, není v dané síti důležitá a může naopak ještě její výsledek zhoršovat. Proměnná není významná ani tehdy, je-li její hodnota sice větší než jedna, ale mnohonásobněkrát menší, než u zbylých proměnných. V tomto případě sice daná proměnná nezhoršuje výsledek sítě, ale ani ho nijak výrazně neovlivňuje v pozitivním smyslu. Je tedy možné použít model bez tohoto vstupu a výsledky budou téměř shodné jako u modelu, kde je tento vstup použit.[11]

Záložka „Vlastní predikce“

Tuto záložku můžeme využít později k predikci hodnot pro nové vstupy. Kliknutím na tlačítko „Vlastní vstupy“ můžeme přidat hodnoty vstupních proměnných a po kliknutí na „OK“ se zobrazí výstup pro jednotlivé uchované sítě. Kliknutím na tlačítko „Vlastní predikce“ zobrazíme tabulku s našimi vstupy a jejich výstupy do sešitu.

3.3 Algoritmus výpočtu neuronové sítě pro regresi v systému STATISTICA

Je zřejmé, že v případě regrese pracujeme dle daného problému s rozdílnými intervaly hodnot. Zřídka lze ovšem reálný problém řešit na intervalu $(0,1)$. A právě v tomto intervalu počítá systém STATISTICA. Software nejprve hodnoty převede na interval $(0,1)$, následně neuronová síť pracuje pouze s tímto intervalem a nakonec je opět proveden předpočet hodnot do původního intervalu.

Následující část se zabývá postupem výpočtu v systému STATISTICA. Nejedná se o univerzální postup pro neuronové sítě v různých programech, každý software užívá jiné metody.

STATISTICA nepracuje během výpočtu se vstupy, potažmo hodnotami, v rozsahu celé reálné osy, ale pouze s hodnotami v intervalu $(0,1)$. Proto se nejprve provede přepočtení vstupů na tento interval. Přepočtení se provádí pomocí lineární interpolace.

Z „Tabulky výsledků“, ve které máme uloženy krom cílové a predikované hodnoty také vstupy, vypíšeme pomocí záložky „Statistiky“ a dále „Základní statistiky“ minimální a maximální hodnoty pro všechny vstupy a cílovou proměnnou. Následně minimální hodnotě přiřadíme hodnotu 0 a maximální 1. Poté vstupům přiřadíme přepočítanou hodnotu $vstup_{prepocitany}$ dle vzorce:

$$vstup_{prepocitany} = vstup \cdot \delta - min \cdot \delta, \quad (3.1)$$

kde δ počítáme jako:

$$\delta = \frac{1}{min - max}, \quad (3.2)$$

kde min je minimální hodnota daného vstupu a max maximální.

Poté vypočteme běžným způsobem výstup $vystup_{prepocitany}$ z neuronové sítě, který opět musíme převést pomocí lineární extrapolace na celou reálnou osu. To provedeme pomocí vzorce:

$$vystup = \frac{vystup_{prepocitany} + \delta \cdot min}{\delta}, \quad (3.3)$$

kde δ počítáme opět dle vztahu 3.2 s rozdílem, že za hodnoty min a max dosazujeme minimum a maximum cílové proměnné.

Kapitola 4

Predikční modely pro plánování provozu zařízení na energetické využití odpadu

Modely vzniklé v rámci této bakalářské práce jsou součástí rozsáhlejšího projektu se společností ZEVO Malešice. Cílem bylo vytvořit model pro predikci předávaného elektrického výkonu do přenosové soustavy. Aby bylo možné tohoto cíle dosáhnout, bylo nutné namodelovat jednotlivé technologické bloky provozu. Většina modelů poskytovala dobré výsledky při použití regresní analýzy. Avšak u modelu vyrobené páry nebyla lineární regrese tak úspěšná, a proto bylo vyzkoušeno, jestli neuronové sítě nebudou dávat lepší výsledky. U Laktosu to bylo zejména z důvodu kategoriálních dat, pro které je regrese složitá. Právě těmito dvěma modely se zabývá tato kapitola.

4.1 ZEVO Malešice

ZEVO Malešice je největší ze tří českých spaloven komunálního odpadu a nachází se na území Prahy 10. Její zpracovatelská kapacita je asi 310 tun odpadu za rok.

Základem technologického zařízení jsou 4 kotle s válcovými rošty, přičemž každý může za hodinu spálit až 15 t tuhého komunálního odpadu (TKO) a vyrobit 36 t páry s parametry 235 °C a 1,37 MPa. Běžně jsou v provozu 3 kotle. Část vyrobené páry se používá pro vlastní spotřebu v technologii, část pokrývá nepravidelnou poptávku firmy Laktos (běžně asi 6 t/h, maximum je 18 t/h). Zbytek páry jde na odběrovou turbínu, kde část je odebrána do soustavy centrálního zásobování teplem (neregulovaný odběr) a ze zbylé části se vyrábí elektrická energie. Kombinovaná výroba elektrické energie a tepla je známá pod názvem kogenerace. Díky kogenerační jednotce, která je v provozu od roku 2011, je spalovna schopna využívat svoji kapacitu na 100 %. Zařízení disponuje rozsáhlým systémem čištění spalin, jehož na první pohled viditelnou částí je 177,5 metru vysoký komín, sloužící pro odvod spalin. [13] [14]

4.2 Hodnocení kvality modelů

Kvalita modelů byla hodnocena zejména podle koeficientu korelace R a průměrné absolutní chyby MAE (mean absolute error). Korelační koeficient udává vztah mezi skutečnou (pozorovanou) hodnotou a tou, která je výstupem z modelu. Byl zvolen z důvodu

porovnání kvality jednoduchých lineárních a nelineárních regresních modelů a neuronových sítí. Průměrná absolutní chyba má posloužit k lepší představě provozovatele zařízení o kvalitě modelu, protože udává průměrnou chybu predikce v t/h.

V následujících dvou částech je matematické zavedení koeficientu korelace R a průměrné absolutní chyby MAE .

4.2.1 Korelační koeficient

Koeficient korelace $\rho(X_1, X_2)$ je definován dle vztahu:

$$\rho(X_1, X_2) = \begin{cases} \frac{C(X_1, X_2)}{\sqrt{D(X_1)D(X_2)}} & \text{je-li } D(X_1) \neq 0, D(X_2) \neq 0 \\ 0 & \text{jinak,} \end{cases} \quad (4.1)$$

kde reálné číslo $C(X_1, X_2)$ nazýváme *kovariancí* náhodných veličin X_1 a X_2 , která je definovaná:

$$C(X_1, X_2) = E([X_1 - E(X_1)] \cdot [X_2 - E(X_2)]). \quad (4.2)$$

$E(X)$ označuje střední hodnotu veličiny X a $D(X)$ její rozptyl. Korelační koeficient ρ má několik důležitých vlastností:

1. $\rho(X_1, X_2) \in \langle -1, 1 \rangle$
2. $\rho(X_1, X_1) = 1$
3. Jestliže $\rho = 1$, pak existuje $a, b \in \mathbb{R}, b > 0$ takové, že $X_2 = a + bX_1$ s pravděpodobností 1.
4. Jestliže $\rho = -1$, pak existuje $a, b \in \mathbb{R}, b < 0$ takové, že $X_2 = a + bX_1$ s pravděpodobností 1.
5. Jsou-li náhodné veličiny X_1 a X_2 nezávislé, pak jsou náhodné veličiny nekorelované.

Vlastnost 5 však neplatí naopak. Tedy to, že jsou veličiny nekorelované ještě neznamená, že jsou nezávislé, protože korelace udává pouze míru lineární závislosti. Je zřejmé, že čím větší je koeficient korelace, tím vyšší je lineární závislost daných veličin. [15]

4.2.2 Průměrná absolutní chyba

Průměrná absolutní chyba, označovaná MAE , slouží k posouzení toho, jak dobrá je předpověď daného modelu vůči skutečným naměřeným datům. Pro její výpočet používáme vztah:

$$MAE = \frac{1}{n} \sum_{i=1}^n |d_i - y_i| = \frac{1}{n} \sum_{i=1}^n |e_i|, \quad (4.3)$$

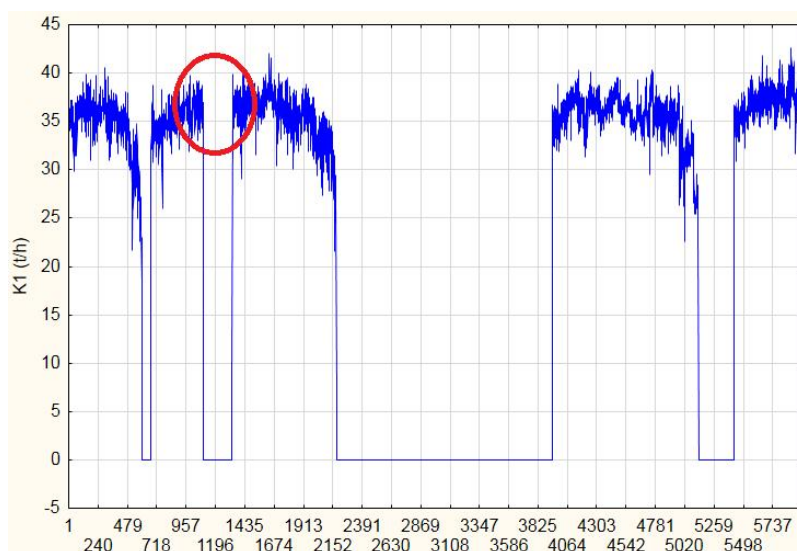
kde d_i je naměřená hodnota a y_i je výstup z modelu. Je zřejmé, že výsledek je tím lepší, čím je průměrná absolutní chyba nižší. [16]

4.3 Model kotlů

Model byl vytvářen pro každý ze čtyř kotlů kotel zvlášť. Výstupem modelů je přirozeně množství páry, které v daném kotli vzniká spalováním odpadu. Predikce množství páry je velmi důležitá pro další technologické bloky, protože přímo ovlivňuje jejich činnost.

4.3.1 Analýza vlivů faktorů

Jelikož dávkování odpadu je stále přibližně stejné, není nutné jej uvažovat. Dalším možným faktorem byla vlhkost v ovzduší a srážky. Avšak prozkoumáním dat, které poskytl ČHMÚ, nebyla zjištěna výraznější závislost. Na základě konzultace s operátorem jsem se nakonec rozhodla vytvářet modely v závislosti na stavu v minulých dnech a to z důvodu, že parní výkon kotle se od najetí po nějakou dobu zvyšuje, poté pracuje v pásmu svého nominálního výkonu a po určité době se začne výkon kotle snižovat až musí být kotel opět odstaven a vyčištěn. Jak je vidět na obr. 4.1, této odstávce vždy předchází pokles množství produkované páry. Červený kroužek na tomtéž obrázku označuje odstávku z plného provozu, která se provádí z důvodu jiné technické poruchy na zařízení bloku. Protože se v tomto případě jedná o nestandardní situaci, kterou nelze předpovídat, tyto havarijní stavy nejsou v modelu zahrnuty. Před samotným sestavováním modelu bylo provedeno čištění dat a byly odstraněny všechny nestandardní stavy a nulové, tj. kotel nebyl v provozu (technici obvykle vědí, že kotel bude odstaven i další den).



Obrázek 4.1: Provoz kotle 1

Jelikož se předpověď vytváří den dopředu (tedy v pondělí na úterý), nelze použít stav z minulého dne (ještě ho neznáme). Finální model je sestaven v závislosti na průměrném množství páry vyrobené dva, tři a čtyři dny zpět (tedy v případě modelu na úterý je to neděle, sobota a pátek). Názvy proměnných jsou dva, tři a čtyři, kde dva znamená dva dny zpět (tedy neděle), tři znamená tři dny zpět a čtyři čtyři dny zpět.

Tabulky 4.1 a 4.2 znázorňují výsledky analýzy globální citlivosti, která byla popsána v kapitole 3.2.4. Jak je vidět, pro kotle 1 a 2 má proměnná 2 převládající vliv. U kotle 2 se dokonce může zdát, že proměnná čtyři může výsledky ovlivňovat v negativním smyslu (zhoršení výsledků). Ovšem vzhledem k výsledkům globální citlivosti pro kotle 3 a 4 musíme uvažovat všechny proměnné.

Sítě	1. kotel			2. kotel		
	Dva	Tři	Čtyři	Dva	Tři	Čtyři
1. MLP	2,766136	1,250416	1,039759	2,245024	1,043169	0,999777
2. MLP	3,184567	2,041692	1,269800	2,399716	1,010878	0,999389
3. MLP	2,869413	1,092370	1,013182	2,315356	1,027000	1,000061
4. MLP	3,413589	1,291963	1,068944	2,459391	1,041771	1,008264
5. MLP	2,548565	1,764987	1,083742	2,515259	1,042486	1,036464
6. MLP	3,642586	1,263135	1,969836	2,711306	1,167838	1,018914
7. MLP	2,936764	3,185177	1,171908	2,489374	1,069275	1,055181
8. MLP	2,790148	1,171048	1,035594	2,386571	1,032166	0,999995
9. MLP	2,960204	1,666652	1,018990	2,588641	1,016514	1,028412
10. MLP	3,284185	1,529943	1,044777	2,296889	1,025639	1,000002
Průměr	3,039616	1,625738	1,171653	2,440753	1,047674	1,014646

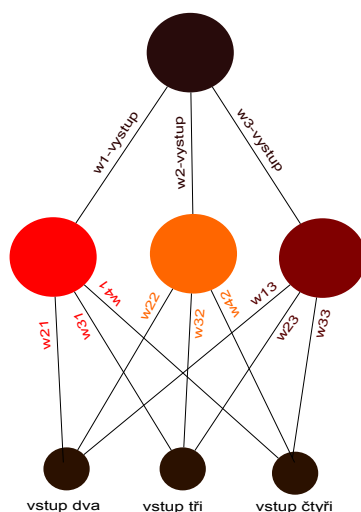
Tabulka 4.1: Globální analýza citlivosti – kotel 1 a 2

Sítě	3. kotel			4. kotel		
	Dva	Tři	Čtyři	Dva	Tři	Čtyři
1. MLP	1,237491	1,173311	1,107320	1,05302	3,97033	1,189956
2. MLP	1,290481	1,166648	1,232581	5,58192	3,94317	4,001754
3. MLP	1,253121	1,194958	1,161017	18,48793	6,27985	3,975271
4. MLP	1,102483	1,180942	1,035013	1,52553	4,41117	3,078147
5. MLP	1,237560	1,245599	1,060556	1,04657	4,65130	1,129529
6. MLP	1,237907	1,282688	1,066236	99,19534	12,53134	6,717080
7. MLP	1,233319	1,263486	1,074357	6,15582	6,50225	4,536913
8. MLP	1,218108	1,318401	1,062200	1,91787	5,92640	3,528066
9. MLP	1,259781	1,287404	1,068219	1,16428	4,51402	1,389111
10. MLP	1,278141	1,192728	1,149831	4,91708	4,28142	2,738899
Průměr	1,234839	1,230617	1,101734	14,10454	5,70113	3,228473

Tabulka 4.2: Globální analýza citlivosti – kotel 3 a 4

4.3.2 Neuronové sítě zvolené pro modely jednotlivých kotlů

V kapitole 3 bylo napsáno, že STATISTICA nabízí 2 modely neuronových sítí – RBF a vícevrstevnou perceptronovou síť. Pro všechny čtyři modely kotlů byla zvolena vícevrstevná perceptronová síť. Architekturu sítí všech modelů můžeme vidět na obrázku 4.2. Jak je zřejmé, u všech čtyř modelů se jedná o síť se třemi vstupy, jednou skrytou vrstvou, která čítá tři neurony, a o jednom výstupním neuronu. Váhy jednotlivých spojení jsou uvedeny v příloze A (tabulka A.1). Přehled aktivačních funkcí pro skryté a výstupní neurony jednotlivých modelů je vidět opět v příloze v tabulce A.2.



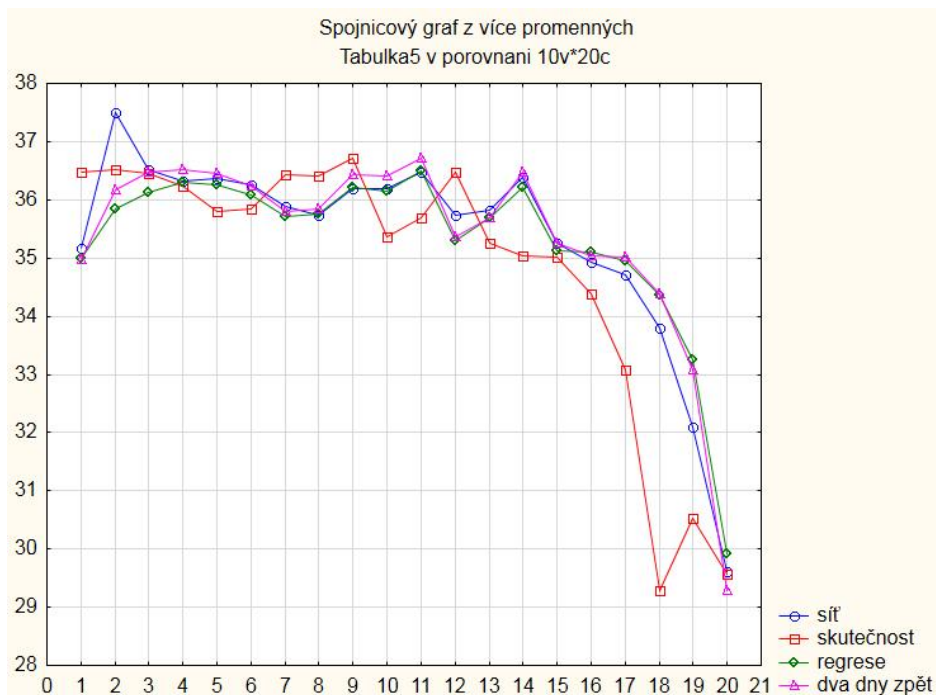
Obrázek 4.2: Model kotlů – architektura sítí

4.3.3 Porovnání neuronové sítě a lineární regrese

Podle tabulky 4.3 je zřejmé, že neuronová síť dává lepší predikci než lineární regresní analýza. Z korelačních koeficientů se dá usuzovat, že v modelu nejsou výraznější nelineární vazby, a proto je lineární regrese skoro stejně dobrá. Obrázek 4.3 ukazuje spojnicový graf z dvaceti po sobě jdoucích dní. Červená křivka představuje skutečná data, modrá namodelovaná neuronovou sítí a zelená pochází z lineární regrese. Z obrázku je zřejmé, že v první části jsou výsledky obou modelů porovnatelné, v závěrečné části je chyba obou modelů velká, ale v případě neuronové sítě je přece jen o něco nižší. Fialová křivka ukazuje, že predikce by byla možná i pomocí množství páry dva dny zpět, které v ustáleném stavu dává skoro stejně dobré výsledky jako neuronová síť a lineární regrese. To lze vidět na obrázku 4.4, kde mají skutečné hodnoty relativně malý rozptyl.

	Lineární regrese		Neuronová síť	
	R	MAE	R	MAE
1. kotel	0,74	0,94	0,8	0,83
2. kotel	0,77	1,14	0,78	1,05
3. kotel	0,66	0,72	0,74	0,65
4. kotel	0,92	0,78	0,9	0,79

Tabulka 4.3: Porovnání lineární regrese a neuronové sítě



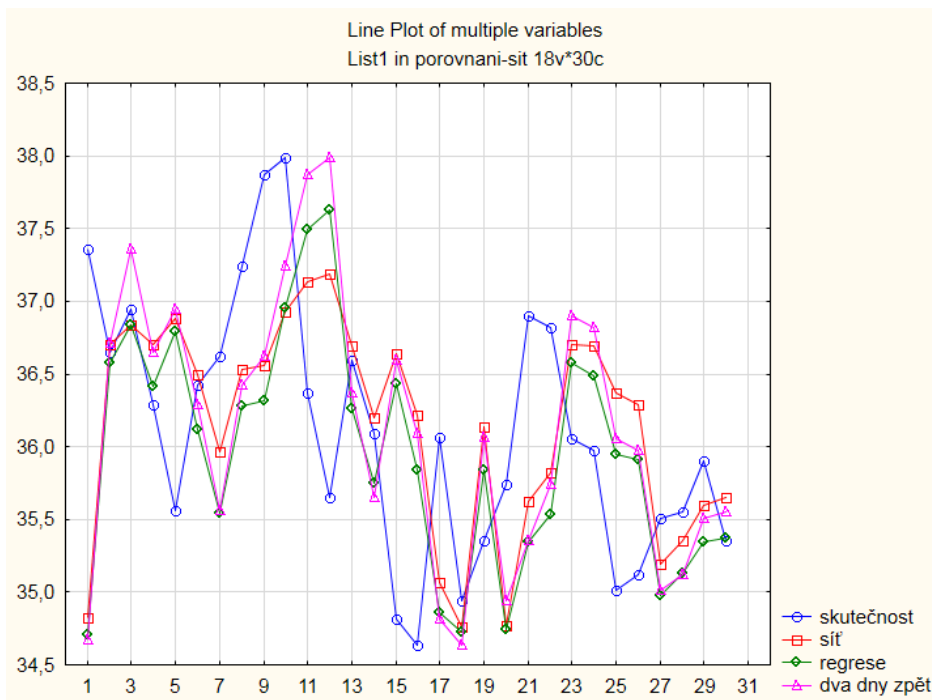
Obrázek 4.3: Průběh skutečných dat, modelovaných pomocí neuronové sítě a pomocí lineární regrese

Jak již bylo zmíněno na začátku kapitoly, pro tento model je velmi důležitá co nejpřesnější predikce z důvodu, že množství páry vyrobené v kotlích vstupuje do dalších modelů. Jedním z nich je například model turbíny. Je zřejmé, že čím méně přesná bude předpověď páry vyrobené v kotlích, tím horší bude i předpověď výroby elektrické energie na turbíně. Počáteční chyba se tak bude kumulovat v dalších modelech, které mají množství páry jako vstupní parametr – nezávislou proměnnou, a to povede k velké chybě konečné predikce elektrické energie předávané do přenosové soustavy. Síť dává o něco lepší výsledky, takže v případě, že není problém s její větší výpočtovou náročností, není důvod pro její nepoužití. Myslím si, že v případě modelu, který přímo neovlivňuje další předpovědi a jeho kvalita není oproti neuronové síti výrazně horší, by bylo použití lineární regrese dokonce lepší z důvodů jednoduchosti a větší rozšířenosti této metody. Dle mého názoru by byla lineární regrese lepší i z hlediska dalšího použití modelu např. v optimalizačním problému.

V modelu kotlů je použita neuronová síť z důvodu, jak již bylo zmíněno výše, větší přesnosti. Protože se jedná o simulační model, nepředstavuje větší výpočtová náročnost neuronové sítě žádnou komplikaci.

4.4 Model odběru páry do Laktos

Firma Laktos, vyrábějící mléčné výrobky, používá páru ze spalovny ve své technologii. Odběr tohoto subjektu je značně nepravidelný, ale poptávku je nutné vždy splnit. Obvykle je odebíráno relativně malé množství celkově vyrobené páry (asi 7 – 14 % celkové produkce). Avšak i v tomto případě je nutné dosáhnout co nejpřesnějšího modelu z důvodu zachování kvality dalších modelů, do kterých tato predikce vstupuje. Výstupem modelu je zřejmě množství páry odebrané firmou Laktos.



Obrázek 4.4: Ustálený stav - srovnání

4.4.1 Analýza vlivu faktorů

Analýzou provozních dat byly zvoleny vstupy sítě. Výsledný model závisí na tom, o jaký den v týdnu se jedná, na hodině v daném dni a na venkovní teplotě. Vliv dnů a hodin je dán pravděpodobně tím, že v některé dny a hodiny vyžaduje technologie zvýšené množství páry (např. začátek pracovní doby v pracovní dny). Jak je vidět na obrázku 4.5, množství odebrané páry (parovody na ose y) roste s klesající venkovní teplotou. Závislost je zjevně nelineární.

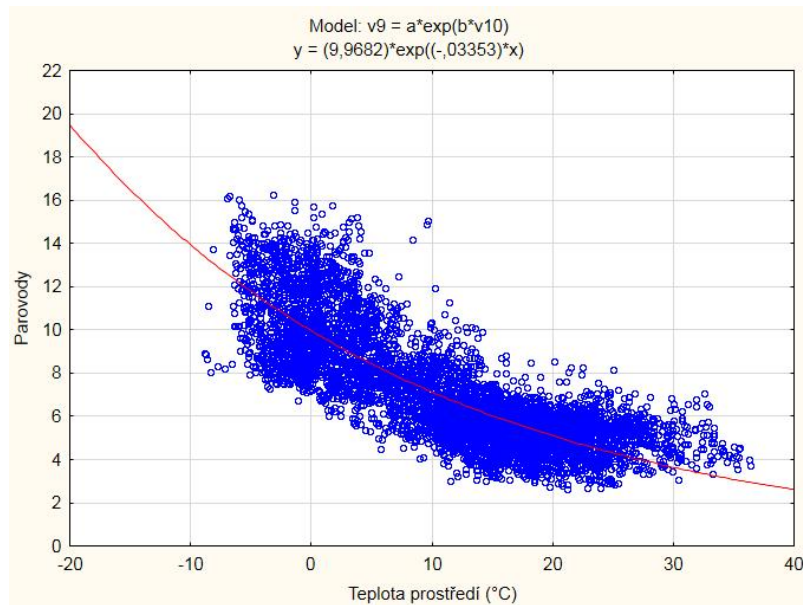
Dny v týdnu a hodiny v průběhu dne jsou kategoriální proměnné, jak už však bylo zmíněno v kapitole 3, jednotlivé kategoriální hodnoty vstupují do sítě jako spojité vstupy, které však nabývají hodnoty 0 (v případě, že se nejedná o daný den/hodinu) nebo 1 (daný den/hodina). Výsledný model má tedy celkem 32 spojitých vstupů (7 dní + 24 hodin + teplota).

Dle analýzy globální citlivosti (tabulka 4.4) lze vidět, že největší vliv má jednoznačně teplota prostředí a výrazně menší, nikoli zanedbatelný, den a hodina.

4.4.2 Výsledný model

Výsledný model má mean absolute error $MAE = 0,84837$ t/h a korelační koeficient $R = 0,9$.

Pro tento model je opět použita vícevrstevná perceptronová síť, která má 32 vstupů, jednu skrytou vrstvu o 4 neuronech a jeden výstup. Váhy jednotlivých spojení jsou uvedeny v příloze B v tabulkách B.2, B.3, B.4, B.5 a B.6, aktivační funkce rovněž v příloze B v tabulce B.1.



Obrázek 4.5: Model Laktos – závislost odběru na venkovní teplotě

Sítě	Teplota prostředí	Den	Hodina
1. MLP	5,57970647	1,97622002	1,47599379
2. MLP	5,09359387	1,96768512	1,46235745
3. MLP	5,81626921	2,03846786	1,51642179
4. MLP	5,30393405	2,00258718	1,50074825
5. MLP	5,95847004	2,06771031	1,58214562
6. MLP	5,6851316	1,93317432	1,45952183
7. MLP	5,51521508	2,0435194	1,51742492
8. MLP	5,10453146	1,95928194	1,46095361
9. MLP	5,90551313	2,04210767	1,55073446
10. MLP	5,48499491	1,92553406	1,41812219
Průměr	5,54473598	1,99562879	1,49444239

Tabulka 4.4: Globální analýza citlivosti – model Laktos

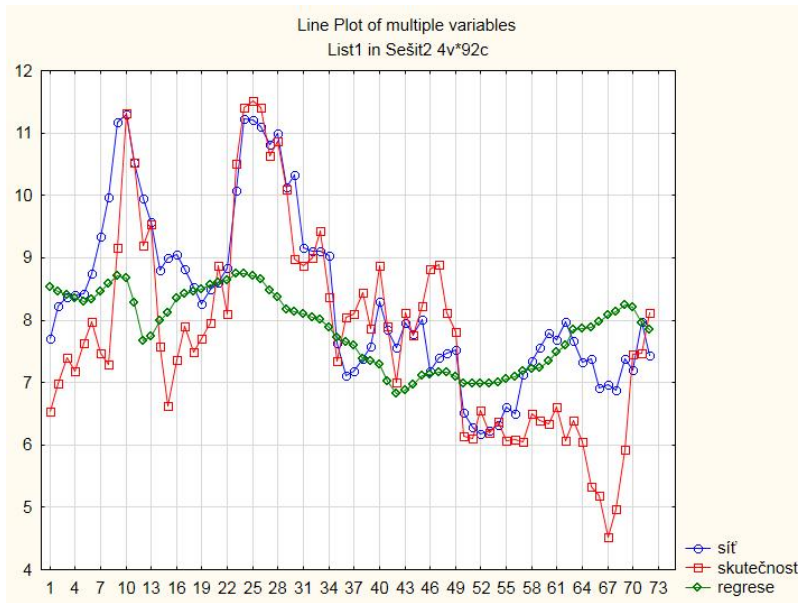
4.4.3 Porovnání neuronové sítě a nelineární regrese

K modelu pomocí neuronových sítí bylo přistoupeno zejména kvůli kategoriálním vstupům, které se obtížně zpracovávají pomocí klasické regrese. V tomto konkrétním případě by to znamenalo vytvořit a následně implementovat 168 regresních funkcí - model pro každou hodinu každého dne.

Vyzkoušen byl také nelineární regresní model v závislosti na teplotě (exponenciální závislost). Výsledná regresní rovnice je ve tvaru: $odber = 9,9682 \cdot e^{-0,03353 \cdot teplota}$. Grafické vyjádření této závislosti lze vidět na obrázku 4.5. Koeficient korelace tohoto modelu je $R = 0,82$ a průměrná absolutní chyba $MAE = 1,2$ t/h. Porovnáním těchto hodnot s výsledky pro síť, které jsou v části 4.4.2 je vidět, že nelineární model bez uvažování kategoriálních proměnných dává horší výsledky.

Na obrázku 4.6 je vidět porovnání skutečných a namodelovaných dat. Červená křivka ukazuje skutečná data, modrá namodelovaná pomocí neuronové sítě a zelená pochází z modelu nelineární regrese (exponenciální funkce). Z obrázku je jasně vidět, že neuronová síť kopíruje daleko lépe skutečnost než nelineární regrese a rovněž že kategoriální vstupy

mají velký vliv. Z obrázku je rovněž vidět, že neuronová síť špatně predikuje stav s nízkým odběrem.



Obrázek 4.6: Model Laktos – porovnání skutečných a modelovaných dat

Kapitola 5

Implementace sítí do prostředí VBA

Jak již bylo uvedeno v kapitole 3.3, STATISTICA počítá pouze se vstupy na intervalu (0, 1). Proto nejen u výše popsaných problémů je nutné nejprve provést škálování na tento interval z širšího intervalu, který popisuje daný problém. Pro další použití neuronových sítí byla vytvořena aplikace ve VBA, která oproti běžným algoritmům neurových sítí, používá škálování. Aplikace pracuje pouze s neuronovými sítěmi s jednou skrytou vrstvou a to z důvodu, že je vytvořena k implementaci neuronových sítí ze STATISTICY a ty mají vždy jen jednu skrytou vrstvu. Velký přínos této aplikace vidím ve schopnosti generování kódu pro zadanou síť. Vygenerovaný kód je pro jazyk VBA, avšak stačí jen úprava syntaxe a kód lze použít v libovolném programovacím jazyku. Podobnou možnost sice umožňuje i STATISTICA, ovšem je nutné zakoupit rozšířenou licenci, ovšem generování kódu ve VBA není možné.

Prostředí VBA bylo zvoleno z důvodu zamýšleného dalšího použití v simulačním modelu, který je vytvořen právě ve VBA.

5.1 Funkce aplikace

Jak je vidět na obrázku 5.1, uživatel zadá počet vstupů, počet skrytých neuronů, nakopíruje ze STATISTICY sloupec vah, minima a maxima. Použitá aktivační funkce je volena ze dvou seznamů, které nabízí standardně používané funkce z programu. Výpočet se provede po stisknutí tlačítka „Výpočet“.

počet vstupů:	3	váhy	vstupy:	34,98025	minimum vstupy:	29,29162	maximum vstupy:	38,60492	minimum výstup:	29,29162	maximum výstup:	38,60492
počet neuronů ve skryté vrstvě:	3	-0,86145		34,18555	31,80858			38,60492				38,60492
		0,79453		35,44914	32,55190			37,99106				
		0,19516										
		-2,63669										
výstupní funkce skrytých neuronů:	Hyperbolický tangens	0,18083										
		2,00924										
výstupní funkce výstupního neuronu:	Hyperbolický tangens	-2,56287										
		-0,98859										
		-0,24422										
výstup z neuronové sítě:	34,9292	0,64811										
		-1,99351										
		0,86623										
		0,42310										
		0,73478										
		0,77999										

Váhy vkládejte v pořadí váhy vstupů k jednotlivým skrytým neuronům (obojivzestupně), poté biasy ke skrytým neuronům (vzestupně). Následují váhy jednotlivých skrytých neuronů k výstupnímu (vzestupně) a jeho bias.

V případě programu Statistica stačí nakopírovat sloupec vah.

Obrázek 5.1: Aplikace pro výpočet neuronových sítí v prostředí VBA

Na začátku procedury pro výpočet je ošetřeno zadání špatného počtu vah, vstupů,

minim či maxim vzhledem k zadanému počtu skrytých neuronů a vstupů. Počet vah musí odpovídat dle vztahu:

$$pocetvah = vstupy \cdot skryte + 2 \cdot skryte + 1, \quad (5.1)$$

kde první člen jsou váhy vstupů ke skrytým neuronům, druhý bias skrytých neuronů a váhy k výstupnímu neuronu a poslední určuje bias výstupního neuronu. Počet vstupů, minim a maxim musí zřejmě odpovídat počtu vstupů. V posledním sloupci musí být vždy dvě hodnoty. V případě zadání špatného počtu některého z údajů se na obrazovce objeví dialogové okno upozorňující uživatele, že u daných hodnot není zadán správný počet.

Pokud je počet všech vstupů v pořádku, proběhne výpočet a jeho výsledek je zobrazen na obrazovce.

Druhé tlačítko „Vygeneruj kód“ slouží k vygenerování textového souboru s kódem konkrétní neuronové sítě, která je právě zadána na vstupu programu. Hlavní přínos vidím v tom, že tato funkce značně usnadňuje přenos sítě do VBA. Je možnost kód vložit jako uživatelskou funkci a přes makro ji spouštět přímo v sešitě nebo funkci použít v dalším VBA kódu. Implementace do dalšího VBA kódu bude použita i v případě modelů vytvořených v této práci. Navíc tento kód je vysoce univerzální, protože stačí jen změny v syntaxi a může fungovat v mnoha dalších programovacích jazycích. Ukázkou vygenerovaného souboru můžeme vidět na obrázku 5.2.

```
'bias pro skryty neuron 1
w( 10) = -0,244220859706786
'vaha vystupniho neuronu od neuronu 1
w(13) = 0,866228623100613
'bias pro skryty neuron 2
w( 11) = 0,6481111172599414
'vaha vystupniho neuronu od neuronu 2
w(14) = 0,423097688906227
'bias pro skryty neuron 3
w( 12) = -1,99350827046858
'vaha vystupniho neuronu od neuronu 3
w(15) = 0,73478437596597
'bias pro vystupni neuron
w(16) = 0,779987807941459

'vypocet delta pro prepocet vstupu.
For a = 1 To 3
DeltaPole(a) = 1/ (max(a) - min(a))
Next a

'Prepocet vstupu
For b = 1 To 3
If (vstup(b) >= max(b)) Then
VstupyPrepocitane(b) = 1
ElseIf (min(b) < vstup(b) < max(b)) Then
VstupyPole(b) = vstup(b) * DeltaPole(b) - min(b) * DeltaPole(b)
Else
VstupyPrepocitane(b) = 0
EndIf
Next b

'vypocet potencialu skrytych neuronu
For i = 1 To 3
For j = 1 To 3
PotencialPole(i) = PotencialPole(i) + vstupyPrepocitane(j) * w(i,j)
Next j
Next i
```

Obrázek 5.2: Vygenerovaný textový soubor

Aplikace a její zdrojový kód se nachází na přiloženém CD.

Kapitola 6

Závěr

Cílem této práce bylo vytvořit určené modely technologie zařízení na energetické využití odpadu ZEVO Malešice pomocí modulu „Neuronové sítě“ v softwaru STATISTICA a vytvořit nástroj pro jejich implementaci do simulačního modelu.

Neuronové sítě jsou pro predikci v oblasti energetiky poměrně často využívaný nástroj. Já jsem v praktické části vytvořila model pro množství vyrobené páry a množství páry odebrané externím subjektem.

Stěžejní částí této práce jsou kapitoly 4 a 5, ve kterých jsou popisovány modely vzniklé v praktické části a jejich implementace do programovacího jazyka VBA. Kapitoly 2 a 3 se věnují teorii neuronových sítí a ovládání modulu „Neuronové sítě“.

Kapitola 4 se zabývá modely vytvořenými v praktické části. Vstupními faktory do modelu pro produkci páry jsou stavy několik dní zpět, konkrétně dva, tři a čtyři. Model neuronových sítí byl porovnán také s jednoduchým lineárním modelem. V tomto modelu doporučuji spíše použití neuronové sítě, protože dává lepší predikci než lineární regrese. Například pro kotel 1 je korelační koeficient neuronové sítě $R = 0,8$ a průměrná absolutní chyba $MAE = 0,83$ t/h, pro lineární regrese jsou to hodnoty $R = 0,74$ a $MAE = 0,94$ t/h. Model je určen pro použití v simulačním nástroji, a proto nevádí jeho větší výpočetní náročnost. Avšak pokud by byl model dále použit v optimalizační úloze, zvolila bych raději lineární regresi. Bohužel, ani jeden z modelů (neuronová síť, regrese) nedokáže postihnout náhlé poklesy či vzrůsty v množství vyrobené páry. Informace o výsledné neuronové síti lze najít v příloze A.

Model odběru páry externí firmou má jako nezávislé vstupní parametry teplotu venkovního prostředí, hodinu a den v týdnu. Obecně odběr roste s klesající teplotou a je vyšší v určité hodiny a dny (např. na začátku pracovní doby pracovních dní). Závislost na teplotě je exponenciální, a proto i regresní model je uvažován nelineární. V případě nelineární regrese by kvůli kategoriálním proměnným bylo nutné vytvořit a následně implementovat 168 modelů. Jednoduchý nelineární regresní model, který závisí jen na teplotě (exponenciální závislost), dává horší predikci než neuronová síť. Pro neuronovou síť je korelační koeficient $R = 0,9$ a průměrná absolutní chyba $MAE = 0,84$ t/h, pro nelineární regrese jsou to hodnoty $R = 0,82$ a $MAE = 1,2$ t/h. Z těchto důvodů je u odběru do Laktos lepší použití neuronové sítě. Informace o výsledné neuronové síti lze najít v příloze B.

Poslední kapitola 5, ale velmi důležitá část mé práce, se zabývá implementací neuronových sítí do prostředí VBA. Prostředí VBA bylo zvoleno z důvodu zamýšleného dalšího použití v simulačním modelu, který je vytvořen právě ve VBA. Aplikace usnadňuje implementaci neuronových sítí, vytvořených ve STATISTICE, do simulačních modelů ve VBA. Hlavní funkcí programu je generování textového souboru s kódem neuronové sítě na vstupu. Tento kód pak lze jednoduše použít buď v jiném zdrojovém kódu ve VBA nebo

jej lze vložit jako uživatelskou funkci a spouštět přes makro přímo v sešitě. Velkou výhodou je také to, že stačí jen jednoduchá úprava syntaxe a kód lze generovat v libovolném programovacím jazyce.

Literatura

- [1] KRÖSE, Ben, Patrick van der SMAGT. *An introduction to Neural Network*. The University of Amsterdam: 1996.
- [2] ŠÍMA, Jiří, Roman NERUDA. *Teoretické otázky neuronových sítí*. 1. vydání. Praha: MatfyzPress, 1996. ISBN 80-85863-18-9.
- [3] BOERRE, C. George. *General psychology (The Neuron)*[online]. ©2003, 2009 [cit. 11. 4. 2014]. Dostupné z: <http://webspaceship.edu/cgboer/theneuron.html>.
- [4] ZÁLESKÁ, Kateřina. *Klasifikace obrazů pomocí umělých neuronových sítí*. Brno, 2011. Bakalářská práce. Masarykova univerzita, Přírodovědecká fakulta, Centrum pro výzkum toxických látek v životním prostředí.
- [5] KALOGIROU, Soteris A. Application of artificial neural-networks for energy system. *Applied energy* [online]. 2000, ročník 67, č. 1-2 [cit. 12. 4. 2014]. ISSN: 0306-2619. Dostupné z: <http://www.sciencedirect.com/science/journal/03062619/67/1-2>.
- [6] KALOGIROU, S., E. Mathioulakis, V. Belessiotis. Artificial neural networks for the performance prediction of large solar systems. *Renewable Energy* [online]. 2014, číslo 63 [cit. 20. 4. 2014]. ISSN: 0960-1481. Dostupné z: <http://www.sciencedirect.com/science/journal/09601481>.
- [7] MOHANRAJ, M., S. Jayaraj, C. Muraleedharan. Applications of artificial neural networks for refrigeration, air-conditioning and heat pump systems—A review. *Renewable and Sustainable Energy Reviews* [online]. 2012, ročník 16, č. 2 [cit. 12. 4. 2014]. ISSN: 1364-0321. Dostupné z: <http://www.sciencedirect.com/science/journal/13640321/16/2>.
- [8] AZADEH, A., R. Babazadeh, S. M. Asadzadeh. Optimum estimation and forecasting of renewable energy consumption by artificial neural networks. *Renewable and Sustainable Energy Reviews* [online]. 2013, ročník 17 [cit. 12. 4. 2014]. ISSN: 1364-0321. Dostupné z: <http://www.sciencedirect.com/science/journal/13640321/17>.
- [9] EKICI, Betül Bektas, U. Teoman Aksoy. Prediction of building energy consumption by using artificial neural networks, *Advances in Engineering Software* [online]. 2009, ročník 40 [cit. 20. 4. 2014]. ISSN: 0965-9978. Dostupné z: <http://www.sciencedirect.com/science/journal/09659978>.
- [10] *StatSoft (Časová řada v modulu Neuronové sítě)* [online]. ©2004-2014 [cit. 11. 4. 2014]. Dostupné z: http://www.statsoft.cz/file1/PDF/newsletter/13_08_07_StatSoft_Casova_rada_v_modulu_neuronove_site.pdf.

- [11] JIŘINA, Marcel. *Jak na neuronové síť v programu STATISTICA NEURONOVÉ SÍŤE*. 1. vydání. Praha: StatSoft, 2003. ISBN 978-80-904033-0-7.
- [12] BISKUP, Roman. *Možnosti neuronových sítí*. Praha, 2009. Disertační práce. Česká zemědělská univerzita v Praze, Provozně ekonomická fakulta, Katedra systémového inženýrství.
- [13] *Pražské služby a. s.* (Energetické využívání odpadu) [online]. ©2014 [citováno 11. 4. 2014]. Dostupné z: <http://www.psas.cz/index.cfm/sluzby-firmam/zarizeni-pro-energeticke-vyuzivani-odpadu/energeticke-vyuzivani-odpadc5af/>.
- [14] *Institut plánování a rozvoje hlavního města Prahy* (ZEVO Malešice - kogenerační jednotka a zařízení DeNOx) [online]. ©2013 [citováno 12. 4. 2014]. Dostupné z: <http://www.iprpraha.cz/cs/mestske-investice/detail/1290/zevo-malesice-kogeneracni-jednotka-a-zarizeni-denox>.
- [15] HRDLIČKOVÁ, Zuzana. *Matematika ONLINE* (Náhodný vektor a jeho charakteristiky)[online]. ©2005 [citováno 11. 4. 2014]. Dostupné z: <http://mathonline.fme.vutbr.cz/Nahodny-vektor/sc-1153-sr-1-a-145/default.aspx>.
- [16] StatSoft. *StatSoft Electronic Statistics Textbook* (How To Identify Patterns in Time Series Data: Time Series Analysis) [online]. [citováno 11. 4. 2014]. Dostupné z: <http://www.statsoft.com/Textbook/Time-Series-Analysis>.

Příloha A

Výsledné neuronové sítě - model kotlů

Spojení	Váha kotel 1	Váha kotel 2	Váha kotel 3	Váha kotel 4
w21	0,63183	-4,15311	-3,23405	12,30209
w31	-0,86144	-5,05523	-0,12430	2,23713
w41	0,79453	-3,60587	-3,11296	-3,50017
w22	0,19516	2,19951	-5,65101	16,07213
w32	-2,63669	3,68312	-8,96063	1,04888
w42	0,18082	1,71270	-5,78028	-5,04612
w23	2,00923	0,92263	-4,73943	-1,47424
w33	2,56287	1,11670	-3,26892	48,89285
w43	-0,98858	0,86967	-3,38542	-10,81274
bias pro skrytý neuron 1	-0,24422	0,05659	1,43566	-5,36936
bias pro skrytý neuron 2	0,64811	-0,67786	4,87663	-5,92450
bias pro skrytý neuron 3	-1,99350	0,05222	1,48890	-13,43138
w1-vystup	0,86622	1,00457	-1,66218	5,61741
w2-vystup	0,42309	2,29668	-2,39675	-4,29564
w3-vystup	0,73478	-0,36102	2,93079	-0,55575
bias výstupní neuron	0,77998	0,63763	-0,38603	0,06331

Tabulka A.1: Váhy spojení – model kotle 1, 2 a 3

	1. kotel	2. kotel	3. kotel	4. kotel
Skryté neurony	tanh	identita	logistická sigmoida	logistická sigmoida
Výstupní neuron	tanh	logistická sigmoida	exponenciální funkce	identita

Tabulka A.2: Aktivační funkce – model kotle 1, 2 a 3

Příloha B

Výsledné neuronové sítě - model Laktos

	Funkce
Skryté neurony	logistická sigmoida
Výstupní neuron	identita

Tabulka B.1: Aktivační funkce – Laktos

Spojení	Váha
Teplota prostředí - neuron 1	-28,85131
Hodina(0) - neuron 1	-0,27013
Hodina(1) - neuron 1	0,01681
Hodina(2) - neuron 1	-0,0285
Hodina(3) - neuron 1	0,0194
Hodina(4) - neuron 1	-0,0731
Hodina(5) - neuron 1	-0,0410
Hodina(6) - neuron 1	0,0156
Hodina(7) - neuron 1	-0,1768
Hodina(8) - neuron 1	-0,3707
Hodina(9) - neuron 1	-0,1512
Hodina(10) - neuron 1	0,4165
Hodina(11) - neuron 1	0,2019
Hodina(12) - neuron 1	0,3444
Hodina(13) - neuron 1	0,4967
Hodina(14) - neuron 1	0,2885
Hodina(15) - neuron 1	0,3143
Hodina(16) - neuron 1	0,5048
Hodina(17) - neuron 1	0,0739
Hodina(18) - neuron 1	0,5120
Hodina(19) - neuron 1	0,3376
Hodina(20) - neuron 1	0,5261
Hodina(21) - neuron 1	0,5831
Hodina(22) - neuron 1	0,0270
Hodina(23) - neuron 1	0,0270
Den(Po) - neuron 1	0,2624
Den(Ut) - neuron 1	0,2740
Den(St) - neuron 1	0,6702
Den(Ct) - neuron 1	0,8498
Den(Pa) - neuron 1	0,3166
Den(So) - neuron 1	0,8328
Den(Ne) - neuron 1	0,3607

Tabulka B.2: Váhy spojení – model Laktos I

Spojení	Váha
Teplota prostředí - neuron 2	-11,1375
Hodina(0) - neuron 2	3,3758
Hodina(1) - neuron 2	0,1613
Hodina(2) - neuron 2	0,7873
Hodina(3) - neuron 2	0,0206
Hodina(4) - neuron 2	-1,2679
Hodina(5) - neuron 2	-0,9634
Hodina(6) - neuron 2	-1,8216
Hodina(7) - neuron 2	-0,5854
Hodina(8) - neuron 2	-2,9824
Hodina(9) - neuron 2	-0,5277
Hodina(10) - neuron 2	0,1122
Hodina(11) - neuron 2	0,5817
Hodina(12) - neuron 2	-3,2245
Hodina(13) - neuron 2	-2,3044
Hodina(14) - neuron 2	0,7252
Hodina(15) - neuron 2	1,3753
Hodina(16) - neuron 2	0,6315
Hodina(17) - neuron 2	0,6837
Hodina(18) - neuron 2	0,6876
Hodina(19) - neuron 2	-0,9114
Hodina(20) - neuron 2	1,0599
Hodina(21) - neuron 2	0,3061
Hodina(22) - neuron 2	2,3946
Hodina(23) - neuron 2	5,2082
Den(Po) - neuron 2	-0,5455
Den(Ut) - neuron 2	1,7244
Den(St) - neuron 2	4,7790
Den(Ct) - neuron 2	6,7798
Den(Pa) - neuron 2	3,7296
Den(So) - neuron 2	-8,1375
Den(Ne) - neuron 2	-4,9150

Tabulka B.3: Váhy spojení – model Laktos II

Spojení	Váha
Teplota prostředí - neuron 3	-14,1940
Hodina(0) - neuron 3	-0,6621
Hodina(1) - neuron 3	1,2759
Hodina(2) - neuron 3	-0,2689
Hodina(3) - neuron 3	-0,3344
Hodina(4) - neuron 3	-1,3128
Hodina(5) - neuron 3	1,2327
Hodina(6) - neuron 3	1,8742
Hodina(7) - neuron 3	1,6247
Hodina(8) - neuron 3	1,6690
Hodina(9) - neuron 3	2,3848
Hodina(10) - neuron 3	2,6037
Hodina(11) - neuron 3	1,9451
Hodina(12) - neuron 3	1,9551
Hodina(13) - neuron 3	1,3483
Hodina(14) - neuron 3	1,2818
Hodina(15) - neuron 3	1,8166
Hodina(16) - neuron 3	-1,7798
Hodina(17) - neuron 3	-1,8470
Hodina(18) - neuron 3	-3,9931
Hodina(19) - neuron 3	-2,3199
Hodina(20) - neuron 3	-0,9741
Hodina(21) - neuron 3	-4,0647
Hodina(22) - neuron 3	-2,8112
Hodina(23) - neuron 3	-2,3220
Den(Po) - neuron 3	0,6003
Den(Ut) - neuron 3	1,0056
Den(St) - neuron 3	0,6226
Den(Ct) - neuron 3	1,0850
Den(Pa) - neuron 3	0,4348
Den(So) - neuron 3	-2,5026
Den(Ne) - neuron 3	-2,8622

Tabulka B.4: Váhy spojení – model Laktos III

Spojení	Váha
Teplota prostředí - neuron 4	-16,6627
Hodina(0) - neuron 4	-4,2884
Hodina(1) - neuron 4	0,8729
Hodina(2) - neuron 4	-3,9610
Hodina(3) - neuron 4	-4,7799
Hodina(4) - neuron 4	-3,7406
Hodina(5) - neuron 4	1,5994
Hodina(6) - neuron 4	2,3276
Hodina(7) - neuron 4	2,4835
Hodina(8) - neuron 4	3,0092
Hodina(9) - neuron 4	3,3466
Hodina(10) - neuron 4	3,2820
Hodina(11) - neuron 4	2,9082
Hodina(12) - neuron 4	2,9161
Hodina(13) - neuron 4	2,6465
Hodina(14) - neuron 4	2,5127
Hodina(15) - neuron 4	2,7390
Hodina(16) - neuron 4	1,4865
Hodina(17) - neuron 4	0,6161
Hodina(18) - neuron 4	-3,8367
Hodina(19) - neuron 4	-0,1409
Hodina(20) - neuron 4	-3,9660
Hodina(21) - neuron 4	-5,5573
Hodina(22) - neuron 4	-1,1996
Hodina(23) - neuron 4	-2,9457
Den(Po) - neuron 4	0,2439
Den(Ut) - neuron 4	0,3857
Den(St) - neuron 4	0,2369
Den(Ct) - neuron 4	0,3850
Den(Pa) - neuron 4	0,0896
Den(So) - neuron 4	-1,2330
Den(Ne) - neuron 4	-1,8679

Tabulka B.5: Váhy spojení – model Laktos IV

Spojení	Váha
bias neuron 1	3,5330
bias neuron 2	3,4773
bias neuron 3	-1,6250
bias neuron 4	-1,8338
neuron 1 - výstup	0,3742
neuron 2 - výstup	0,0811
neuron 3 - výstup 4	-0,4757
neuron 4 - výstup	0,8469
bias výstup	0,1109

Tabulka B.6: Váhy spojení – model Laktos V