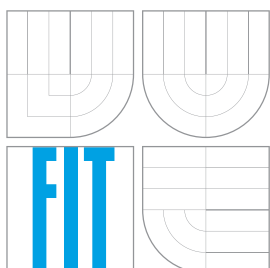


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

PREDIKCE DEŠTĚ Z METEORADARU

PREDICTION OF RAINING BY METEORADAR

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MICHAEL VLČEK

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. IGOR SZÓKE, Ph.D.

BRNO 2015

Abstrakt

Tato práce se zabývá předpovědí počasí s využitím meteoradarových snímků a některých dalších souvisejících faktorů prostřednictvím výpočetního modelu neuronové sítě. Klade si za cíl prozkoumat možnosti predikce pomocí tohoto modelu a experimentálně stanovit co nejúspěšnější konfiguraci modelu pro vykonávání vybrané činnosti.

Abstract

This thesis deals with rain prediction using information from meteoradar images and some other relevant factors through the computational model of a neural network. It focuses on exploring different prediction possibilities using this model and defining the most successful model configuration to fulfill the chosen task.

Klíčová slova

Předpověď deště, počasí, meteoradar, radarové snímky, zpracování obrazu, neuron, umělé neuronové sítě, predikční model, dopředná neuronová síť se zpětným šířením chyby, učící algoritmy

Keywords

Rain prediction, weather, meteoradar, radar images, image processing, neuron, artificial neural networks, prediction model, feed-forward backpropagation neural network, learning algorithms

Citace

Michael Vlček: Predikce deště z meteoradaru, bakalářská práce, Brno, FIT VUT v Brně, 2015

Predikce deště z meteoradaru

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Igora Szóke, Ph.D. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Michael Vlček
19. května 2015

Poděkování

Tímto děkuji za vstřícnost, trpělivost a hlavně návrhy a poskytnutí prostředků potřebných pro zpracování bakalářské práce panu Ing. Igorovi Szóke, Ph.D. Zároveň chci poděkovat panu Ing. Lukáši Burgetovi, Ph.D., a panu Ing. Janu Pešánovi, za jejich trpělivost a velmi cenné rady týkající se knihovny theano a obecně neuronových sítí, které pomohly tuto práci uskutečnit.

© Michael Vlček, 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

1	Úvod	3
2	Problematika předpovědi z meteorradaru	4
2.1	Radarové snímky	4
2.2	Možnosti predikce	5
2.3	Zpracování obrazu - DCT transformace	6
3	Umělé neuronové sítě	8
3.1	Historický kontext	8
3.2	Princip neuronových sítí	9
3.2.1	Neuron	9
3.2.2	Neuronová síť	12
3.3	Učení neuronové sítě	13
3.3.1	Obecný popis učení	13
3.3.2	Crossvalidace	14
3.3.3	Další parametry učení	15
3.3.4	Vícevrstvá perceptronová síť jako klasifikátor	15
4	Zpracování dat a implementace	17
4.1	Formát a zpracování radarových snímků	17
4.2	Extrakce snímkových dat	18
4.3	Tvorba trénovacích množin	19
4.4	SoftmaxBox	20
4.5	Metody ověření úspěšnosti	21
5	Experimentální fáze	23
5.1	Stále a modifikované parametry sítě	23
5.2	Vliv jednotlivých faktorů na úspěšnost předpovědi	24
5.2.1	Velikost vstupního vektoru	25
5.2.2	Počet klasifikovaných tříd	26
5.2.3	Rozsah předpovědi	27
5.2.4	Vliv DCT transformace	28
5.3	Porovnání s ostatními pracemi	28
6	Závěr	31
A	Obsah DVD	35
B	Manual	36

Kapitola 1

Úvod

Předpověď počasí je přirozeným předmětem lidského bádání již od pravěku. Detekce budoucích změn prostřednictvím různých smyslových vjemů je neocenitelnou vlastností členů drtivé většiny suchozemských živočišných druhů, a na včasném zpozorování extrémních změn často závisí jejich přežití. Mnoho druhů napříč faunou (ale i florou) přizpůsobuje své chování nadcházejícím změnám, z čehož jsou znalejší lidé schopni pozorováním vydedukovat, jaké je čeká počasí. Kupříkladu snížení tlaku před deštěm způsobí, že se hmyz koncentruje v nižších polohách, což často pocítujeme častějším bodáním komárů a jiného hmyzu. Obecně známé tvrzení, že vlaštovky létají níže před deštěm, je až důsledkem přesunu hmyzu jakožto jejich potravy blíže k zemi.

Smyslové rozpoznávání příchozích změn nás provází i dnes, a ačkoliv je velmi cennou schopností pro jedince, společnost zejména v posledním století vyžaduje přesnější a dlouhodobější predikce. S příchodem meteoradarů lze sledovat vývoj mnoha parametrů vývoje počasí na velkém území, a to hlavně co se týče srážek. Radarové snímky velmi dobře přibližují aktuální počasí dostatečně abstraktním způsobem. Z posloupnosti radarových snímků pořízených v krátkém časovém intervalu potom lze vysledovat přibližný vzor chování daného srážkového mraku různými metodami a učinit předpověď.

Tato práce se zabývá předpovědí deště na základě zmiňované posloupnosti radarových snímků z minulosti. Tento problém je zde řešen pomocí výpočetního modelu umělé neuronové sítě, která je díky svým vlastnostem aplikovatelná téměř ve všech odvětvích lidské činnosti. Cílem je vytvořit aplikaci, která je schopná úspěšné krátkodobé predikce a důkladně ověřit její úspěšnost. Problematika předpovědi z meteorradaru včetně metody zpracování obrazu je řešena v kapitole 2. Bližší informace o neuronových sítích najdete v kapitole 3. Veškeré informace o vlastní implementaci, způsobu extrakce a přípravy dat pro využití v neuronové síti a metodách ověření úspěšnosti jsou v kapitole 4. Kapitola 5 pak tvoří jádro celé práce a zabývá se experimentálním ověřením způsobilosti neuronových sítí pro účel předpovědi srážek, zkoumá vliv jednotlivých faktorů na úspěšnost předpovědi, a nakonec poskytuje porovnání s ostatními, podobnými pracemi. Poslední kapitola zhrnuje dosažené výsledky a popisuje možná rozšíření této práce.

Kapitola 2

Problematika předpovědi z meteoradaru

Tato kapitola shrnuje teoretické základy potřebné k pochopení principů tvorby radarových snímků, jejich interpretace, a metody využití pro následovné zpracování snímku do formátu vyhovujícího pro tvorbu trénovacích množin.

2.1 Radarové snímky

Nejprve si krátce vysvětlíme, jakým způsobem nám radarové snímky poskytují informace o srážkách. Pojem radar vznikl z termínu “*Radio Detection and Ranging*” a jedná se o elektronické zařízení, které slouží k identifikaci, zaměření a určení vzdálenosti objektů pomocí krátkých elektromagnetických vln. V případě meteoradaru je prostřednictvím antény vyslán soustředěný vysokofrekvenční paprsek o vlnové délce 1-10 cm přibližně po jednu mikrosekundu. Zmíněná vlnová délka je zvolena z toho důvodu, že při odpovídajících frekvencích u částic, které chceme pozorovat (kapky, vločky, kroupy, aj.), dochází k jevu zvanému *Rayleighův rozptyl*. Tento jev způsobuje, že se část energie vyslaného paprsku odrazí (přesněji řečeno zpětně rozptýlí) od těchto částic zpátky směrem k meteoradaru, přičemž intenzita této energie závisí na počtu, velikosti a typu srážkových částic [3]. Po vyslání každého paprsku stanice slouží jako přijímač, přičemž naslouchá odraženým částem signálu, které v předešlém paprsku vyslal. Čas potřebný pro zachycení těchto signálů je přibližně 1 milisekunda, což je dáno dobou, za kterou je vyslaný paprsek schopen urazit cestu z radaru na hranici jeho dosahu. Tato vzdálenost se pohybuje v řádu stovek kilometrů (maximální dosah meteoradaru Praha v Brdech je 256 km [13]). Navracené signály z každého paprsku se poté analyzují podle jejich intenzity (*míra reflektivity*) a směru, aby mohly být použity pro sestavení snímku. Avšak vzhledem k tomu, že radary nejsou schopny pozorovat velikost kapek, nelze použít *teoretickou definici reflektivity*. Zjednodušená definice, kterou využívají meteoradary v Česku, zní:

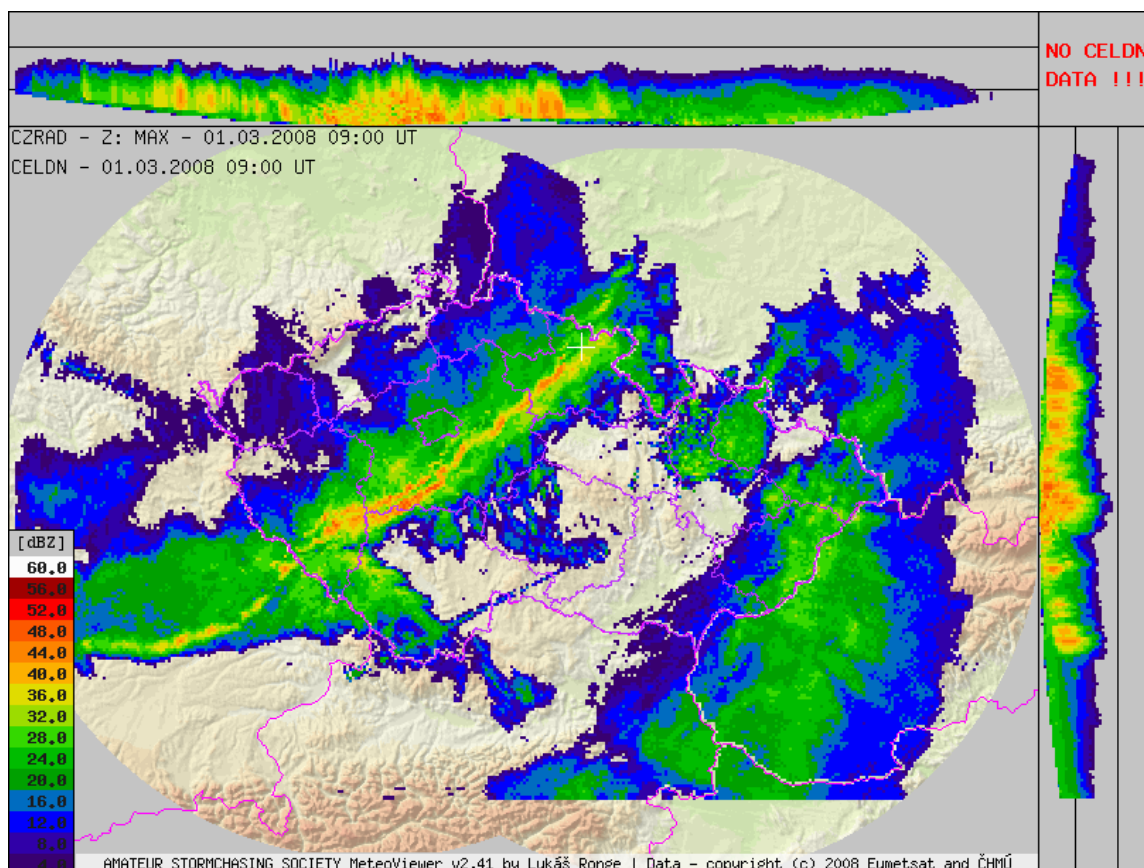
$$Z = \frac{P_r}{P_l} R^2, \quad (2.1)$$

kde P_r je přijatý výkon, P_l je meteorologický potenciál radaru (přístrojová konstanta) a R je vzdálenost cíle. Jednotkou reflektivity je mm^6m^{-3} , přičemž se častěji užívá její vyjádření na logaritmické škále v jednotkách dBZ . Tato funkce se velmi často používá k odhadu intenzity srážek pomocí *Marshall-Palmerova* vzorce:

$$Z = aR^b, \quad (2.2)$$

kde a, b jsou konstanty a R je intenzita srážek. Pro R v mm/h a Z v mm^6m^{-3} byly vyhovujícími, experimentálně zjištěnými hodnotami určeny hodnoty $a = 200, b = 1.6$. [5]

Z takto nashromážděných hodnot odrazivosti je pak sestaven samotný radarový snímek. Toto sestavení se však musí potýkat s velkým množstvím rušivých elementů. Se zvyšující vzdáleností od meteorradaru se navíc zhoršuje přesnost měření, kvůli stále snižující se intenzitě zpětně odražených paprsků. Po zohlednění těchto faktů a co největší eliminaci rušivých elementů je již snímek ve finální podobě (obr. 2.1).



Obrázek 2.1: Radarový snímek z přechodu fronty nad Trutnovem z 1. 3. 2008, 10:00 SEČ, převzato z [12]

2.2 Možnosti predikce

Jak již bylo řečeno v úvodu, je možné z posloupnosti radarových snímků s dostatečně krátkým intervalem mezi pořízením jednotlivých snímků a zobrazeným viditelným průběhem deště vysledovat vzor chování srážkového mraku a učinit předpověď. Pro tento účel existuje několik postupů, které vykazují různé úspěšné předpovědi počasí. V současné době jsou po celém světě používány velmi sofistikované a robustní systémy se schopností velmi detailní studie možností vývoje počasí na základě velkého množství faktorů. Implementace těchto systémů je však utajená. Tato bakalářská práce se zaměřuje na řešení pomocí neuronových sítí, které je věnována kapitola 3.

2.3 Zpracování obrazu - DCT transformace

Aby bylo možné data ze snímků použít, musí být nejdříve zpracována. Konkrétní postup při zpracování jednotlivých snímků bude popsán v kapitole 4. kapitole, je ovšem nutné alespoň krátce uvést metodu použitou pro kompresi a převod spojitého vstupu na vstup s reálnými koeficienty, který je pro účely trénování neuronové sítě (kapitola 3.3) vhodnější. Jedná se o DCT transformaci.

DCT transformace (*discrete cosine transform*) je technika převodu signálu na elementární frekvenční komponenty [8]. Jinými slovy pro účely zpracování obrazu spočívá princip této transformace ve vyjádření jednotlivých malých částí obrazu (obvykle se jedná o bloky velikosti 4x4 až 32x32 pixelů při JPEG kompresi) ve formě sumy sinusoid s různými frekvencemi a amplitudami. Tato metoda je například použita při kompresi v obrázcích formátu JPEG. Existuje 8 variant této metody, z nichž jsou často používány pouze 4, zde bude později popsána pouze její varianta *DCT-II*, jelikož je právě tato využívána pro tvorbu trénovacích množin. DCT se v mnoha ohledech podobá *diskrétní Fourierově transformaci*, pracuje však pouze s reálnými čísly. Další využitelnou metodou je *KL transformace* (*Karhunen-Loève transform*), která je ovšem výpočetně mnohem náročnější, přičemž DCT má podobné vlastnosti při zachování nižší výpočetní náročnosti, a to hlavně co se týče zahuštění energie (přesněji řečeno soustředění nejdůležitějších komponent obrazu do méně koeficientů než u původního obrazu), což je právě díky *psychovizuální redundanci*¹ velmi žádoucí pro obrazovou kompresi. Rovnice jednodimenzionální DCT transformace k -tého vzorku bloku X o N vzorcích vypadá takto:

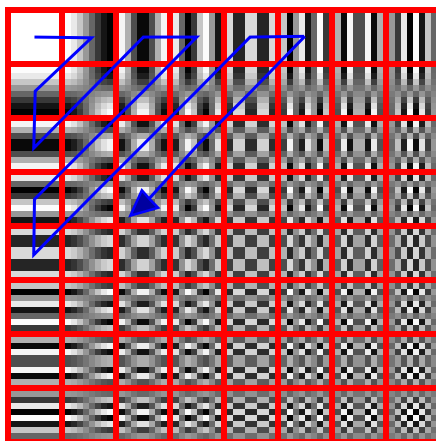
$$X_k = \sum_{n=0}^{N-1} x_n \cos \left[\frac{\pi}{N} \left(n + \frac{1}{2} \right) k \right] \quad k = 0, \dots, N - 1. \quad (2.3)$$

Z rovnice lze tedy vyčíst, že stejně jako DFT operuje DCT nad konečnou posloupností vzorků a jejím výstupem je stejně velký vektor, jako byl vektor vstupní. Pro účely této práce je však nutné upravit tuto rovnici na její dvoudimenzionální tvar, jelikož potřebujeme zpracovat obrazový vstup, čehož dosáhneme kompozicí dvou jednorozměrných rovnic (*row-column algoritmus*):

$$X_{k_1, k_2} = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x_{n_1, n_2} \cos \left[\frac{\pi}{N_1} \left(n_1 + \frac{1}{2} \right) k_1 \right] \cos \left[\frac{\pi}{N_2} \left(n_2 + \frac{1}{2} \right) k_2 \right] \quad (2.4)$$

Tímto je získána transformovaná matice bloku obrazu, z níž lze inverzní transformací získat původní blok, tudíž ještě nedošlo k žádné ztrátě obrazových dat. Klíčovou vlastností však je, že jsou hodnoty uspořádány podle frekvence tak, jak je ilustrováno na obrázku 2.2. Čím více v transformované matici postupujeme do vyšších indexů, tím vyšší frekvence je v daných hodnotách zachycena a tím méně jsou tyto hodnoty podstatné pro zpětné vykreslení obrazu. V důsledku je tedy možné tyto hodnoty naprosto zanedbat bez podstatného narušení obrazu, a tím významně snížit výslednou velikost dat při zachování podstatných rysů bloků, čehož lze využít využít i pro účely neuronové sítě, jak bude popsáno později.

¹Jedná se o část informace, jejíž nepřítomnost vlivem nedokonalosti oka nepostřehneme.



Obrázek 2.2: Dvourozměrná matice s jednotlivými koeficienty různých frekvencí. Korelace bloku s těmito koeficienty je výstupem transformace (jednotlivé hodnoty mohou být i záporné). Důležitost koeficientů je tím vyšší, čím je nižší frekvence, kterou představují. Modrá šipka ukazuje koeficienty od nejnižší frekvence vlevo nahoře dále.

Kapitola 3

Umělé neuronové sítě

Umělé neuronové sítě jsou v této práci zvoleny jako výpočetní model pro provádění predikce. V této kapitole jsou popsány teoretické základy neuronových sítí, jak co se týče krátké zmínky historického vývoje, tak hlavně i principů, na kterých jsou neuronové sítě založeny. Je zde vysvětlena podstata umělého neuronu, analogie jeho struktury se strukturou biologického neuronu, koncept vytváření neuronové sítě z umělých neuronů a způsob jejího učení. V závěru kapitoly je zběžně popsán konkrétní typ neuronové sítě využívaný v bakalářské práci.

3.1 Historický kontext

Ač se metodologie neuronových sítí prosazuje relativně krátce, její kořeny sahají až do 40. let minulého století. V roce 1943 Warren McCulloch (neurofyziolog) a Walter Pitts (matematik) zveřejnili první práci zabývající se popisem způsobu fungování neuronů v mozku. Aby vyobrazili svou představu co nejlépe, sestavili model pomocí elektrických obvodů ilustrující prostou neuronovou síť, jejíž popis (orientované grafy s ohodnocenými hranami) se stále využívá pro neuronové sítě. Tito pánové ve svém pojednání o chování nervové soustavy živých organismů tedy nejen podali vysvětlení příčin tzv. *podmíněných reflexů*, ale také zavedli *normu grafického znázorňování struktury umělých neuronových sítí*. Na tuto práci navázal Donald Hebb v roce 1949, když ve své práci poukázal na fakt, že *synapse* mezi neurony jsou posíleny pokaždé, když jsou použity, což je naprosto zásadní koncept pro způsob, jakým je člověk schopen se učit. Další milník nastal až v roce 1959, kdy Bernard Widrow a Marcian Hoff ze Standfordské univerzity vyvinuli modely *ADALINE* a později *MADALINE* (*Multiple ADaptive LINear Elements*), z nichž druhý jmenovaný byl první neuronovou sítí využitou pro řešení reálného problému. *MADALINE* byl postaven na principu *třívrstvé feed-forward sítě* s jednou skrytou vrstvou, a využívá se dodnes pro filtraci ozvěny na telefonních linkách.

Ač měly neuronové sítě zpočátku úspěch, jejich rozvoj zastínila *von Neumannova architektura*. Navíc příslib úžasných nových technologií z akademických řad založených na neuronových sítích zůstal nevyplněn kvůli velmi omezeným možnostem tehdejší výpočetní techniky. Výzkum čelil kritice a většina jeho financování se v důsledku přesunula jinam. Toto období trvalo až do roku 1982, kdy byl vývoj znovu obnoven, a od té doby věnujeme neuronovým sítím čím dál větší pozornost. Tak jako v minulosti je stále v cestě omezení výkonu výpočetní techniky, i když se tento problém postupem času odbourává. V současnosti se neuronové sítě používají v mnoha odvětvích informačních technologií. V tomto

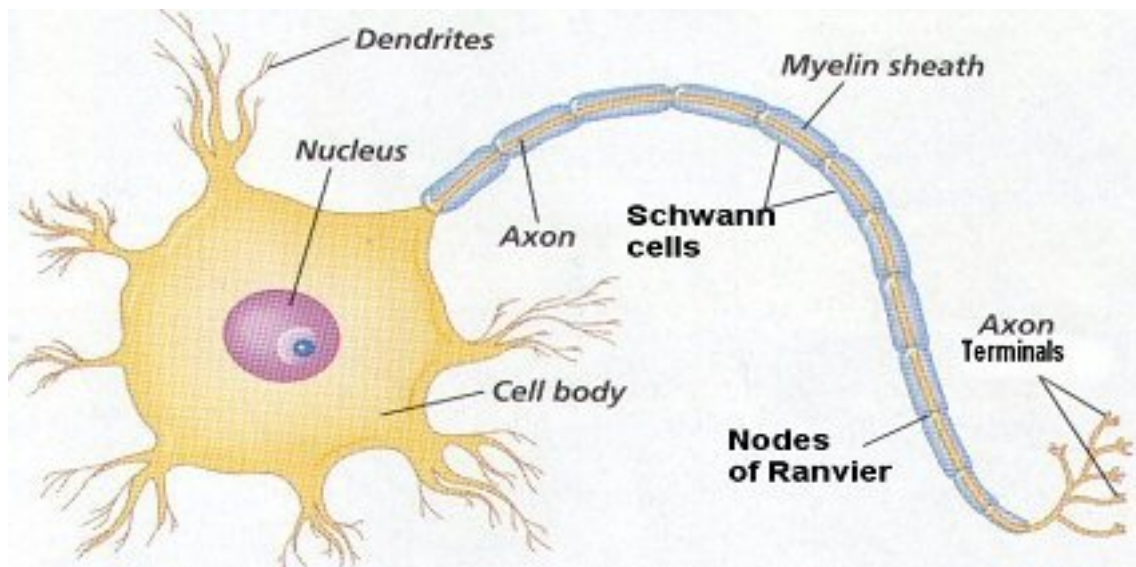
ohledu lze zmínit vysoké investice do výzkumu od společností *Google* (např. pro optimalizaci obrovských datových center, rozpoznání objektů v obraze, ...), *IBM* (vývoj nové architektury založené na neurosynaptických jádrech, diametrálně odlišné od von Neumannovské) nebo nezávislý *Human Brain Project* (výzkum simulace lidského mozku, na kterém má spolupracovat až 135 institucí v rámci celé Evropy [4]). Obecně se využívají například pro rozpoznávání psaného textu, predikce sázek a sázkových kurzů, vývoje finančního trhu a v neposlední řadě také při předpovědi počasí.

3.2 Princip neuronových sítí

Umělý neuron představuje základní prvek umělé neuronové sítě. Je do značné míry inspirován neurony v mozku, hlavně co se týče struktury, a zakládá se i na stejných principech z hlediska způsobu přenosu vstupního a výstupního signálu. Tato kapitola se zabývá objasněním funkce a struktury obou typů neuronů a definicí neuronových sítí včetně jejich základní klasifikace.

3.2.1 Neuron

Neuronové sítě jsou inspirovány svým biologickým vzorem v mozku, jak lze vydedukovat z historického vývoje v předchozí kapitole. Abychom pochopili, jak taková umělá neuronová síť vlastně vznikla a funguje, je potřeba popsat alespoň zběžně prvky jejího biologického ekvivalentu. Základním stavebním kamenem mozku, jako centra nervové soustavy, je *neuron*. Neuron můžeme definovat jako funkční a anatomickou jednotku nervového systému [11]. Jeho struktura, z níž vycházejí i umělé neurony, se skládá ze *somy* (tělo buňky podmiňující vznik a šíření vzruchu), *dendritů* (krátké dostředivé výběžky přijímající nervový vzruch jako vstupní informaci), a *axonu* (dlouhý výběžek, na svém konci se bohatě větví, typicky vysílá informaci k jiným neuronům). Neurony si mezi sebou předávají informaci posíláním elektrických nebo chemických impulsů prostřednictvím takzvaných *synapsí*, což jsou struktury tvořící hranici mezi koncovými výběžky axonu vysílajícího neuronu a dendrity přijímajícího neuronu. Pokud neuron přijme dostatečné množství energie z příchozích impulsů, tělo nervové buňky rozhodne o vystřelení vlastního impulsu po axonu, který je propagován synapsemi do dalších neuronů. Strukturu neuronu můžeme vidět na obrázku 3.1.



Obrázek 3.1: Struktura neuronu, převzato z [1]

Umělé neuronové sítě představují způsob, jak napodobit schopnost učení mozku tak, aby z jeho vlastností mohly těžit i stroje, a patří tedy mezi modely *strojového učení* (*machine learning*). Je ovšem velmi důležité poznamenat, že ač se umělé neuronové sítě do jisté míry inspiřují svou biologickou předlohou, jsou k imitaci fungování mozku stále velmi daleko. Z fyzického neuronu se umělý neuron inspiřoval třemi hlavními vlastnostmi:

- *Synapse a synaptické váhy* - některé synapse mají větší vliv na výsledný impuls, než jiné
- *Souhrnný podnět* - suma impulsů ze všech dendritů rozhoduje o propagaci impulsu z daného neuronu dále
- *Binární rozhodování* - buď je impuls vyslán, nebo ne, žádná míra pro vyslání signálu neexistuje (platí pouze pro skokové aktivační funkce, viz dále)

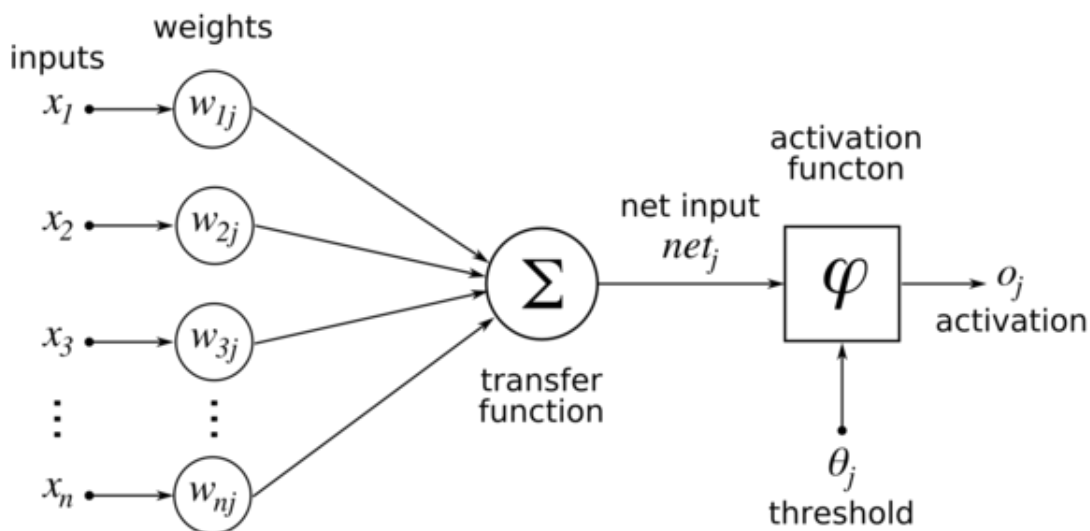
Přejdeme tedy k samotné struktuře umělého neuronu. Jednoduchý model umělého neuronu se skládá z několika hlavních součástí: *vektoru vstupních hodnot* (tyto vstupní hodnoty představují přichodící impulsy z dendritů), *vektoru vah* (které určují schopnost jednotlivých vstupů ovlivnit výstupní hodnotu, viz. sekce 3.3), *agregační funkce* (která provádí součet nad váženými vstupními hodnotami), *prahové hodnoty* (typicky je zastoupena dalším vstupem o hodnotě -1 a proměnlivé váze) a *přenosové funkce* (určující transformaci výstupu agregační funkce pro propagaci výstupní hodnoty ven z neuronu). Obecný tvar *výstupní funkce výpočtu hodnoty excitace* (vzruchu) y_j pro j -tý neuron vypadá následovně:

$$y_j = \varphi\left(\sum_{k=0}^n x_{jk} w_{jk} + \theta_j\right) \quad (3.1)$$

kde:

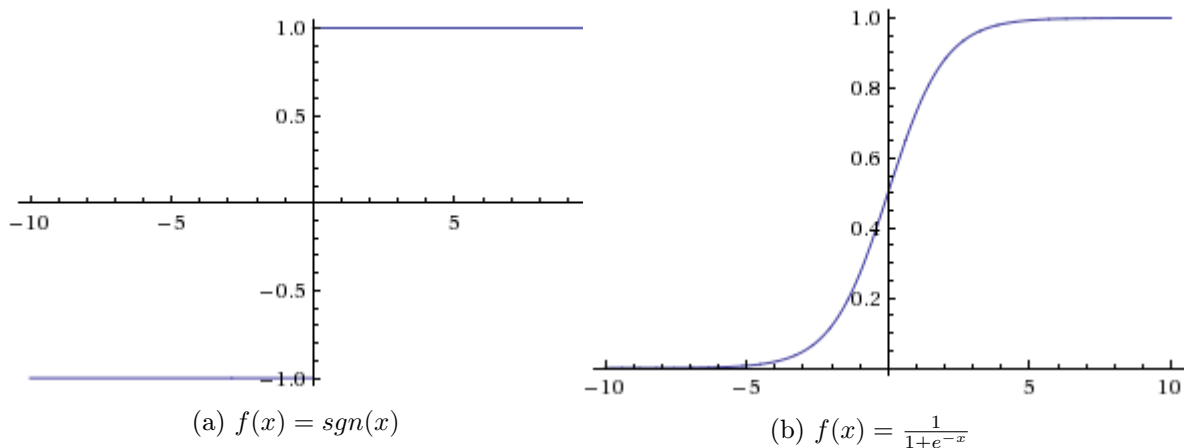
- φ je přenosová (aktivační) funkce,
- $\sum_{k=0}^n x_{jk} w_{jk}$ je agregační funkce,

- x_{j_k} je k -tá hodnota z vektoru vstupních hodnot pro daný neuron j ,
- w_{j_k} je k -tá hodnota z vektoru vah pro daný neuron j ,
- θ_j je prahová hodnota pro daný neuron j ,
- n je celkový počet hodnot ve vstupním vektoru.



Obrázek 3.2: Struktura umělého neuronu, převzato z [9]

V případě modelu s jediným vstupem by se jednalo jen o skalární vstupní hodnotu a váhovou hodnotu, agregační funkce by mohla být zanedbána. *Blokové schéma umělého neuronu* je na obrázku 3.2. Existuje několik typů transformačních funkcí využitelných v praxi, z nichž zde blíže popíšu pouze dvě základní. Historicky prvním typem těchto funkcí jsou *skokové aktivační funkce*, které nejpřesněji odpovídají modelu skutečného neuronu, protože jejich výstup je definován pouze nad dvěma hodnotami (obvykle 0 a 1 nebo -1 a 1). Pro vstupy menší než je daná mez vrací obvykle nekladnou hodnotu, pro ostatní kladnou. Příkladem takové funkce může být například $sgn(x)$ na obrázku 3.3a. Neuronu klasifikující výstupy pouze na dvě možné hodnoty se nazývají *perceptrony*. Druhým typem je takzvaná *sigmoidální aktivační funkce*. Její hlavní výhodou je, že má derivaci v celém oboru reálných čísel a zároveň jsou její derivace jednoduché, proto je vhodnější pro užití v neuronových sítích než většina podobných funkcí (čistě z matematického a potažmo optimalizačního hlediska). Její tvar a průběh je na obrázku 3.3b.

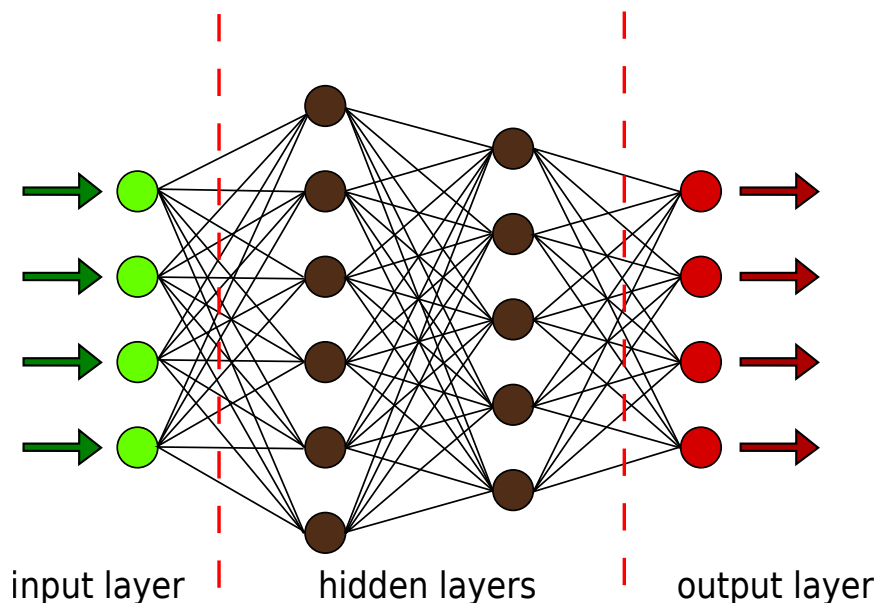


Obrázek 3.3: Průběhy obou aktivačních funkcí

3.2.2 Neuronová síť

Za umělou neuronovou síť se obecně považuje taková struktura pro distribuované paralelní zpracování dat, která se skládá z jistého, obvykle velmi vysokého, počtu vzájemně propojených neuronů. Každý z nich může současně přijímat libovolný konečný počet různých vstupních dat. Na další neurony může předávat libovolný konečný počet svého jediného, avšak obvykle velmi rozvětveného výstupu. Každý neuron transformuje vstupní data na výstupní podle jisté přenosové funkce. Přitom se též může uplatnit obsah jeho lokální paměti. [11]

Pro vytvoření jednoduché neuronové sítě stačí propojit několik neuronů dohromady. Abstraktní popis těchto struktur je prováděn pomocí orientovaných grafů s ohodnocenými hranami. Tyto umělé neuronové sítě mohou být realizovány jak v podobě hardwaru, kde neuron představuje komponentu počítače, nebo softwaru. Mohou být založeny na různých topologiích a učících algoritmech. Drtivá většina sítí je však založena na *vrstevnaté topologii*. Síť je rozdělena do vrstev, kde výstupy jedné vrstvy jsou vstupy vrstvy následující. Vstupy první (vstupní) vrstvy a výstupy poslední (výstupní) vrstvy tvoří *rozhraní neuronové sítě*, všechny vrstvy mezi vstupní a výstupní vrstvou se označují jako *skryté vrstvy*. Počet vrstev a počet neuronů v jednotlivých vrstvách tvoří tzv. *konfiguraci sítě*. Příklad sítě s vrstevnatou topologií je na obrázku 3.4.



Obrázek 3.4: Neuronová síť s vrstevnatou topologií a dvěma skrytými vrstvami. Informace se ve feed-forward neuronové síti přenáší pouze směrem ze vstupní vrstvy do vrstvy výstupní.

Hlavní rozdělení neuronových sítí určuje způsob přenášení informací. V tomto směru rozdělujeme neuronové sítě na *dopředné (feed-forward)* a *rekurentní (recurrent)*. Ve feed-forward sítích se informace šíří pouze jedním směrem, a to ze vstupní vrstvy přes případné skryté vrstvy až do vrstvy výstupní. Naproti tomu v rekurentních neuronových sítích jsou vazby mezi neurony tvořeny *cyklickými vazbami*, což vynucuje *uchovávání stavu* neuronové sítě. Signál se nešíří pouze od vstupní vrstvy směrem k výstupní, ale prostřednictvím *rekurentních neuronů* dochází i k zpětnovazebnímu přenosu informace zpět z vyšších vrstev do nižších. V tomto textu se ale důkladněji věnuji jen specifické formě dopředné neuronové sítě, protože je použita pro řešení problému předpovědi počasí.

3.3 Učení neuronové sítě

Učení upravuje neuronovou síť skrze její parametry podle zpracovávaného vstupu tak, aby její výstup co nejvíce odpovídal požadovanému výstupu, a tím plnila účel, pro který je trénována. Vstupní vektor a požadovaný výstup tvoří dohromady *trénovací množinu*. Tyto trénovací množiny se dále dělí na *trénovací část* a *crossvalidační část*. Trénovací část slouží k samotnému trénování prostřednictvím modifikace vah a prahů, zatímco crossvalidační část pomáhá ověřit při *crossvalidaci*, zda nedošlo k *přetrénování* neuronové sítě. Průchod obou částí kolekce trénovacích množin se nazývá *epocha*.

3.3.1 Obecný popis učení

Učení neuronové sítě je proces, při kterém síť *modifikuje váhy a prahy* u vstupů jednotlivých neuronů takovým způsobem, aby se výsledná výstupní funkce co nejvíce blížila požadovanému tvaru [10]. Tento požadovaný tvar není nijak předem definovaný, protože buď není známý, nebo není vhodný pro implementaci. *Cílem procesu učení* je tedy v podstatě aproximovat funkci, která co nejlépe vystihuje požadované chování (tedy poskytuje pro

dané vstupy odpovídající požadované výstupy). Způsob, jakým učení probíhá, je jedním z nejdůležitějších klasifikátorů neuronových sítí. Hlavním rozlišením je vlastnost, zda se jedná o neuronovou síť s učitelem (*supervised learning*) nebo bez učitele (*unsupervised learning*). Vzápětí si je blíže popíšeme.

Typicky se vstupy neuronu zpracovávají tak, že se provede vážený součet všech vstupních hodnot, přičte se práh a na výsledek se aplikuje aktivační funkce, která rozhodne o výstupu neuronu (viz. podkapitola 3.2.1). U neuronových sítí s učitelem je však požadavkem pro učení sítě i další vstup, a tím je očekávaný výsledek. Spolu s vstupním vektorem tvoří očekávaný výsledek *trénovací množinu*. Tyto trénovací množiny pochopitelně nabývají takové podoby, aby poskytovaly veškerá data potřebná pro trénování v jednotném a zpravidla dobře strojově zpracovatelném formátu. Počet takových trénovacích množin by měl mnohonásobně přesáhnout počet neuronů v síti, aby došlo k dostatečnému natrénování sítě. Tyto trénovací množiny je však nutné ještě dále rozdělit na dvě části, a to trénovací část a crossvalidační část. Úsek, ve kterém dojde k jedné iteraci zpracování všech trénovacích množin jak z trénovací, tak crossvalidační části, je označován jako *epocha*. Jak zapadá crossvalidace do procesu trénování je popsáno v podkapitole 3.3.2.

Konkrétní algoritmus, který do kategorie učení neuronové sítě spadá, a bude využit pro účely této práce, se nazývá *backpropagation algoritmus*, neboli *algoritmus zpětného šíření chyby*. Pomocí tohoto algoritmu se pro minimalizaci *chybové (energetické) funkce* používají *gradientní metody*. Nejčastěji je chybová funkce $E(y, d)$ definována jako *kvadratická chyba mezi požadovaným a skutečným výstupem*:

$$E(\vec{y}, \vec{d}) = -\frac{1}{2}(\vec{d} - \vec{y})^2 \quad (3.2)$$

kde: \vec{y} je výstupní vektor neuronové sítě a \vec{d} je požadovaný výstupní vektor. Druhou častou formou chybové funkce je cross entropy error, který vypadá takto:

$$E(\vec{y}, \vec{d}) = -\sum_{k=0}^n (\ln(d_k) - y_k) \quad (3.3)$$

kde: d_k je hodnota v požadovaném výstupním vektoru a y_k je hodnota výstupního vektoru neuronové sítě. Obě tyto funkce jsou použitelné a volba mezi těmito algoritmy proběhne na základě experimentů.

3.3.2 Crossvalidace

Při trénování se může stát, že se síť začne místo zobecňování zaměřovat přímo na danou kolekci trénovacích množin, což by neblaze ovlivnilo výslednou natrénovanou neuronovou síť. Tento jev je důsledkem syndromu *přetrénování (overtraining)*, který vzniká při předložení příliš velkého počtu opakujících se trénovacích množin. I když se výsledky trénování postupně zlepšují vzhledem k stále důkladnějšímu zapamatování detailních rysů v trénovacích množinách (jelikož jsou v každé epoše použity stejné trénovací množiny), ztrácejí se informace o obecných rysech, což může značně ovlivnit kvalitu předpovědi výsledné neuronové sítě. Proto je nutno provést ověření, zda došlo k syndromu přetrénování, a případně opravit váhy a prahy na předchozí hodnoty. Toto ověření se označuje jako *crossvalidace*.

Pro crossvalidaci je před samotným trénováním vyhrazeno určité množství trénovacích množin, které není použito pro žádné jiné účely při trénování. Jakmile jsou v dané epoše vyčerpány všechny trénovací množiny z trénovací části, crossvalidace vyhodnotí natrénování neuronové sítě zpracováním všech trénovacích množin z crossvalidační části, a porovná

výsledek s výsledkem z předchozí epochy. Pokud došlo ke zlepšení, nově natrénované parametry jsou ponechány, uloženy pro příští crossvalidaci a trénování pokračuje další epochou. Pokud ke zlepšení nedošlo, je proveden rollback k parametrům z předchozí epochy, což obvykle provází buď snížení learning rate (česky faktor učení, podkapitola 3.3.3), nebo ukončení trénování.

3.3.3 Další parametry učení

I po zvolení učícího algoritmu a chybové funkce je stále několik faktorů, které přímo ovlivňují jak přesnost, tak rychlost učení. Dvěma z těchto faktorů jsou *počet neuronů ve skrytých vrstvách* a *počet skrytých vrstev*. Celkový počet vrstev značně souvisí se složitostí řešeného problému. Například často zmiňovaným mezníkem schopnosti *jednovrstevné perceptronové sítě* je výpočet logické funkce *XOR*, což implikuje, že jsou tyto sítě použitelné pouze na množiny, které jsou lineárně separovatelné. Matematicky byl tento fakt dokázán v práci M. Minského a S. Paperta, oba však svůj správný poznatek neoprávněně zobecnili na neuronové sítě vůbec. Jak se dnes s odstupem let často konstatuje, nesou díky tomu svůj díl viny na pozdržení vývoje neuronových sítí, které nastalo od konce šedesátých až do poloviny let osmdesátých [11]. Vícevrstvé sítě už jsou schopny tento problém snadno řešit. Počet neuronů v jednotlivých skrytých vrstvách oproti tomu umožňuje roztrždit vstupy z předchozí vrstvy do ekvivalentního počtu vzorů, tudíž čím více neuronů v dané vrstvě je, tím lépe od sebe dokáže rozpoznat jednotlivé vzory. Je důležité poznamenat, že vstupní vektory jsou totožné pro všechny neurony ve skrytých vrstvách a výstupní vrstvě. Toto však znamená, že před prvotním použitím neuronové sítě je nutné nainicializovat váhy na rozdílné hodnoty v rámci jednotlivých neuronů (všechny neurony by v opačném případě rozpoznávaly stále jeden a ten samý vzor), a to zpravidla buď náhodně, nebo cíleně na základě předchozího trénování. V této práci je zvolen náhodný způsob inicializace, jelikož je značná část práce věnována experimentování s různými konfiguracemi sítí, pro které nejsou k dispozici hodnoty z předchozího trénování.

Dalším faktorem přímo ovlivňujícím kvalitu i rychlost učení je tzv. *learning rate*. Tento koeficient snižuje velikost změn vah a prahů u každého neuronu, které zavádí učící algoritmus podle vypočítané chyby oproti očekávanému výstupu neuronové sítě. Pokud je však learning rate příliš vysoký, může dojít k oscilaci a síť se nebude efektivně učit. Na druhou stranu velmi nízký learning rate učení značně zpomaluje. K nalezení optimálního learning rate parametru je nutné dojít experimentální metodou. S learning rate úzce souvisí i parametry pro jeho adaptivní snižování při blížení se optimálnímu natrénování, tedy tolerance, při jakém maximálním rozdílu v natrénování mezi dvěma epochami se má learning rate začít snižovat, a koeficient určující, o jakou hodnotu se má případně learning rate každou další epochu snižovat. Tyto parametry jsou opět zvoleny na základě experimentů.

Trénování je také nutné v určitém momentě ukončit. Ukončení může být zajištěno maximálním počtem epoch, tento faktor ovšem nebere v potaz žádnou konkrétní informaci o skutečném posunu v natrénování sítě. Oproti tomu lze však využít parametru tolerance. Trénování je ukončeno, pokud již ani půlení learning rate nesplní parametr tolerance. Po natrénování sítě jsou všechny podstatné parametry uloženy a proces trénování je ukončen.

3.3.4 Vícevrstvá perceptronová síť jako klasifikátor

Tento typ umělé neuronové sítě je zde využit pro predikci deště. Narozdíl od rekurentních sítí se v dopředné síti informace šíří pouze jedním směrem, a to od vstupní vrstvy přes skryté vrstvy do vrstvy výstupní, tak jak je ilustrováno v obrázku 3.4. Tato vlastnost jí

umožňuje využívat učící algoritmus back-propagation, jak je popsán v podkapitole 3.3.1. V této práci ji představuje síť s maximálně dvěma skrytými vrstvami, jelikož rané experimenty ukázaly, že třetí skrytá vrstva značně zhorší výsledky předpovědi.

Aby však byla síť kompletní, musí jednoznačně určit formu svého výstupu. Vzhledem k využití sítě pro klasifikaci do několika tříd (*multinomiální logistická regrese*) je použita funkce *softmax* v neuronech výstupní vrstvy. Jejím úkolem je vyjádřit výstup funkce jako posloupnost pravděpodobnostních hodnot vyjadřujících pravděpodobnosti příslušnosti k jednotlivým třídám sumujících do hodnoty 1.0. Pokud máme například nadefinovány tři výstupní třídy a síti je předložen korektní vstup, výstup je prezentován ve formě tří hodnot, které určují pravděpodobnost příslušnosti vstupu k první, druhé a třetí třídě. V ideálním případě by síť jednoznačně určila pravděpodobnost 1.0 pro jednu ze tříd a vyloučila pravděpodobnost ostatních dvou. Přibližování tomuto ideálnímu stavu je cílem trénovacího procesu.

Kapitola 4

Zpracování dat a implementace

Tato kapitola je zaměřena na vysvětlení způsobu zpracování obrázků poskytnutých fakultní databází a popisu implementace jak neuronové sítě, tak případně i dalších pomocných skriptů. Veškerý kód je napsán v jazyce *Python 2.7.3* s využitím distribuce *Anaconda 2.1.0*, a s použitím jak standardních knihoven, tak i knihovny *Theano* pro optimalizaci trénovacího procesu.

4.1 Formát a zpracování radarových snímků

Veškeré obrázky byly pořízeny organizací ČHMÚ (Český hydrometeorologický ústav), a poskytnuty k použití z fakultní databáze. Tyto snímky zaznamenávají několik konkrétních dní v časech od 00:00 do 23:50, ve kterých je zastoupena dostatečná srážková aktivita. Jednotlivé snímky jsou pořízeny s desetiminutovým intervalem. Každý snímek obsahuje data z obou českých radarů, tedy jak z radaru Brdy v Čechách, tak z radaru Skalky u Protivanova na Moravě. Příklad takového snímku je na obrázku 2.1. Prvním krokem pro vytvoření trénovacích množin je analýza formátu obrázků. Pro tyto účely je vhodné, aby byl obrázek v dobře zpracovatelném formátu, nejlépe bez komprese, aby se zkrátila extrakce dat, jelikož tvorba různých forem trénovacích množin je součástí experimentování, a je značně časově náročná. Všechny přidělené obrázky jsou v jednotném formátu, a to v *PNG palette-based* formátu. Formát PNG používá dvoustupňovou bezztrátovou kompresi (fáze *filtering* a *deflate* [7]), s přidanou třetí fází v podobě palety, která však je dobře zpracovatelná. Pro dekompresi prvních dvou fází je použita knihovna *PyPNG*. Poté, co je tato dekomprese provedena, je však stále na snímkových datech aplikovaná specifická paleta.

Paleta v počítačové grafice označuje dostupný rozsah zobrazitelných barev na monitoru počítače. Obrázky však bývají založené na paletě hlavně z důvodů komprese, a to tak, že je v metadatech vyhrazen specifický prostor pro definici použitých barev, načež se samotná data v obrázku odkazují právě na tuto paletu. PNG palety podporují barevné hloubky v rozsahu 1, 2, 4 a 8 bitů, což odpovídá maximálně 2, 4, 16 nebo 256 barvám namapovaným v paletě. Nejdříve tedy extrahujeme paletu z metadat některého z obrázků. Paleta všech obrázků byla v tomto případě stejná napříč všemi přidělenými obrázky, a její podoba je ilustrována na obrázku 4.1. Je tedy extrahována, a po oříznutí obrázku kvůli irelevantním a nepotřebným souřadnicím (v obrázku 2.1 lze jasně vidět nepotřebná data) a specifikaci cílových souřadnic pro všechny obrázky mohou být jednotlivé obrázky zpracovány.

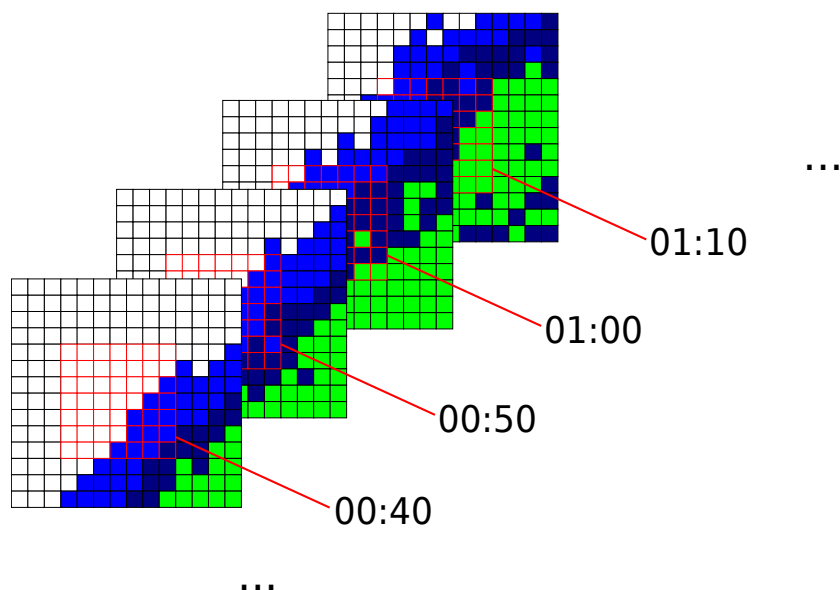


Obrázek 4.1: Zobrazení palety přidělených obrázků. Každý vertikální pruh vyobrazuje jeden index palety. Jak je vidět, velká část palety zůstává nevyužita, protože 7-bitová barevná hloubka palety není podporována.

4.2 Extrakce snímkových dat

Po získání palety, provedení dekomprese a určení cílových souřadnic je již možné začít extrahovat konkrétní data z jednotlivých snímků. Nejprve je však nutné specifikovat, jak mají tyto data být naformátována, s čímž úzce souvisí i podoba trénovací množiny. Základním předpokladem a zvoleným způsobem předpovědi je predikce v konkrétním bodě, který představuje jeden pixel. Předpověď v daném bodě je závislá na vývoji případné fronty v rámci krátké historie v okolí daného pixelu, a tedy vstupní vektory pro neuronovou síť by měly informaci o okolí pixelu obsahovat. Tudíž je potřeba pro každou cílovou souřadnici extrahovat z každého snímku hodnotu odrazivosti (reprezentovanou konkrétní barvou) v daném pixelu a jeho okolí o předem specifikované velikosti. Tato velikost je ovšem také předmětem experimentů, proto není stanovena na pevnou hodnotu a bude konkrétně určena až v kapitole 5. Při vlastním procesu extrakce je nejdříve celý obrázek dešifrován podle palety a hodnoty v jednotlivých pixelech jsou normalizovány. Přitom jsou eliminovány nežádoucí pixely nesouvisející s odrazivostí, například hranice států. Poté u každého konkrétního cílového pixelu a jeho okolí dochází nejprve k sestavení jeho okolí a následovnému zápisu do odpovídajícího souboru pro daný snímek spolu s informací o souřadnici cílového pixelu.

Pro každý obrázek je tedy vytvořený soubor, který obsahuje popis okolí všech cílových pixelů v daném snímku. Samostatný obrázek ale neobsahuje dost informací pro sestavení vstupního vektoru trénovací množiny spolehlivého predikčního modelu, protože v něm nelze vydedukovat informaci o pohybu fronty. Pro sestavení vstupního vektoru je tedy nutné propojit více za sebou jdoucích obrázků dohromady. V další fázi je proto z vytvořených souborů s extrahovanými daty sestaven soubor pro každý den, který váže okolí pixelů jednotlivých cílových souřadnic ze všech obrázků v jednom dni dohromady. Výsledný soubor pro konkrétní den tedy obsahuje vždy všechna okolí pro každou cílovou souřadnici v rámci celého dne, jak je vidět na obrázku 4.2. Z těchto souborů je již možné tvořit samotné trénovací množiny.

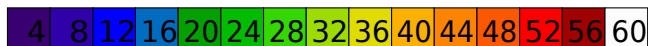


Obrázek 4.2: Tento obrázek ilustruje, jak jsou jednotlivé části snímku získány (vyjmutím určitého okolí bodu ze snímků, jak je na obrázku vyobrazeno červeně ohraničenými pixely) a seskládány dohromady do sekvencí, ze kterých jsou později tvořeny vstupní a výstupní vektory trénovacích množin. Hodnoty reflektivity vybraných okolí jsou extrahovány a konkaténovány napříč všemi obrázky z daného dne.

4.3 Tvorba trénovacích množin

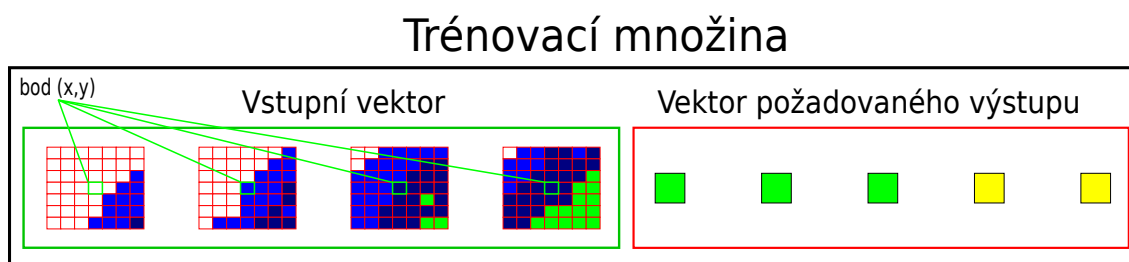
Trénovací množiny mají několik parametrů, které je nutné při jejich vytváření zohlednit. Hodnoty téměř všech z nich jsou opět předmětem experimentů, a tedy zde bude nastíněna jen jejich funkčnost. Jedním z těchto parametrů je počet po sobě jdoucích okolí, která budou využita pro tvorbu vstupního vektoru trénovací množiny. Jinými slovy se jedná o počet snímků, jejichž omezené okolí poskytneme neuronové síti pro provedení predikce. Zvolení optimálního počtu je nutné provést experimentálně. Příliš velký počet snímků by zahrnul neuronovou síť nepodstatnými informacemi a výrazně by zvětšil vstupní vektory, což by se negativně projevilo nutností úměrně zvětšit neuronovou síť se všemi důsledky s tím spojenými (např. potřeba většího množství trénovacích množin pro natrénování, což způsobí prodloužení trénování, atd.). Naopak nízký počet může výrazně snížit úspěšnost predikce.

Další dva parametry přímo souvisí s formou požadovaného výstupu neuronové sítě. Neuronová síť je používána jako klasifikátor, proto by očekávaný výstup měl mít formu vektoru tříd. Parametry pak určují, kolik takových tříd bude existovat a jakým způsobem budou namapovány na hodnoty reflektivity. Radarové snímky rozlišují celkem 15 barevně odlišených hodnot reflektivity, jak lze vidět na obrázku 4.3. Pro účely predikce není nutné pro každou předpovídanou hodnotu přidělovat specifickou třídu. Celkový počet tříd byl zvolen také experimentálně, a to tak, aby měl alespoň základní výpovědní hodnotu (tedy minimálně dvě třídy), a přitom nebyl příliš vysoký pro efektivní natrénování sítě. Mapování hodnot na třídy bylo provedeno na základě úvahy, a tedy byly hodnoty logicky rozděleny podle intenzity deště. Podle toho, jak dlouhou předpověď chceme provést, je ještě zvolena délka vstupního vektoru, a pak už nic nebrání ve tvorbě trénovacích množin.



Obrázek 4.3: Barevná škála jednotlivých tříd reflektivity, které jsou zaznamenány na radarových obrázcích v jednotkách dBZ.

Trénovací množiny jsou sestaveny ze souborů, jejichž vytvoření je popsáno v předchozí kapitole. Podle parametrů je pro každou trénovací množinu vyjmut kus časové sekvence okolí určitého bodu. Ten je rozdělen na vstupní část, která je ponechána vcelku, a výstupní část, ze které jsou vybrány pouze konkrétní body bez svého okolí a přetransformovány z hodnot reflektivity na třídy. Před samotným použitím je však ještě vstupní vektor transformován pomocí 3D-DCT transformace. Tato transformace vstupní vektor přetransformuje z 3-D matice přirozených čísel (dimenze reprezentují čas, souřadnici x a souřadnici y) na komprimovanou 3-D matici reálných čísel, která pro umělou neuronovou síť představuje lépe zpracovatelný vstup, a díky svým vlastnostem zahušťuje informaci ze vstupního vektoru do menšího prostoru, což se pozitivně projeví hlavně na velikosti vstupní vrstvy sítě. Kompresce sice způsobí pozorovatelnou ztrátu dat, zahuštění energie plynoucí z této komprese je však stále převažujícím kladným faktorem při rozhodování o volbě DCT transformace.



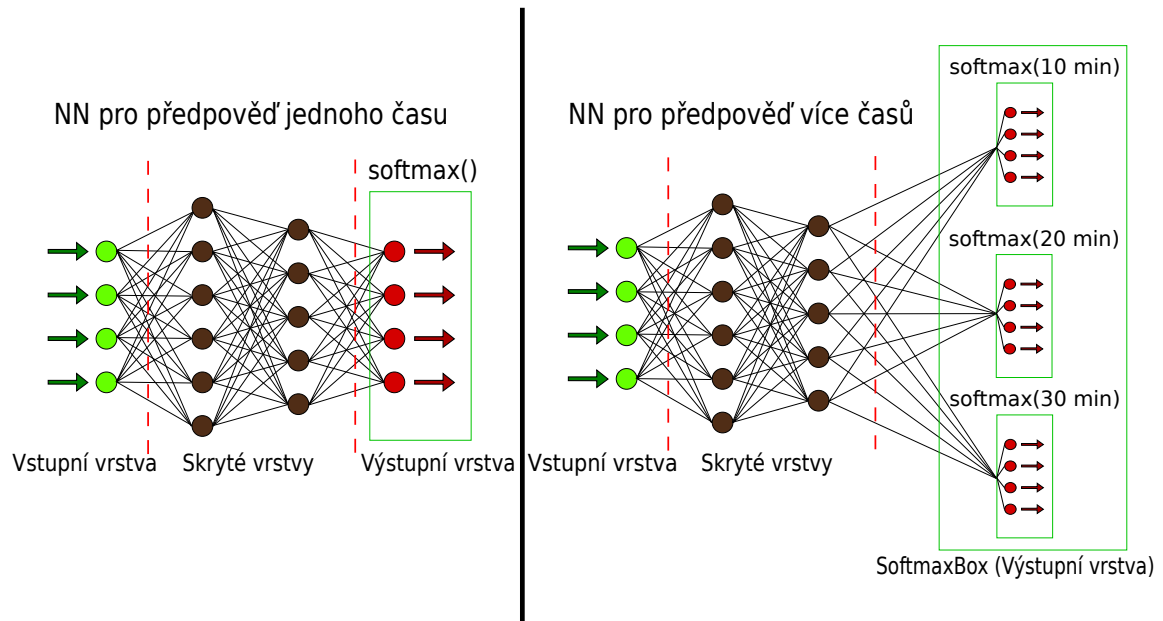
Obrázek 4.4: Příklad trénovací množiny pro vybraný bod (x,y) před DCT transformací vstupního vektoru. Její vstupní vektor se skládá z několika v čase po sobě jdoucích vektorizovaných okolí vybraného bodu. Výstupní vektor představuje několik skalárních indexů tříd, do kterých byly namapovány hodnoty pixelů vybraného bodu.

Při vytváření trénovacích množin je však nutné ještě zohlednit vhodné rozložení zastoupení jednotlivých tříd. V tomto ohledu bylo vyzkoušeno několik způsobů, nejvíce se však osvědčil faktor tolerance srážkové aktivity vstupního vektoru. Při sestavování trénovací množiny extrakcí posloupností několika okolí z celodenních sekvencí je ověřováno, jestli se v jednotlivých okolicích vyskytuje jakákoliv srážková aktivita. Pokud se na určitém počtu snímků (daném právě hodnotou faktoru tolerance) ze vstupního vektoru ve vytvářené trénovací množině nevyskytuje alespoň nějaká srážková aktivita, trénovací množina je zahozena. Kupříkladu pokud vstupní vektor obsahuje okolí z 5 po sobě jdoucích snímků, a tolerance je nastavena na 3, tak pokud ve vstupním vektoru není více než 3 srážkově aktivních okolí, skript zahodí trénovací množinu.

4.4 SoftmaxBox

Jak je již popsáno v kapitole 3.3.4, neuronová síť je využívána jako klasifikátor. V tomto ohledu ale vyvstává problém pro předpověď více časových intervalů, jelikož výstupní vrstva s jedním softmaxem poskytuje pouze jednu předpověď. Tento přístup značně omezuje využití

sítě, a vynucuje vytvoření celých sítí pro předpověď více intervalů, což je značně neefektivní způsob. Proto je výstupní vrstva ve většině experimentů rozšířena způsobem, který je ukázán na obrázku 4.5. Veškeré neurony z předchozí vrstvy jsou propojeny se všemi neurony jednotlivých softmaxů. Při porovnávání výsledků této sítě s požadovaným výstupem je pak každému softmaxu předkládán požadovaný výstup pro různý čas v budoucnosti. Tento přístup natrénuje každý softmax pro předpověď jiného časového úseku, s tím, že jsou chybové a validační funkce počítající přesnost předpovědi (popsána v další kapitole) počítané pro každý softmax zvlášť a jednotlivé výsledky uspořádány do vektoru.



Obrázek 4.5: Výstupní vrstva v síti SoftmaxBox je rozšířena, obsahuje tolik softmaxů, kolik časů má předpovídat.

4.5 Metody ověření úspěšnosti

Se sestavením sítě a připravenými daty vyvstává otázka, jak bude evaluována úspěšnost předpovědi natrénované sítě. Ať už je natrénována jakýmkoliv způsobem, pokud nedosahuje požadovaných výsledků podle validace, je v konečném důsledku k ničemu. Vybraná ověřovací funkce je pak použita při trénování pro průběžný orientační pohled na pokrok trénovacího a crossvalidačního procesu, a při testování pro určení úspěšnosti předpovědi finální podoby sítě. V případě trénování s učitelem je ověřovací funkce poměrně přímočará, poněvadž je k dispozici přesný tvar výstupu, který je od neuronové sítě očekáván. Tato funkce tedy vypadá následovně:

$$acc = \sum_{i=0}^n \frac{r_i}{n}; r_i = \begin{cases} 1 & \text{pro } y_i = t_i \\ 0 & \text{jinak} \end{cases}$$

kde:

- y_i je i -tý vzorek výstupu sítě,

- t_i je i -tý vzorek požadovaného výstupu,
- r_i označuje i -tou shodu výstupu sítě a požadovaného výstupu,
- n určuje počet vzorků pro výpočet úspěšnosti.

Další metodou pro evaluaci je rekonstrukce předpovědi celého obrázku a jeho vizuální porovnání s předpokládaným výstupem. Předpověď je v takovém případě provedena pro všechny lokace a výsledek v podobě tříd zpětně namapovaných na vhodné hodnoty je promítnut do nového obrázku. Tato metoda je velmi užitečná pro rozpoznání situací, ve kterých byla předpověď nejvíce úspěšná, a ve kterých byla naopak často vyhodnocena špatně. Z obou těchto metod byla mnohem častěji využívána jen ověřovací funkce, jelikož rekonstrukce obrázků je nesrovnatelně časově náročnější.

Kapitola 5

Experimentální fáze

Experimentální fáze představuje jádro celé práce. Cíl spočívá v modifikaci konfigurace sítě, vstupních dat a požadavků na výstupní data tak, aby bylo dosahováno co nejlepších výsledků. V této kapitole budou představeny popis měněných a neměných parametrů sítě, jednotlivé ověřované konfigurace, způsoby evaluace jejich úspěšnosti, vliv jednotlivých faktorů na výsledky předpovědi a nakonec i srovnání s ostatními, podobnými řešeními.

5.1 Stále a modifikované parametry sítě

Jak už bylo řečeno, neuronová síť obsahuje mnoho parametrů, které přímo či nepřímo ovlivňují rychlost učení, úspěšnost predikce, velikost sítě a další vlastnosti. Proto je vhodné stanovit, které z nich reprezentují model, jehož ověřováním se tato práce zabývá, a které budou předmětem modifikace v rámci experimentů.

Způsob učení neuronové sítě je základním stavebním kamenem sítě, a nebude v průběhu experimentování měněn. Pro účely práce se tedy bude jednat o feed-forward vícevrstvou perceptronovou síť, využívající učící algoritmus back-propagation. Tento typ sítě je využíván jako klasifikátor, což je rozhodnutí provedené na základě rady vedoucího práce. V důsledku je zvolena klasifikační funkce softmax pro výstupní vrstvu sítě.

Samotná struktura je velmi významným a často modifikovaným faktorem a v případě vstupní vrstvy úzce souvisí s velikostí vstupního vektoru. Pokud není explicitně specifikováno jinak, vstupní vektor obsahuje následující hodnoty:

- Data z pěti po sobě jdoucích obrázků. Počet zvolen na základě výsledků z rané experimentální fáze.
- Okolí 29x29 pixelů z každého využitého obrázku pro cílový pixel před DCT transformací. Toto okolí poskytuje dostatečné množství informací pro učinění předpovědi při zvladatelném objemu dat, co se týče výpočetního výkonu a kapacitních možností.
- Matice 9x9 hodnot získaných DCT transformací okolí 29x29 pixelů. Tento počet je často měněn v průběhu experimentování.

Vstupní vektor také ovlivňuje potřebný počet neuronů ve skrytých vrstvách, jelikož rozšiřuje množství rozpoznatelných vzorů s přibývajícím množstvím využitých dat. Experimentálně bylo zjištěno, že neuronová síť je efektivně schopna využít pro tyto účely jednu až dvě skryté vrstvy, což je dáno nízkou úrovní komplexity předávaných vstupních vektorů.

Množství těchto vzorů je ovšem značné, a proto je velikost těchto vrstev také rozhodujícím, často měněným faktorem, a je uváděna při každém experimentu.

Velikost výstupního vektoru je přímo úměrná požadované délce předpovědi. Jak je objasněno v sekci 5.2.3, i tento faktor může ovlivnit úspěšnost predikce. Důležitější je ovšem počet a podoba tříd, které má klasifikátor rozpoznávat. Volba počtu tříd, stejně jako jejich podoba, je převážně založena na instinktivním rozdělení do fuzzy množin, což je ideální pro využití v neuronových sítích. Třídy byly zvoleny podle tří různých schémat:

$$\begin{aligned}
 2 \text{ třídy: } & \begin{cases} 0 & \text{pro } Z < 4\text{dBZ} \\ 1 & \text{pro } Z \geq 4\text{dBZ} \end{cases} & 3 \text{ třídy: } & \begin{cases} 0 & \text{pro } Z < 4\text{dBZ} \\ 1 & \text{pro } 4\text{dBZ} \leq Z < 20\text{dBZ} \\ 2 & \text{pro } Z \geq 20\text{dBZ} \end{cases} \\
 4 \text{ třídy: } & \begin{cases} 0 & \text{pro } Z < 4\text{dBZ} \\ 1 & \text{pro } 4\text{dBZ} \leq Z < 12\text{dBZ} \\ 2 & \text{pro } 12\text{dBZ} \leq Z < 32\text{dBZ} \\ 3 & \text{pro } Z \geq 32\text{dBZ} \end{cases}
 \end{aligned}$$

Neuronová síť ve všech zde zmíněných experimentech čítá maximálně dvě vrstvy, jelikož větší počet sítí není schopna efektivně využít. Počet neuronů v jednotlivých vrstvách však představuje důležitou proměnnou, a podléhá častým modifikacím v rámci jednotlivých experimentů. Jejich počet je zde vždy explicitně uveden. S celkovou strukturou sítě pak souvisí i další faktory, jako learning rate, maximální počet epoch, apod. Tyto jsou pak voleny buď individuálně právě na základě struktury sítě, nebo jsou ponechány na hodnotách, které vykazují pokrok při trénování sítě.

Důležité je také zmínit rozdělení trénovacích množin. Určitá část je totiž vyhrazena pro trénování, další pro crossvalidaci, a poslední část se využívá při testování/evaluaci. V tomto ohledu byl poměr jednotlivých částí rozdělen na základě rady. 70% všech trénovacích množin byl vyhrazen čistě pro trénování, jako nejdůležitější a nejnáročnější části na počet potřebných trénovacích množin. Další 10% bylo přiřazeno crossvalidaci, a zbylých 20% slouží k evaluaci, jak dobře se síť naučila předpovídat srážky.

5.2 Vliv jednotlivých faktorů na úspěšnost předpovědi

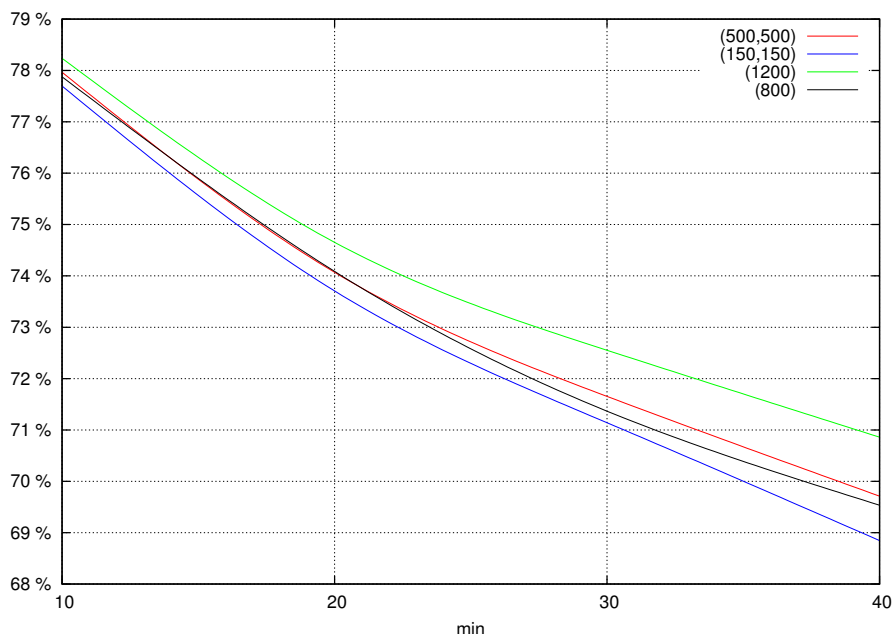
V této části již přecházíme k vyhodnocování experimentů. Cílem této části je prozkoumat vlastnosti různých konfigurací s různými parametry sítě, abychom došli k nejúspěšnější konfiguraci. Experimenty jsou prováděny nezávisle na sobě, každý trénovací proces probíhá na nové náhodně nainicializované síti. Všechny zde uvedené experimenty jsou prováděny nad dostatečně obsáhlým množstvím dat, parametry sítě jsou udržovány přinejmenším v mezích určených výpočetním výkonem použitého počítače a zvladatelností experimentů z časového hlediska. Data vybraná pro trénování jsou stále stejná (myšleno tak, že jsou zvolena ze stále stejné kolekce obrázků), mění se pouze jejich podoba při extrakci dat a formování trénovacích množin (počet a namapování tříd, velikost okolí cílového pixelu, apod.). Stejně tak není v žádném ohledu měněn hardware, na kterém experimenty probíhají.

Pro ucelenost a pozdější reference je uveden první regulérní experiment, provedený na finální podobě způsobu trénování, s využitím následujících parametrů:

- Vstupní vektor využívá data z pěti po sobě jdoucích obrázků, každá část je tvořena

okolím 29x29 pixelů. Z této posloupnosti je následovně DCT transformací získána 3D matice 5x5x5 hodnot, což je finální podoba vstupního vektoru.

- Výstupní vrstva je tvořena strukturou SoftmaxBox popsanou v kapitole 4.4. Síť je navrhována a natrénována tak, aby byla schopna předvídat srážky pro 10, 20, 30 a 40 minut do budoucna.
- Síť je využita jako klasifikátor do 4 různých tříd.



Obrázek 5.1: Úspěšnosti předpovědi jednotlivých konfigurací neuronové sítě pro 10, 20, 30 a 40 minut. V závorkách jsou uvedeny velikosti skrytých vrstev, počet hodnot v závorce odpovídá počtu skrytých vrstev.

Jak je vidět z grafu, úspěšnosti nejkratších, desetiminutových předpovědí se pro 4 klasifikované třídy pohybují kolem 78%, s pozdějšími předpověďmi však úspěšnost stále klesá. Toto zjištění potvrzuje validitu předpovědního modelu - čím déle do budoucna chceme určit předpověď, tím je její správné určení náročnější a složitější. Se stejnými parametry byla sestavena síť bez struktury SoftmaxBox ve výstupní vrstvě, aby bylo ověřeno, zda SoftmaxBox vrstva nemá negativní vliv na výsledky jednotlivých předpovědí. Výsledky obou typů konfigurací neuronové sítě vykazovaly pro krátké předpovědi podobné hodnoty úspěšnosti, ale byl vyzorován mírný negativní vliv SoftmaxBox vrstvy na výsledky delších předpovědí. Podrobnějším zkoumáním tohoto jevu se zabývá sekce 5.2.3.

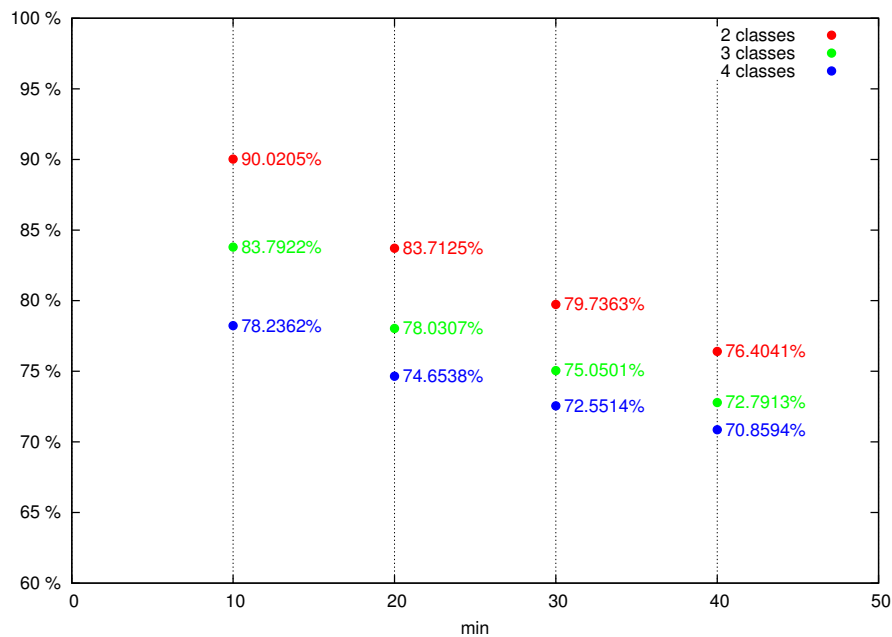
5.2.1 Velikost vstupního vektoru

Velikost vstupního vektoru přímo ovlivňuje natrénování sítě. Je výchozím bodem pro vyladění pravidel, podle kterých se na radarových obrázcích přesunují fronty. Čím je vstupní vektor větší (a to jak z pohledu velikosti okolí, tak i počtu obrázků zahrnutých ve vstupním vektoru), tím má neuronová síť více informací pro určení přesnější předpovědi. Tento fakt však vyžaduje přizpůsobení konfigurace sítě, a to hlavně z hlediska velikosti vstupní vrstvy

a skrytých vrstev, s čímž je spjato i delší trénování takové sítě a nutnost vytvoření mnohem většího počtu trénovacích množin. Použitá okolí by proto měla poskytnout co nejvíce informací o vývoji potenciální fronty v co nejmenším počtu koeficientů. Počet obrázků použitých pro vytvoření konkrétní trénovací množiny byl však omezen. Experimenty prokázaly, že neuronová síť dokáže vydedukovat pohyb fronty z 3 po sobě jdoucích okolí, větší počet snímků již nepomáhá k provedení předpovědi, a je spíše ke škodě, protože zbytečně nafukuje vstupní vrstvu neuronové sítě.

5.2.2 Počet klasifikovaných tříd

Dalším důležitým faktorem je stanovení počtu a namapování jednotlivých tříd. Tímto však efektivně omezujeme diverzitu rozpoznatelných výsledků, a proto se ve většině případů s tímto parametrem musí manipulovat jen velmi obezřetně a odůvodněně. V případě předpovědi počasí však je možné tento parametr modifikovat s větší volností. Hlavním smyslem této práce je předpověď srážek, ke které nám stačí v nejjednodušší formě stanovit dvě třídy (jednu pro stav "prší", druhou pro "nepřší"). Rozšíření tohoto počtu již závisí na tom, do jaké míry chceme identifikovat výskyty různých intenzit deště. V tomto ohledu se jeví více než dostatečné definovat maximálně 4 třídy, označující jeden stav beze srážek a tři stavy pro mírnou, střední a vysokou intenzitu srážek. Více tříd je pro předpověď na této úrovni poměrně přebytná, a velké množství tříd značně komplikuje proces trénování a zhoršuje schopnost sítě dojít ke správné klasifikaci. Současně toto zvýšení požadavků implikuje nutnost značného zvětšení skrytých vrstev se všemi s tím spojenými důsledky a problémy.



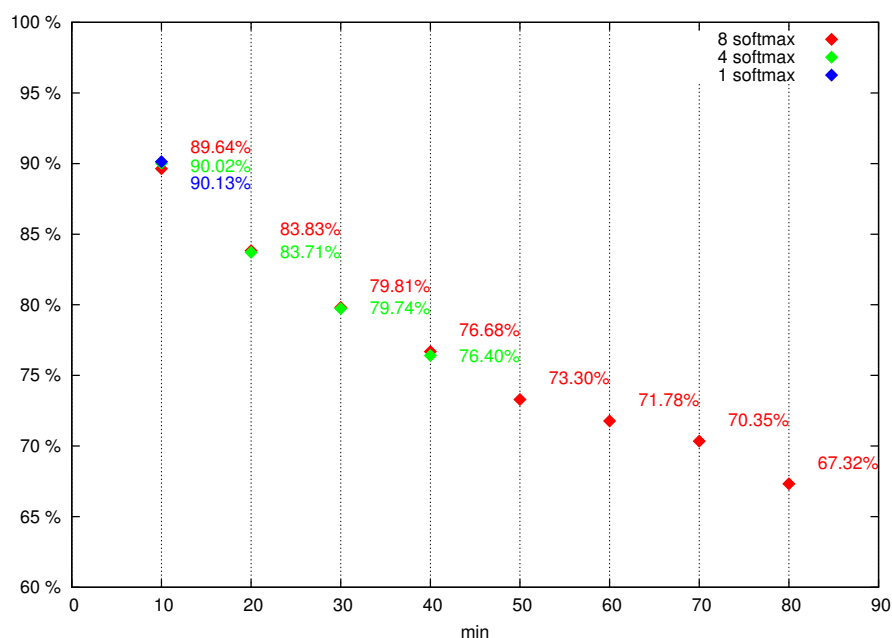
Obrázek 5.2: Vyjádření vlivu počtu tříd na celkovou úspěšnost předpovědi.

Tento experiment spíše než pro ověření faktu, že více tříd ke klasifikaci znamená nižší celkovou úspěšnost, ověřuje správné namapování jednotlivých tříd a rozdělení trénovacích množin tak, aby všechny klasifikované třídy měly co nejrozměrnější zastoupení v celé kolekci trénovacích množin. Tento záměr má za cíl natrénování sítě pro rozpoznávání všech definovaných tříd. Kdyby měla některá třída velmi minoritní zastoupení ve vektorech oče-

kávaných výsledků u kolekce trénovacích množin, síť by odvodila nízký vliv této třídy na předpověď, neměla by dostatek podkladů pro detekci této třídy a v důsledku by upřednostnila ve svém výstupu některou z více zastoupených tříd, což by vedlo k poklesu úspěšnosti u natrénované sítě.

5.2.3 Rozsah předpovědi

Rozsah předpovědi, jak ve smyslu délky, tak počtu provedených předpovědí v rámci jedné umělé neuronové sítě, je také podstatnou součástí předpovědního modelu. Architektura sítě (viz. sekce 4.4) povoluje provést více předpovědí v různých časových oknech prostřednictvím jedné neuronové sítě. S tím se ovšem pojí další potenciální problém. V důsledku sestavení sítě s rozšířenou výstupní vrstvou v podobě SoftmaxBoxu by mohla síť vykazovat horší výsledky pro jednotlivé časy než dedikovaná síť, která je zaměřena pouze na konkrétní čas. Proto je součástí experimentální fáze i toto prověření úspěšnosti předpovědi pro obě architektury výstupní vrstvy. Všechny ostatní parametry byly v obou případech ponechány na stejných hodnotách.

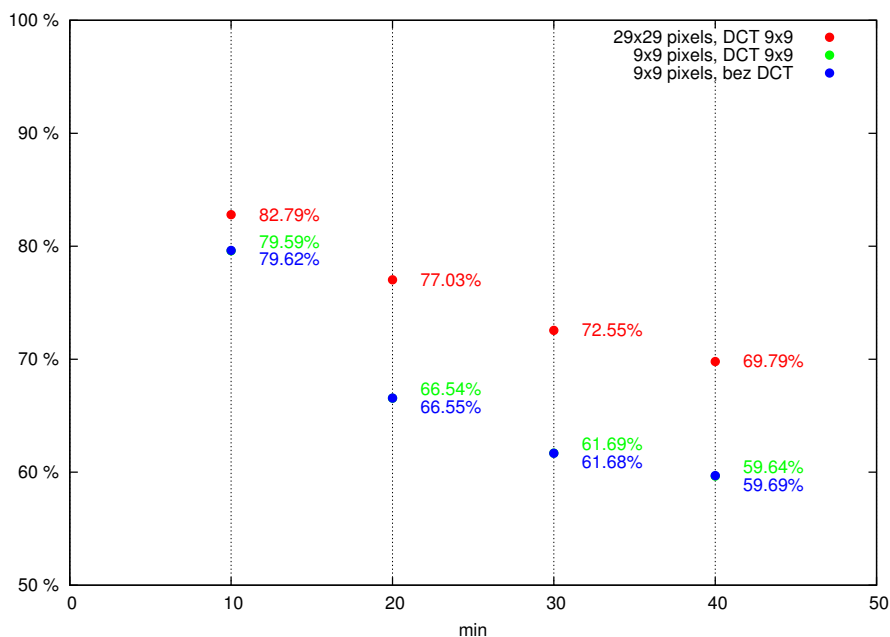


Obrázek 5.3: V grafu jsou zachyceny úspěšnosti jednotlivých sítí, které využívají 8, 4, resp. 1 softmax.

Experiment neukázal žádný výrazný rozptyl v úspěšnosti pro síť s různým rozsahem předpovědi, pokud začínají nejkratší, desetiminutovou předpovědí. Ověřením rekonstrukcí obrázku však bylo zjištěno, že všechny tyto softmaxy poskytují dosti podobné výstupy, a pokles úspěšnosti plyne téměř výlučně z pohybu srážkových mraků. Nejkratší předpověď tedy v podstatě určovala výstup celé sítě pro všechny časy předpovědi, což samozřejmě negativně ovlivňuje hlavně delší předpovědi. Proto jsou dále používány pouze síť, které se zaměřují na konkrétní čas.

5.2.4 Vliv DCT transformace

Posledním zmíněným parametrem, který je zde detailněji prozkoumán, je vliv DCT transformace. Tato transformace nepopíratelně pomáhá při kompresi jednotlivých okolí do menšího počtu koeficientů při zahuštění energie, a transformuje hodnoty z přirozeného definičního oboru čísel (popisujících hodnoty odrazivosti v jednotlivých pixelech) na reálné koeficienty DCT transformace. Otázkou však stále zůstává, zda a do jaké míry tato transformace (v případě bezztrátové transformace) ovlivňuje výsledek předpovědi.



Obrázek 5.4: Vliv DCT transformace. Legenda v pravém horním rohu grafu vyjadřuje velikost okolí v pixelech jako součást vstupního vektoru trénovacích množin před DCT transformací a přítomnost DCT transformace i s cílovou velikostí matice hodnot vstupního vektoru po transformaci, pokud byla provedena.

Sama o sobě DCT transformace nevykazuje žádné změny ve výsledcích predikce. Její přínos je ovšem mnohem markantnější při kompresi informace. Tato transformace vyžaduje náročnější přípravu trénovacích dat. V důsledku zahuštění energie však je výsledná síť mnohem menší, ač sbírá informace z většího okolí.

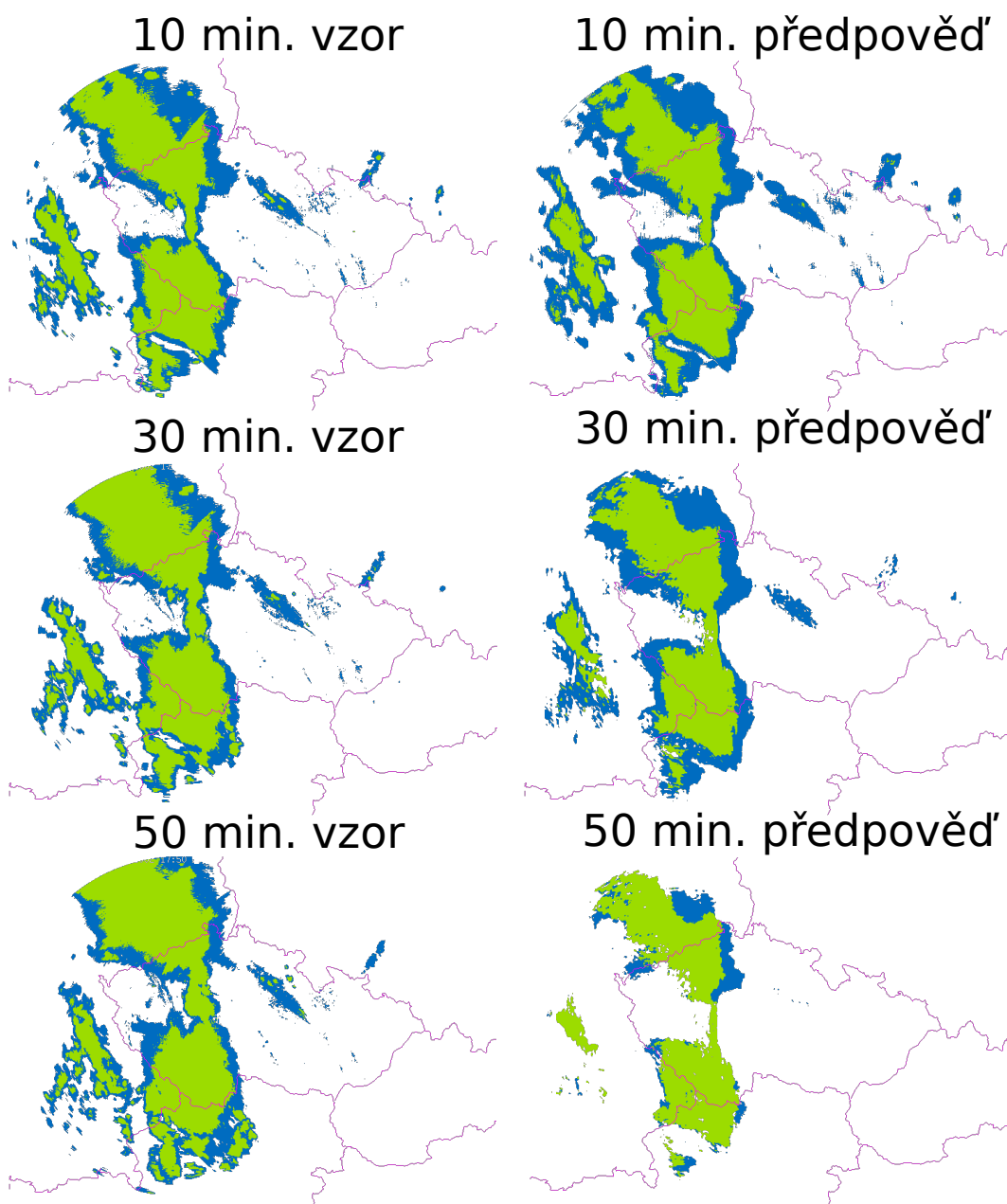
5.3 Porovnání s ostatními pracemi

V roce 2009 byly vypracovány dvě bakalářské práce zabývající se podobným přístupem k předpovědi počasí. Proto je vhodné porovnat výsledky této práce s výsledky, které tyto dvě práce poskytly. V bakalářské práci Lukáše Putny [6] a Petra Gerycha [2] jsou dostatečně detailně popsány parametry (respektive obě práce popisují výlučně některé podstatné parametry, a jejich kombinací je získán dostatek informací pro napodobení experimentu), aby bylo možné tyto dvě metody alespoň přibližně porovnat s metodou používanou v této práci. Obě tyto práce byly porovnávány se vzájemně dohodnutými parametry, a to tak, že byla vybrána konkrétní série obrázků, na základě nichž byla provedena předpověď pro všechny body na mapě. Úspěšnost takového modelu přímo souvisí s množstvím srážek zachycených

na konkrétních obrázcích. Pokud kterémukoliv z těchto modelu předložíme sérii obrázků, na kterém se nepohybují žádné srážkové fronty, budou všechny modely vykazovat až 100% úspěšnost. Naopak série obrázku s bohatou a rozmanitou srážkovou aktivitou dokáže silně zamávat s úspěšností všech metod. Je nutné použít naprosto stejné obrázky, aby porovnání bylo co nejpřesnější.

Předpověď je tvořena pomocí obrázků ze dne 10.07.2002 do 17 hodin, a předpověď je tvořena pro časy po 17. hodině (10-minutová předpověď odpovídá času 17:10, atd.). Po sestavení předpovědních obrázků jsou porovnány jednotlivé pixely určující srážkovou aktivitu předpovědních a skutečných obrázků. Procentuální úspěšnosti vyjadřují, do jaké míry se daný předpovědní obrázek shoduje se svým vzorem nejdříve v rámci všech tříd, a poté je spočítána shoda u jednotlivých tříd zvlášť. Výsledky tohoto srovnávání jsou v následující tabulce.

	Kritérium úspěšnosti	N. síť (Putna)	Rozpozn. struktur	N. síť (Vlček)
10 minut	Celková úspěšnost	90.56%	90.71%	91.08%
	Úspěšnost třídy 1	95.31%	96.41%	95.37%
	Úspěšnost třídy 2	55.09%	61.49%	67.97%
	Úspěšnost třídy 3	88.91%	73.15%	81.11%
20 minut	Celková úspěšnost	88.76%	87.91%	90.58%
	Úspěšnost třídy 1	95.14%	94.98%	97.18%
	Úspěšnost třídy 2	40.68%	50.41%	55.56%
	Úspěšnost třídy 3	86.28%	66.19%	74.50%
30 minut	Celková úspěšnost	87.79%	85.65%	88.12%
	Úspěšnost třídy 1	95.06%	93.68%	96.21%
	Úspěšnost třídy 2	34.53%	42.27%	46.26%
	Úspěšnost třídy 3	84.73%	60.18%	67.15%
40 minut	Celková úspěšnost	86.39%	83.39%	85.54%
	Úspěšnost třídy 1	94.31%	92.69%	95.14%
	Úspěšnost třídy 2	26.61%	35.42%	37.62%
	Úspěšnost třídy 3	83.55%	53.10%	60.23%
50 minut	Celková úspěšnost	84.35%	81.75%	84.42%
	Úspěšnost třídy 1	92.93%	91.99%	98.35%
	Úspěšnost třídy 2	26.10%	26.22%	6.14%
	Úspěšnost třídy 3	81.94%	49.12%	54.36%



Obrázek 5.5: Tento obrázek ukazuje porovnání mezi skutečností, reprezentovanou snímky nalevo, a předpověďmi provedenými neuronovou sítí pro všechny body na obrázku a následně sestavenými do snímků vpravo. Jak je vidět, korelace mezi skutečností a předpovědí je pro nejkratší předpovědi velmi vysoká, s požadavkem na delší předpověď však úspěšnost sítě viditelně klesá.

Kapitola 6

Závěr

Umělé neuronové sítě jsou stále velmi rapidně rozvíjejícím se typem strojového učení, a tato bakalářská práce si kladla za cíl experimentálně ověřit jejich použitelnost v oboru predikce srážek pomocí meteoradarových snímků. Implementace komerčních předpovědních systémů je velmi často utajená, a tedy evaluace úspěšnosti předpovědi čistě na základě již fungujících a mnohem robustnějších modelů není možná.

Byl tudíž vytvořen vlastní model. Pro účely vypracování této práce byly poskytnuty radarové obrázky z minulosti v rozsahu 7 dní z různých časových období. Byl navržen a realizován postup pro zpracování a rozšifrování těchto obrázků, extrakce potřebných dat pro natrénování sítě, jejich seskládání do logicky uspořádaných sekvencí a následné vytvoření trénovacích množin. Po rozdělení trénovacích množin do trénovací, crossvalidační a testovací části byla natrénována vlastní neuronová síť. Pak následovala dlouhá experimentální fáze, která měla za úkol prověřit vliv změn jednotlivých parametrů sítě na úspěšnost předpovědi. Prostřednictvím kalibrace parametrů sítě byla pak vybrána ideální konfigurace, a ta byla porovnána s podobnými řešeními, které v minulosti řešili mí kolegové.

Přestože řešení v této práci vykazuje dobré výsledky z pohledu celkové úspěšnosti předpovědi, stále nedosahuje takových výsledků, jakých by mohlo dosahovat i s využitím pouze poskytnutých dat, a to hlavně co se týče delších předpovědí. To je dáno velmi dlouhou dobou vytváření trénovacích množin s obsáhlejším okolím až v řádu dní. I přesto však lze vyzorovat potenciál ve zlepšení oproti práci Lukáše Putny, hlavně co se týče využití vlastnosti zahuštění energie pomocí DCT transformace s následnou kompresí. Časový tlak tedy nedovolil blíže prozkoumat možnosti konfigurací využívajících mnohem větší okolí.

Z pohledu praktického řešení zde hrála roli jedna důležitá překážka - poskytnuté obrázky jsou pořizovány s odstupem 10 minut, zatímco aktuální obrázky z CHMI jsou získávány s 15-minutovými intervaly. Toto zhoršuje efektivní použití neuronové sítě, která byla natrénována na poskytnutých obrázcích. Aktuální obrázky však mají velmi podobnou formu jako poskytnuté obrázky, a síť může být poměrně jednoduše přetrénována prostřednictvím aktuálnější databáze obrázků na aktuální předpovědi.

Nabízejí se také mnohá možná rozšíření využití, kupříkladu by mohla být síť využita jako mobilní aplikace, která by prostřednictvím GPS (nebo jiného lokalizačního prostředku) získala informaci o poloze, a provedla by na základě posloupnosti aktuálních radarových snímků předpověď pro současnou polohu uživatele. Při vysoké pravděpodobnosti nadcházejícího výskytu srážek by pak informovala uživatele o možné přicházející frontě. Rozšíření se dá uplatnit i na zvýšení úspěšnosti předpovědi prostřednictvím zahrnutí dalších faktorů jako vstup neuronové sítě. Mohlo by se jednat například o orografickou mapu, která by mohla usnadnit detekci vzniku srážek při přechodu oblačnosti do vyšších nadmořských výšek v

důsledku kondenzace vlhkosti při chladnutí vzduchu. Tato rozšíření by však vyžadovala opět zvětšení neuronové sítě v kombinaci s mnohem větší základnou dat pro trénování.

Literatura

- [1] BrainU, University of Minnesota Department of Neuroscience and Department of Curriculum and Instruction: About Neurons [online].
http://brainu.org/files/tn_about_neurons.pdf, 2000-2011 [cit. 2015-01-17].
- [2] GERYCH, Petr: Predikce deště pomocí memeteorologického radaru. Technická zpráva, FIT VUT v Brně, 2009.
- [3] Government of Canada - Environment Canada: About Radar [online].
<http://www.ec.gc.ca/meteo-weather/default.asp?n=2B931828-1>, 2014-08-06 [cit. 2014-12-28].
- [4] Human Brain Project: Human Brain Project [online].
<https://www.humanbrainproject.eu/>, duben 2012 [cit. 2015-01-17].
- [5] KRÁČMAR, Jan: Meteorologické radiolokátory [online].
http://www.chmi.cz/files/portal/docs/meteo/rad/info_radar/, 1997-2011 [cit. 2014-12-29].
- [6] PUTNA, Lukáš: Predikce deště pomocí memeteorologického radaru. Technická zpráva, FIT VUT v Brně, 2009.
- [7] ROELOFS, Greg: *PNG: The Definitive Guide*. 2002-2003.
- [8] WATSON, Andrew B.: Image Compression Using the Discrete Cosine Transform. *Mathematica Journal*, ročník 4, č. 1, 1994: s. 81-88.
- [9] Wikibooks: Artificial Neural Networks [online].
http://en.wikibooks.org/wiki/Artificial_Neural_Networks/Activation_Functions, 2015-01-11 [cit. 2015-01-17].
- [10] WILSON, Robert A. KEIL, Frank C.: *The MIT Encyclopedia of the Cognitive Sciences*. The MIT Press, 2001, iSBN 0-262-73144-4.
- [11] ŠNOREK, Miroslav: *Neuronové sítě a neuropočítače*. Praha: ČVUT, první vydání, 1996, iSBN 80-010-1455-X.
- [12] Český hydrometeorologický úřad: Radarový snímek z přechodu fronty nad Trutnovem z 1. 3. 2008, 10:00 SEČ [online].
<http://www.bourky.com/pozorovani/cyklona-emma-a-derecho/>, 2008 [cit. 2014-12-28].

- [13] Český hydrometeorologický úřad: Parametry meteorologických radarů ČHMÚ [online]. http://www.chmi.cz/files/portal/docs/meteo/rad/info_czrad/, 2010-2011 [cit. 2014-12-28].

Příloha A

Obsah DVD

DVD obsahuje veškeré zdrojové texty společně s návodem na použití. Dále zde je technická zpráva bakalářské práce jak v zdrojové, tak zkompilované formě, několik obrázků pro vzorovou předpověď, pomocné obrázky potřebné pro vytvoření předpovědi a složková hierarchie připravená pro veškeré operace nad obrázky, extrahovanými daty, trénovacími množinami a provedenými předpověďmi.

Příloha B

Manual

Manuál se nachází v kořenovém adresáři DVD v souboru "manual". Obsahuje veškeré informace potřebné k využívání příložených skriptů.

Příloha C

Konfigurační soubor

Konfigurační soubor se nachází v adresáři “nn_training.v6” a je uložen jako soubor “features_config.py”. Obsahuje důležité parametry, které musí být nastaveny před použitím většiny skriptů.

```
# features config
diameter_in_pixels = 4 # prumer okoli kolem vybraneho bodu
compression_factor = 0 # komprese okoli, kdyz je 0, neprovadi se komprese (neni
    nutna)
pixel_surrounding_area = (2 * diameter_in_pixels + 1)**2

# training set config
training_set_size = input_vector_size = 5
label_set_size = 12
DCT_onedim_size = 9 # DCT komprese - velikost matice po kompresi

# Promenna tolerance filtruje trenovaci mnoziny, snazi se eliminovat nepouzitelna
    data z obrazku.
tolerance = 3
file_split_period = 200

nclasses = 3 # Pocet klasifikovanych trid
nSoftmaxes = 4 # Pocet predpovidanych casu (v ramci jedne neuronove site)
hidden_layer_sizes = (950,)
n_hidden_layers = len(hidden_layer_sizes) # Pocet skrytych vrstev

FECHAINEDPATH = "/media/michael/AEB07285B0725435/NN_RESOURCES/features_chained/"
TRSETPATH = "/media/michael/AEB07285B0725435/NN_RESOURCES/training_sets/"
```
