VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTEMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

PŘEDPOVĚĎ POČASÍ NA ZÁKLADĚ RŮZNÝCH ZDROJŮ

WEATHER FORECAST BASED ON DIFFERENT SOURCES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — Jakub Kříž
AUTHOR

VEDOUCÍ PRÁCE — Ing. Tomáš Novotný
SUPERVISOR

BRNO 2015

Abstrakt

Práce je zaměřena na metody umělé inteligence pro zpřesnění předpovědi počasí. Teoretická část se zabývá službami, které jsou použity jako zdroj předpovědí, a popisem metod umělé inteligence, které by bylo možné použít včetně teorie neuronových sítí. Podrobně pak metoda časových řad, kterou jsem se rozhodl implementovat. Práce dále popisuje návrh a implementaci serverové části aplikace, která slouží k ukládání dat a části aplikace, která slouží k vyhodnocení a zobrazení nových výsledků pomocí umělé inteligence.

Abstract

This thesis is mainly focused on methods of artificial intelligence for getting more accurate weather forecast. Theoretical part addresses services, which are used as the source for forecasts, and a description of methods of artificial intelligence, which could be used for this kind of work including the theory of neural networks. The time series method, which I decided to use, is descripted in greater detail. The thesis also describes a draft and implementation of the server application, which is designed for saving the data and the application, which uses the method of artificial intelligence to calculate new forecast and displays it.

Klíčová slova

Číselné řady, neuronové sítě, meteorologie, předpověď počasí

Keywords

Time series, neural networks, meteorology, weather forecast

Citace

Kříž Jakub: Předpověď počasí na základě různých zdrojů, bakalářská práce, Brno, FIT VUT v Brně, 2014/2015

Předpověď počasí na základě různých zdrojů

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Tomáše Novotného a uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Jakub Kříž
18. května 2015

Poděkování

Chtěl bych poděkovat svému vedoucímu panu Ing. Tomáši Novotnému za odborné vedení a cenné rady při tvorbě této práce.

©Jakub Kříž, 2014/2015
Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah 1
1 Úvod 3
2 Počasí a jeho předpověď 4
2.1 Formát předpovědí 4
2.1.1 XML 4
2.1.2 HTML 4
2.1.3 Bitmapa 4
2.2 Služby poskytující předpověď počasí 5
2.2.1 Celodenní předpovědi 5
2.2.2 Hodinové předpovědi 6
2.2.3 Aktuální stav počasí 7
2.2.4 Místa předpovědí 7
3 Neuronové sítě 9
3.1 Inspirace biologií 9
3.2 Umělý neuron 9
3.3 Strojové neuronové sítě 10
3.4 Historie 11
3.5 Parametry neuronových sítí 11
3.5.1 Rychlost učení a počet opakování 11
3.5.2 Velikost neuronové sítě 12
3.6 Časové řady a predikce (time series) 12
3.6.1 Metody predikce časových řad 12
3.7 Knihovny C# 15
3.7.1 OpenCvSharp 15
3.7.2 EmguCV 15
3.7.3 NeuronDotNet 15
3.7.4 OxyPlot 16
4 Návrh serverové aplikace 17
4.1 Databáze 17
4.1.1 Tabulky databáze 17
4.1.2 Atributy tabulek 18
5 Návrh výpočetní aplikace 19
5.1 Struktura aplikace 19
5.2 Zvolená knihovna pro neuronové sítě 19

5.3 Využití uložených dat 20
5.3.1 Vstupy a výstupy neuronové sítě 20
5.3.2 Velikost neuronové sítě 20
5.3.3 Struktura neuronové sítě 21
5.3.4 Tvorba trénovací sady 21
5.4 Zobrazení výsledků 22
5.4.1 Zobrazení předpovědi 22
5.4.2 Grafy 22
6 Implementace serverové aplikace 23
6.1 Vytvoření a manipulace s databází 24
6.2 Analýza dat 24
6.2.1 HTML předpovědi 24
6.2.2 Grafické předpovědi 24
6.3 Ukládání do databáze 25
7 Implementace Neuronové sítě 26
7.1 Postup implementace 26
7.1.1 WPF aplikace 27
8 Testování 29
8.1 První fáze – Optimalizace 29
8.1.1 Vyřazení nevhodných služeb 29
8.1.2 Zkouška masivní sítě 30
8.2 Druhá fáze – Síť s více typy hodnot 31
8.2.1 Návrh experimentu 31
8.2.2 Výsledky experimentů 31
8.2.3 Výsledek a závěr experimentu se silou větru 32
8.3 Třetí fáze – Porovnání sítě se službami 32
8.3.1 Návrh testu 33
8.3.2 Brno 34
8.3.3 Kladno 35
8.3.4 Třebíč 36
8.3.5 Kunovice 37
8.3.6 Špindlerův Mlýn 38
8.3.7 Karlovy Vary 39
8.3.8 Závěr Experimentu 39
9 Závěr 40
A. Obsah CD 43
Příklady dat předpovědí 44

1 Úvod

Počasí nás jako lidi vždy ovlivňovalo, proto se ho od pradávna snažíme předvídat a měnit. Přestože přivolat déšť už dokážeme, je pro nás přesná předpověď počasí čím dál důležitější. Ať už se jedná o řízení vzletu letadel, plánování dovolené nebo řízení chytré domácnosti, je počasí jeden z elementů, který je vhodné brát v potaz.

V dnešní době už nepozorujeme nízko létající vlaštovky, ani rosničky skákající po žebříku. Přesto však i v době meteostanic a složitě počítaných předpovědí, nejsme kolikrát schopni získat správnou předpověď na jeden nebo více dní dopředu.

V této práci budu využívat toho, že každá služba, poskytující předpověď počasí používá různé, nebo modifikované technologie k získání údajů a následným výpočtům. Každá taková technologie má své pro a proti. Bylo proto vybráno několik služeb, u kterých se pozoruje předpověď pro 6 měst České republiky. Záměrně byla vybrána města s rozdílnou nadmořskou výškou a polohou. Předpokladem je, že každá služba (technologie) bude dominovat v určité lokalitě a my tak budeme schopni vypočítat co nejpřesnější předpověď podle vah informací jednotlivých služeb v určité nadmořské výšce, čase a poloze.

Nejprve tedy bude popsána teorie k předpovědi počasí a úvod do neuronových sítí. Následně bude popsán návrh 2 aplikací, se kterými budu pracovat a jejich implementace. Závěrem práce pak bude testování neuronové sítě.

Kapitola 1 – Teorie

2 Počasí a jeho předpověď

Počasí je okamžitý stav ovzduší, které je dáno stavem atmosférických jevů v dané oblasti [2]. Udává, zda je horko nebo zima, sucho nebo vlhko, jasno nebo oblačno. Téměř všechny jevy počasí, jako je déšť, mlha nebo tornáda se objevují ve troposféře, což je nejnižší část atmosféry tzn. začínající na zemském povrchu. Toto počasí je ovlivňováno počasím stratosféry (vyšší část atmosféry). Tento mechanismus však ještě není dobře znán.

Počasí je řízeno rozdílnou vlhkostí a teplotou v různých částech Země, které jsou nejvíce ovlivněné pozicí a natočením Země ke Slunci. Dlouhodobý rozdíl pak tvoří podnebí, zatímco krátkodobý počasí.

Předpověď počasí je pak proces aplikující vědecké poznatky a technologie k tomu, aby odhadl stav atmosféry v určitém (budoucím) čase a místě [22].

2.1 Formát předpovědí

2.1.1 XML

XML je rozšiřitelný značkovací jazyk textového formátu, který definuje formát dokumentů, který je lidsky čitelný. Je definován pomocí W3C XML 1.0 specifikací. Nahrazuje složitější SGML [12]. Podobně jako JSON či YAML slouží k serializaci dat. Jedním z hlavních cílů je jednoduché a čitelné šíření dat po internetu.

2.1.2 HTML

Hyper Text Markup Language je standardním jazykem používaným na internetu pro webové stránky. Je psán formou elementů, má počáteční tag <HTML>, a každý z těchto elementů je ukončen. Webový prohlížeč je pak vybaven schopností číst a zobrazovat jednotlivé HTML elementy.

2.1.3 Bitmapa

Jedná se o grafický element (rastrový obrázek), reprezentován řadou pixelů, proto občas referován jako pixmap. Každý pixel z pole, ze kterého se obrázek skládá, pak reprezentuje jeden bod grafického elementu. Pro každý tento bod je definována RGB barva ve formě třech čísel.

2.2 Služby poskytující předpověď počasí

Na internetu je možné nalézt spoustu předpovědí a to v různých formách. Problém je pak najít přesně to co hledáte v určité formě a s vhodnou licencí. Dohromady stahuji počasí z 9 služeb. Z toho 4 jsou předpovědi po dnech ve formátu XML. Ty většinou poskytují větší počet dat s menší přesností právě proto, že předpovídají na celé dny a ne na určitou hodinu. Každá z těchto služeb bude podrobněji popsána v podkapitole Celodenní předpovědi.

Pro další předpovědi, tentokrát hodinové, jsem se rozhodl použít služby poskytující předpověď na webu a to jak textové tak grafické. Bylo to víceméně proto, že bylo potřeba větší množství služeb. Co se týče XML a JSON, dají se z nich data čitelně a lehce získávat. Proto jsem s nimi chtěl původně pracovat. Služeb, které však zdarma poskytují přesnější předpověď v těchto formátech, není mnoho. Pro hodinové předpovědi jsem tedy použil 4 webové služby, ze dvou z nich bude analyzován HTML soubor a u 2 dalších grafické zobrazení počasí na mapě České republiky.

Následuje přehled služeb, které jsem použil jako zdroj předpovědí a dat o současném stavu počasí. Ty jsou rozděleny do 3 kategorií podle jejich použití na celodenní, hodinové předpovědi a na službu, ze které jsou získávány informace o aktuálním stavu počasí.

2.2.1 Celodenní předpovědi

Všechny předpovědi těchto služeb jsou, mimo jiné, formátovány v XML, které jsem použil, a je tak velmi jednoduché dostat z nich informace, které si pro naši potřebu žádáme.

2.2.1.1 YAHOO

Yahoo! Inc. je americká společnost založená v roce 1994, poskytující emailové, vyhledávací a další služby, mezi kterými je webová služba, poskytující předpověď počasí pro mobilní aplikace [23]. Zdarma pak poskytuje pro osobní použití jednoduchý XML soubor s předpovědí na 5 dní dopředu. Tato předpověď má ale jen omezená data (teplota a oblačnost).

2.2.1.2 Wunderground

Weather Underground, hostující stránku wunderground.com, je komerční služba poskytující předpověď počasí pro většinu měst světa. Je zde možnost stahovat počasí zdarma omezená na daný formát souboru XML, který není až tak přehledný, a s omezeným počtem přístupů. Poskytuje předpovědi na 3 dny dopředu s velkým množství dat.

2.2.1.3 Google

Google je další americká společnost, která se proslavila především díky svému vyhledávači. Později také díky prohlížeči a aplikacím, které poskytuje. Jedna z těchto aplikací byla také zaměřená na počasí.

Tato služba poskytuje velmi přehledný XML soubor s mnoha užitečnými údaji a to na týden dopředu, také však s omezeným počtem přístupů za den.

2.2.1.4 Wordweather

Wordweather je webová služba poskytující předpověď počasí ve formátu XML na 3 dny dopředu a to s omezeným počtem dat.

2.2.2 Hodinové předpovědi

Předpokladem hodinových předpovědí je, že předpovídají na několik hodin v rámci jednoho dne, tedy ne na celý den. Služby poskytují předpověď v HTML nebo grafickém (PNG) formátu.

2.2.2.1 Meteocentrum

Meteocentrum je česká služba, která poskytuje stručnou předpověď počasí na 4 dny dopředu v grafickém resp. HTML formátu a to po 3 resp. 1 hodině.

2.2.2.2 In-počasí

Další česká služba. Komerčně poskytuje počasí v různých formátech i grafické podobě. Co se týče poskytování služeb zdarma, poskytuje pouze předpověď na webových stránkách v HTML formátu a to na každé 3 hodiny dne na 5 dní dopředu.

2.2.2.3 Slunečno

Česká služba tvořící předpověď na základě dat, které poskytuje organizace NOAA. Tyto předpovědi pak poskytuje v grafické podobě. Pro každou hodinu na další 2-3 dny je pak uložen PNG obrázek, který na mapě České republiky zobrazuje údaje pomocí barevného spektra. Tyto údaje mají však větší rozptyl, jako třeba teplota, která má v barevném spektru vyhrazenou barvu pouze pro každou možnou sudou hodnotu. Využívá ale jiného výpočtu a bude tedy zajímavé pozorovat tuto službu v porovnání s dalšími.

2.2.2.4 Aladin (ČHMÚ)

Jedná se o Český hydrometeorologický ústav poskytující grafické předpovědi modelu „Aladin“. Jedná se o podobné mapy jako u služby popsané v 2.1.2.3 Slunečno s barevným spektrem. Tentokrát poskytuje předpovědi po 6 hodinách a to až na 3 dny dopředu. Nová předpověď je pak tvořena 4x denně. Každá předpověď se nahrává postupně, takže je možné, že v určitý čas je nahrána pouze část poslední předpovědi.

2.2.2.5 Medard

Služba poskytující předpovědi počasí pro Českou republiku a Evropu. Podobně jako Aladin jí poskytuje v grafické podobě a využívá modelu WRF (NCAR).

2.2.3 Aktuální stav počasí

2.2.3.1 In-počasí meteostanice

Jedná se o další službu serveru In-počasí, která poskytuje aktuální stav počasí pozorované na jedné z desítek meteostanic po České republice. Aktualizace probíhá asynchronně, většinou několikrát do hodiny.

2.2.4 Místa předpovědí

Místa, která budou použita pro tvorbu předpovědní neuronové sítě, byla vybrána tak, aby pokryla velkou část republiky, tzn. tak aby místa nebyla blízko u sebe, a zároveň aby místa, pro které bude předpověď tvořena, měla různou nadmořskou výšku. Byla vybrána města se střední rozlohou, díky které se místo meteostanice, ze které jsou sbírána data v určité oblasti, nemůže tolik lišit od oblasti, pro kterou je předpověď určena.

2.2.4.1 Brno

Nejrozsáhlejší město, u kterého bude předpověď tvořena, a proto se také dají očekávat větší odchylky způsobené divergencí místa meteostanice a zón předpovědí. Město se nachází v Jihomoravském kraji s nadmořskou výškou 190 až 479 metrů nad mořem a s meteostanicí na Kraví hoře, ve výšce 280 metrů nad mořem.

2.2.4.2 Třebíč

Menší město z kraje Vysočina. Průměrná nadmořská výška 405 metrů nad mořem s meteostanicí ve výšce 486 metrů nad mořem. U tohoto města je předpokládáno velmi ideální měření vzhledem k jeho rozloze a nadmořské výšce.

2.2.4.3 Špindlerův Mlýn

Velmi malé městečko z Královéhradeckého kraje s velkou nadmořskou výškou v průměru 718 a meteostanicí ve výšce 800 metrů nad mořem. Zde bude zajímavé pozorovat přesnost jednotlivých služeb v „extrémních“ výškách.

2.2.4.4 Kunovice

Malé městečko Zlínského kraje s velmi malou nadmořskou výškou v průměru 198 metrů a meteostanicí ve výšce 185 metrů nad mořem. Zde je opačný extrém od Špindlerova Mlýna, co se týče nadmořské výšky.

2.2.4.5 Kladno

Město ve Středočeském kraji s průměrnou nadmořskou výškou 381 metrů a meteostanicí ve stejné nadmořské výšce. Co se týče nadmořské výšky, je tedy ideálním kandidátem pro předpověď. To se ale může změnit, pokud by jeho rozloha měla u předpovědí hrát velkou roli.

2.2.4.6 Karlovy Vary

Menší město statisticky podobné městu Třebíč možná ještě s ideálnějšími podmínkami. Toto město se však nachází na druhé straně republiky a to v západních Čechách. Průměrná nadmořská výška 447 metrů nad mořem s meteostanicí ve stejné výšce.

3 Neuronové sítě

Neuron je základní nervová buňka živých organismů. Ta pomocí elektrických signálů dokáže předávat informace dalším neuronům. Neurony, které jsou mezi sebou provázány, tvoří jako celek tzv. neuronovou síť. Neurony jsou závislé na vnějších vstupech, jako může být i jiný neuron. Neuron může být jinak citlivý na jednotlivé vstupy [24].

Neuronové sítě v počítačovém světě jsou algoritmy inspirované biologickými nervovými spojeními, jako například u lidí či zvířat. Samozřejmě s mnohem jednodušší architekturou. Základní myšlenkou je tedy modelovat biologický neuronový systém fungující v přírodě jako model pro strojové učení [9].

3.1 Inspirace biologií

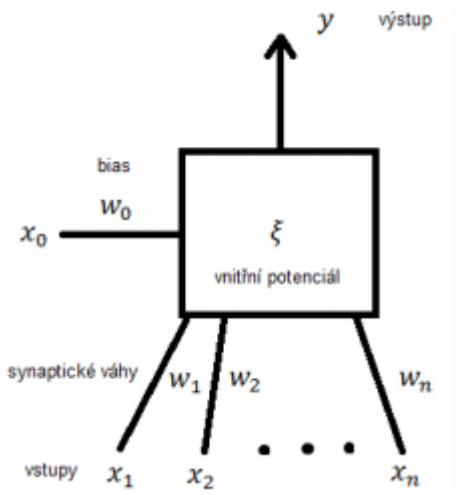
Z biologického hlediska je nervová soustava tvořena nervovými buňkami, neurony, které vedou vzruchy pomocí rozvětveného propojení s dalšími neurony. Každý neuron tak spolupracuje se svými sousedními neurony, které mezi sebou šíří signály.

Neuron má svoji vnitřní aktivitu, která reaguje na dráždění receptory, nebo jinými neurony a pokud přesáhne určitou mez (danou především předchozími zkušenostmi) vysílá signál, který se pak dále šíří po nervové soustavě, což je možné registrovat jako elektrický děj. Informace není pouze 0,1 (vysílá nebo nevysílá), závisí pak i na frekvenci vysílání a dalších věcech, jako je druh mediátoru (neurotransmitéru). V celé soustavě pak dochází k tzv. sbíhavosti resp. rozbíhavosti, kdy jeden neuron přijímá, resp. vysílá signály z resp. k více neuronům [24].

3.2 Umělý neuron

Umělý neuron je zjednodušená verze neuronu popsaného v kapitole 3.1. Jeden z nejznámějších umělých neuronů je RBF neuron (perceptron) [10], který je vyobrazený na obrázku 3.1. Ten se skládá ze:

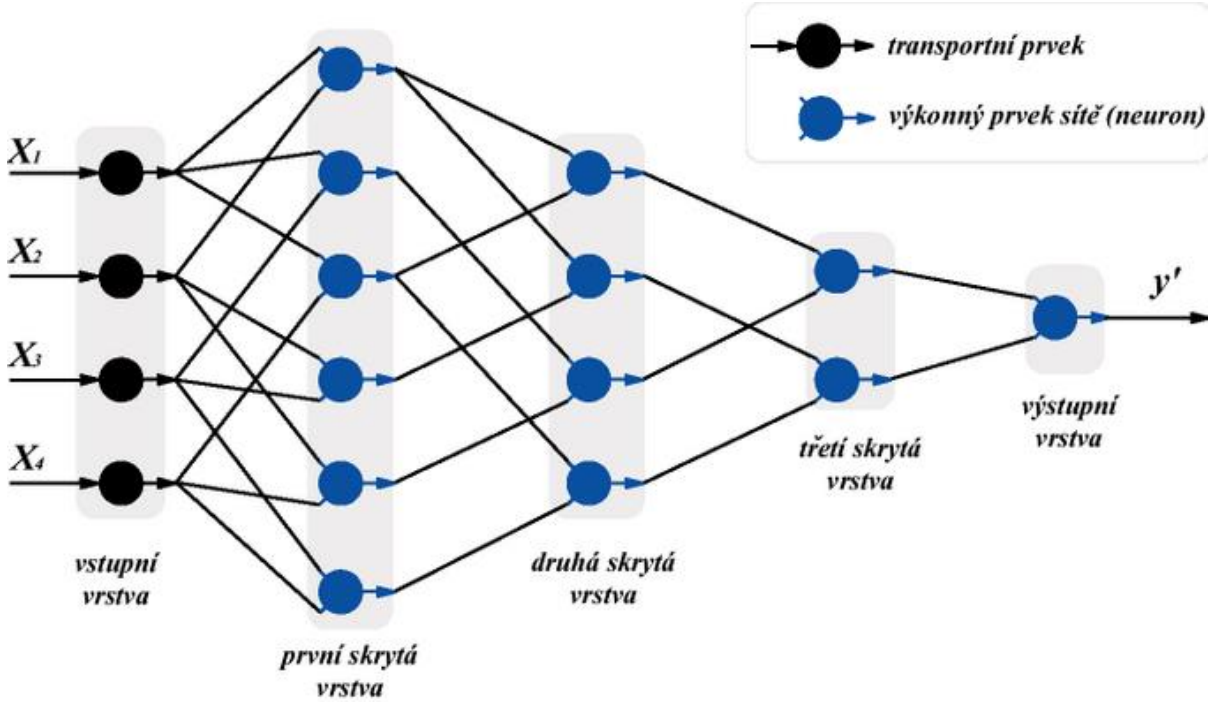
- **Vstupů**, z kterých má každý svou váhu, která se může při procesu učení měnit.
- Tzv. **biasu**, který rozhoduje o tom, jestli má být informace dále zpracovávána nebo se jedná o bezvýznamnou informaci, která se zahazuje.
- **Těla neuronů,** která tvoří jeden z typů tzv. aktivační, nebo také přenosová funkce.



Obrázek 3.1: Umělý neuron [10].

3.3 Strojové neuronové sítě

Tyto algoritmy získávají informace a učí se ze statistik (zkušeností) a na základě obvykle vysokého množství vložených dat, dokáží odhadovat a přiblížit se k požadovaným výsledkům. Neuronové sítě jsou reprezentovány propojením tzv. neuronů, které dokáží vyhodnocovat vstupy a jsou schopny strojového učení.



Obrázek 3.2: Neuronová síť [26].

Mezi výhody neuronových sítí patří jednoduchá implementace, díky které lze dosáhnout relativně dobrých výsledků. V dnešní době patří mezi nejpřesnější víceúčelové aproximátory

diferenciálních funkcí. Nevýhodou je pak praktická nemožnost přetrénování již existující sítě s novými daty a potřeba relativně velkého množství trénovacích dat.

3.4 Historie

Neuronové sítě jsou založené na tzv. modelu „threshold logic“, česky - prahu logiky, zformulované v roce 1943 neurofyziologem W. McCullochem a matematikem W. Pittsem, který modeluje neuronové sítě na základě matematických algoritmů. Tento model se pak dále zkoumá zvlášť z biologického a strojového hlediska [25].

V roce 1949 byla kanadským psychologem D. Hebbem sepsána *The Organization of Behavior*, hypotéza učení na základně nervové plasticity – zesílení často používaných nervových spojů (cest) a spojování nervů, které se aktivují ve stejný čas. Tato hypotéza se začala aplikovat 1948 s Turingovy B-typové stroje.

Na začátku 50. let, díky výkonnějším strojům, proběhly první pokusy o simulaci neuronové sítě. V roce 1959 byl pak vyvinut model ADALINE na rozpoznávání proud bitů z telefonu.

Roku 1962 nová učící technologie přidává váhu informacím šířících se po neuronové síti. I přes své úspěchy neuronových sítí, tradiční von Neumannova architektura přebírá počítačovou scénu a vývoj neuronových sítí tak na několik let upadá.

1982 nový přístup k neuronovým spojením, jako k obousměrným spojením, předtím byly používány pouze jednosměrná spojení neuronů. Ve stejném roce byla také poprvé použita vícevrstvá síť tzv. „Hybrid network“.

V dnešní době se neuronové sítě používají v množství aplikací v množství různých přístupů. Budoucnost neuronových sítí pak záleží hlavně na směru, kterým se vydá vývoj hardwaru [25].

3.5 Parametry neuronových sítí

Každá neuronová síť, její efektivita učení a přesnost výstupů je ovlivněna několika parametry, které budou dále teoreticky rozebrány.

3.5.1 Rychlost učení a počet opakování

Parametr rychlosti učení je ve většině případů upravován podle množství trénovacích dat. Ovlivňuje rychlost ale hlavně kvalitu. Vyšší hodnota umožní rychleji se učit avšak na úkor kvality. Parametr rychlosti učení je nastavován v intervalu (0,1), kdy efektivní učení, až na výjimky, probíhá okolo hodnot (0,1;0,4). Počet opakování je pak proces, kdy se stejnými trénovacími daty neuronová síť „krmí“ několikrát, pokud je to pro dané řešení vhodné je možné „krmit“ neuronovou síť i s drobnými odchylkami ve vstupních datech a tím tak rozšířit trénovací sadu. Větší počet opakování pak znamená delší, ale kvalitnější trénování dané neuronové sítě.

3.5.2 Velikost neuronové sítě

Velikost neuronové sítě odráží počet neuronů, tedy jak počet vstupů, tak počet neviditelných vrstev a počet neuronů v těchto vrstvách. Počet neviditelných vrstev a počet neuronů v těchto vrstvách by měl být úměrný počtu vstupů a výstupů neuronové sítě, např. pro neuronovou sít se třemi vstupními a jedním výstupním neuronem by bylo zbytečné mít čtyři skryté vrstvy s 60 neurony v každé z nich. V zásadě pak platí, že čím větší neuronová síť, tím je potřeba větší počet trénovacích dat, aby síť pracovala efektivně.

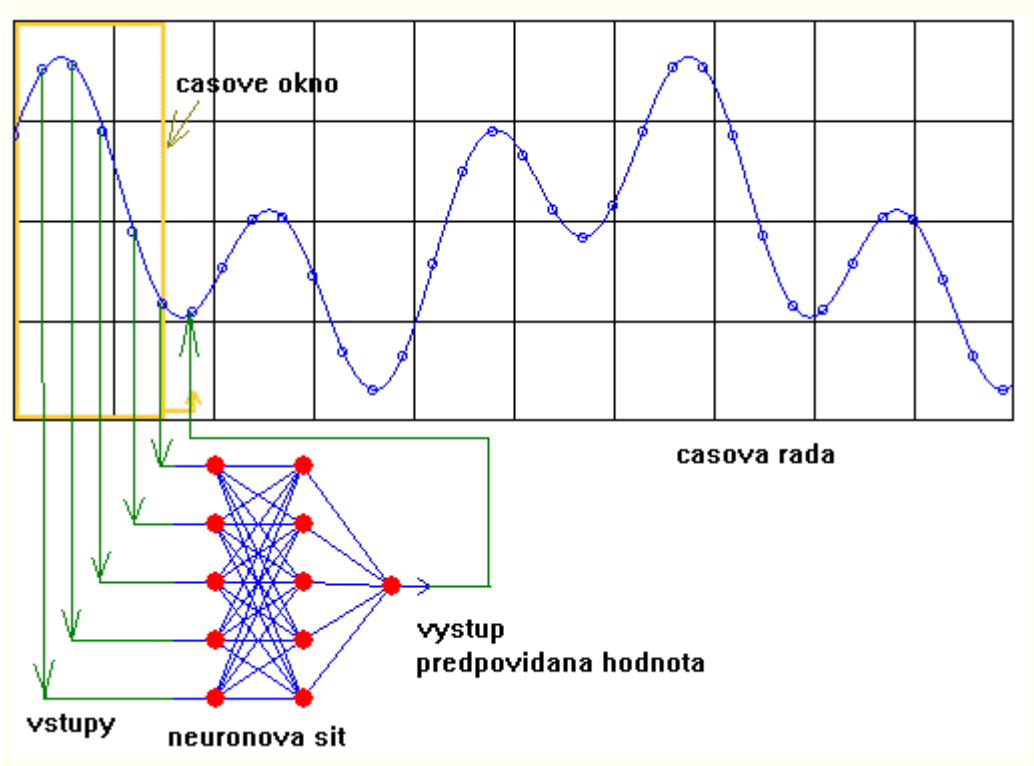
3.6 Časové řady a predikce (time series)

Časové řady jsou sekvence vektorů nebo skalárních součinů závislé na čase (t). Prvky těchto vektorů pak mohou být jakékoliv měřitelné jevy jako teplota, rychlost větru atd. Je to metoda zaměřená na předpověď podle předchozích zkušeností to znamená, že pokud trénovací množina bude t_1,t_2 a t_3, výpočet hodnoty v čase t_4 bude záviset právě na přesnosti předchozích tří událostí. Čím víc se pak čas trénovací množiny liší od času aktuálně pozorovaného stavu, tím je předpověď nepřesnější.

Existují tzv. deterministické číselné řady a to jsou řady, u kterých lze předpovídat průběh. Ty ale nejsou příliš časté a většina řad, k jejichž řešení se využívají neuronové sítě je vystavena náhodným jevům, které mění jejich průběh.

3.6.1 Metody predikce časových řad

Zpracováváme naměřené údaje pomocí časově závislé proměnné x_t (t=1,2,...) a chceme předpovědět hodnotu proměnné v budoucím čase. Pro predikci je možné použít několik typů sítí, např. síť typu backpropagation, ART, Marksovu síť a další. V dalším textu budou popsány.

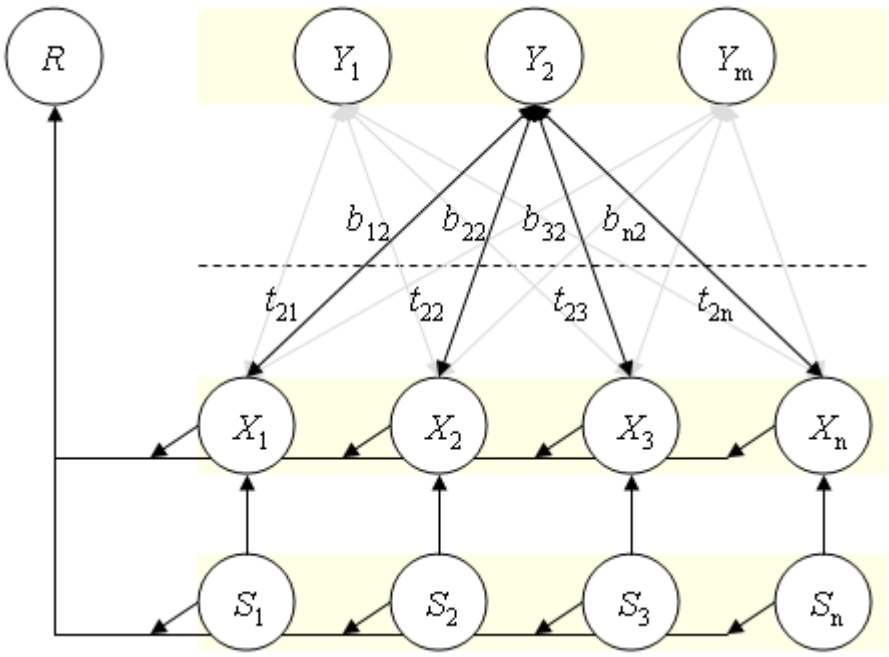


Obrázek 3.3 : Proces učení pomocí časového okna [4].

Metoda časového okna slouží k vytvoření trénovací množiny. *N* vstupů se přivede na neuronovou síť s vstupní vrstvou o *N* neuronech a ta vyhodnotí další výsledek. Poté se časové okno posune o jednu hodnotu a proces se opakuje.

3.6.1.1 Adaptive resonance theory (ART)

Adaptive resonance theory je teorie S. Grossberga a G. Canpentera o tom, jak mozek zpracovává informace. Vykresluje několik neuronových sítí, které používají více typů učících metod. Používají se zejména pro hledání cest nebo predikci. Existuje několik typů ART rozdělené podle složitosti [18].



Obrázek 3.4 : ART síť [18].

Učící model typicky obsahuje porovnávací pole a rozeznávací pole. Porovnávací pole bere vstupní vektor a transformuje ho k nejbližší shodě v rozeznávacím poli. Existuje zde reset modul, který kontroluje, zda byla shoda dostatečně „blízko“ a pokud ne (určuje se porovnáním s parametrem vigilance, který musí být dostatečně vysoký) je tento neuron potlačen. Vysoký parametr vigilance vyústí ve velmi detailní paměť, zatímco nízký spíše obecnou [18].

Nevýhodou je pomalejší trénovací proces, který potřebuje velké množství trénovacích dat oproti ostatním metodám. Některé typy ART sítí jsou také závislé na pořadí vstupních dat, což není žádoucí jev.

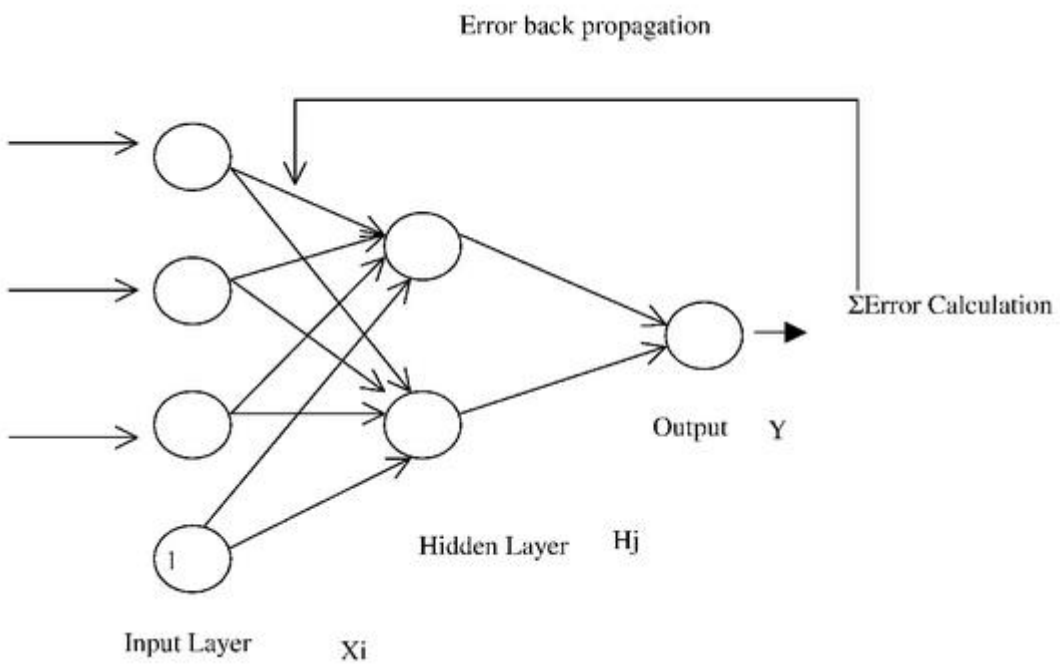
3.6.1.2 Back-propagation

Back-propagation je běžná metoda trénování neuronových sítí. Běžnou metodou pro měření rozdílu mezi odhadem O a reálným výsledem V je počítání čtvercové chyby (rozdílu) E.

$$E = (O - V) \tag{3.1}$$

Učení pak probíhá v 5ti krocích [20]:

1. Inicializace všech vah na náhodnou malou hodnotu
2. Předložení nového vzoru X = x1x2 . . . xn a odpovídajícího výstupu D = d1d2 . . . dm. Trénovací vzory vybíráme náhodně.
3. Výpočet aktuálních výstupů sítě.
4. Adaptace vah.
5. Opakování procesu učení. Pokud je celková chyba větší než ε, pokračujeme krokem 2. Jinak skončíme.



Obrázek 3.5: Back-propagation model [19].

3.7 Knihovny C#

Zde bude popsáno několik knihoven pro C# implementující práci s neuronovými sítěmi. Všechny tyto knihovny jsou kompatibilní s Mono. Dále zde bude přiblížena knihovna pro tvorbu grafů v aplikaci WPF (Windows Presentation Foundation), tedy grafickou aplikaci pro .NET užívající jazyk XAML.

3.7.1 OpenCvSharp

OpenCvSharp je modelován na nativním OpenCV jazyku C/C++. Mnoho tříd implementuje IDisposal. Je to multiplatformní knihovna – může běžet na OS Windows, Linux i Mac. Knihovna umožňuje volat nativní funkce OpenCV [14].

3.7.2 EmguCV

EmguCV je wrapper knihovny OpenCV, díky kterému je možné volat funkce OpenCV z .NET jazyků, mezi které patří i C#, s kterým bude pracováno. Je to poměrně čerstvá knihovna s počátkem vývoje v roce 2008.

3.7.3 NeuronDotNet

NeuronDotNet je knihovna pro neuronové sítě napsaná v C#. Poskytuje prostředí pro pokročilé programátory umělé inteligence ke tvorbě mnoha typů strojových neurálních sítí a jejich užití [15].

3.7.4 OxyPlot

Multiplatformní knihovna pro .NET. Obsahuje velké množství grafů s možností dynamického přidávání dat do grafu. OxyPlot byla vytvořena v roce 2010 jako jednoduchá WPF komponenta primárně zaměřená na dvoudimenzionální systém [21].

Kapitola 2 - Návrh

4 Návrh serverové aplikace

Serverová aplikace bude sloužit k ukládání předpovědí do databáze, jejíž návrh je popsán níže. Pro každou internetovou službu poskytující předpověď bude mít aplikace metodu, která bude údaje z těchto předpovědí zpracovávat. Dále bude vytvořena třída, do které se tyto data budou vkládat a následně odesílat do databáze.

Jak bylo naznačeno v kapitole 2.1, budou předpovědi rozděleny do několika skupin a podle toho se také budou ukládat do různých tabulek databáze. Zároveň se budou ukládat v různých intervalech, které jsem zvolil jednohodinové pro hodinové předpovědi a aktuální počasí a šestihodinové pro celodenní předpovědi.

Jako první bude potřeba vyřešit ukládání aktuálního stavu počasí, aby bylo s čím porovnávat. Jak už bylo zmíněno v kapitole 2.1.3, byla k tomu použita služba In-počasí. Tuto službu jsem vybral, protože je velmi přehledně udělaná a obsahuje velké množství údajů a to pro každé z pozorovaných míst. Jediným problémem bylo to, že poskytované počasí bylo ve formátu HTML. Všechny údaje jsou však na jednom řádku tohoto souboru, takže s tím nebyl větší problém, pokud se ovšem struktura souboru nezmění. Následně bude implementováno ukládání předpovědí ze serverů a přidání všech služeb do další tabulky v databázi.

Vzhledem k potřebě získávání dat z grafických předpovědí, budou navíc implementovány funkce pro:

- Stažení obrázku
- Zjištění barvy pixelu obrázku
- Automatickou tvorbu kódu pro přepínač hodnot barevného spektra
- Určení hodnoty z barvy podle serveru

4.1 Databáze

Do databáze se budou ukládat jednotlivé předpovědi z různých serverů. Ukládání dat bude mít na starosti serverová aplikace a jejich další zpracování aplikace implementující neuronovou síť.

4.1.1 Tabulky databáze

Pro tuto práci byla vybrána databáze MySQL. Ta se bude skládat z následujících tabulek:

- Tabulka celodenních předpovědí
- Tabulka hodinových předpovědí

- Tabulka služeb

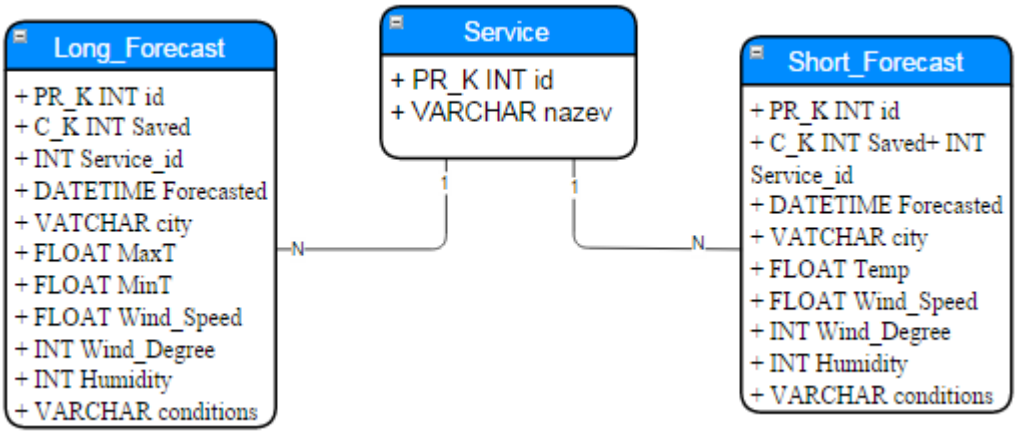
Tabulka služeb bude obsahovat názvy a možná i url jednotlivých služeb. Do tabulky Hodinových předpovědí se pak bude ukládat i aktuální stav počasí. Poznávat se bude právě podle indexu služby, který je uložen jako cizí klíč.

4.1.2 Atributy tabulek

Atributy tabulek počasí budou následující:

- INT id – primární klíč, automaticky inkrementován.
- INT Saved – představuje, kolik hodin dopředu byla tato předpověď uložena.
- INT Service_id – cizí klíč pro tabulku service
- DATETIME Forecasted – datum a čas na kdy je předpovězeno
- VARCHAR city – místo předpovědi
- FLOAT MaxT – maximální teplota za den (pouze u celodenních předpovědí)
- FLOAT MinT – minimální teplota za den (celodenní)
- FLOAT Temp – teplota v určité hodině (u hodinových a aktuálního počasí)
- FLOAT Wind_Speed – rychlost větru
- INT Wind_Degree – směr větru
- INT Humidity – vlhkost v %
- VARCHAR conditions – podmínky počasí (oblačno, Slunečno…)

Celá databáze pak bude vypadat takto:



Obrázek 4.1: Struktura databáze.

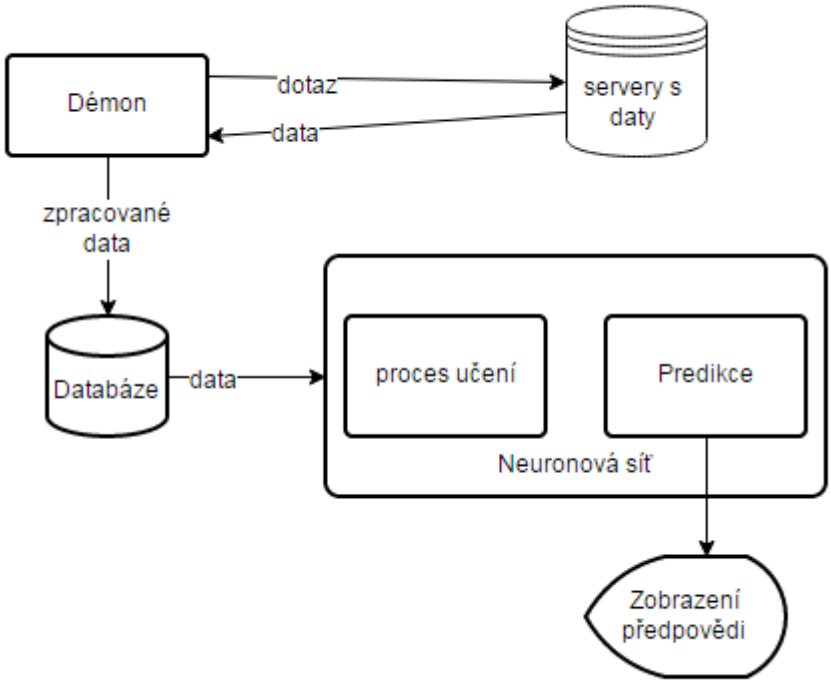
5 Návrh výpočetní aplikace

5.1 Struktura aplikace

Tato aplikace bude implementovat strojové učení pomocí neuronové sítě.

Bude se skládat ze tří částí:

- Zpracování dat z databáze a trénování neuronové sítě
- Vyhodnocení nových dat a vytvoření vlastní předpovědi
- Vyobrazení nových předpovědí a grafů



Obrázek 5.1 : Vyobrazení struktury aplikace.

5.2 Zvolená knihovna pro neuronové sítě

Pro implementaci umělé inteligence jsem si vybral NeuronDotNet, která je popsána výše. Vybral jsem si ji, protože je velmi obsáhlá a běží na různých platformách včetně linuxu přes Mono. Původně jsem uvažoval i o OpenCVsharp, který je také možné použít v Monu na linuxu a dalších platformách. Vzhledem k tomu, že nemám zkušenosti ani s jednou knihovnou (včetně OpenCV), pro umělou inteligenci vybral jsem si NeuronDotNet, protože je více používaná a tak předpokládám i více materiálů a intuitivnější funkcionalitu.

5.3 Využití uložených dat

Další částí návrhu neuronové sítě je určit možné vstupy a výstupy této neuronové sítě. Byla tedy vytvořena tabulka 5.2, která vyobrazuje množinu dat, která byla získána „těžbou dat“.

Služby	Teplota	Rychlost větru	Směr větru	Srážky [mm]	Vlhkost [%]	Tlak [HPa]
Aktuální počasí	*	*			*	*
Yahoo	*					
World Weather	*	*	*	*		
Google	*	*	*		*	
Wunderground	*	*			*	
Meteocentrum	*	*		*		
In-počasí	*	*		*		*
Aladin	*			*		
Slunečno	*			*		

Obrázek 5.3 : Služby a jejich data.

5.3.1 Vstupy a výstupy neuronové sítě

Z tabulky 6.1 je vidět, že pro výstup neuronové sítě lze použít čtyři hodnoty a to teplotu, rychlost větru, vlhkost a tlak a to proto, že tyto hodnoty jsou obsaženy v aktuálním počasí, tzn. v tabulce s reálnými hodnotami v daném čase. Tlak má pak pouze jednu předpovědní službu a proto pro nás není zajímavý. Ostatní hodnoty by pak neměli být s čím srovnány, protože nejsou obsaženy v aktuálním počasí, a proto je možné je použít pouze jako vstup do neuronové sítě kdy ovlivňují výstup jiných hodnot, např. srážky ovlivňují jak teplotu, tak rychlost větru, tak i vlhkost. Směr větru nebude v aplikaci používán vzhledem k nemožnosti kontroly a k jeho nízkému vlivu na výsledek dalších hodnot. Nejzajímavější hodnotou je pak teplota, která je asi nejžádanější hodnotou u předpovědi počasí, a kterou obsahují všechny předpovědní služby. Tím pádem bude nejlehčí porovnávat efektivitu jednotlivých služeb podle ní, a proto na ní bude zaměřena většina testů.

5.3.2 Velikost neuronové sítě

Aby byla předpověď co nejpřesnější a zároveň na co nejdelší dobu, bude neuronová síť využívat co největší počet služeb. Jak již bylo zmíněno v kapitole 4, jsou předpovědi rozdělené na dlouhodobé/celodenní a krátkodobé/hodinové. Všechny krátkodobé služby pak předpovídají alespoň na 36 hodin dopředu a všechny dlouhodobé na 3 dny dopředu, otázkou je zdali bude lepší vytvořit 2 neuronové sítě, tedy jak pro dlouhodobé předpovědi, tak pro krátkodobé, nebo zvolit variantu jedné masivnější neuronové sítě, do které by byly zakomponovány oba druhy předpovědí. K této otázce se budu vracet na začátku testování.

Předpověď tedy bude probíhat jak pro konkrétní hodiny, tak i pro celé dny. Krátkodobé, nebo také hodinové předpovědi, budou předpovídat na následujících 36 hodin a to po 3 hodinách po následujících 12 hodin a poté po 6 hodinách až do 36 hodin. Předpovědi pak budou odrážet počasí v celých hodinách dělitelných 3, tzn. 12:00, 15:00 atd. 36 hodinový interval byl zvolen kvůli předpovědním službám, které na delší dobu nepředpovídají, a proto by další výsledky nebyly až tak zajímavé, právě kvůli použití malého počtu předpovědních služeb.

5.3.3 Struktura neuronové sítě

Neuronová síť se bude skládat ze vstupů tvořených předpovědními službami a záznamy o počasí za poslední 3 hodiny. Každá předpověď pak předpovídá na dalších 8 bodů v čase. Do vstupů budou vkládány jen prvky počasí, které budou relevantní k danému testování, případně zvolené uživatelem, který by aplikaci používal. Podobně pak budou voleny i výstupy. Pokud tedy budeme brát počet prvků počasí pro vstupy jako *x* a počet prvků pro výstup jako *y* bude se dát počet neuronů vypočítat jako:

$$\boldsymbol{počet_neuron}\text{ů} = (\mathbf{3}+\mathbf{8})\boldsymbol{x} + \mathbf{8}\boldsymbol{y} + \boldsymbol{počet_neuronu} * \boldsymbol{počet_skrytych_vrstev} \quad (5.1)$$

Tato struktura však bude použita jen za předpokladu, že vložení dalších typů hodnot zlepší předpověď hodnot již předpovídaných. Např. pokud předpověď teploty bude zlepšena po přidání hodnot předpovědí vlhkosti do neuronové sítě. Druhá možnost je vytvořit pro každou hodnotu vlastní neuronovou síť.

5.3.4 Tvorba trénovací sady

Jak bylo řečeno v předchozí kapitole, budou krátkodobé předpovědi předpovídat po 3 hodinách. Aladin, který předpovídá po 6 hodinách, bude tedy dále zpracován a to tak, že pro hodiny ve kterých předpověď není k dispozici, bude vložena nová předpověď jako průměr dvou sousedních předpovědí.

Předpovědi všech služeb se stáhnou do struktur programu, se kterými se bude dále pracovat. Druhou možností bylo využít SQL dotazů. Těch by bylo ale za jedno spuštění programu potřeba tolik, že by se úspora paměti na úkor rychlosti a zatížení sítě nevyplatila.

Pro vytvoření jednotky trénovací sady je potřeba simulovat použití neuronové sítě pro předpověď. Navíc je pak potřeba poskytnout neuronové síti výsledky jako ideální výstupy sítě, aby se měla z čeho učit. Kvůli službám, které nepředpovídají na každou hodinu, by bylo zbytečné aktualizovat předpověď z neuronové sítě každou hodinu a mohlo by to zhoršit její efektivnost i přes větší trénovací sadu. Aktualizace tedy bude probíhat každé 3 hodiny a to v hodiny dělitelné třemi. K tomu musí být také uspůsobena tvorba trénovací sady.

5.4 Zobrazení výsledků

Aby měla neuronová síť nějaký smysl a hlavně aby bylo s čím porovnávat výkonost jednotlivých neuronových sítí, je třeba nějakým způsobem zobrazovat výsledky. Jelikož bude potřeba zjistit vztah mezi neuronovou sítí s několika předpovědními službami, bude nejlepší použít nejen zobrazení další předpovědi, ale i vytvoření a zobrazení grafů, které ukáží, jak blízko má neuronová síť k jednotlivým předpovědním službám. K zobrazení pak bude potřeba vytvořit grafickou aplikaci. Pro C# pak bude nejjednodušší použít WPF aplikaci se základním uživatelským prostředím.

5.4.1 Zobrazení předpovědi

Pro zobrazení předpovědi bude použita krátkodobá neuronová síť, která bude na 48 hodin dopředu předpovídat více typů hodnot podle toho, které zvolí uživatel. Na 3 dny dopředu pak budou zobrazovány maximální a minimální teploty z dlouhodobých předpovědí.

5.4.2 Grafy

Jelikož bude potřeba porovnávat jednotlivé neuronové sítě a zároveň pozorovat, k jaké předpovědní službě má neuronová síť jaký vztah, bude potřeba vytvořit 2 typy grafů. Pro porovnání bude nejlepší použít bodový graf, který bude tvořen na základě odchylek předpovědí dvou neuronových sítí oproti reálnému stavu počasí. Pro graf, který odhalí vztah neuronové sítě s předpovědními službami, pak bude použit graf s lineárním propojením bodů služeb a neuronové sítě.

Kapitola 3 – Implementace

6 Implementace serverové aplikace

Jak už jsem se zmínil v předcházející kapitole má serverová aplikace na starosti ukládání dat z předpovědí do databáze. Po výběru serverů. Serverová aplikace pak běží na školním serveru Ant2. Vzhledem k tomu, že byla aplikace napsána v C# ve Windows bylo pro použití na linuxovém serveru nutné instalovat Mono i s příslušnými knihovnami.

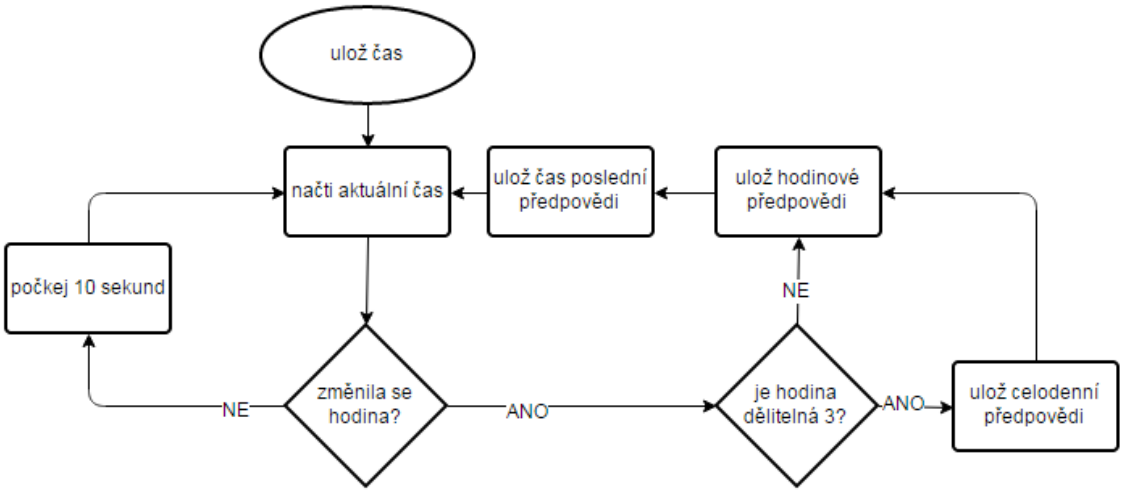
Použité externí knihovny:

- System.Xml.Linq – pro zpracování XML souborů
- System.Drawing – pro zpracování grafických předpovědí
- MySql.Data – pro manipulaci s databází

Serverová aplikace je rozdělena do 5 tříd:

- Program – obsahuje funkci *main*
- Vars – reprezentuje data počasí a obsahuje metody pro jejich ukládání do databáze
- Forecast – třída obsahující statické metody pro předpovědi
- Database – obsahuje statické metody pro inicializaci a práci s databází
- Clrs – obsahuje statické funkce, které vrací hodnotu barvy podle barevného spektra

main funkce je založena na nekonečném cyklu, jehož jednoduchá logika je vyobrazena na diagramu 6.1.



Obrázek 6.1: Logika programu.

6.1 Vytvoření a manipulace s databází

O vytvoření/resetování a vypisování zpráv z databáze se starají metody třídy Database. Ta obsahuje metodu *Ini()*, která pomocí SQL příkazů vytvoří tabulky a naplní tabulku *services* údaji o službách. Dále obsahuje metodu *Restart()*, která se stará o smazání a znovuvytvoření tabulek databáze. A několik dalších metod, jejichž hlavní funkcionalita byla určena pro testování. Pro inicializaci spojení s databází pak používá metod třídy *Vars*.

6.2 Analýza dat

Jak už bylo řečeno, data jsou získávána z 3 formátů a to z XML, HTML a grafického (PNG). S XML pro celodenní předpovědi, není větší problém vzhledem ke knihovně System.Xml, díky které je zpracování těchto dat opravdu jednoduché a nebyly s ním větší problémy, až na službu Wunderground, která poskytuje velmi obsáhlý, ale také nepřehledný soubor XML. Horší to pak bylo s HTML a PNG - grafickými předpověďmi, které jsou popsány níže.

6.2.1 HTML předpovědi

S HTML je největší problém v tom, že každá služba implementuje úplně rozdílnou strukturu HTML souboru a že se musí načíst celá stránka a z ní vybrat právě ty informace, které potřebujeme. V Původní implementaci, bylo využito toho předpokladu, že se data nachází vždy na stejném řádku souboru. Tento předpoklad se ale ukázal jako špatný. Navíc při sebemenší změně HTML kódu by toto zpracování zřejmě začalo vykazovat nechtěné výsledky.

Proto bylo v druhé a poslední implementaci využito vyhledávání, díky kterému je možné se zaměřit na stěžejní body HTML souboru a podle nich se orientovat. Nevýhoda této varianty ale je, že je hardwarově náročnější.

Vzhledem k tomu, že analyzovaný řádek dat většinou obsahuje velké množství informací pro formátování případně jiné, byla implementována pomocná funkce *deleteChars()*, která z řetězce maže vše, co není '-' nebo číslo.

6.2.2 Grafické předpovědi

Přehled pomocných funkcí:

- Last – zjišťuje nejaktuálnější kompletní předpověď
- generateCode – generuje kód pro switch barevného spektra
- GetPixel – vrací RGB barvu určitého pixelu
- pozice/poziceAladin – vrací x a y souřadnici daného města na dané mapě

Pro grafické předpovědi byly vybrány služby Meteocentrum, Slunecno a Aladin. Z těch byly nakonec použity jen 2 a to Slunečno a Aladin, protože Meteocentrum poskytovalo mapy s proměnným spektrem barev pro hodnoty a to bez jasných pravidel této změny čehož jsem si všiml až potom co jsem službu začal implementovat.

Pro zpracování spektra barev pro tyto 2 servery je použita funkce *generateCode()* ve třídě *Forecast*, která vytvoří kód pro *switch* podle barev spektra dané služby. Tento kód je pak použit ve třídě *Clrs*, u dané metody služby.

Funkce *GetPixel()* implementovaná ve třídě *Forecast*, se stará o načtení obrázku ze serveru jako bitmapy a o navrácení barvy pixelu, jehož poloha je dána právě vybraným místem, jehož název se posílá do funkce *pozice()*, pro službu Slunečno a *poziceAladin()* pro službu Aladin, která vrací pozici x a y na obrázku. Barva jako RGB string se pak posílá do funkce třídy *Clrs* podle toho, o jakou službu se jedná. Pokud funkce nenajde hodnotu v rámci spektra barev, kterou má uloženou, opakuje se proces s posunutí pozice pozorovaného pixelu o jeden bod doprava a to i několikrát. V zásadě dokud se nenarazí na analyzovatelný výsledek. Toto je potřeba kvůli číslům, nebo hraničním bodům, které jsou v obrázku zobrazeny a je možné, že se v době pozorování nachází v bodě, ze kterého jsou získávána data.

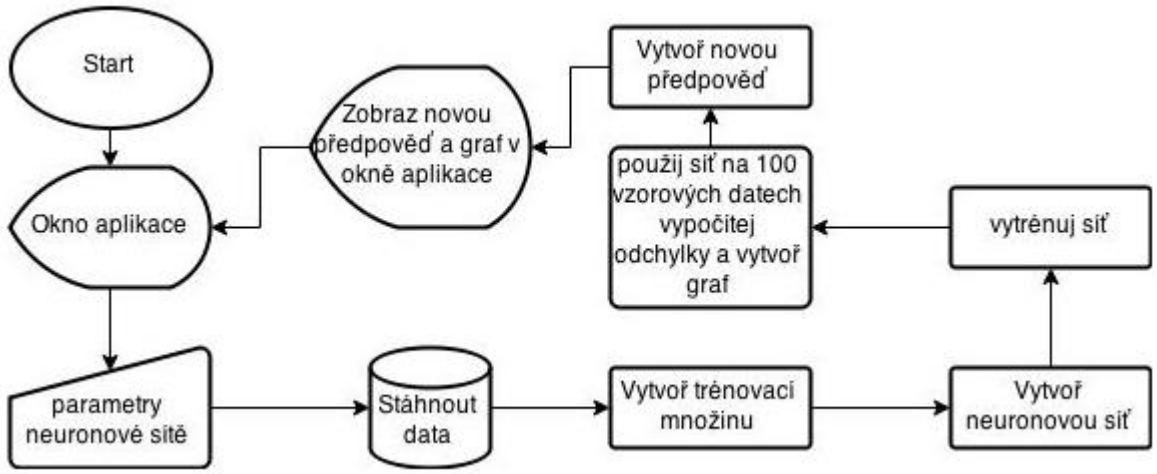
6.3 Ukládání do databáze

Databázi popsanou v předchozí kapitole jsem inicializoval na školním serveru Ant2. Do ní jsem vložil všechny tabulky popsané v kapitole 4.1. Ukládání zanalyzovaných dat má pak na starosti třída *Vars*, která drží informace o všech datech analyzovaných v jednom průchodu XML, tzn., že neobsahuje pole a každý objekt této třídy je schopný naráz nést pouze údaje na předpověď určitého času a místa. Tu je pak schopný díky metodám *insertSF()*, pro hodinové předpovědi a *insertLF()*, pro celodenní předpovědi, uložit do databáze. Pro uložení celé předpovědi, na několik dní a míst, je pak nutné objekt třídy *Vars* několikrát zresetovat a uložit do nich nové údaje, které se před dalším resetem uloží do databáze.

7 Implementace Neuronové sítě

Jak již bylo zmíněno, pro vytvoření neuronové sítě byla použita knihovna *NeuronDotNet*. Dále byla použita knihovna *OxyPlot* a to na tvorbu grafů ve WPF aplikaci. Vytvořený program se skládá z těchto tříd:

- Normalizace – statická třída sloužící k normalizaci vstupů a výstupů z neuronové sítě
- Databaze – třída obsluhující komunikaci s databází, přesněji k výběru dat
- NN – třída, která uchovává data z databází a tvoří neuronovou síť nebo sítě
- LongNN – třída zodpovědná za neuronovou síť pro dlouhodobé předpovědi
- MainViewModel – třída obsahuje konstruktor na vložení dat do grafu
- MainWindow – třída, kterou je řízen obsah okna aplikace i děje programu



Obrázek 7.1: Logika WPF aplikace.

7.1 Postup implementace

Na začátku byla aplikace tvořena jako konzolová aplikace tedy bez tříd *MainViewModel* a *MainWindow*. S touto aplikací proběhlo několik prvních testů. Poté jsem se rozhodl vytvořit grafickou aplikaci, která umožnuje alespoň částečné testování bez změn ve zdrojovém kódu.

Nejprve byla implementována třída Databaze, která se stará jak o stažení dat potřebných pro neuronovou síť, tak i o zjištění hraničních hodnot, které jsou dále použity pro normalizaci hodnot vkládaných do neuronové sítě. Dále byla vytvořena statická třída Normalizace, která si uchovává hraniční hodnoty a obsahuje metody pro převod hodnot na normalizované hodnoty, které se vkládají do neuronové sítě a naopak.

Nejdůležitější třídou je pak třída NN, která reprezentuje neuronovou síť a zároveň si uchovává všechny data předpovědí, které jsou využívány pro trénování neuronové sítě. Jejím základem jsou

metody pro vytvoření trénovací sady, vytvoření konkrétní neuronové sítě, metoda pro samotné trénování a metoda pro vytvoření nové předpovědi. Později byla přidána ještě třída LongNN, která reprezentuje samostatnou neuronovou síť a pracuje s maximální a minimální teplotou v rámci celého dne. Nakonec byla vytvořena třída MainViewModel, s jejíž proměnnými jsou provázány prvky tvořící grafy ve WPF aplikaci.

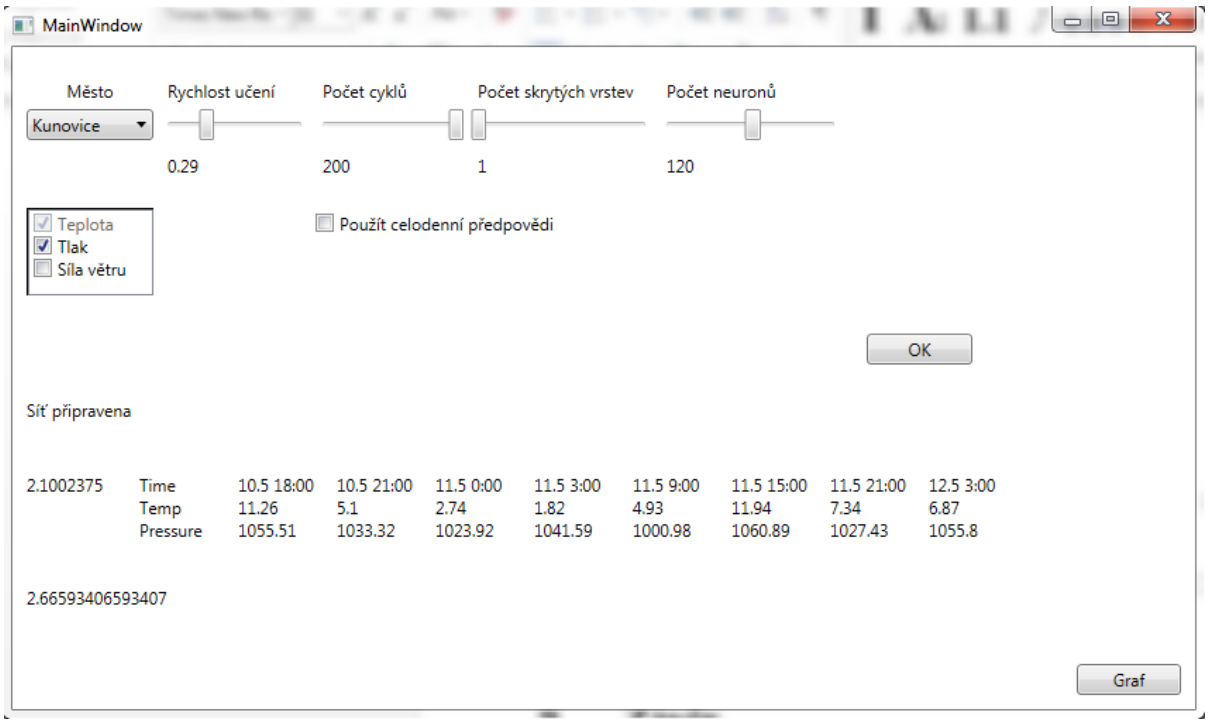
7.1.1 WPF aplikace

WPF aplikace byla vytvořena pro jednodušší testování neuronové sítě a hlavně kvůli vytváření grafů. Není tedy určena pro běžné uživatele, a proto není ani zcela optimalizována ani testována. Tato aplikace byla implementována zároveň s testováním, kde také budou části, které byly do aplikace přidávány kvůli jednotlivým testům popsány. První věcí, která byla do aplikace přidána, byla možnost měnit parametry neuronové sítě dynamicky bez zásahu do kódu programu.

Lineární neuronová síť užívající metodu backpropagation má tyto proměnné parametry:

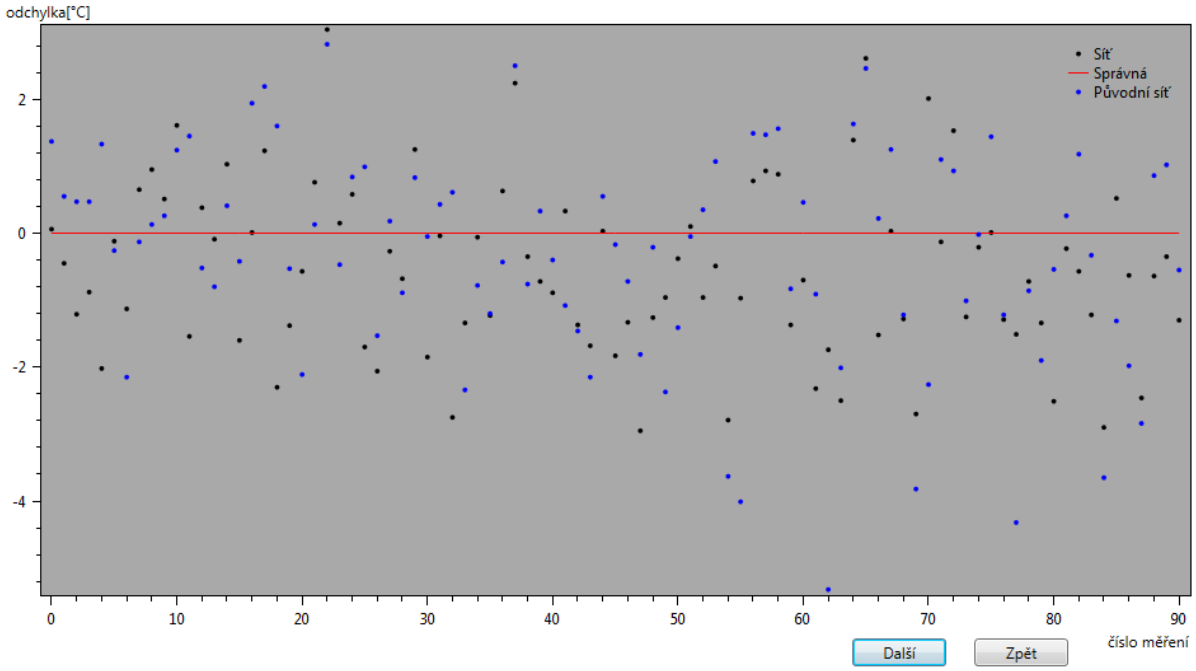
- Rychlost učení: od 0.01 po 0.99
- Počet cyklů učení: od 1 po 200
- Počet vstupů a výstupů
- Počet skrytých vrstev od 1 do 3 a počet jejich neuronů
- Místo předpovědi – města popsána v kapitole 2.2.4

Dále bylo přidána možnost ovládat do jisté míry vstupy a výstupy neuronové sítě pro krátkodobé předpovědi. Později byla tato možnost implementována pouze pro hodnoty prospívající předpovědi. Konečná aplikace pak vypadá takto:



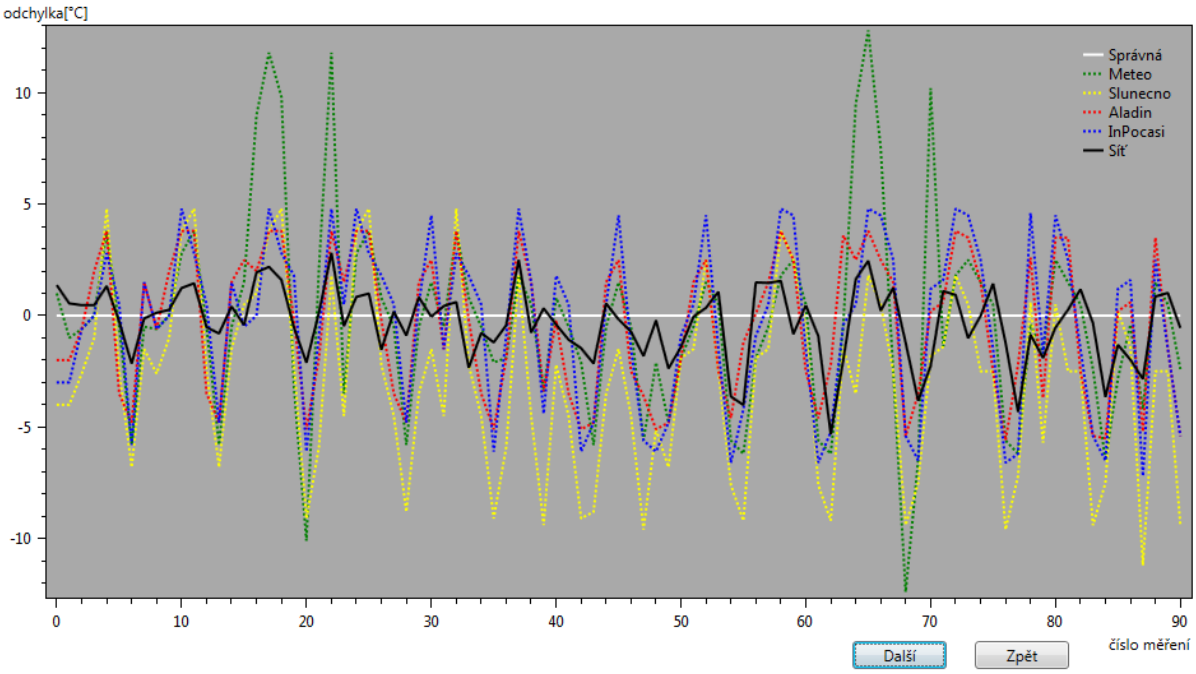
Obrázek 7.2: Aplikace na testování.

Pro porovnání výkonosti dvou sítí s různými parametry pak slouží první graf, který je tvořen z bodů vytvořených dvěma naposledy vytvořenými neuronovými sítěmi. Příklad:



Obrázek 7.3: Příklad porovnání 2 sítí.

Pro Porovnání výkonosti neuronové sítě se službami poskytujícími počasí slouží druhý graf, který bere odchylky v předpovědích jednotlivých služeb a neuronové sítě v devadesáti časových intervalech. Příklad:



Obrázek 7.4: Příklad porovnání neuronové sítě se službami.

8 Testování

Tato fáze probíhala ještě za dokončování implementace konzolové aplikace a pokračovala implementací WPF aplikace.

8.1 První fáze – Optimalizace

Cílem této fáze bylo rozhodnout o velikosti sítě, jejích vstupech a výstupech a vytvořit tak optimální neuronovou síť, na které budou probíhat další testy.

8.1.1 Vyřazení nevhodných služeb

Účelem první fáze testování bylo zjistit, zda se vyplatí u všech předpovídaných míst (měst) použít všechny předpovědi, tedy jestli tyto předpovědi, jako vstupy do neuronové sítě nemají negativní účinek na přesnost předpovědí neuronové sítě. K tomuto účelu byla vytvořena metoda, která počítala průměrnou odchylku předpovědí, a to jak předpovědí neuronových sítí tak i předpovědní z předpovědních portálů uložených v databázi.

8.1.1.1 Porovnání přesností služeb

Protože bylo třeba zjistit, jestli některá z předpovědí není příliš nepřesná, a tím pádem škodlivá, byly nejdříve vypočítány hodnoty průměrných odchylek u předpovědních služeb a to na 200 vzorových datech. Pro porovnání byla použita hodnota teploty, která se vyskytuje ve všech předpovědích. Tyto testy ukázaly, že jedna z předpovědních služeb má ve svých předpovědích teploty velké chyby a to v průměru mírně přes 3 stupně na dalších 36 hodin zatímco ostatní služby byly většinou mírně přes 2 stupně.

8.1.1.2 Test neuronové sítě

Bylo třeba rozhodnout, zda tuto službu vyřadit jako škodlivou. Tato služba byla jednou ze služeb předpovídající hodinové počasí. Byly tedy vytvořeny dvě neuronové sítě pro hodinovou předpověď. První neuronová síť obsahovala předpovědi teplot ze všech čtyřech služeb předpovídajících na 36 hodin. Druhá obsahovala pouze tři, které měly přijatelnou odchylku předpovědí. Obě tyto neuronové sítě měly až na počet vstupů stejné parametry. Porovnávat se pak měly průměrné odchylky v přesnosti těchto dvou sítí vypočítané na stu vzorových dat.

8.1.1.3 Závěr

Jak se ukázalo, neuronová sít číslo 1, tedy ta užívající všechny služby byla po 3 pokusech zhruba o jednu desetinu přesnější, než neuronová síť číslo 2. Služba tedy neuronové síti prospěla i přes své velké odchylky v předpovědích a dále se tedy budou používat všechny předpovědní služby.

8.1.2 Zkouška masivní sítě

Jak již bylo zmíněno v kapitole 5.3.2, je třeba rozhodnout, zda na řešení hodinových a celodenních předpovědí bude lepší použít dvě menší neuronové sítě, nebo jednu masivnější neuronovou síť. I když spolu tyto data jistým způsobem souvisí, je u nich jiná logika a je tedy otázkou, zda se jí dokáže neuronová síť přizpůsobit. Vzhledem k tomu, že s větším množstvím předpovědních služeb vzniká větší riziko výpadku jednoho ze serverů a tudíž ztrátou některých zkušebních dat, je nutné brát v úvahu menší odchylku, vzhledem k nevelkému počtu naměřených dat. V tomto testu bude znovu experimentováno pouze s teplotou.

8.1.2.1 Způsob porovnání

Z předchozího testování byla již implementována neuronová síť pouze pro krátkodobé předpovědi. Rozhodl jsem se proto, že ji využiji k porovnání s novou „masivní“ sítí využívající jak krátkodobé, tak dlouhodobé předpovědi. První neuronová síť implementující pouze krátkodobé předpovědi měla 35 vstupních neuronů v podobě 4 služeb předpovídajících teplotu na dalších 36 hodin v 8 intervalech a hodnoty teploty z třech předcházejících hodin. Mezi vstupní a výstupní vrstvou byla jediná skrytá vrstva se 70 neurony. Masivnější neuronová síť měla navíc 4 služby pro celodenní předpovědi, předpovídající na další 3 dny maximální a minimální teplotu. To znamená dalších 24 vstupních neuronů dohromady tedy 59 a dalších 6 výstupních tedy 14. Skrytá vrstva zůstala se 70 neurony. Další parametry byly pro obě sítě stejné. Rychlost učení 0.3 a jako město pro testování jsem zvolil Brno.

8.1.2.2 Neoptimalizovaný test a možnosti optimalizace

První neuronová síť měla průměrnou odchylku ve výstupní předpovědi 1,7 stupňů celsia, zatímco průměrná odchylka u masivnější sítě byla 1,85 stupňů. Masivnější síť se tedy zdála být nevhodná. Bylo zde ale několik důvodů proč by mohla mít masivnější síť problémy. Jako první je velikost trénovacích dat, kterých by potřebovala mít více než původní síť. Druhý důvod pak mohl být malý počet neuronů ve skryté vrstvě. Třetím důvodem pak malý počet skrytých vrstev. Rozhodl jsem se proto zkusit test znovu s optimálnější sítí.

8.1.2.3 Optimalizovaný test

Kvůli nedostatku trénovacích dat byl zvětšen počet cyklických průchodů trénovacích dat ze 100 na 200. Počet neuronů ve skryté vrstvě byl zvýšen na 118 tedy dvojnásobek vstupních neuronů. Počet skrytých vrstev zůstal 1. Byl proveden experiment i se dvěma vrstvami, který ale výsledek pouze zhoršil.

8.1.2.4 Závěr

Výsledné hodnoty nové neuronové sítě měly odchylku 1,82 stupňů tedy ani po vyřešení možných problémů masivnější sítě nebyla síť dostatečně efektivní. Proto budou pro další použití tvořeny dvě menší neuronové sítě místo jedné masivní.

8.1.2.5 Doplnění

Test byl zopakován s městem Kladno s podobným výsledkem. Po dokončení testu byly vytvořeny 2 nezávislé sítě, které s podobnými parametry jako byly použity u tohoto testu, dosáhly odchylek neuronové sítě 1,7 stupňů u hodinových a 1,55 stupňů u celodenních předpovědí.

8.2 Druhá fáze – Síť s více typy hodnot

Prozatím program pracoval pouze s jedním typem hodnot, který byl proměnný. Program však zatím neumožňoval vkládat do sítě více typů hodnot do vstupů a výstupů. Do okna aplikace byla tedy přidána možnost vybrat hodnoty, které do neuronové sítě budou vstupovat a dále možnost určit výstupy neuronové sítě. Otázkou se znovu stalo, zda se předpověď hodnot zlepší při použití větší sítě, kdy spolu hodnoty nějakým způsobem souvisí, nebo udělat pro každý typ hodnoty vlastní neuronovou síť. Z důvodů zajímavějších výsledků, přehlednější aplikace a více možností pro vstup a výstup byla tato možnost implementována pouze pro krátkodobé předpovědi.

8.2.1 Návrh experimentu

Protože neexistují reálné záznamy srážek, ale jen vlhkosti, od kterých zase neexistují krátkodobé předpovědi, byly provedeny experimenty 2. První experiment obsahoval jako vstup teplotu a srážky a jako výstup teplotu a vlhkost, která je srážkami velmi ovlivněná. Druhý experiment měl stejné vstupy, ovšem jako výstupy byly použity pouze předpovědi teploty. Síť z druhého experimentu měla za úkol zjistit, zda je možné zpřesnit výsledky jedné hodnoty tím, že neuronová síť bere v úvahu i další údaje.

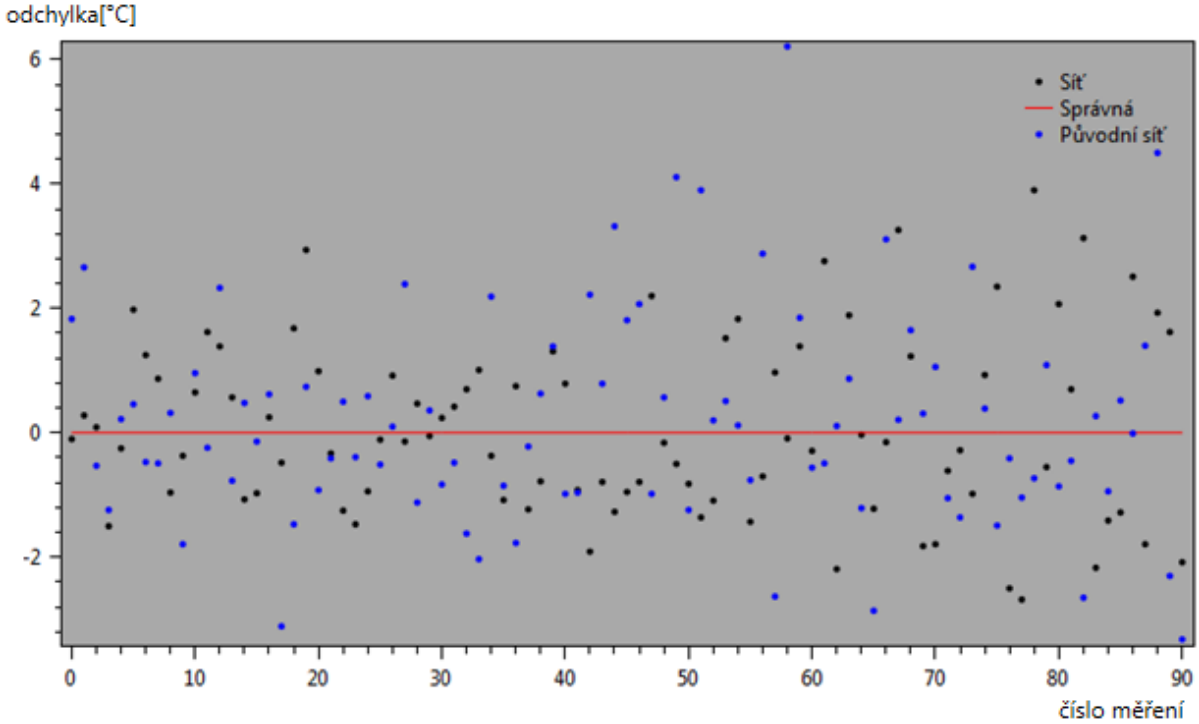
8.2.2 Výsledky experimentů

První i druhý experiment dosáhl výsledků, které byly mnohem horší než u sítí, které fungovali pouze s jedním typem hodnot. Je docela možné, že jen proto, že trénovacích dat nebylo tolik, aby mohly dostatečně „nakrmit“ větší neuronovou síť. Další možností bylo to, že jako výstup byl použitý jiný typ hodnoty než pro výstup z neuronové sítě, srážky a vlhkost, což mohlo neuronovou síť lehce zmást. Proto jsem se rozhodl udělat ještě jeden experiment a to pro neuronovou síť, obsahující teplotu a sílu větru.

8.2.3 Výsledek a závěr experimentu se silou větru

Jak se ukázalo při použití síly větru jako vstupu a výstupu neuronové sítě místo použití srážek a vlhkosti, zlepšila se průměrná předpověď teploty o jednu desetinu stupně. Problém pak nastal po zjištění, že i síť, která pracuje jen se srážkami a vlhkostí dosahuje průměrné odchylky u výstupní hodnoty, kterou byla vlhkost, 15%, tedy že se jedná o absolutně nepřijatelnou a nepoužitelnou předpověď. Tato hodnota byla tedy z testování vyřazena a proto, že zůstaly už jen 2 hodnoty a to teplota a síla větru.

Bohužel nejsou k dispozici další data, která by byla předpovídaná více službami s krátkodobými předpověďmi. Přesto byla přidána ještě hodnota tlaku, kterou předpovídá jediná služba. I předpověď z jedné služby však pozitivně ovlivňuje výsledek předpovědi. Rozdíl je vidět na dalším obrázku:



Obrázek 8.1 : Teplota a všechny hodnoty.

Na obrázku 8.1 je vidět porovnání přesnosti předpovědi teploty modré neuronové sítě obsahující pouze vstupy a výstupy teploty a druhé, černé neuronové sítě obsahující hodnoty teploty, síly větru a tlaku. Průměrná odchylka druhé sítě je zhruba o 4 desetiny nižší než odchylka sítě obsahující pouze hodnoty teploty.

8.3 Třetí fáze – Porovnání sítě se službami

V této fázi je již realizována relativně optimalizovaná neuronová síť. Dostáváme se tedy k původně zamýšlenému testování a to jaký mají služby vztah k jednotlivým geografickým místům a jaký má neuronová síť vztah k jednotlivým službám v rámci každého místa. Pro toto porovnání bylo třeba

naimplementovat další část aplikace a to nový graf, ve kterém budou figurovat i předpovědní služby. Myšlenkou je to, že podle rozdílu předpovědi neuronové sítě od jednotlivých služeb bude možné určit, které předpovědní službě dává neuronová síť větší a menší váhu.

8.3.1 Návrh testu

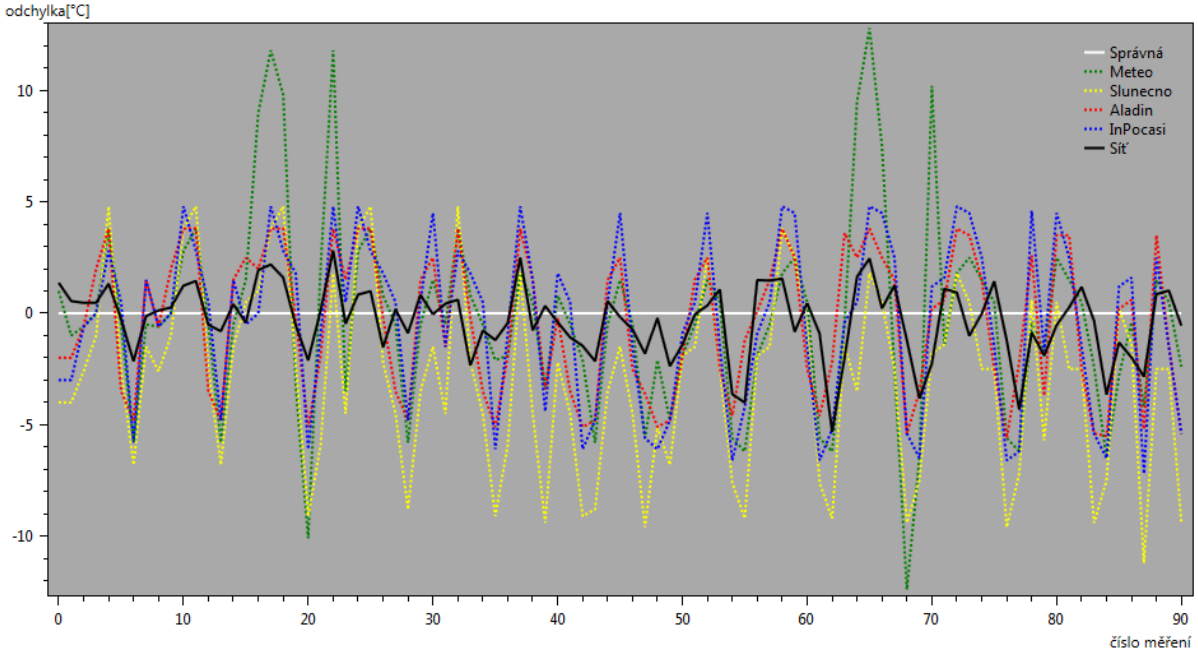
Pro tuto fázi byl vytvořen nový typ grafu popsaný v návrhu, kapitola 5.4.2. Aby bylo dosaženo požadovaných výsledků, byla implementována neuronová síť o stejných parametrech pro každé z předpovídaných míst. Protože se síť pokaždé naučí pracovat trochu jinak, bylo měření pro každé z míst zopakováno třikrát.

Tato síť pracovala pouze s teplotou a tlakem. Sít pracovala s následujícími parametry:

- Rychlost učení: 0.29
- Počet cyklů trénování: 200
- Skrytých vrstev: 1
- Počet neuronů skryté vrstvy: 120

8.3.2 Brno

Jako první místo bylo vybráno Brno, na kterém byla prováděna většina předchozích testování. Protože zde nejsou garantovány ideální podmínky, byly očekávány velké odchylky předpovědních služeb a tak i předpovědní neuronové sítě. Na následujícím obrázku lze vidět odchylky jednotlivých služeb a odchylky neuronové sítě.

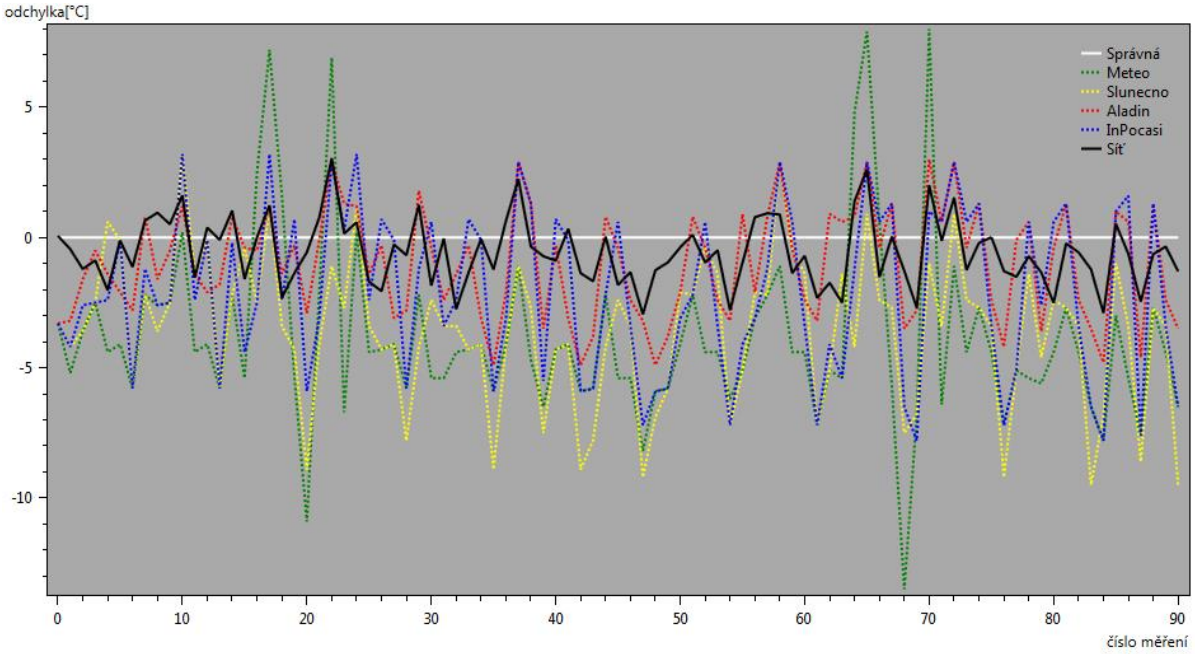


Obrázek 8.2 : Služby a neuronová síť Brno.

Podle obrázku 8.2 je patrné, že všechny předpovědní služby předpovídají relativně přesně až na Slunečno, což bylo řešeno v kapitole 8.1. Je zde i vidět proč je služba i přes velké odchylky přínosná. Její divergence je totiž téměř vždy ve stejném směru jako u ostatních předpovědních služeb akorát s větší odchylkou. Služba Meteocentrum má také občas problémy a to spíše v předpovědích na delší dobu. Na neuronové síti, která má v tomto případě průměrnou odchylku 1,9 stupňů, lze pak pozorovat, že je přesnější než všechny služby z kterých čerpá a že je používá všechny, tedy že se nenaučila používat pouze nejpřesnější z nich. Nejpřesnější byla služba Aladin s průměrnou odchylkou 2,2 stupňů. Průměrná odchylka služeb pak byla 3 stupně.

8.3.3 Kladno

Další vybrané místo tentokrát s předpokladem ideálních podmínek. Tentokrát na druhé straně republiky. Odchylky jsou vidět na dalším obrázku:

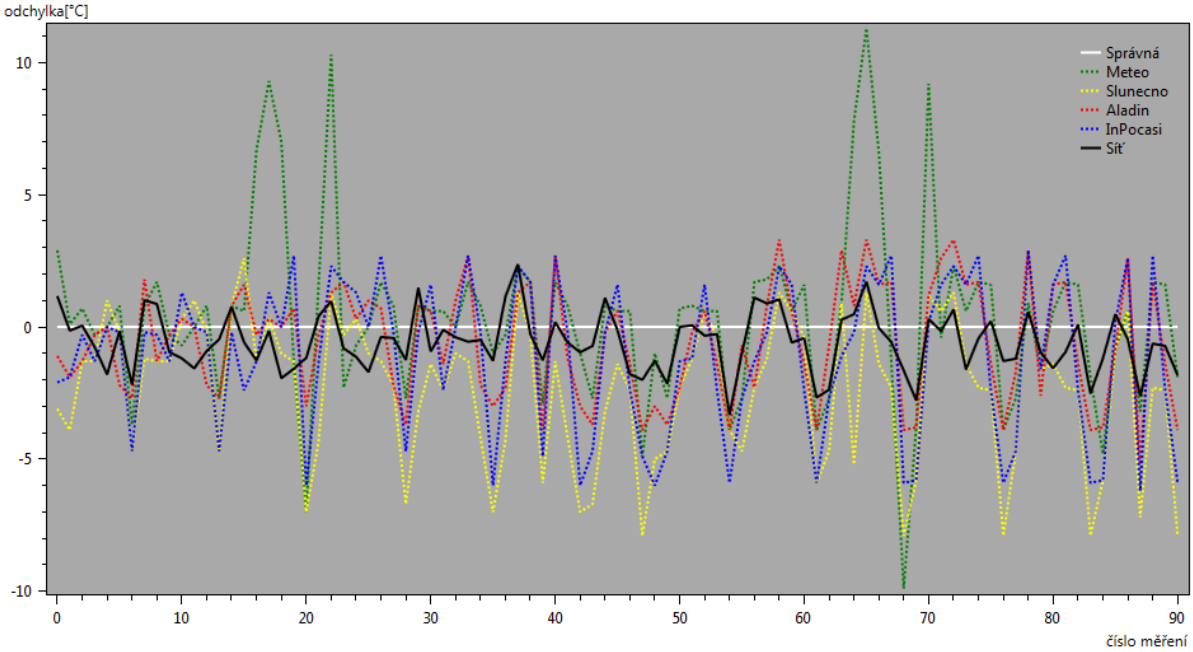


Obrázek 8.3 : Služby a neuronová síť Kladno.

Jak lze na obrázku 8.3 vidět předpokládané ideální podmínky se potvrdili a většina předpovědních služeb je mnohem přesnější, než tomu bylo u Brna. Rozdíl je tu u služeb, kdy u Brna dosahovaly až na službu Slunečno ostatní služby podobně dobrých výsledků, zde je vidět že nejpřesnější předpovědní službou je Aladin s průměrnou odchylkou 1,69 stupňů. Služba Aladin má v tomto případě i o 2 setiny přesnější předpověď než neuronová síť. Neuronová síť dosáhla průměrné odchylky 1,71. Průměrná odchylka všech služeb je ale 2,7 stupňů. Ani zde nelze pozorovat větší závislost sítě na jedné službě.

8.3.4 Třebíč

Další město s ideálními předpoklady a pozicí nedaleko města Brno, tedy ideální k porovnání s předpověďmi Kladna. Následuje obrázek s odchylkami:

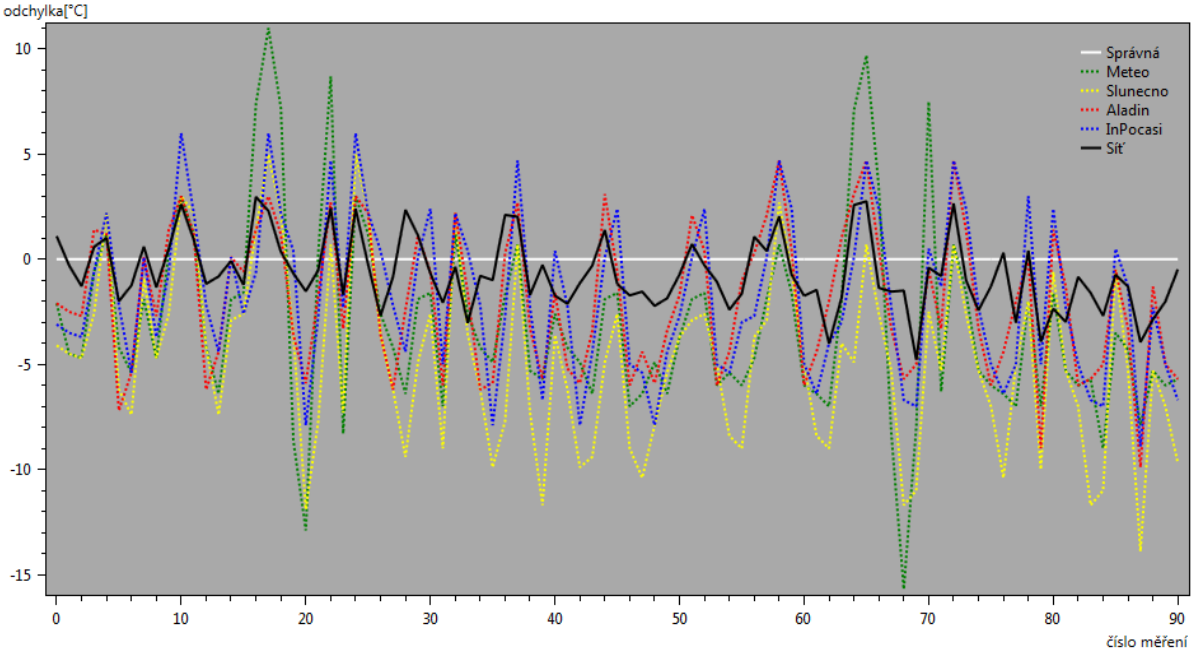


Obrázek 8.4 : Služby a neuronová síť Třebíč.

Z obrázku 8.4 je patrné, že předpovědní služby svou přesností převyšují jak Brno, tak Kladno. Jsou zde vidět velké odchylky služby Slunečno a Meteocentrum stejně, jak tomu bylo i u Brna. Průměrná odchylka služeb je pak 2 stupně, což je zatím nejlepší výsledek. Je to dáno velmi přesnými předpověďmi služby Aladin, s průměrnou odchylkou 1,37 stupňů. Přesto průměrná odchylka neuronové sítě je 1,8 stupňů což překonává všechny ostatní služby ale je mnohem horší než služba Aladin. Zároveň je také horší než neuronová síť pro Kladno, která měla k dispozici horší předpovědi služeb. Rozdíl lze posoudit neutrálně vzhledem k tomu, že zde oproti Kladnu existuje pouze jedna služba s dobrými předpověďmi, zatímco ostatní zaostávají. Je vidět, že službě Aladin tato poloha sedla, bohužel se neuronová síť nedokázala naučit na nejlepší poskytovanou předpověď a to ani po změně parametrů a využívala i méně přesných předpovědí.

8.3.5 Kunovice

S novým geografickým místem přichází Kunovice. Toto místo je nejníže položené ze všech pozorovaných míst a je proto zajímavé pro porovnání s ostatními městy.

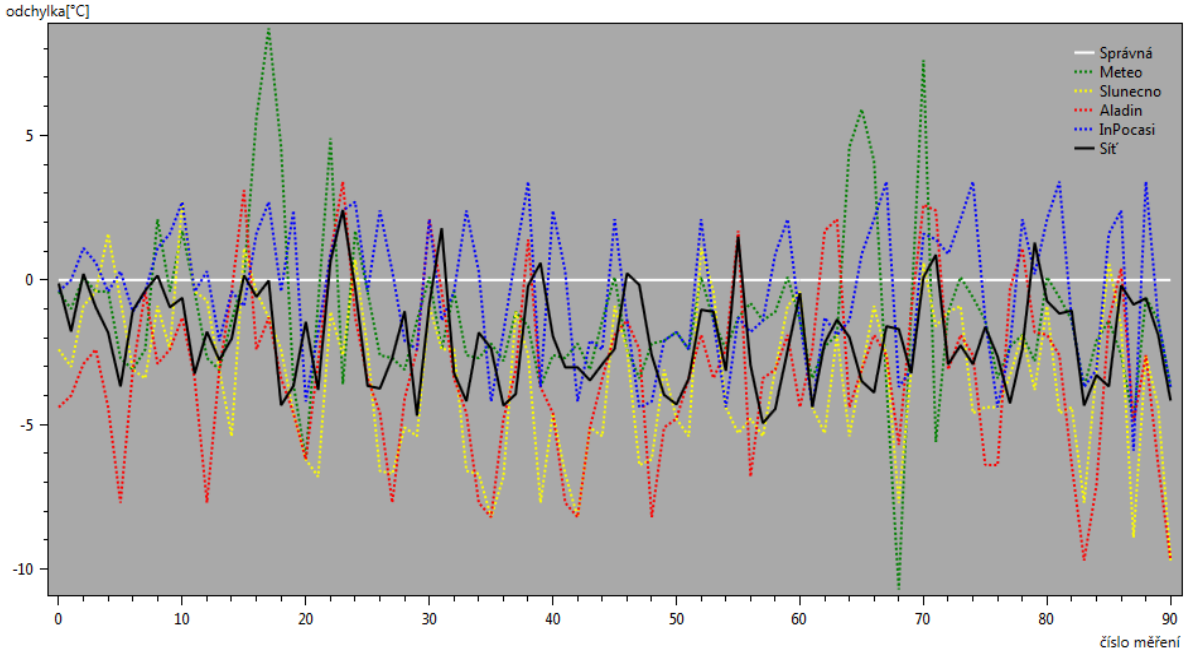


Obrázek 8.5 : Služby a neuronová síť Kunovice.

Již z obrázku 8.5 lze pozorovat velké odchylky jednotlivých služeb. Průměrná odchylka služeb je na tomto místě 3,5 stupňů, což je zatím nejvíce ze všech pozorovaných míst. Průměrná odchylka sítě je 1,9 stupňů, což je vzhledem k nepřesnosti služeb velmi dobrý výsledek. Je vidět že na služby měla velký vliv lokální nízká nadmořská výška v tomto případě nízká a žádná z nich tu nepodává dobrý výkon. V dalším testu bude možnost porovnat výkonost služeb ve vysoké nadmořské výšce.

8.3.6 Špindlerův Mlýn

Místo s nejvyšší nadmořskou výškou a nejsevernější geografickou pozicí.

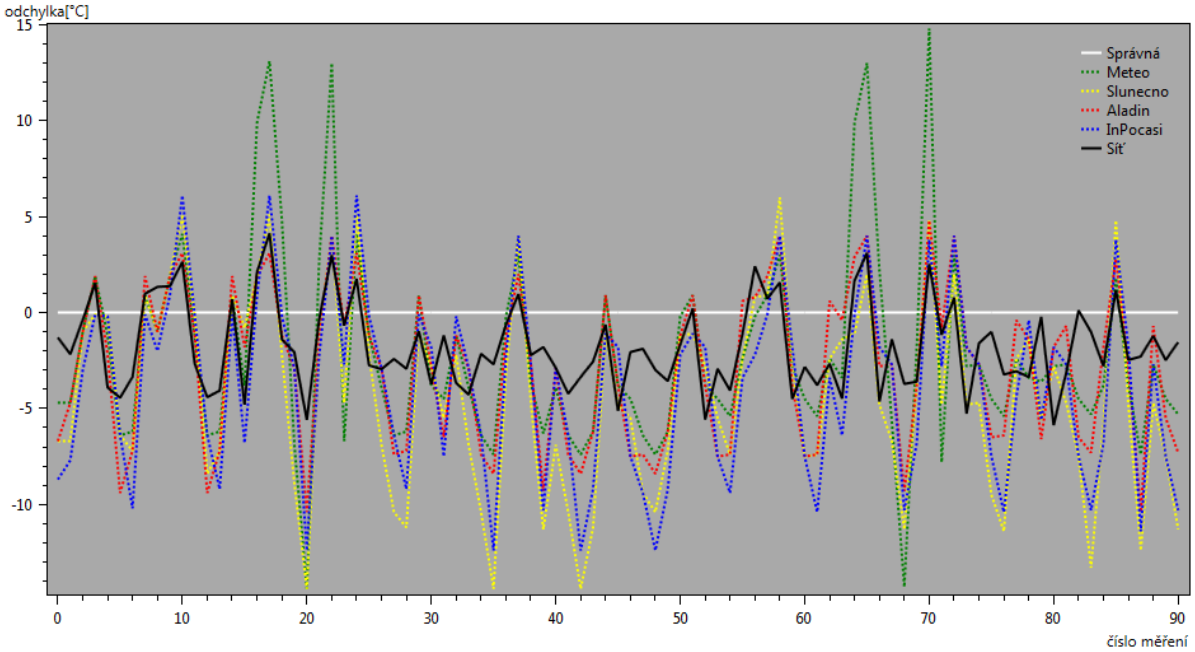


Obrázek 8.6: Služby a neuronová síť Špindlerův Mlýn.

Z obrázku 8.6 lze vypozorovat, že služby si mírně prohodily role. Nejméně přesná služba je zde Aladin a nejpřesnější je InPočasí s průměrnou odchylkou 1,8 stupňů. Neuronová síť dosahuje odchylek 2 stupně ze vstupů služeb o průměrné odchylce 2,7 stupňů. Problémem je pak jejich nepravidelnost. Když se podíváme na obrázek 8.7 lze vidět, že jednotlivé předpovědní služby nemají žádné pravidlo vzdálenosti od ostatních předpovědí, které by mohla neuronová síť použít. Jak je tedy vidět na předpověď počasí má větší vliv nadmořská výška než geografická pozice.

8.3.7 Karlovy Vary

Poslední město s ideálními podmínkami.



Obrázek 8.7: Služby a neuronová síť Karlovy Vary.

I přes očekávané ideální stavy zde byly předpovědi velmi nepřesné. Neuronová síť byla nejpřesnější ze všech služeb i přes svou průměrnou odchylku 2,36 stupňů. Předpovědní služby měly průměrnou odchylku 4,2 stupňů. Zde je vidět vliv geografické pozice na přesnosti předpovědí.

8.3.8 Závěr Experimentu

Očekáváné zlepšení předpovědí pomocí neuronové sítě bylo bohužel potvrzeno jen u poloviny pozorovaných míst. Nejvíce síť zklamala u Třebíče, kde by možná bylo vhodné použít pro novou předpověď teploty pouze službu Aladin.

9 Závěr

V rámci této bakalářské práce jsem se seznámil s neuronovými sítěmi a jejich metodami. Pro předpovídání na základě jiných předpovědí jsem vybral 6 geografických míst po České republice. Dále jsem vybral 8 internetových služeb, předpovídající počasí v daných oblastech a jednu službu, která poskytuje aktuální hodnoty počasí v daných oblastech. V první části práce jsem vytvořil serverovou aplikaci, která data z těchto služeb zpracovává a ukládá do databáze umístěné na školním serveru.

Z metod neuronových sítí jsem si vybral metodu zpětného učení (backpropagation), nad kterou jsem síť implementoval. Když se databáze naplnila dostatkem dat, tedy asi po 3 měsících ukládání, začal jsem s testováním. Pro testování jsem vytvořil grafickou WPF aplikaci, kterou jsem postupně dotvářel podle potřeb.

V první části testování jsem se pokusil o optimalizaci vstupů a výstupů neuronové sítě a vybral jsem tři hodnoty, které budou v rámci předpovědí měřeny.

V další fázi jsem porovnával výsledky předpovědí jednotlivých služeb s výsledky předpovědi neuronové sítě. Předpovědi neuronové sítě pak byly ve většině případů přesnější než předpovědi jednotlivých služeb bez toho, aby neuronová síť byla na některé z těchto služeb více závislá, což se ale u dvou z oblastí ukázalo jako nevýhoda.

Nevýhodou neuronové sítě je relativně dlouhé učení, které jí bez možnosti uložení, která knihovna, kterou jsem použil, neměla k dispozici, znemožňuje pohodlně použít jako klientskou aplikaci. Jako rozšíření by se tedy dalo uvažovat implementace neuronové sítě jako serverové aplikace, ze které by klienti získávali předpovědi, nebo použití jiné knihovny, které ukládání neuronové sítě umožňuje.

Literatura

[1] DOSTÁL, D. 1997. *Předpovídání časových řad*. Diplomová práce. FEL ČVUT.

[2] Taháky. *Počasí a podnebí* [online]. [cit. 2015-01-03]. Dostupné z: http://tahaky.lam.cz/

[3] DUFOUR, J-M. 2003. *Introduction to time series analysis* [online]. [cit. 2015-01-05]. Dostupné z: http://www2.cirano.qc.ca/~dufourj/Web_Site/ResE/Dufour_1998_C_TS_IntroductionTS.pdf

[4] OBITKO, Marek. *Obitko: Učení neuronové sítě* [online]. [cit. 2015-01-11]. Dostupné z: http://www.obitko.com/tutorials/predpovidani-neuronovou-siti/uceni-neuronove-site.html

[5] GERSHENSON, Carlos. *Artificial Neural Networks for Beginners* [online]. [cit. 2015-01-12]. Dostupné z: http://arxiv.org/ftp/cs/papers/0308/0308031.pdf

[6] DUDKA, Kamil. 2008. *Studijní materiály* [online]. [cit. 2015-01-12]. Dostupné z: http://dudka.cz/study/files/min/sfc.txt

[7] DHTechnologies. *An Introduction to Artificial Neural Networks* [online]. [cit. 2015-01-12]. Dostupné z: http://www.dhtusa.com/media/NeuralNetworkIntro.pdf

[8] SHIFFMAN, Daniel. 2012. *The nature of code: Neural Networks* [online]. [cit. 2015-01-12]. Dostupné z: http://natureofcode.com/book/chapter-10-neural-networks/

[9] Charles University. *Neuronové sítě* [online]. [cit. 2015-01-12]. Dostupné z: http://cgg.mff.cuni.cz/~pepca/prg022/mucha/

[10] VYŠKOVSKÝ, Roman. 2013. *Predikce časových řad pomocí umělých neuronových sítí*. Bakalářská práce. MUNI SCI.

[11] NASA. *What is wheather* [online]. [cit. 2015-01-15]. Dostupné z: http://www.nasa.gov/mission_pages/noaa-n/climate/climate_weather.html#.VLqgSyuG8Xw

[12] W3C. *XML* [online]. [cit. 2015-1-15]. Dostupné z: http://www.w3.org/XML/

[13] tvorba-webu.cz. *XML* [online]. [cit. 2015-1-15]. Dostupné z: http://www.tvorba-webu.cz/xml/

[14] BRADSKI, Gary R. 2008. *Learning OpenCV* [online]. Sebastopol: O'Reilly, xvii, 555 s. [cit. 2015-05-13]. ISBN 978-0-596-51613-0. Dostupné z: http://cs.haifa.ac.il/~dkeren/ip/OReilly-LearningOpenCV.pdf

[15] Sourceforge. *NeuronDotNet* [online]. [cit. 2015-01-15]. Dostupné z: http://sourceforge.net/projects/neurondotnet/

[16] CARPENTER, G.A. a Grossberg S. 2003. *Adaptive Resonance Theory* [online]. [cit. 2015-02-20]. Dostupné z: http://cns.bu.edu/Profiles/Grossberg/CarGro2003HBTNN2.pdf

[17] GROSSBERG, S. 1987. *Competitive learning: From interactive activation to adaptive resonance* [online]. [cit. 2015-2-20]. Dostupné z: http://www.cns.bu.edu/Profiles/Grossberg/Gro1987CogSci.pdf

[18] MC CULLOCK, John. *Adaptive Resonance Theory* [online]. [cit. 2015-02-20]. Dostupné z: http://mnemstudio.org/neural-networks-adaptive-resonance-theory.htm

[19] MEDINA, Gonzalo. *Drawing neural network* [online]. [cit. 2015-02-25]. Dostupné z: http://tex.stackexchange.com/questions/162326/drawing-back-propagation-neural-network

[20] KAČENKA, Petr. *Neuronové sítě* [online]. [cit. 2015-02-25]. Dostupné z: http://mks.mff.cuni.cz/library/NeuronoveSitePK/NeuronoveSitePK.pdf

[21] Oxyplot.org : *Introduction* [online]. [cit. 2015-03-19]. Dostupné z: http://docs.oxyplot.org/en/latest/introduction/introduction.html

[22] Slunecno.cz: *Jak vzniká předpověď počasí* [online]. [cit. 2015-03-27]. Dostupné z: http://www.slunecno.cz/clanky/jak-vznika-predpoved-pocasi-100

[23] Forbes: *Yahoo* [online]. [cit. 2015-04-08]. Dostupné z: http://www.forbes.com/companies/yahoo/

[24] DLOUHÁ, Lenka: *Perioperační péče o pacienta s tumorem mozku při využití intraoperační MR* [online]. [cit. 2015-04-08]. Dostupné z: https://is.cuni.cz/webapps/zzp/download/130069190

[25] Žáci Sophomore Collage. *Neural Networks: History* [online]. [cit. 2015-04-18]. Dostupné z: http://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/History/index.html

[26] Automatizace.cz: *GMDH neuronová síť* [online]. [cit. 2015-05-01]. Dostupné z: http://automatizace.hw.cz/gmdh-neuronova-sit-filtr-pro-digitalni-zpracovani-signalu

Seznam příloh

A. Obsah CD

- Zdrojový kód bakalářské práce (.docx) a elektronická verze ve formátu PDF
- Program pro stahování dat do databáze
- Program pro tvorbu neuronových sítí z dat v databázi
- Uživatelský návod k programům

B. Příklady dat předpovědí

YAHOO (XML)


```
<yweather:forecast day="Fri" date="9 Jan
2015" low="42" high="44" text="Heavy Rain/Wind" code="40"/>
<yweather:forecast day="Sat" date="10 Jan
2015" low="35" high="53" text="AM Rain/Wind" code="12"/>
<yweather:forecast day="Sun" date="11 Jan
2015" low="34" high="38" text="Mostly Cloudy" code="28"/>
<yweather:forecast day="Mon" date="12 Jan
2015" low="39" high="40" text="Cloudy/Wind" code="24"/>
```


IN-Počasí (HTML)

<td>5.9 °C</td>

<td>47 km/h JZ</td>

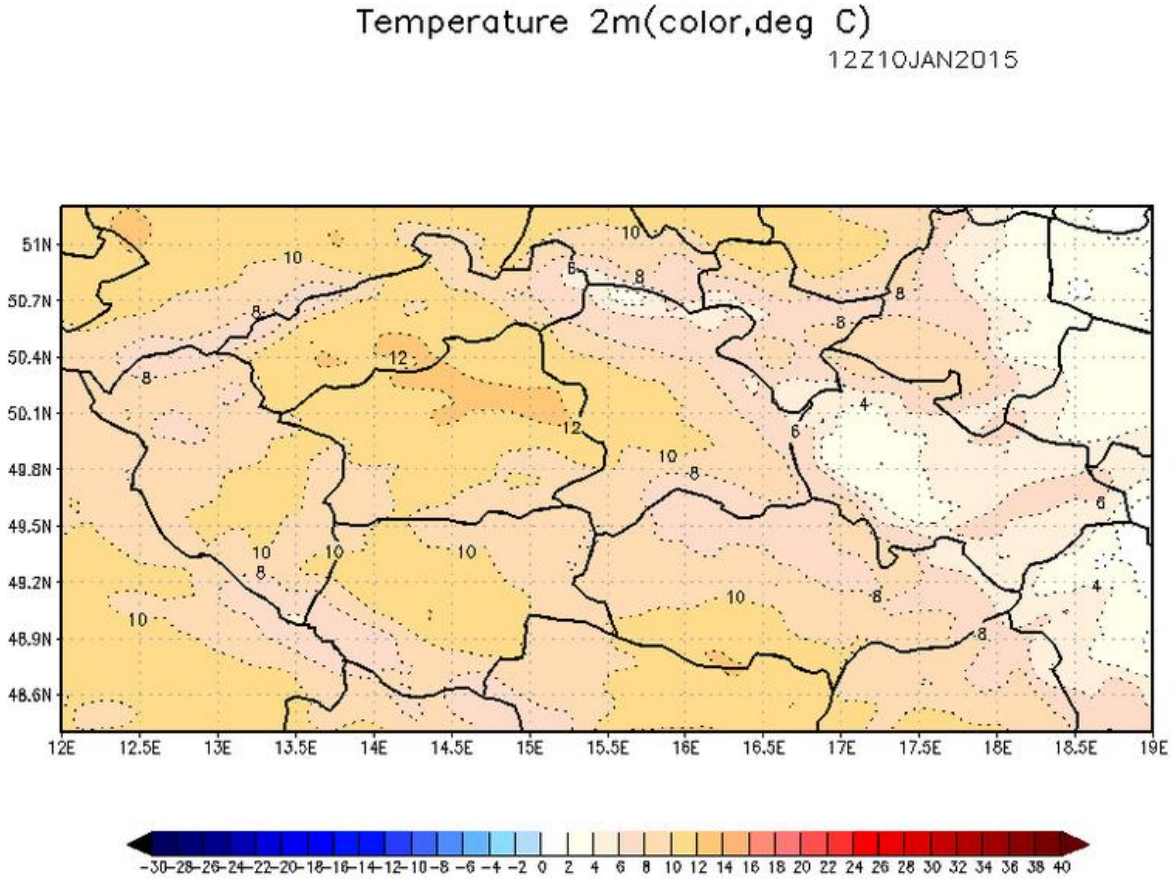
<td>83%</td>

<td>1014.1 hPa</td>

<td>2.2 mm</td>

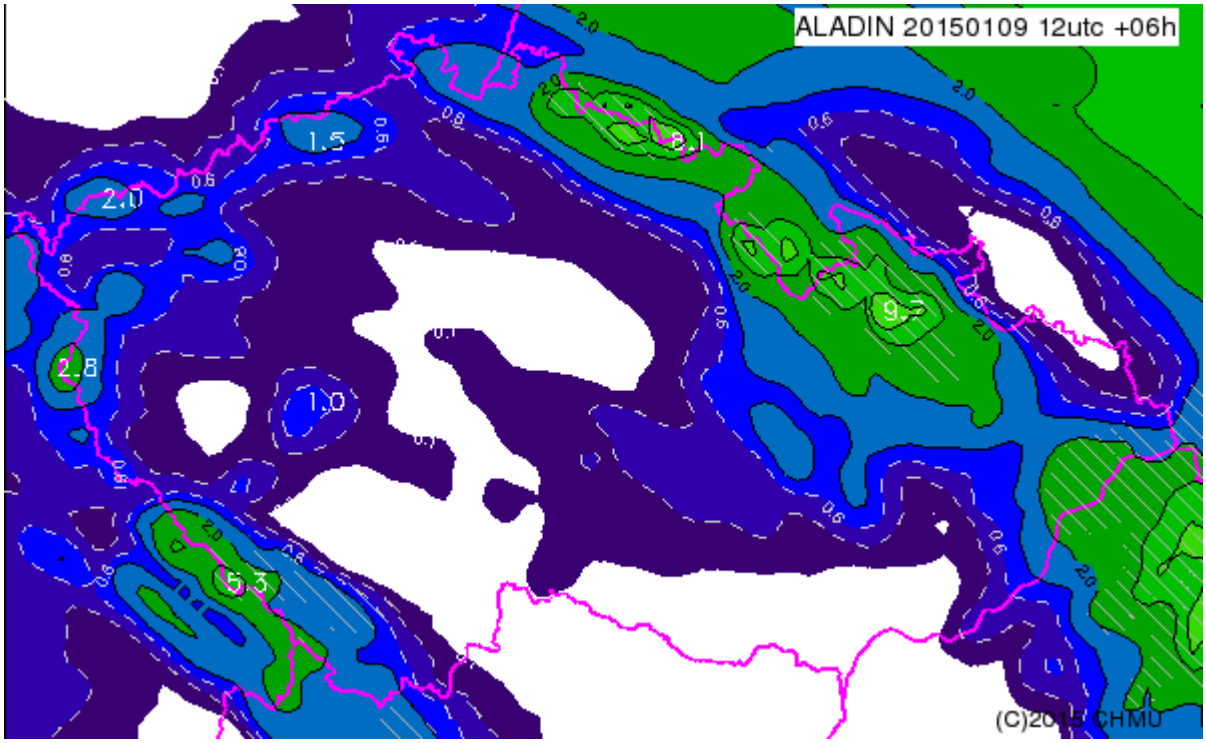
</tr>

Slunecno - teplota (PNG)



Obrázek B. 1: Slunečno mapa teploty.

Aladin - srážky (PNG)



Obrázek B.2: Aladin mapa srážek.