

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 POMOCÍ METEOROLOGICKÉHO
RADARU

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

LUKÁŠ PUTNA

BRNO 2009



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 POMOCÍ METEOROLOGICKÉHO RADARU

RAIN PREDICTION USING METEO-RADAR

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

LUKÁŠ PUTNA

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. IGOR SZÖKE

BRNO 2009

Abstrakt

Tato bakalářská práce se zabývá návrhem a implementací systému pro krátkodobou předpověď deště za pomoci meteorologického radaru. Zmiňuje některé postupy při řešení a zejména využití vícevrstvé neuronové sítě, čemuž se celá práce dále věnuje. Vícevrstvé neuronové sítě mají dobré klasifikační schopnosti a dokáží aproximovat jakoukoliv nelineární funkci. V práci je navrženo, jak předpověď pomocí neuronové sítě funguje, postup při úpravě dat z meteorologického radaru pro vstup sítě. Dále se práce zabývá přístupem k trénování sítě a možnými metodami vyhodnocení výsledků testů. Na závěr jsou zhodnoceny konkrétní výsledky, dosažené testováním neuronové sítě, a je navrženo další možné vylepšení postupu.

Abstract

This work deals with the concept and implementation of short term rain prediction system using meteo-radar. Some basic methods are mentioned and then artificial neural networks are discussed and used for solution. It is proposed how the prediction system using neural networks works. The process of preparing radar data, training neural network with the data and a few scoring methods are discussed. There are shown some experimental results and several improvements are devised at the end of this paper.

Klíčová slova

Predikce deště, krátkodobá předpověď počasí, umělá neuronová síť, meteorologický radar, zpracování obrazu, trénování neuronových sítí.

Keywords

Rain prediction, short term weather prediction, artificial neural network, meteorological radar, image processing, neural network training.

Citace

Putna Lukáš: Predikce deště pomocí meteorologického radaru, bakalářská práce, Brno, FIT VUT v Brně, 2009

Predikce deště pomocí meteorologického radaru

Prohlášení

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

.....
Lukáš Putna
18.května 2009

Poděkování

Touto cestou bych chtěl poděkovat svému vedoucímu Ing. Igoru Szökemu za ochotu při konzultacích, četné rady a návrhy při řešení a zpracování této práce. Dále bych chtěl také poděkovat Ing. Františkovi Grézlovi, Ph.D. za cenné rady ohledně neuronových sítí a Petru Gerychovi za spolupráci při testování.

© Lukáš Putna, 2009

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

Úvod	2
1 Radarové snímky a možnosti predikce	4
1.1 Radarové měření a jeho produkt	4
1.2 Možnosti řešení	5
1.3 Umělé neuronové sítě	6
2 Trénování neuronové sítě	8
2.1 Neuronová síť obecně	8
2.2 Vícevrstvá perceptronová síť	10
2.3 Neuronová síť při předpovědi	11
2.4 Trénování	14
2.5 Crossvalidace	14
2.6 Rychlost učení	15
3 Příprava dat	17
3.1 Historie dat a předpověď	17
3.2 Okolí bodu a snížení rozlišení dat	18
3.3 Rozdělení intenzit do tříd	20
3.4 Vstupní a výstupní vektory	21
4 Testování	23
4.1 Celková úspěšnost	23
4.2 Úspěšnost třídy	24
4.3 Bodování a složitější návrhy hodnocení	24
4.4 Metody výpočtu chyby	26
4.5 Grafický výstup předpovědi	26
5 Výsledky experimentů	28
5.1 Velikost neuronové sítě, faktor rychlosti učení	28
5.2 Závislost na počtu trénovacích vektorů	29
5.3 Predikce v časech	30
5.3.1 Předpověď na dobu 10 až 50 minut	31
5.3.2 Předpověď na 60 minut	31
5.3.3 Předpověď více jak na 60 minut	33
5.4 Porovnání řešení s jinou metodou	33
6 Závěr	36
Použitá literatura	37

Úvod

Prognózou počasí se lidstvo zabývá už dlouhé věky. Přece jen počasí ovlivňuje naše každodenní bytí na Zemi a určuje podmínky pro život. Pojem počasí sám o sobě shrnuje celkový stav atmosféry, která se skládá z velkého množství různých faktorů a atmosférických jevů, v určitém místě a čase. Všechny tyto vlastnosti a děje lze jednotlivě popsat fyzikálními rovnicemi. Problém nastává v tom, že toto velké množství meteorologických prvků se výrazně navzájem ovlivňuje. Vyjádřit celkový stav atmosféry, na základě veškeré interakce a jejích fyzikálních vlastností, je stále v této počítačové době nemožné. Z pohledu meteorologie a meteorologických předpovědních modelů, ať se jedná o počítačové modely, nebo o předpověď např. na základě pozorování, je důležité, vybrat ze všech faktorů atmosféry ty nejdůležitější a z těch teprve vytvořit zjednodušený model atmosféry, na základě kterého se potom odvodí předpověď dalšího vývoje počasí nebo konkrétního jevu. A zde právě vzniká problém s úspěšností předpovědi. Nejsou zahrnuty všechny vlastnosti, a proto nelze nikdy očekávat přesnou odpověď. Složitá otázka se týká samotného vyhodnocení předpovědi, toho, jak vyjádřit míru úspěšnosti.

Snahou zakomponovat co nejvíce meteorologických prvků do předpovědi, které dnes již převážně řeší výkonné počítače, vznikají složité numerické výpočetní modely. Bývají to jedny z nejsložitějších programů na světě vůbec, nejen co se výpočetního času týče. Díky neustálému vývoji modelů, výpočetní síly počítačů, nebo například propojení výstupů různých modelů, se předpověď počasí neustále zlepšuje. Právě zmíněná kombinace výsledků hraje v této práci velký význam, resp. určuje jeden ze zájmů pro krátkodobou predikce deště.

Předpověď počasí lze rozdělit do několika kategorií podle doby, na jak dlouho dopředu je předpovídáno. Tato práce se zabývá tou nejkratší používanou předpovědí, navíc specializovanou na dešťové srážky. Meteorologie za takovou předpověď považuje předpověď na dobu v řádu minut až několik málo hodin. Většinou se právě jedná o odvození nějakých vlastností ze snímků meteorologického radaru. Z mnoha účelů takové předpovědi, je asi nejpodstatnější myšlenka, využívat výsledky velmi krátkodobé předpovědi pro předpověď dlouhodobější. Tedy, že numerické výpočetní modely např. pro předpověď několik málo dní dopředu, využívají predikci vývoje deště nebo třeba větru v několika následujících hodinách. Dalším podstatným účelem je systém včasného varování před pohromami, které vznikají velice rychle a často se nedají předpovědět dlouho dopředu. Jde hlavně o detekci konvektivních bouří, které bývají často doprovázeny dalšími jevy jako jsou tornáda a vichřice, kroupy a přívalové srážky, z nichž potom mohou vznikat přívalové povodně. V neposlední řadě krátkodobá předpověď nalezne uplatnění pro obecnou informovanost o počasí, což by se z vědeckého hlediska mohlo zdát zanedbatelné, z pohledu možného komerčního využití v dopravě, např. plánování letového provozu, nebo v médiích to je podstatné.

Metod pro krátkodobou předpověď deště na základě radarových snímků existuje několik. Osvědčené postupy jsou chráněny organizacemi, které je používají, a nebývá k nim zpravidla volný přístup. Tato práce se zabývá netradičním řešením, kterým byl problém již několikrát řešen, ale které se v běžné praxi zatím skoro nepoužívá. Hlavní myšlenkou je využití neuronových sítí, které mají vynikající schopnosti zobecňování. Vývoj dešťových srážek záleží na několika faktorech, jako je rosný bod, směr větru apod., v tomto případě bylo využito pouze radarových snímků. V případě, že takových snímků bylo k dispozici větší množství, bylo možné zformulovat jistou abstrakci chování a vývoje dešťových mraků. Z radarových snímků zobecnit a vytvořit vjem o chování dešťových mraků vyzdvižením důležitých faktorů, které lze z radarových snímků vyzorovat, jako je např. směr nebo rychlost pohybu dešťových mraků. Touto abstrakcí je právě myšlena umělá neuronová síť, která

k tomu má dobré předpoklady. Tématem spojení neuronových sítí a předpovědí deště se zabývá kapitola první.

Podrobnější popis neuronových sítí a samotný průběh jejich trénování se nachází v kapitole druhé. Do procesu trénování patří určení faktoru učení nebo průběžná kontrola výsledků sítě mezi jednotlivými iteracemi trénování.

Jedním z nejpodstatnějších kroků je příprava a předzpracování radarových snímků pro trénování neuronové sítě a dále pro testování na reálných situacích. Celkový postup a možné varianty jsou podrobně rozebrány v kapitole třetí. Je potřeba vybrat vhodné radarové snímky a vytvořit z nich trénovací a testovací vektory. S tím souvisí některé nutné nebo volitelné úpravy, jako je například zmenšení rozlišení obrázků apod.

Kapitola čtvrtá navrhuje možnosti testování reálných situací a zejména pak metody a postupy pro vyhodnocení a porovnání výsledků. Tyto metody jsou zde přesně definovány, protože je jich využito v další kapitole. V kapitole páté jsou potom uvedeny výsledky některých testů spolu s komentářem.

Práce nenavazuje na žádné další projekty, stejné zadání bylo souběžně řešeno mým kolegou samostatně a zcela jiným přístupem. V páté kapitole bude také uveden krátký popis tohoto přístupu, porovnání výsledků s přístupem popsáním v této práci na konkrétních příkladech a diskutovány klady a zápory obou metod. V závěru jsou pak výsledky celkově zhodnoceny a jsou navrženy některé mechanismy pro vylepšení celého postupu a pokračování v práci.

1 Radarové snímky a možnosti predikce

Aby bylo možné předpovídat déšť na základě radarových snímků, je zapotřebí jejich pochopení a správná interpretace. Budou nastíněny některé základní přístupy k predikci dešťových srážek na základě radarových obrazových produktů a zejména bude krátce popsáno využití neuronových sítí při řešení této úlohy. Bude potřeba vysvětlit, na jakém principu a jak vůbec neuronové sítě fungují. Jelikož se jedná o rozsáhlou látku s velkým množstvím různých teorií, bude popsán jen základní princip.

1.1 Radarové měření a jeho produkt

Meteorologické radiolokátory využívají faktu, že srážkové částice rozptýlené v atmosféře jsou schopny odrazet radiovlny v určitém pásmu vlnových délek, konkrétně v centimetrovém pásmu (mikrovlny). Elektromagnetická energie je vysílána parabolickou anténou vysílače radaru ve tvaru úzkého svazku do atmosféry. Zde dochází k odrazu energie od srážkových částic, ale také bohužel od různých nemeteorologických cílů např. letadel. Část odražené energie je zpět přijata anténou, následně zesílena a detekována přijímačem radaru. Intenzita cíle je dána množstvím odražené energie, která je přímo úměrná radiolokační odrazivosti Z . Tu lze určit pomocí následující rovnice:

$$P_r = Pl \times \left(\frac{Z}{R}\right)^2, \quad (1.1)$$

kde P_r je výkon přijatý přijímačem, Pl je meteorologický potenciál radaru (konstanta) a R je vzdálenost cíle. Jednotkou odrazivosti je mm^6/m^3 , v praxi se používá logaritmická jednotka dBZ . Poloha cíle je určena podle polohy a natočení antény a doby mezi vysláním a příjmem signálu.

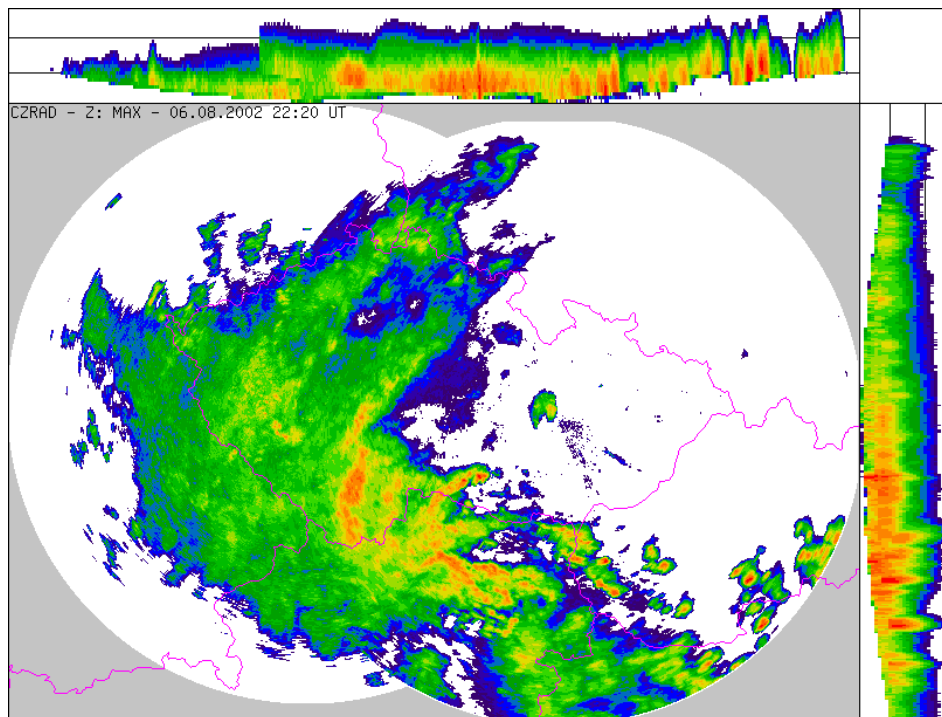
Okamžitá odrazivost cílů v atmosféře je zobrazena ve formě obrázku, který potom zobrazuje okamžité horizontální, ale i vertikální rozložení v prostoru dosahu radaru pomocí barevné stupnice. Horizontální rozložení dat bývá v buňkách o reálných rozměrech $1 \times 1 km$, přičemž jeden obrazový bod (1 pixel) odpovídá jedné buňce. Vertikální rozložení bývá po $1 km$. Radar má celkový dosah maximálně $250 km$. Naměřená data bývají ve větších vzdálenostech od radaru podhodnocena a v kratších zase nadhodnocena. Je to způsobeno zakřivením Země, na větší vzdálenost se energetický paprsek od Země vzdaluje a není schopen zachytit srážky v nižších výškových hladinách, ve kterých má měření větší účinnost. Naopak v blízkosti radaru nedochází k zachycení srážek ve větších výškách a dochází tak k nadhodnocení údajů. Dalo by se říci, že data zjištěná radarem nejsou vždy zcela přesná. K dalším odchylkám přispívá například vlastní šum a nestabilita vysílače radaru, nebo již zmíněné odrazy nemeteorologických cílů. Radary však poskytují okamžitý přehled rozložení intenzit srážek na velkém území v dobrém časovém rozlišení (většinou po 10ti minutách).

Intenzitu srážek I je možné zjistit z radiolokační odrazivosti Z , protože k ní má přímý vztah. Pro výpočet lze použít *Marshall–Palmerův* vzorec:

$$Z = a \times I^b, \quad (1.2)$$

kde a a b jsou experimentálně zjištěné konstanty ($a=200$, $b=1,6$). Je potřeba připomenout, že barevné rozložení v radarových snímcích se týká odrazivosti Z , nikoliv intenzity srážek I . Dále je nutné si při používání radarových snímků uvědomit, že vzhledem k jednotce odrazivosti Z , která má exponenciální charakter, je i intenzita srážek na odrazivosti závislá *exponenciálně*. [1]

Konkrétní radarové snímky, použité v této práci, jsou poskytnuty od Českého hydrometeorologického ústavu. Příkladem jednoho takového snímku je Obrázek 1.1.



Obrázek 1.1 Radarový snímek, (Zdroj: ČHMÚ, 2002)

Barevná stupnice odrazivostí má 15 stupňů s krokem 4dBZ. Tabulka 1.1 vypovídá o barevném rozložení odrazivostí spolu s přibližně odpovídajícím rozsahem intenzity srážek:

Z[dBZ]	4	8	12	16	20	24	28	32	36	40	44	48	52	56	60
I[mm/h]	0 – 1				1 - 10				10 - 100						

Tabulka 1.1 - Barevné rozdělení odrazivostí

1.2 Možnosti řešení

Přístupů k předpovědi srážek z dat meteorologického radaru existuje jistě mnoho. V praxi používané metody mají mnohdy svá označení nebo jména. V různých variacích však většinou používají některý z možných základních principů, které bych zde rád alespoň zhruba zmínil.

Jednou ze základních myšlenek je sledování intenzit srážek v jedné buňce a na základě vývoje v nějakém sledovaném čase odvodit, jaké srážky budou následovat. Metoda je to primitivní a výsledky by tomu také pravděpodobně odpovídaly. Tímto řešením bychom nepoznali spoustu

důležitých momentů. Nepříklad v situaci, kdy se v buňce dlouhou dobu nevyskytují žádné srážky, nelze také žádné srážky předpovědět. K místu se může rychle blížit silná bouřka, kterou touto metodou nejsme schopni zjistit.

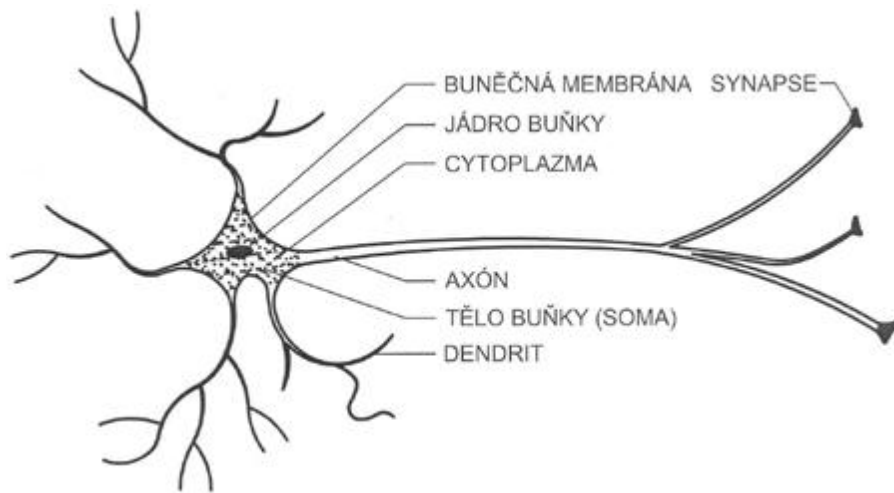
Mnohem vhodnější je sledovat vývoj v jistém okolí buňky. Dále je nutné zjistit jakým směrem se srážky, pokud se v okolí nějaké nacházejí, pohybují. Metod, jak směr pohybu z obrazu zjistit, je mnoho a zde nebudou popisovány, protože k předpovědi byl nakonec zvolen naprosto odlišný přístup. Dále je nutné identifikovat jednotlivé struktury srážek a na základě směru a rychlosti jejich pohybu vypočítat přibližně novou polohu. Zde bychom mohli skončit a při testování bychom zjistili, že metoda je pro krátkou dobu (cca. desítky minut) předpovědi relativně přesná, resp. použitelná (míře úspěšnosti bude věnována samostatná kapitola, navíc její ohodnocení je vcelku nejednoznačné). Pokud by nám šlo o předpověď na delší dobu (v řádu hodin), museli bychom do výpočtu zakomponovat i samotný vývoj srážkových struktur, např. přidávání na intenzitě nebo vypršení mraku.

Zcela odlišný přístup nabízí neuronové sítě. Jejich výhodou jsou schopnosti, které nabízejí, zejména schopnost reprezentace nelineárních funkcí, lze je využít i pro řešení předpovědi deště. Není nutné složitě rozpoznávat srážkové struktury, vypočítávat jejich rychlost a další potřebné faktory. Vystávají zde některé důležité podmínky, které je nutné splnit. Pro vytvoření zobecnění vývoje srážek je zapotřebí mít k dispozici dostatečný počet reprezentativních dat, které by obsahovaly většinu situací, které mohou v atmosféře nastat, nebo alespoň ty, které bychom chtěli umět předpovídat. Další nevýhodou je mnohonásobně větší výpočetní náročnost, než při předpovídání na základě principů popsaných výše. Náročný přitom není jen proces samotného trénování sítě, ale i příprava velkého množství vektorů nutného pro trénování.

1.3 Umělé neuronové sítě

Umělé neuronové sítě jsou inspirovány jejich biologickými ekvivalenty, které jsou základem veškerých informačních systémů živých organizmů. Podstatou neuronových sítí je modelování a napodobení funkce biologických neuronových sítí. Jejich využití nacházíme v případech, kdy lze úloha velmi těžko nebo dokonce vůbec popsat matematicky nebo je její matematické řešení příliš náročné, kdy případná algoritmicizace je téměř nemožná. Neuronové sítě mají vynikající schopnost učit se, tedy vytvořit zobecnění problému na základě předkládaní vzorů. Analogicky jako v biologickém světě, lze později využít znalosti získané učením a zhodnotit další podobnou situaci v řešené úloze. Na rozdíl od matematického popisu problému toto lze navíc i bez znalosti algoritmickeho řešení konkrétní úlohy. Neuronové sítě mají uplatnění hlavně v řešení složitých nebo nelineárních systémů. Škála oblastí použití je velká, lze je využít pro rozpoznávání, analýzu signálů, kompresi a expanzi, kódování a dekodování, transformace signálů a pro predikci časových řad a rozhodování. [2]

Základním prvkem biologické neuronové sítě je neuron. Jeho schopnosti jsou zpracování, uchování a přenos informací ve formě signálů. Schéma biologického neuronu zobrazuje Obrázek 1.2. Neuron se v základech neliší od ostatních typů buněk, skládá se také z buněčné membrány, cytoplazmy a jádra. Od jiných buněk se odlišuje četnými a bohatě rozvětvenými výběžky. Jsou dvojího druhu, první se nazývají axony druhé dendrity. Axony jsou dlouhé a silné a umí rychle přenést informace v podobě nervových impulsů. Na konci se větví na další výběžky zvané synapse, které zajišťují přenos informace k dalším neuronům. Axony jsou výstupem neuronu, který je synapsí spojen s dalšími neurony. Naopak dendrity mají funkci vstupu, protože vedou informace do buňky. Jsou slabší než axony a také mnohem kratší.



Obrázek 1.2 - biologický neuron, (Převzato z [3])

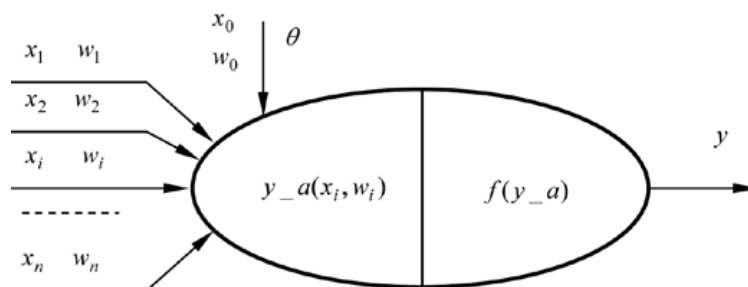
Spojením mnoha neuronů pomocí synapsí vzniká neuronová síť. U člověka je to například mozková kůra, která je tvořena 13 až 15 miliardami neuronů. [3]

2 Trénování neuronové sítě

Na počátku je nutné alespoň v základech pochopit teorii neuronové sítě, která bude trénována pro předpověď deště, a dále zvolit její *topologii*. Samotné trénování neuronové sítě probíhá v několika iteracích zvaných *epocha*. V každé takové iteraci jsou na vstup sítě vystaveny *vstupní trénovací vektory*. Váhy propojující jednotlivé neurony jsou pozměňovány metodou *zpětného šíření chyby*, která je vypočítána jako rozdíl mezi výstupem neuronové sítě a *vzorovým vektorem*, představujícím požadovaný výstup. Při trénování sítě je potřeba měnit mezi jednotlivými epochami *faktor učení*, v závislosti na průběžných výsledcích, aby nedocházelo k nežádoucímu efektu *přetrénování*. Tomu zabraňuje postup zvaný *Crossvalidace*.

2.1 Neuronová síť obecně

Jak již bylo řečeno, umělé neuronové sítě hledají inspiraci v sítích biologických. Poroto i základním stavebním prvkem umělé sítě je umělý neuron. Biologické funkce buňky jsou nahrazeny funkcemi matematickými. Ukázkou matematického modelu představuje Obrázek 2.1. Existuje množství různých variací, které se liší právě těmito matematickými funkcemi. Obrázek 2.1 zobrazuje tzv. základní model neuronu (někdy nazýván podle svých autorů *McCulloch-Pittsův*).



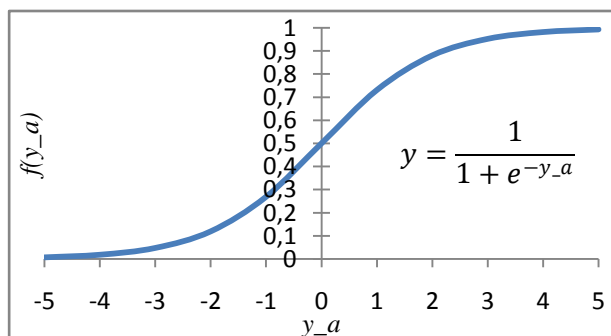
Obrázek 2.1 - Schéma umělého neuronu, (Převzato z [4])

Vstup do neuronu představují signály x_i , které převážně nabývají hodnoty z oboru reálných čísel. Podle umístění neuronu v síti jsou to pak vstupy, které jsou zároveň výstupy jiných neuronů, nebo jsou to vstupy z vnějšího okolí. Dohromady spolu tvoří vstupní vektor $X = [x_1, \dots, x_n]$. Jednotlivé vstupy neuronů jsou dále ovlivňovány váhami spojení w_i , které ve své podstatě určují průchodnost dané cesty a tím také to, jak velké působení bude mít vstup neuronu na jeho výstup. Vhodným nastavením vah může být docílena shoda mezi požadovaným výstupem a výstupem neuronu. Nastavováním nejvhodnějších vah se zabývá proces učení. Prahová hodnota θ neuronu se aktivně podílí na výstupu z neuronu, často slouží k posunutí signálu do aktivační funkce. Tuto funkci ještě předchází agregační funkce $y_a(x_i, w_i)$, která slouží ke sloučení signálů x_i vektoru X . Vektor X je v čase k přetransformován na skalární signál y_a , který pokračuje do aktivační funkce neuronu. Agregační funkci lze matematicky vyjádřit následovně:

$$y_a(k) = \sum_{i=0}^n x_{i(k)} \cdot w_{i(k)} , \quad (2.1)$$

Aktivační funkce $f(y_a)$ převádí vstupní potenciál y_a na výstupní hodnotu z neuronu. Konkrétních forem přenosových funkcí je více a vždy je potřeba použít některou vhodnou podle typu

aplikace neuronové sítě. Nejčastěji používanou je funkce *skoková* nebo např. *sigmoidní*, jejíž průběh je vidět v grafu 2.1. [4]



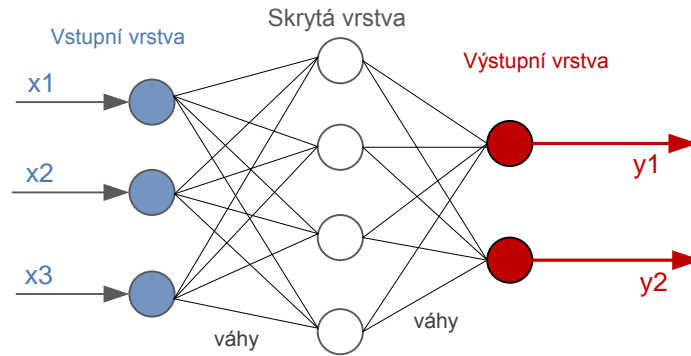
Graf 2.1- Průběh sigmoidní funkce

Spojení několika neuronů vytváří neuronovou síť. Ta se dá popsat jakýmkoliv orientovaným grafem, kde vrcholy jsou tvořeny neurony a hrany zmíněnými propojeními s jistou vahou. Způsob, jakým jsou neurony propojeny, tzn. který je spojen se kterým apod., se nazývá *topologie* sítě. Existuje více typů topologií. Skoro vždy se však používá vrstevnatá architektura. Neurony jsou rozděleny do vrstev, které se podle pozice označují jako vstupní, jedna nebo více skrytých a výstupní. Počet vrstev spolu s počty neuronů ve vrstvách tvoří *konfiguraci* sítě. Podle toku signálu se síť dělí na rekurentní a dopředné, kterých bude dále využíváno. Neurony z každé vrstvy jsou spojeny s neurony (většinou všemi) vrstvy sousední. Proces učení potom spočívá v postupném nastavování vah těchto spojení a lze ho popsat jako algoritmus, který určuje jakým způsobem budou váhy nastavovány. Základem učících algoritmů bývá tzv. *Hebbův zákon učení*[Donald Hebb,1949] a lze ho vyjádřit následující rovnicí:

$$\Delta w_{ij} = \alpha \cdot x_i(k) \cdot x_j(k) , \quad (2.2)$$

která udává změnu váhy mezi dvěma neurony i a j jako zesílení v případě, že jsou oba neurony v čase k aktivní, a v případě, že jsou oba v neaktivním stavu, jako zeslabení, kde α je důležitá konstanta udávající rychlost učení. Učení může probíhat jako učení s učitelem nebo učení bez učitele podle způsobu aplikace. Učení s učitelem je v praxi použito, pokud existují dvojice vektorů vstupních a výstupních, kde vstupní vektor je zpracován neuronovou sítí a její výstup může být porovnán se vzorem. Váhy mezi neurony jsou potom modifikovány na základě chyby zjištěnou jako rozdíl mezi požadovanou a získanou hodnotou.[5]

Při řešení předpovědi deště bude využito vícevrstvé dopředné neuronové sítě. Její zjednodušené schéma zobrazuje Obrázek 2.2. Proces trénování dopředné neuronové sítě s učitelem spočívá v opakovaném přikládání vzorků na vstup sítě. Tyto vzorky se nazývají trénovací množina a musí být dostatečně reprezentativní, aby bylo dosaženo chtěného chování sítě. Průchod celé trénovací množiny je označován jako epocha.[6] Jednotlivé epochy se opakují tolikrát, kolikrát určuje zvolený postup při učení, většinou do té doby, dokud se výsledky sítě zlepšují, nebo je dosaženo dostatečného zmenšení chyby mezi výstupem sítě a výstupem požadovaným. Průběžné výsledky je vhodné zjišťovat na tzv. *Crossvalidační* množině, což je část trénovací množiny, na které se učení neprovádí, a která slouží jen pro ověření funkčnosti sítě na datech, která síť při učení neviděla. Během jednotlivých epoch trénování je vhodné měnit konstantu α , určující rychlost učení, podle průběžných výsledků.



Obrázek 2.2 - Schéma neuronové sítě

Pro učení dopředných sítí s učitelem existuje algoritmus zpětného šíření chyby E , tzv. *zpětně propagační algoritmus* (z angl. *Back Propagation Algorithm*). Je založen na její minimalizaci, vyjádřené jako kritériální funkce ve tvaru

$$E(p_e) = \sum_{k=1}^N e(p_e, k), \quad (2.3)$$

kde p_e je pořadové číslo epochy trénování, N je počet vzorků v trénovací množině, k je pořadové číslo takové množiny a $e(p_e, k)$ je přírůstek kritériální funkce mezi jednotlivými vzorky v trénovací množině, a kde

$$e(p_e, k) = \frac{1}{2} \sum_{h=1}^m \varepsilon_h^2(p_e, k) = \frac{1}{2} \sum_{h=1}^m (y_{s_h}(k) - y_{M_h}(p_e, k))^2, \quad (2.4)$$

kde m je počet výstupů sítě, h je h -tý výstup, $\varepsilon_h^2(p_e, k)$ jsou přírůstky kritériální funkce od jednotlivých výstupů, $y_{M_h}(p_e, k)$ jsou hodnoty výstupů sítě a $y_{s_h}(k)$ jsou požadované hodnoty na výstupu.[6] Chyba je potom propagována zpět přes všechny neurony, na jejímž základě jsou pozměněny váhy vstupů konkrétního neuronu tak, aby tato chyba byla minimalizována. Pro každý uzel i v předcházející vrstvě uzlu j je potom vypočítána nová chyba e_i na základě chyby e_j a váhy spojující uzly i a j . Takto je vytvořena chyba e pro každý uzel. Nová váha w' uzlu i , spojující ho s uzlem v předcházející vrstvě (v případě vstupní vrstvy vnějším vstupem), je potom vypočítána na základě váhy stávající w , chyby e_i , konstanty α udávající rychlost učení a hodnoty y , která je výstupem předchozího uzlu s nímž je neuron spojen váhou w . Aktivační funkce neuronu je $f(y_a)$ a y_a je výstupem agregační funkce, tedy vstupní potenciál neuronu. Výpočet nové váhy lze potom popsat následovně:

$$w = w' + \alpha \cdot e_i \frac{df(y_a)}{dy_a} y \quad (2.5)$$

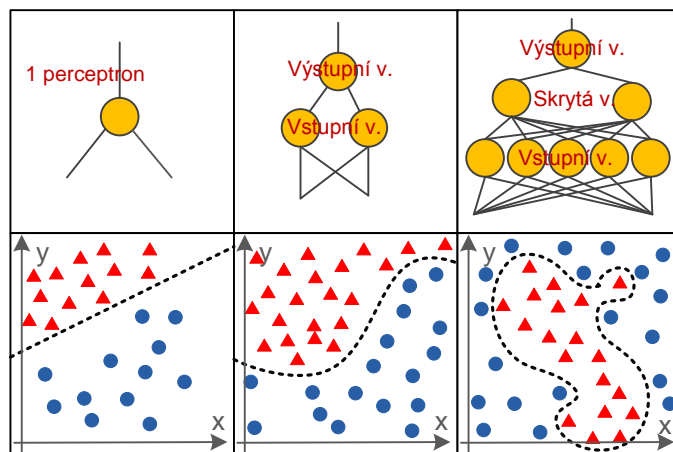
2.2 Vícevrstvá perceptronová síť

Neuronové sítě jsou využívány pro rozpoznávání (klasifikaci). Klasifikace sestává ze dvou fází. V první se získají informace z předloženého vzoru. Těmito informacemi jsou vlastnosti předkládaných vzorů, jako může být například délka nebo barva nějakého objektu. Ve druhé fázi se

na základě získaných znalostí určuje zařazení konkrétního vzoru do skupiny(třídy) vzorů s podobnými vlastnostmi. Při klasifikaci je často potřeba získat hned několik vlastností vzoru naráz. Tyto potom tvoří tzv. vektor vlastností, jejichž počet udává dimenzi, ve které se klasifikace provádí.[7]

Základním prvkem dopředných neuronových sítí je *perceptron*. Perceptron již sám o sobě může pracovat jako jednoduchý binární klasifikátor. Počet vstupů perceptronu určuje dimenzi klasifikace. Pokud jsou vzory v takové dimenzi oddělitelné jednou rovinou, lze je pomocí samotného perceptronu klasifikovat a rozdělit do dvou tříd. Jinak řečeno, jeden perceptron může udávat lineární hranici v klasifikačním prostoru mezi dvěma skupinami vzorů. Při spojení více perceptronů ve dvou vrstvách, kdy neurony vrstvy vstupní jsou spojeny s neurony vrstvy výstupní, lze předkládané vzory klasifikovat do tříd, mezi kterými může být hranice v klasifikačním prostoru určena např. polynomem. Perceptronová síť, která obsahuje jednu nebo více vrstev skrytých, je potom schopna aproximovat zcela libovolnou nelineární funkci, která určuje hranici mezi třídami. Taková síť se nazývá *Vícevrstvá perceptronová síť*, označuje se *MLP*(z angl. *Multilayer Perceptron Network*) a je dále využita při předpovědi deště.

Následující obrázek (Obrázek 2.3) zobrazuje využití různých vrstev perceptronů pro účely klasifikace. Vzory mají dvě vlastnosti, které určují hodnoty na osách x a y . Jsou dvou druhů a účelem učení je nalezení hranice mezi nimi, aby bylo možno v budoucnu správně klasifikovat na neviděných vzorech.



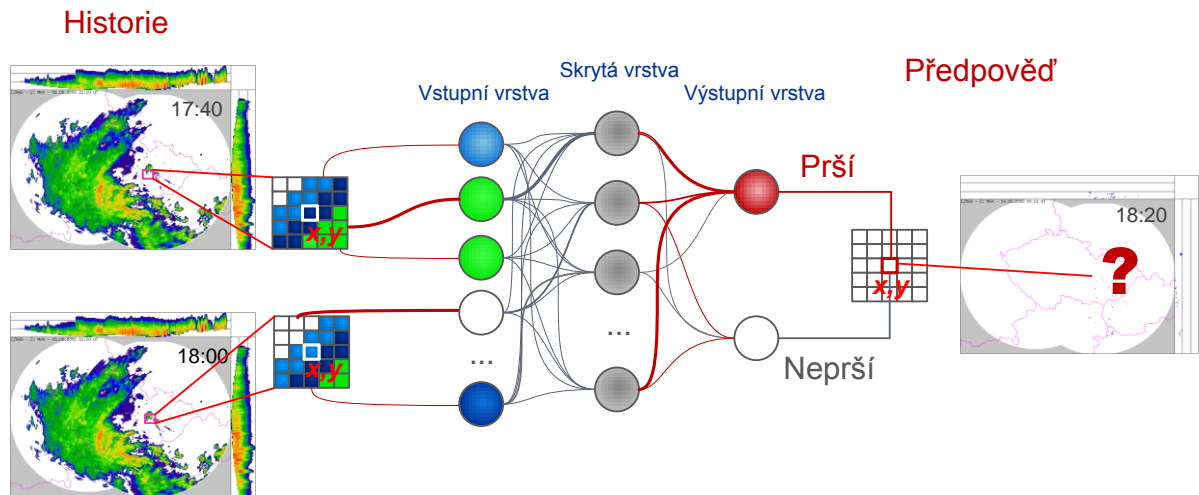
Obrázek 2.3 - Perceptron při klasifikaci

2.3 Neuronová síť při předpovědi

Na počátku je nutné zvolit topologii neuronové sítě a definovat, v jaké podobě budou síti předány informace s radarových snímků. Pro řešení byl vybrán zatím nejpoužívanější typ sítě, dopředná perceptronová vícevrstvá síť (viz. předchozí kap.) s jednou skrytou vrstvou. Snahou je, zvolit co nejmenší konfiguraci sítě kvůli časové náročnosti, ale také složitosti učení a přípravy dat, přitom dostatečně velkou tak, aby síť byla schopna vstřebat požadované znalosti.

Informace z radarových obrázků jsou převedeny na číselné hodnoty způsobem využívajícím fakt, že každá barva odpovídá určité hodnotě odrazivosti srážek v atmosféře. Základní myšlenkou celého systému je přivést na vstup neuronové sítě číselné hodnoty odrazivosti z několika radarových měření, která byla naměřena v určitém čase před okamžikem, který bude předpovídán (tato data

budou dále označována jako *historie*, předpovídaný okamžik jako *předpověď* v *budoucnosti*). Ve výstupní vrstvě neuronové sítě bude požadována informace, zda bude či nebude pršet, při pokročilejším řešení pak i informace o intenzitě srážek. Z důvodu zjednodušení celého systému a následně možného praktického řešení bude předpověď probíhat pro jednu buňku, odpovídající reálné ploše 1x1 km a zároveň jednomu obrazovému bodu v radarových snímcích na pozici x,y , kterou bude takový bod dále obecně popisován. Rovněž na vstup nebudou vystavena veškerá data z radarových snímků v *historii*, nýbrž jen jistý počet buněk okolo buňky předpovídané. Teorii přípravy dat se bude zabývat kapitola 3., kde bude mimo jiné vysvětleno, jak velké toto okolí musí být. Na obrázku 2.4 je názorně ukázáno nyní popsané řešení.



Obrázek 2.4 - Neuronová síť při předpovědi

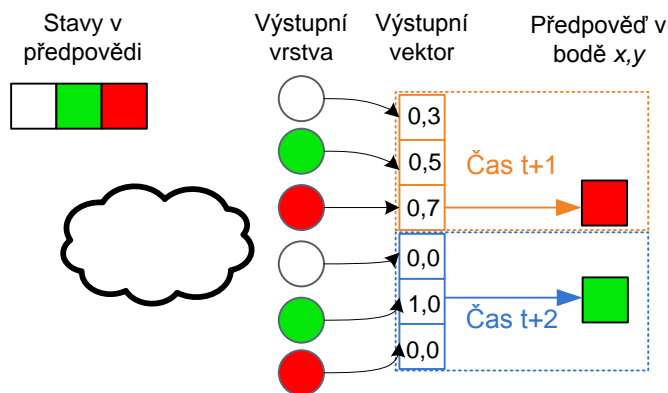
Obrázek 2.4 zobrazuje dvouvrstvou neuronovou síť¹. Jejím vstupem jsou části dvou radarových snímků naměřených v časech 17:40 a 18:00. Barevná informace ze snímků je převedena na číselnou hodnotu. Natrénovaná síť má nastaveny váhy mezi jednotlivými neurony ve vrstvách, které jsou v obrázku pro ilustraci znázorněny a popsány modrým písmem. Výstupní vrstva obsahuje dva neurony, které zde jsou za účelem určení stavu buňky v čase 18:20, v tomto konkrétním případě pro určení, zda *Prší* nebo *Neprší*. V případě, že je hodnota výstupu prvního neuronu vyšší, znamená to, že síť předpověděla stav *Prší* pro červeně zvýrazněnou buňku.

Přesný tvar výstupního vektoru sítě bude popsán v kapitole 3. o tvorbě dat, protože má stejný tvar jako vzorový trénovací vektor. Je však vhodné zde zmínit, jak je vytvořen. Je sestaven z hodnot výstupů výstupních neuronů. Důležité je, kolika neurony je tvořena výstupní vrstva a jaký mají význam. Ve výstupní vrstvě se nachází tolik neuronů, kolik chceme rozlišovat stavů v předpovědi. Například podle obrázku 2.4, kdy mohou nastat pouze dva stavy (*Prší*, *Neprší*), jsou ve výstupní vrstvě neurony také dva. Každý potom reprezentuje jeden stav. Ve výstupním vektoru budou dvě hodnoty, přitom vyšší z nich označuje stav neboli třídu do které neuronová síť případ zařadila, a která je potom označena za předpověď. Stavů předpovědi může být přitom více, kde potom záleží na konkrétním požadavku. Lze například předpovídat stavy tři, kdy jeden by znamenal, že v buňce neprší, druhý by znamenal, že poprchá, a třetí by značil silný déšť.

Takto je předpověď vytvořena pro jeden okamžik v budoucnu. Je ale možné vytvořit předpověď i na několik takových okamžiků zároveň. Ve výstupní vrstvě je potom tolik neuronů, kolik může nastat stavů ve všech okamžicích předpovědi. Tyto neurony jsou logicky rozděleny podle doby. V každém takovém logické úseku je vybrán neuron s nejvyšším výstupem, který značí předpověď pro

¹ vstupní vrstva není započítána

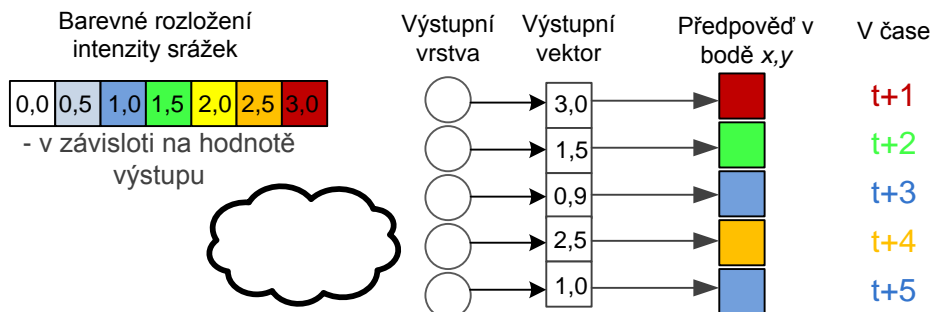
danou dobu. V takovém případě neuronová síť klasifikuje případ do více tříd najednou. Obrázek 2.5 je částečně podobný předchozímu obrázku 2.4, avšak je více zaměřen na výstupní vektory. Jedná se o předpověď na dva okamžiky v budoucnu v časech $t+1$ a $t+2$. Tři stavy v předpovědi jsou pro názornost rozlišeny barevně.



Obrázek 2.5 - Výstup neuronové sítě při předpovědi

Při běžném použití neuronových sítí pro klasifikaci je často za výstupní vrstvu přidána nelineární funkce zvaná *SoftMax*, která normalizuje jednotlivé výstupy tak, aby se jejich součet rovnal číslu 1. V praxi má tato funkce význam takový, že výstup jednotlivých neuronů po zpracování funkcí *SoftMax* přímo udává pravděpodobnost, s jakou by neuronová síť zařadila konkrétní případ do jaké třídy. Problém nastává, pokud je třeba vzor zařadit do více tříd zároveň, což je právě případ předpovědi deště na více okamžiků (viz Obrázek 2.5). Při takovém použití funkce *SoftMax* nelze výstup považovat přímo jako pravděpodobnost zařazení do třídy, protože správných tříd je více a hodnoty výstupu jsou mezi těmito třídami rozděleny. Rozhodování mezi vítěznými třídami, tak jak je v této práci popisováno, funguje naprosto stejně s i bez funkce *SoftMax*. Ve výstupním vektoru jsou vítězné třídy označeny podle výstupů mající nejvyšší hodnotu, a nikoliv podle pravděpodobnosti zařazení.

Existuje ještě další možnost využití výstupních neuronů, která zatím skončila pouze ve fázi návrhu a je otázkou dalšího vývoje. Tuto možnost ilustruje Obrázek 2.6. Výstupních neuronů je pouze tolik, kolik je okamžiků v předpovědi. Výstup každého neuronu potom neurčuje třídu intenzit srážek, nýbrž přímo samotnou hodnotu intenzity. V takovém případě by již funkce *SoftMax* byla nepoužitelná.

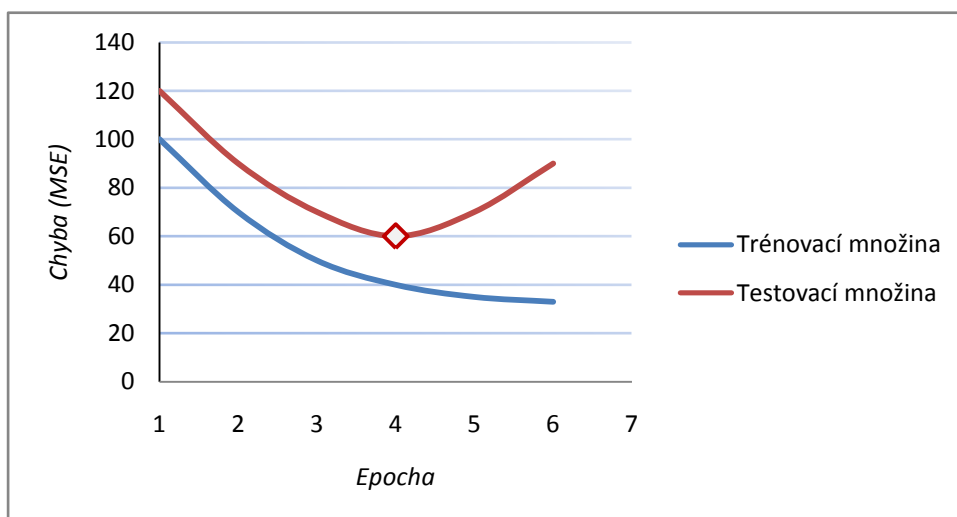


Obrázek 2.6 - Výstup určující intenzitu srážek

2.4 Trénování

Nejpodstatnější částí při řešení úlohy je natrénovat takovou neuronovou síť pro žádanou předpověď. Trénování se provádí předkládáním dvojic vstupních a vzorových vektorů trénovací množiny. Trénování probíhá na situacích, které jsou již naměřeny a převedeny do podoby radarových snímků, proto je v době trénování k dispozici jak *historie* tak i *předpověď*.

Trénování sítě probíhá v několika epochách, během nichž je nutné sledovat průběžné výsledky. V další části 2.5 bude popsáno jedno z hlavních úskalí neuronových sítí, jehož důsledek je nutné v průběhu trénování pohlídat a předejít mu. Je potřeba také určit na kolika dvojicích vektorů se bude trénovat. To je často individuální a počet je nejlepší zvolit na základě experimentů nebo zkušenosti. Základní poučka říká, že velikost trénovací množiny by měla desetkrát až dvacetkrát přesahovat celkový počet všech spojení mezi neurony sítě. Důležitým faktorem je také počet epoch, ve kterých bude síť trénována. Tento počet lze sice samozřejmě stanovit předem, je ale výhodnější, trénovat neuronovou síť, dokud je schopná se dále učit. Tento poznatek lze získat průběžným ověřováním výsledků v každé iteraci. Kvůli hrozbě přetrénování je ale nutné výsledky zjišťovat na jiných datech než trénovacích. Graf 2.2 obsahuje zvýrazněné místo, kde je vhodné trénování ukončit. Je vidět, že z výsledků získaných na trénovacích datech, by toto místo nebylo poznat. Metoda výpočtu chyby v grafu je označena jako *MSE* a bude podrobně vysvětlena v kapitole o testování (Kap.4.).



Graf 2.2 - Průběžné testování výsledků sítě

2.5 Crossvalidace

Metoda průběžné kontroly výsledků je běžně používána při trénování neuronových sítí. Česky by se tato metoda dala nazvat *křížové porovnání* (volně přeloženo z angl. *Cross-validation*, dále v textu bude označována jako *Crossvalidace*). Její smysl spočívá v průběžné kontrole učení neuronové sítě během jednotlivých epoch trénování, mezi kterými by síť měla produkovat stále lepší výsledky. V případě, že by výsledek předchozí iterace byl lepší, než výsledek aktuální iterace, je to známka tzv. syndromu *přetrénování* (angl. *Overtraining*). V biologickém světě tento stav nastává, pokud dlouhodobý nebo intenzivní trénink není doprovázen dostatečným časem na zotavení a odpočinek. Jistým znakem je potom snížená obecná výkonnost.[8] U umělých neuronových sítí může také nastat

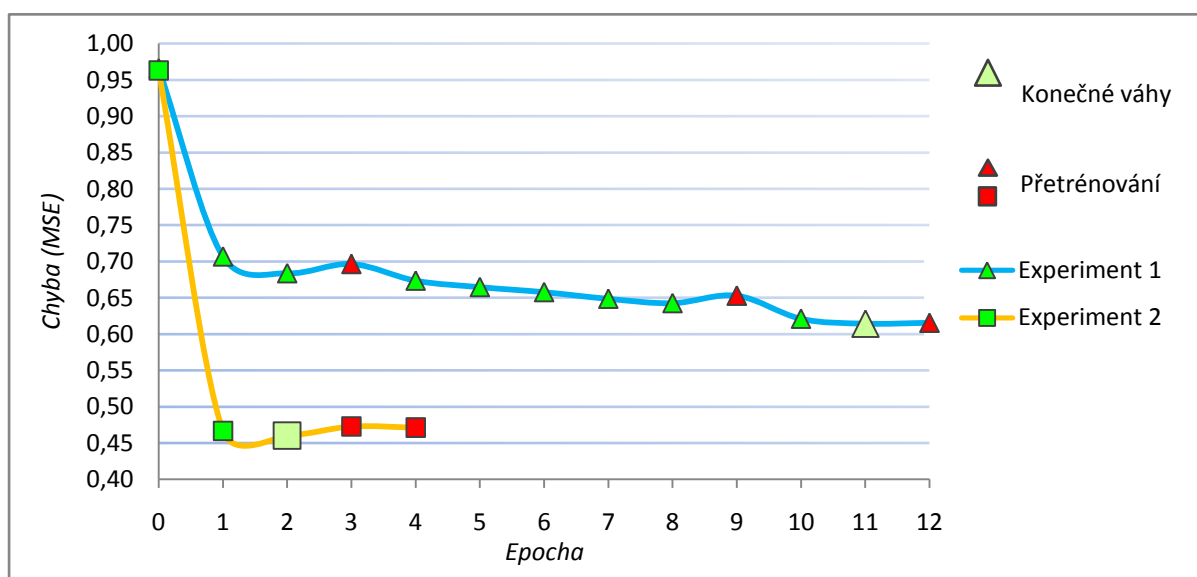
situace, která lze popsat jako stav přetrénování. Příčina vzniku je podobná jako v biologickém světě, avšak důsledky jsou trochu odlišné.

Stav nastává po velmi intenzivním trénování, tedy v případě vysokého faktoru učení α . Bohužel nelze nikdy přesně předem stanovit hranici, kdy dochází k přetrénování, protože se tato hranice v každé iteraci trénování mění. Například v první iteraci je síť úspěšně natrénována s faktorem α nastaveným na hodnotu, která v iteraci páté již může způsobit přetrénování. Z tohoto důvodu je použita metoda Crossvalidace, pomocí níž lze zmiňovaný nežádoucí stav rozpoznat.

Samotná podstata přetrénování spočívá v tom, že neuronová síť se nenaučí obecné charakteristické rysy, nýbrž spíše jemné odlišnosti a detaily, které odpovídají jen trénovací sadě. Neuronová síť si výborně zapamatuje trénovací sadu, ovšem není již schopna zobecnit data neviděná při trénování.[9] Je proto nutné ověřovat průběžné výsledky sítě pomocí sady vektorů, díky níž nebyly nastavovány váhy v průběhu trénování. K tomuto účelu je z trénovací sady vektorů oddělena část, která se nazývá Crossvalidační množina vektorů. Jejím úkolem je vždy pouze ověření výsledků nikoliv samotné trénování.

Váhy, které byly nastaveny v iteraci, ve které došlo k přetrénování, jsou zahozeny, a trénování je nutné buď ukončit, nebo trénovat dále síť natrénovanou v předchozí iteraci, avšak již se sníženým faktorem učení.

Postup ohodnocení průběžných výsledků bude popsán v kapitole 4. V grafu 2.3 je ukázka průběhu trénování dvou různých neuronových sítí, chyba je zde počítána metodou MSE (viz. kap.4.4).



Graf 2.3 - Průběžné testování na Crossvalidační množině

2.6 Rychlost učení

Obecnou snahou je co nejlépe natrénovaná síť. Toho lze dosáhnout co nejvyššími možnými hodnotami faktoru učení. Jelikož ale hrozí riziko přetrénování, zmíněné v předchozí kapitole, je nutné ho nějakým způsobem mezi jednotlivými epochami měnit. K řízení změn tohoto faktoru se používá mnoho algoritmů. Jedním z jednodušších a často používaných je algoritmus s názvem *New-Bob*. [10]

Před započítáním první iterace trénování je nastavena rychlost učení na jistou hodnotu. Následně je provedena první iterace, po které je provedena Crossvalidace a tím je zjištěn mezivýsledek. Kvůli porovnání úspěšnosti hned po první iteraci je nutné předem znát výsledky Crossvalidace ještě před

započetím trénování, ve chvíli, kdy jsou ještě váhy nastaveny náhodně. Pokud je přírůstek úspěšnosti menší jak 0,5%, je nutné snížit rychlost trénování, nejčastěji o $\frac{1}{2}$. V případě, že je výsledek větší alespoň o 0,5%, rychlost učení zůstává stejná. Je provedena další iterace trénování a po ní je opět zjištěn výsledek na Crossvalidační množině. Pro změnu rychlosti opět platí stejné zásady. Trénování probíhá tímto způsobem pořád dále. Pokud nastane situace, že je přírůstek úspěšnosti při Crossvalidaci dvě iterace po sobě menší než 0,5%, trénování je skončeno a nastavení vah považováno za nejlepší možné.

3 Příprava dat

Zdroje dat byly získány od *Českého hydrometeorologického úřadu* ve formě radarových snímků popsaných v kapitole 1.1. Jednalo se o sady obrázků zaznamenávajících různé frontální přechody přes území České republiky. Obrázky obsahovaly informace o srážkových odrazivostech zcela v horizontálním směru, ale i částečně ve směru vertikálním ve formě dvou průmětů prostorem severojižním a západovýchodním směrem. Bylo by možné vytvořit trojrozměrný vjem o intenzitách srážek v prostoru. Protože by zpracování takových informací vyžadovalo poněkud složité operace, bylo využito pouze dat plošného rozložení odrazivosti v horizontálním směru. Použitá oblast byla z barevných stupňů v obraze převedena na číslo a uložena v textové podobě pro lepší manipulaci.

Důležité je upozornit, že radarové snímky nejsou naprosto přesné. Nejen, že nemusí vždy naprosto přesně odpovídat realitě, navíc se v nich čas od času objevují různé vady nebo chyby. Z obrázků bylo na první pohled evidentní, že jeden ze dvou radarů používaných *ČHMÚ* občas postihl výpadek a tím pádem nebyly k dispozici informace z části území a byl narušen celkový průběh vývoje srážek. Při trénování na takových datech by se mohlo stát, že neuronová síť by se mohla naučit situaci, která byla způsobena chybou a ne opravdovým vývojem. Proto bylo nutné taková data úplně vynechat a rozdělit tím tak kontinuitu vývoje srážek v čase. Dále se ve snímcích objevily následky záchvěvů radaru, odrazy od pozemských objektů, či další nemeteorologické cíle jako například letadla apod. Jelikož je jednalo o výjimečné jevy, nebyly tyto chyby nijak odstraněny a byly považovány za šum. Neuronové sítě si navíc dokáží poradit s drobným šumem.

Pro účely trénování i testování byly použity snímky celkem ze 7 dnů od r. 2002 do r. 2007, ve kterých byly záznamy pořizovány s časovým odstupem *10 minut*. V jednom dni bylo *144* radarových snímků, dohromady bylo použito okolo *1000* snímků. Počet je to jistě velký, ale ani tolik obrázků nemůže zaznamenávat většinu situací, které mohou v přírodě v souvislosti s deštěm nastat. Pro správné natrénování neuronové sítě, tedy pro pochopení alespoň většiny dějů, které lze z radarových snímků vyčíst, by byl zapotřebí mnohem větší počet příkladů. Tato práce se spíše zaměřuje na schopnosti neuronových sítí předpovídat dešť, než na natrénování konkrétní neuronové sítě, která by dokázala předvídat dešť v každé situaci.

V použitých snímcích bylo dohromady zhruba 60-70% buněk, ve kterých se nenacházeli žádné srážky. Je potřeba si tento fakt uvědomit při vytváření trénovacích vektorů, protože jinak by se síť v podstatě přetrénovala na předpovídání sucha namísto deště. Případů, kdy se v celé trénovacím vektoru nevyskytovaly žádné srážky, bylo pro trénování sítě použito jen 10%, ostatní takové vektory byly zahozeny. Tím se hodně zmenšil maximální počet vytvořených vektorů, což v konečném důsledku mělo limitující vliv na konfiguraci neuronové sítě, zvláště pro předpověď na delší dobu dopředu.

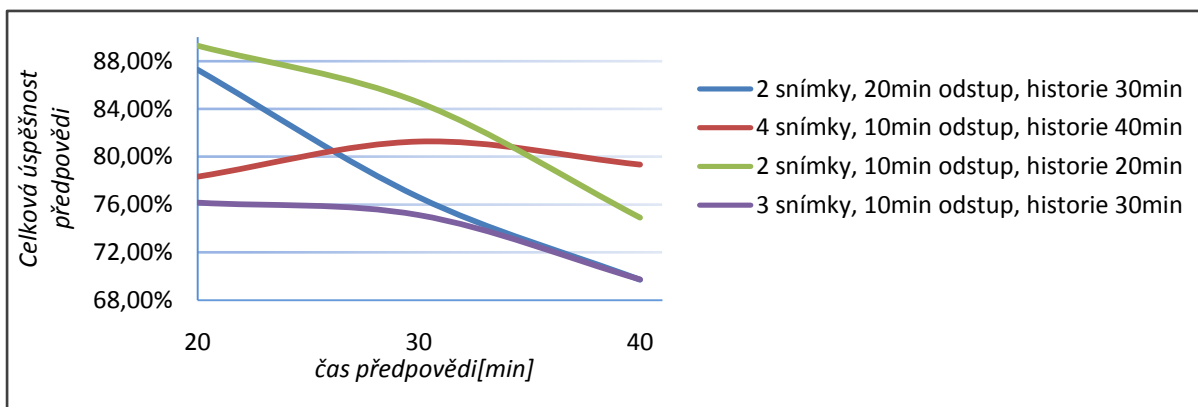
Počet vstupních resp. výstupních neuronů odpovídá velikosti vstupního resp. výstupního vektoru, který je ovlivněn mnoha faktory. Tyto faktory budou dále podrobně popsány.

3.1 Historie dat a předpověď

Jak bylo řečeno, předpověď probíhá jen pro jednu buňku z celé sledované oblasti. Z hlediska časového si lze vybrat. Předpověď lze provádět na dobu v intervalech po 10ti minutách, přičemž je možné jednu neuronovou síť naučit předpovídat na více dob najednou tak, že se zvětší výstupní vektor. Větší počet neuronů ve výstupní vrstvě však zvětšuje chybu předpovědi, proto je vhodnější

pro předpověď na více časových okamžiků natrénovat více neuronových sítí, orientujících se právě na jeden okamžik.

Historií radarových dat je myšlen celkový počet použitých obrázků pro předpověď. Tento počet by se měl odvíjet od doby předpovědi. Důležitým faktorem jsou také časové odstupy mezi jednotlivými snímky. Nejvhodnější časové rozpětí a počet použitých obrázků v historii nelze nijak vypočítat nebo dokázat, lze jej jen odhadnout nebo experimentálně vyzkoušet, viz Graf 3.1. Výpočet celkové úspěšnosti je uveden v kapitole 4.1, tento graf slouží jen pro ilustraci vlivu použité historie při předpovědi.



Graf 3.1 - Vliv použité historie na úspěšnost předpovědi

Není nutné nějakým způsobem neuronové síti předávat informaci o tom, na jak dlouho předpovídáme, jaké je časové rozpětí mezi jednotlivými částmi² vstupního vektoru ani kolik je zkombinováno předpovědí do jedné. Vstupní a výstupní vektory budou předkládány z pohledu sítě nezávisle na tom, co reprezentují. Z pohledu řešitele však musí být všechny trénovací i testovací vektory pro jednu síť vytvořeny vždy se stejnými parametry, což platí i pro všechny vlastnosti zmíněné dále.

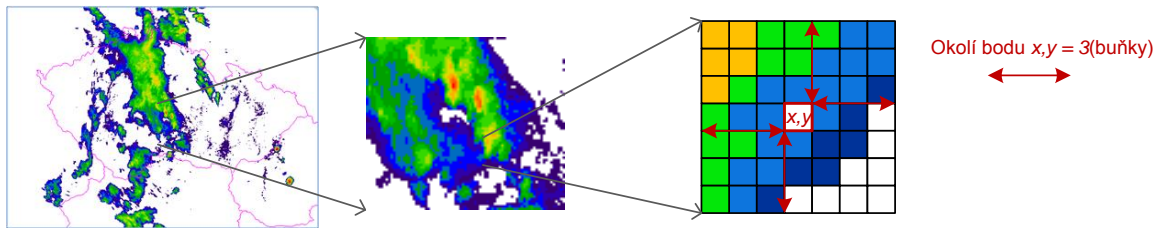
Protože funkce neuronových sítí spočívá v klasifikaci, výstupní vektor je logicky rozdělen na několik tříd. V případě předpovědi zda prší či neprší se jedná o třídy dvě. Aktivnější výstup s větší hodnotou, potom značí předpověď. Je nutné udržovat uspořádání tříd vždy ve stejném pořadí. Při předpovídání intenzit srážek je použito tříd tolik, kolik je předpovídáno různých intenzit a jedna navíc, znamenající nulové srážky. Na výstupu neuronové sítě je potom vybrána pouze jedna taková třída, která je označena za předpověď. V případě předpovídání více časových období je výstupní vektor logicky rozdělen na více bloků, v němž je vždy vybrána jen jedna třída tvořící předpověď pro jednu dobu.

3.2 Okolí bodu a snížení rozlišení dat

Při použití celých radarových snímků by byl počet neuronů ve vstupní vrstvě příliš veliký. Proto je pokaždé zvolen jen určitý počet okolních buněk sledovaného bodu, ze kterého je následně vytvořen vstupní vektor. Na pořadí hodnot nezáleží, ovšem pro jeden experiment musí být buňky každého vektoru přiřazeny těm stejným vstupním neuronům. Obrázek 3.1 ilustruje okolí bodu x,y . Z celého radarového snímku je vybrána pouze část buněk (v obr. jedna buňka odpovídá jednomu čtverečku)

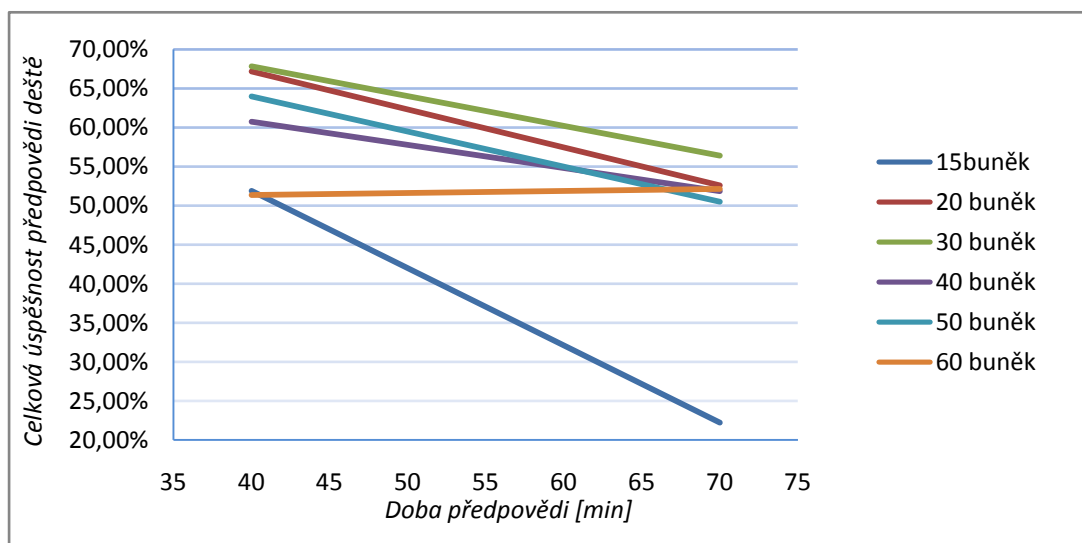
² jednotlivé hodnoty z radarových obrázků použitých v historii jsou nakonec spojeny do jednoho vstupního vektoru

obklopující sledovaný bod, zde konkrétně tři okolní buňky, celkem by bylo z takového snímku použito 49 buněk.



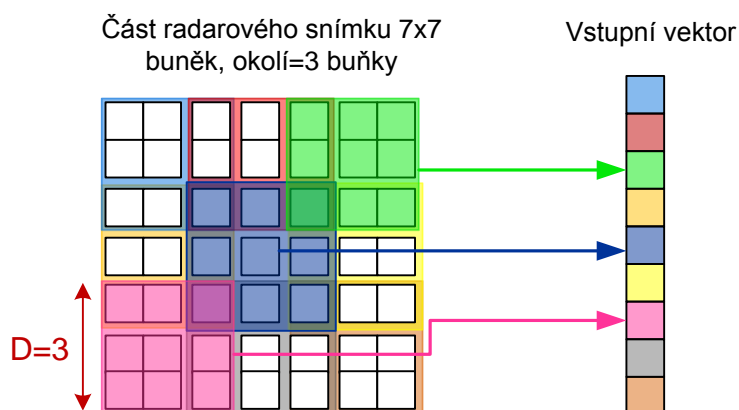
Obrázek 3.1 - Okolí bodu x,y

Na velikosti okolí bodu záleží úspěšnost predikce. Důležitou otázkou proto je, jak velké okolí zvolit. Opět nelze zcela jasně říci, pro jakou dobu predikce je jaká velikost okolí nevhodnější. Je ale vhodné předpokládat, že pro správnou funkčnost systému je zapotřebí použít okolí o takové vzdálenosti od sledovaného bodu, jakou jsou schopny urazit dešťové mraky za dobu pro předpověď, nebo za dobu mezi jednotlivými snímky v historii. Jinak řečeno je potřeba, aby mezi srážkovými útvary, které se nacházejí buď na jednotlivých snímcích v historii nebo nad sledovaným bodem v předpovědi, existovala souvislost, kterou je právě neuronová síť schopna se naučit. Vhodné okolí lze sice přibližně vypočítat pomocí doby mezi jednotlivými časovými okamžiky v systému a odhadu rychlosti dešťových mraků, která je však proměnlivá. Proto se později vhodnějším ukázal opět experimentální odhad. Z grafu 3.2 lze dále vyzorovat, že i příliš velké okolí bodu má na úspěšnost předpovědi negativní vliv.



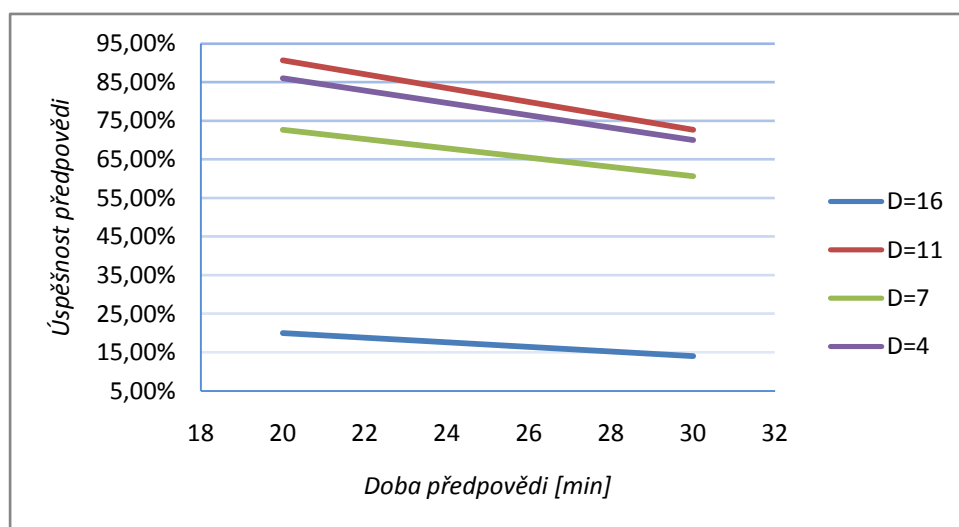
Graf 3.2 - Vliv velikosti okolí na úspěšnost předpovědi

Delší doba předpovědi vyžaduje větší okolí. Při zvětšení použitého okolí se ale vstupní vrstva neuronové sítě zvětšuje kvadraticky. Příliš velká neuronová síť se trénuje hůře než síť menší. Je proto navržen mechanismus zmenšení rozlišení vstupních dat. Nesmí přitom nedocházet ke ztrátě cenných informací tím, že by byly vynechávány některé buňky. Mechanismus zobrazuje Obrázek 3.2. Funguje na principu sloučení několika buněk v jednu hodnotu, která je potom použita ve vstupním vektoru. Hodnota je vypočítána jako aritmetický průměr odrazivosti srážek v buňkách.



Obrázek 3.2 - Mechanismus zmenšení rozlišení vstupních dat

Je nutné zvolit faktor zmenšení rozlišení D - počet buněk, které budou sloučeny. Vliv různých faktorů D byl vypořizován experimentálně viz Graf 3.3. Jeho hodnota byla v experimentech zvolena vždy taková, aby neuronová síť měla rozumnou velikost, přitom vždy také záleží na použitém okolí. V experimentech zobrazených v grafu 3.3 je v okolí použito 15 buněk. Větší síť se trénuje neúměrně delší dobu a je k tomu zapotřebí větší množství dat, které by v některých případech nebylo možné z disponovaných radarových snímků vůbec vytvořit.



Graf 3.3 - Vliv faktoru D na úspěšnost předpovědi

3.3 Rozdělení intenzit do tříd

Vstupní vektor obsahuje zprůměrněné hodnoty odrazivosti. Ve výstupním a vzorovém vektoru je už zapotřebí rozlišit jednotlivé srážkové intenzity. Každé intenzitě ve výstupním vektoru odpovídá jeden výstupní neuron, dohromady tyto neurony určují srážkové třídy. Čím více srážkových tříd je vytvořeno, tím horší jsou potom výsledky předpovědi. Je to způsobeno mimo jiné také tím, že srážkové třídy jsou nerovnoměrně zastoupeny ve výstupních trénovacích vektorech. Kvůli exponenciální závislosti odrazivosti a intenzity srážek je složité dosáhnout rovnoměrného zastoupení. Navíc se v radarových snímcích vyskytují v převážné většině právě nízké hodnoty odrazivosti. Pro

vytvoření dokonalé množiny trénovacích vektorů by bylo zapotřebí velké množství radarových dat. Obecné vyjádření rozdělení intenzit do n tříd dle prahových hodnot $prah_n$ zobrazuje výraz 3.1.

$$třída(Z) = \begin{cases} 1, pro Z < prah_1 \\ 2, pro Z < prah_2 \wedge Z \geq prah_1 \\ \dots \\ n, pro Z \geq prah_{n-1} \end{cases} \quad (3.1)$$

Při řešení této práce bylo méně dat vynahrazeno rozdělením intenzit do malého množství tříd tak, aby trénovací vektory obsahovaly přibližně stejné množství příkladů od každé třídy.

3.4 Vstupní a výstupní vektory

Jaké parametry ovlivňují vstupní a výstupní vektory bylo vysvětleno v předcházejících kapitolách. Konfigurace neuronové sítě je přímo závislá na velikosti vstupních a výstupních vektorů. Velikost vstupního vektoru lze v případě sníženého rozlišení radarových obrázků faktorem D vypočítat podle vzorce 3.2.

$$In = h \cdot \left(\frac{2 \cdot s}{D - 1} \right)^2 \quad (3.2)$$

V případě, že rozlišení není sníženo, když $D=1$, lze použít vzorec 3.3. V rovnicích je použito následujících proměnných: D – faktor snížení rozlišení, s – použité okolí bodu, h – počet snímků v historii a In je počet neuronů ve vstupní vrstvě.

$$In = h \cdot (2s + 1)^2 \quad (3.3)$$

Velikost *Out* výstupní vrstvy odpovídá podle rovnice 3.4 počtu okamžiků v předpovědi *future* násobených počtem *nclasses* tříd srážkových intenzit v předpovědi.

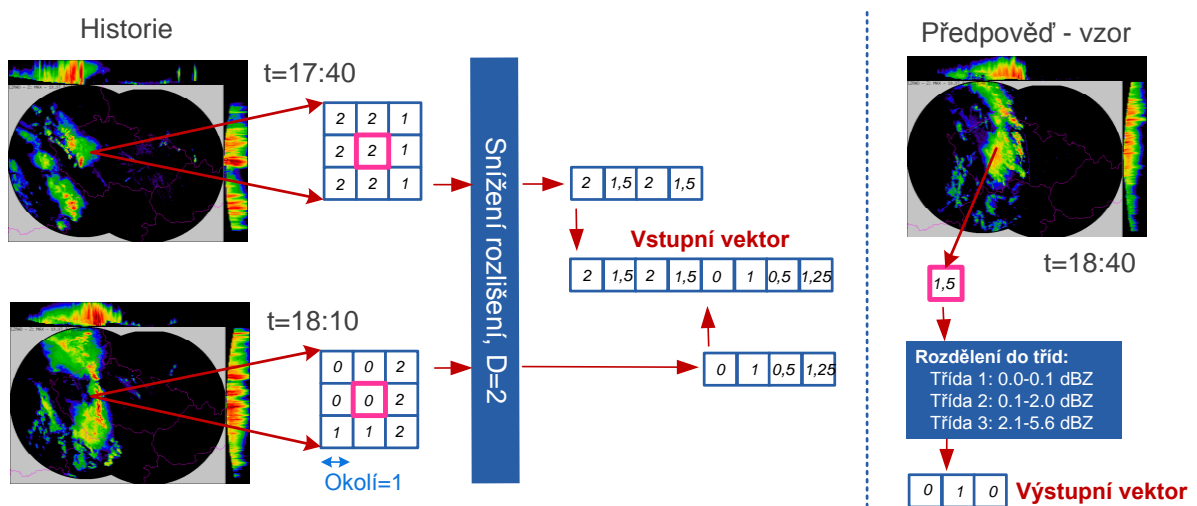
$$Out = future * nclasses \quad (3.4)$$

Počet neuronů ve vrstvě skryté závisí na počtu použitých trénovacích vektorů. Resp. počet trénovacích vektorů je závislý na celkové konfiguraci neuronové sítě. Ve většině případů bylo využito poučky, která říká, že počet neuronů ve skryté vrstvě by měl být větší a nebo alespoň odpovídat počtu neuronů ve vrstvě vstupní. Množství potřebných trénovacích vektorů bylo zase zvoleno podle jiné rady, která říká, že počet trénovacích vektorů by měl odpovídat deseti až dvaceti-násobku počtu parametrů sítě, což je celkový počet vah spojující neurony v síti. Později bylo vyzorováno, že trénovacích vektorů je vhodné použít spíše 20krát než 10krát více, než je počet parametrů sítě.

Při tvorbě trénovacích i testovacích vektorů je postupováno následovně. Nejprve je podle zadané délky historie, doby předpovědi a odstupů mezi jednotlivými snímky vytvořen seznam radarových snímků tvořících kontext pro předpověď. Část je tvořena *historií*, druhou část seznamu tvoří snímky reprezentující *předpověď*. V každém takovém snímku je vybrána jedna stejná buňka, pro kterou jsou vektory vytvořeny. Vstupní trénovací i testovací vektor je sestaven z hodnot ze zvoleného *okolí* buňky, které jsou následně sloučeny dle zadaného parametru D a kvůli sjednocení řádu hodnot vstupů a výstupů vyděleny konstantou 10. Takto zredukované vektory z jednotlivých snímků v *historii* jsou spojeny za sebou. Výstupní vektor je tvořen z jednotlivých zvolených buněk reprezentující předpověď. Úroveň srážkové odrazivosti je porovnána s rozdělením do tříd, z čehož je zjištěna třída intenzit srážek v buňce. Vektor pro jednu buňku potom obsahuje hodnoty zastupující každou třídu intenzit, přičemž zjištěná třída je zastoupena nejvyšší hodnotou. V praxi je pro neaktivní

třídě použita hodnota 0, pro třídu aktivní hodnota 1, přitom bylo již zmíněno, že taková aktivní třída může být pro buňku v jednom snímku jen jedna. Dílčí vektory všech snímků tvořících předpověď jsou spojeny za sebou, což tvoří výstupní trénovací nebo testovací vektor. Celý proces ilustruje Obrázek 3.3, rozdělení odrazivosti do tříd je pouze názorné a je odlišné od již zmíněného příkladu v předchozí kapitole.

Postup výběru jednotlivých buněk je založen na potřebném počtu trénovacích vektorů. Rozměrnější sítě vyžadují takových vektorů mnoho. Bylo nutné v jedné sadě snímků vybrat buněk hned několik a pro každou vytvořit vstupní a výstupní vektory. Rozestupy mezi jednotlivými buňkami byly zvoleny podle potřebného počtu vektorů. Zpočátku byla tendence, aby buňky byly mezi sebou vzdáleny alespoň o velikost *okolí*. Celkový počet vektorů byl nedostačující a tak rozestupy mezi jednotlivými buňkami byly zmenšeny. Problém nastal při snižování rozlišení dat. Zvolené buňky by mezi sebou měli mít odstup alespoň odpovídající velikosti faktoru D, při vzdálenosti D a menší by se sobě vektory čím dál více podobaly a jednotlivým vstupním neuronům by byly předkládány stejné hodnoty z různých vektorů, což by teoreticky nemělo vést ke správnému natrénování. Experimentálně bylo dokázáno, že při vhodně zvoleném faktoru učení α vykazovala neuronová síť, naučená pomocí takto podobných vektorů, při trénování a testování lepší výsledky.



Obrázek 3.3 - Tvorba vstupních a výstupních vektorů

4 Testování

Testování je důležitou částí práce, díky níž lze porovnávat různě natrénované sítě nebo přístupy různých řešení, případně říci, do jaké míry jsou neuronové sítě schopny předpovídat déšť. S testováním přímo souvisí i metody ohodnocení výsledků. Takovými metodami se budou zabývat následující části kapitoly.

Pro úplnost je vhodné zmínit, že testování neuronové sítě je v některých částech podobné jako její trénování. Na vstup jsou přivedeny vstupní vektory, které mají stejnou podobu jako vektory trénovací. I jejich vytváření je stejné. Rozdíl je jen v tom, že pro tvorbu takových vektorů je použita jiná množina radarových snímků, aby byla dokázána obecná funkčnost, nikoliv jen funkčnost pro omezenou sadu, v tomto případě trénovací. Po zpracování vstupního vektoru natrénovanou neuronovou sítí vznikne vektor výstupní, který je následně porovnán s již předem známou předpovědí. Tato předem známá předpověď je převedena do tvaru výstupního vektoru způsobem uvedenými v předchozí kapitole. Nutno dodat, že způsobů porovnání je mnoho a je obtížné zvolit takový, který by podával nejobektivnější zprávu.

Často lze míru úspěšnosti vyjádřit numericky, ovšem právě při řešení předpovědi deště taková úspěšnost může být občas matoucí. Bylo by zapotřebí využití složitějších analytických a statistických metod, aby byla s úspěchem vyjádřena míra podobnosti mezi výstupem neuronové sítě a reálnou předpovědí. Statistický výsledek experimentu, vypočítaný některou z jednodušších metod zmíněných dále, vyjádřený čísly může například udávat úspěšnost stejnou jako u jiného experimentu, přitom pohled na oba výsledky zobrazené graficky ve formě obrázku může zanechat zcela jiný dojem. Proto se také osvědčilo k numerickým výsledkům navíc vytvořit předpověď v grafické podobě. Zhodnocení výsledku pouze lidským pohledem bylo potom zcela jistě subjektivní, ale v mnoha případech pomohlo pochopit nedostatky řešení a přispělo tak k dalšímu vývoji.

4.1 Celková úspěšnost

Asi nejjednodušší a nejintuitivnější použitou metodou bylo zhodnocení předpovědi z pohledu úspěšnosti předpovědi v jednotlivých buňkách binárně, tedy rozdělení na dvě možnosti, zda intenzita deště v předpověděné buňce souhlasí s intenzitou požadovanou. Míra úspěšnosti bylo potom vyjádřena jako průměr úspěšnosti nad všemi buňkami v testovací sadě. Takové zhodnocení bylo časově i výpočetně nenáročné a poskytovalo nejzákladnější pohled na úspěšnost. Problém nastal, když bylo potřeba zjistit jak dobře je předpovězen pouze déšť. Bylo řečeno, že v radarových snímcích obsahuje většina buněk nulové srážky, v některých snímcích se nevyskytoval déšť téměř vůbec. Jelikož předpověď nulových srážek v místě, kde se v daném okolí nevyskytovaly srážky ani v historii předpovědi, dokáže správně vytvořit i hůře natrénovaná síť, úspěch v takových buňkách byl skoro stoprocentní. Z důvodu nepoměru takových buněk k buňkám obsahujícím srážky, byl výsledek značně zkreslen. Přitom v testovací sadě nemohly být takové buňky vynechány, protože takové buňky přece tvoří také předpověď. Navíc v případě, že by byly hodnoceny pouze buňky obsahující srážky, mohl by být v některých případech úspěch stoprocentní i v případě špatně natrénovaná sítě, například když by taková síť předpověděla ve všech buňkách testovací sady déšť, nebyly by testovány buňky ve kterých pršet nemělo. Přes často zkreslený výsledek, podávala tato metoda alespoň základní informace o natrénování sítě z pohledu možného porovnání s jinými sítěmi. Úspěšnost v jednotlivých

buňkách je definována výrazem 4.1, $predicted_class_{x,y,t}$ je předpověděná třída bodu určeného souřadnicemi x, y v čase t a $target_class_{x,y,t}$ je vzorová třída bodu x, y v čase t .

$$match(x, y, t) = \begin{cases} 0, & \text{pokud } predicted_class_{x,y,t} \neq target_class_{x,y,t} \\ 1, & \text{pokud } predicted_class_{x,y,t} = target_class_{x,y,t} \end{cases} \quad (4.1)$$

Celková úspěšnost n testovacích vektorů je definována jako aritmetický průměr hodnocení $match(x, y, t)$ jednotlivých vektorů i reprezentujících body x_i, y_i , kde v jednom výstupním vektoru se může nacházet předpověď na více okamžiků T .

$$Celkova\ uspesnost = \frac{100}{n \cdot T} \cdot \sum_{i=1}^n \sum_{j=1}^T match(x_i, y_i, t_{i,j}) \quad (4.2)$$

4.2 Úspěšnost třídy

Výše uvedená metoda byla pozměněna tak, aby podávala informace o úspěšnosti každé dílčí třídy zvlášť. Buňka, která měla být identifikována jako jistá třída, byla porovnána s výstupem neuronové sítě, přičemž každý takový úspěch či neúspěch nebyl zaznamenán do celkové statistiky, nýbrž pro každou třídu zvlášť. Z takového výsledku bylo jasnější, jaké úspěšnosti dosahovala natrénovaná síť pro předpověď konkrétních srážek a míst bez deště. Úspěšnost je definována podobně jako u předchozí metody (rov. 4.1, rov. 4.2), avšak pro každou třídu c zvlášť.

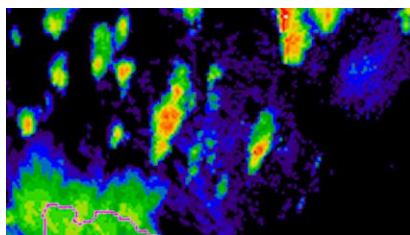
$$hit(x, y, t, c) = \begin{cases} 0, & \text{pokud } predicted_class_{x,y,t} \neq target_class_{x,y,t} \\ 1, & \text{pokud } target_class_{x,y,t} = c \wedge predicted_class_{x,y,t} = target_class_{x,y,t} \end{cases} \quad (4.3)$$

$$Uspesnost\ tridy(c) = \frac{100}{n_c \cdot T} \cdot \sum_{i=1}^n \sum_{j=1}^T hit(x_i, y_i, t_{i,j}, c) \quad (4.4)$$

kde n_c je celkový počet výskytů třídy c v testovací množině n vektorů.

4.3 Bodování a složitější návrhy hodnocení

Hlavním problémem v předchozích částech uvedených metod je jejich orientace jen na jednu buňku. Dešťové srážky jsou však ve všech radarových snímcích velmi členité, jde zpravidla o buňky naznačující okraje dešťových mraků, viz. Obrázek 4.1. Tato členitost se často mění a nelze ji téměř předvídat. Úspěšnost při hodnocení buněk předpovědi nacházejících se na okrajích srážkových struktur byla vždy velmi nízká. Takových buněk je však také mnoho, což přispělo k dalšímu zkreslení výsledků.



Obrázek 4.1 - Ukázka členitosti srážek v radarovém snímku

Přestože je předpověď tvořena právě pro jednu buňku, sousední buňky v předpovědi jednoho snímku by měli dohromady tvořit ucelenější útvary. Z hlediska předpovědi deště pomocí neuronových sítí se potom jeví mnohem důležitější schopnost rozpoznání srážkových struktur jako celku a určení správné tendence jejich vývoje, směru, rychlosti apod., než statistické vyhodnocení úspěšnosti pro každou buňku zvlášť. Takové vyhodnocení je ale nesmírně náročné.

Bylo by zapotřebí rozpoznávat jednotlivé srážkové struktury, jejich pohyb, tendenci ubývání nebo přibývání srážek, rychlost a další. Následné porovnání předpovědi s realitou by jistě podávalo ucelenější zprávu než dříve zmíněné metody. Z důvodu velké náročnosti realizace bohužel zůstalo takové řešení jen v části návrhu.

Jako kompromis byla navržena metoda využívajícího bodování. V každé testované buňce byla vypočítána absolutní hodnota vzdálenosti mezi třídami výstupu sítě a reality, kde třídy byly odstupňovány podle intenzity deště, a kde dvě sousední buňky měli mezi sebou vzdálenost 1. Pro lepší pochopení je uveden názorný příklad – vzdálenost mezi třídou 1 a třídou 3 je 2. Každá vzdálenost je potom ohodnocena jinou bodovou hodnotou. Vzdálenost 0, tedy situace kdy předpověď odpovídá realitě, byla ohodnocena nejvyšší hodnotou. S větší vzdáleností potom bodová hodnota klesala. Bodové ohodnocení všech buněk testovací sady bylo nakonec sečteno a byl z něj vypočítán aritmetický průměr. Při vhodně zvoleném rozložení bodů takové hodnocení potom alespoň trochu zvýhodňovalo předpověď, ve které byly rozpoznány celé srážkové útvary, jejichž okraje se dají obtížně předpovědět, před předpovědí, ve které byly struktury rozpoznány chybně. Metoda je úspěšně využitelná při větším počtu tříd ale je spíše podobná metodám numerickým než výše zmíněnému návrhu. Matematické vyjádření bodovaného ohodnocení je následující.

$$distance(x, y, t) = |predicted_class_{x,y,t} - target_class_{x,y,t}| \quad (4.5)$$

$$Score = \frac{100}{n \cdot T} \cdot \sum_{i=1}^n \sum_{j=1}^T score_table[distance(x_i, y_i, t_{i,j})] \quad (4.6)$$

kde n je celkový počet vektorů testovací množiny, $|predicted_class_{x,y,t} - target_class_{x,y,t}|$ je vzdálenost předpovídané a vzorové třídy bodu x, y v čase t a $score_table[]$ množina bodových ohodnocení podle vzdáleností tříd. Tabulka 4.1 obsahuje příklad bodového hodnocení předpovědi rozdělené do pěti tříd.

Vzdálenost tříd	0	1	2	3	4
Body	1	0,5	0,1	-0,1	-0,5

Tabulka 4.1 - Ukázka bodového hodnocení

4.4 Metody výpočtu chyby

Metodami doposud uvedenými byla vždy zjišťována úspěšnost předpovědi. Pro účely Crossvalidace bylo však vhodné vypočítat spíše chybu. Ve své podstatě by nezáleželo na tom, zda je použita úspěšnost, které by se musela zvyšovat, nebo chyba, která se snižovala. Nicméně chyba byla počítána trochu jiným způsobem vhodným pro Crossvalidaci. Tento způsob výpočtu má název *Průměrná kvadratická chyba* (angl. *Mean Squared Error*, zkratka *MSE*) a je způsobem běžně používaným ve statistice.

Spočívá v počítání aritmetického průměru kvadratických hodnot chyb, které jsou určeny rozdíly mezi odhadovanými hodnotami od hodnotami požadovanými[11]. V případě rozdělení výstupních hodnot do tříd, je chyba vypočítána jako druhá mocnina rozdílu hodnoty každého výstupu neuronové sítě a hodnoty v požadovaném vzoru. Účelem snižování chyby při Crossvalidaci je minimalizace rozdílů mezi třídami ve výstupech neuronové sítě a třídami vzorových situací. Při výpočtu chyby při Crossvalidaci je použit následující vzorec.

$$MSE = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^m (target_{i,j} - predicted_{i,j})^2, \quad (4.7)$$

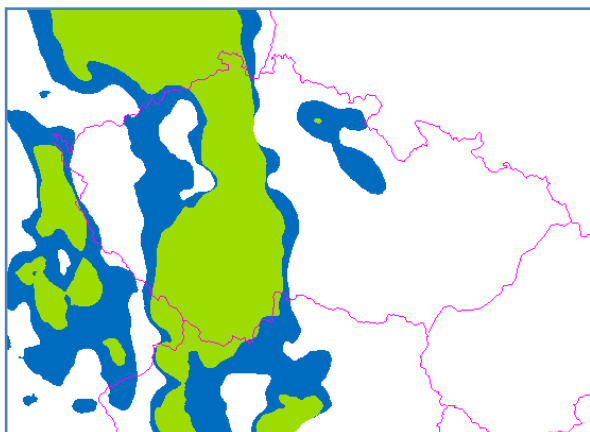
kde n je celkový počet testovacích vektorů, m je délka výstupního vektoru, $target_{i,j}$ odpovídá j -té hodnotě i -tého vektoru ze vzorových vektorů a $predicted_{i,j}$ odpovídá j -té hodnotě i -tého vektoru z vektorů předpověděných. Hodnoty v jednotlivých vektorech tvoří základ pro určení třídy. Bylo již zmíněno, jakým způsobem toto určení probíhá (viz kap. 3.3).

Pro účely Crossvalidace byla občas využívána ještě jedna metoda, založená na dříve zmíněných metodách. Byl spočítán celkový počet případů *succeeded*, kdy předpověď v jedné buňce přesně odpovídala reálné situaci v té samé buňce, a celkový počet případů *failed*, kdy byla situace opačná. Chyba je vypočítána následujícím vzorcem.

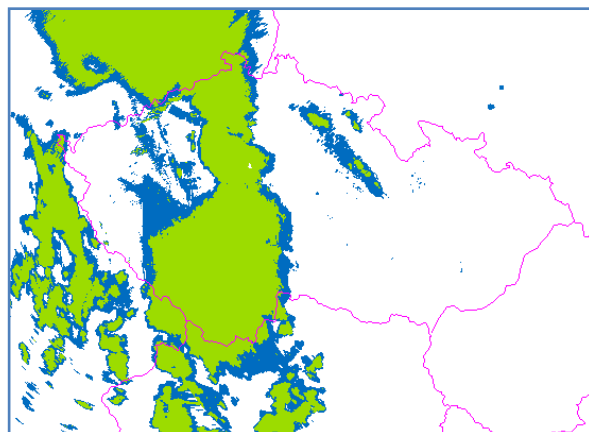
$$E = 100 \cdot \frac{failed}{failed + succeeded} \quad (4.8)$$

4.5 Grafický výstup předpovědi

Z hlediska zjištění a pochopení nedostatků řešení se osvědčilo hodnocení pohledem na mnoho buněk sestavených zpět jako obrázek. Předpověď byla vypočítána neuronovou sítí pro každou buňku v radarovém snímku a následně byla každá buňka zobrazena v obrázku jako jeden pixel, barevně se odlišující od okolí předpověděnou intenzitou srážek. Obrázek 4.2 zobrazuje právě jednu takovou předpověď a pro porovnání Obrázek 4.3 zobrazuje vzor předpovědi.



Obrázek 4.2 - Grafický výstup předpovědi



Obrázek 4.3 - Grafické znázornění vzoru předpovědi

5 Výsledky experimentů

V této kapitole budou zveřejněny některé důležité poznatky, které jsem získal při provádění mnoha experimentů. Většina bude popsána slovně, u některých budou poznatky podpořeny i uvedením výsledků konkrétních experimentů. Metody vyhodnocení experimentů jsou používány tak, jak byly definovány v předchozí kapitole.

Nutno dodat, že skripty pro tvorbu dat byly převážně vytvořeny v jazyce *Perl* a některé další programy v jazyce *C++*. Nástroj s názvem *SNet* pro trénování neuronové sítě mně byl pro účely praktického ověření poskytnut vedoucím této práce a byl již dříve vytvořen na Fakultě informačních technologií Vysokého učení technického v Brně. Příprava dat, trénování a testování neuronové sítě probíhalo paralelně na více počítačích z důvodu velké časové náročnosti těchto úkonů. Parametry prvních pokusů, jako je např. počáteční rychlost učení neuronové sítě apod., byly nastaveny na hodnoty, které se běžně používají na FIT VUT v Brně. S přibývajícím počtem provedených experimentů rostla i moje zkušenost při řešení úloh tak, že jsem později parametry volil co nejvhodněji sám.

V závěru této kapitoly potom budou uvedeny výsledky řešení úlohy této práce mým kolegou, studentem Petrem Gerychem, který zvolil zcela odlišný postup.

5.1 Velikost neuronové sítě, faktor rychlosti učení

Na velikosti neuronové sítě závisí úspěchy při trénování. Příliš velké sítě se trénují velmi dlouho a obtížně, naopak malá síť nemusí být schopná vstřebat všechny potřebné vlastnosti. Bohužel se ale její velikost nedá určit přímo, protože závisí na spoustě dalších faktorů, jako je např. délka předpovědi a na ní závislá velikost okolí, protože velikost sítě je ovlivněna velikostí vstupních a výstupních vektorů.

Pokud by velikost neuronové sítě určovala ostatní faktory, mohlo by se stát, že se tyto faktory nabudou takového rozměru, že vstupní vektor bude např. zkreslený a zkrátka nepoužitelný. V případě, že by velikost sítě byla přímo dána dle zvolených faktorů při tvorbě dat, mohla by tato velikost dojít i do extrémních rozměrů. Je nutné zvolit vhodný kompromis, který zanechává data v použitelné podobě s tím, že vstupní vektor bude mít rozumnou velikost. Rozumná velikost neuronové sítě je relativní pojem a bohužel neexistuje žádné konkrétní omezení z hlediska funkce, jen z hlediska výpočetní náročnosti. Tu je v dnešní době možných paralelních výpočtů také těžké určit.

Vstupní vrstva byla určena dle velikosti vstupního vektoru. Ten byl s ohledem na zachování informace zmenšen již dříve popsaným postupem, protože pro jistou dobu předpovědi bylo nutné využít i nutné okolí bodu, tedy počet různých hodnot. Také počet snímků v historii předpovědi ovlivňoval počet vstupních vektorů. Bylo nutné zvolit co nejmenší počet takový snímků, aby přitom byla zachována souvislost. Velikost výstupní vrstvy byla přímo dána počtem tříd klasifikátoru. Velikost skryté vrstvy byla závislá na počtu trénovacích vektorů, resp. podle velikosti sítě byl vytvořen potřebný počet vektorů.

Z důvodu, že jsem nenašel žádné práce, zabývající se úlohou mnou zvoleným přístupem, byly všechny faktory zjištěny experimentálně. Nejlepších výsledků bylo dosaženo po dlouhém sladění všech možných nastavení. Velikost vstupní vrstvy při různých pokusech se pohybovala v řádech desítek až stovek neuronů. Skrytá vrstva obsahovala přibližně stejný počet neuronů jako vrstva

vstupní. Ve výstupní vrstvě se nacházelo nejčastěji okolo deseti neuronů, samozřejmě v závislosti na počtu požadovaných tříd.

Počáteční velikost faktoru rychlosti učení, tzv. *Learning rate*(angl.), jsem nejčastěji volil dle rad kolegů, kteří neuronové sítě využívají v jiných projektech. Jelikož se tyto hodnoty osvědčily, a protože při zvolení řádově jiné hodnoty byly výsledky horší, zvolil jsem je za hodnoty vhodné a využíval je nadále. Konkrétně se jednalo o hodnotu kolem 0,001.

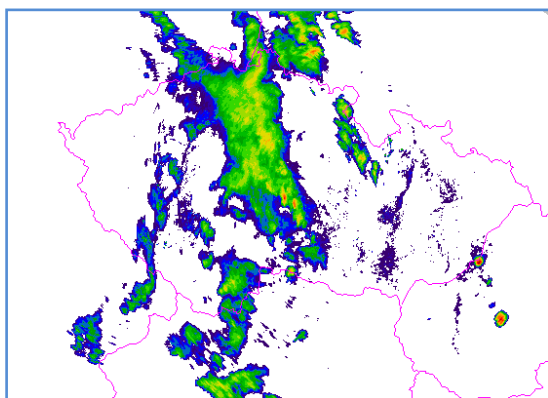
5.2 Závislost na počtu trénovacích vektorů

Úspěšnost predikce silně závisí na počtu vektorů použitých pro trénování. Opět jsem zde pro začátek vyžil rady, že počet trénovacích vektorů by měl být 10x až 20x větší, než počet všech vah neuronové sítě spojující jednotlivé neurony. Experimentálně jsem později došel k závěru, že pro úspěšné natrénování je spíše lepší počet něco málo převyšující 20ti násobek počtu vah, spíše než 10ti násobek.

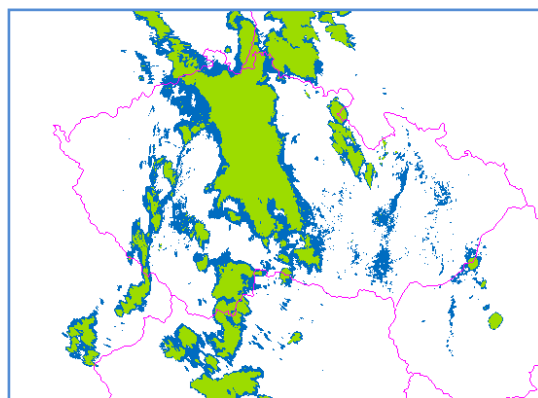
Důležitým faktem se také ukázala nutnost vyváženosti trénovacích vektorů, resp. počtu zastoupení jednotlivých tříd ve výstupních trénovacích vektorech. Bylo vidět, že neuronová síť se nejlépe naučí klasifikovat do tříd, kterých je v množině trénovacích vektorů nejvíce. Při tvorbě takových vektorů potom bylo nutné některé dále nepoužívat pro trénování, pokud výstup takového vektoru měl být klasifikován do tříd, které již byly v trénovací množině zastoupeny v počtu řádově přesahujícím počty zastoupení ostatních tříd. Tento fakt se stal omezujícím, zejména s ohledem na počet radarových snímků, z nichž byly vektory vytvářeny. Opakuji, že vyváženost měla na úspěšnost predikce téměř limitující vliv. Bylo upuštěno od klasifikace předpovědi do mnoha tříd a ve většině případů byly použity třídy dvě, tři či čtyři. Díky vhodnému rozdělení srážkových odrazivostí z radarových snímků potom mohlo být dosaženo optimálního počtu zastoupení jednotlivých tříd. Následuje ukázka jednoho z často používaných rozdělení do tří tříd.

$$třída = \begin{cases} 1, \text{ pro } Z < 4\text{dBZ} \\ 2, \text{ pro } Z \geq 4\text{dBZ a } Z < 20\text{dBZ} \\ 3, \text{ pro } Z \geq 20\text{dBZ} \end{cases} \quad (5.1)$$

Grafická ukázka rozložení do takových tříd je zobrazena na dvou následujících obrázcích, kde první Obrázek 5.1 barevně odlišuje radarem naměřené srážkové odrazivosti a druhý Obrázek 5.2 zobrazuje použitý převod na intenzitu srážek, kde bílou barvou jsou vybarveny buňky *třídy 1*, modře buňky *třídy 2* a zeleně buňky *třídy 3*.



Obrázek 5.1 - Radarový snímek s barevným rozlišením odrazivosti



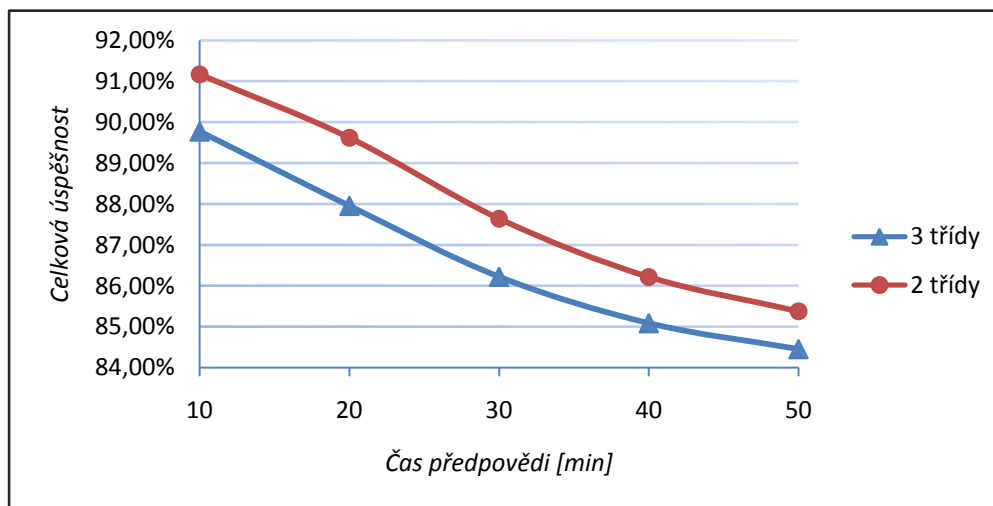
Obrázek 5.2 - Radarový snímek s barevným rozlišením intenzit deště

5.3 Predikce v časech

Předpovědět deště lze provádět pro různý čas dopředu. Z experimentů nepřímo vyplývá, že tento čas není z hlediska řešení pomocí neuronových sítí nijak omezen. Při delších časech předpovědi však úspěšnost rychlým tempem klesá. Je však individuální stanovit nějakou konkrétní hranici úspěšnosti, pod kterou by se dala předpověď považovat za neúspěšnou, záleží totiž na jejím konkrétním využití. Limitujícím faktorem z pohledu řešení je potom jen dosah měření srážkových odrazivostí radarem, protože je nutné pro delší čas předpovědi srážek v jednom místě použít větší počet okolních buněk. Větší okolí si žádá větší snížení rozlišení naměřených dat, což má za následek větší nepřesnosti v předpovědi.

Obecným limitujícím faktorem je hlavně to, že předpověď se provádí jen na základě radarových snímků, tedy informací, v jakém stavu a kde se dešťové srážky před vydáním předpovědi vyskytují. Srážky se však vyvíjí na základě mnoha faktorů, které pro předpověď nejsou k dispozici. Jde například o satelitní snímky oblačnosti, teplotu vzduchu nebo rosný bod naměřený v pozemních meteorologických stanicích. Bez znalosti hodnot těchto faktorů v místech předpovědi a jejich okolí nelze očekávat, že předpověď na delší dobu bude mít velkou úspěšnost. Celkovou tendenci vývoje srážek totiž nelze jen z radarových snímků vypožorovat a nelze ani předpovědět dlouhodobější vývoj.

V jednotlivých experimentech byla doba předpovědi stanovena v rozmezí od 10ti minut do 70ti minut. Bylo již dříve zmíněno, že neuronovou síť lze natrénovat jak pro předpověď v jednom okamžiku, tak pro předpověď ve více časových intervalech. Lepší výsledky přitom vykazovaly neuronové sítě, které byly trénovány jen pro jeden čas, protože se na tento čas mohli více zaměřit. Byly provedeny i pokusy s předpovědi více časů zároveň, především pro účely porovnání. Graf 5.1 shrnuje předpověď vygenerovanou dvěma různými neuronovými sítěmi pro časy 10, 20, 30, 40 a 50 minut dopředu při rozložení intenzit deště do tří resp. dvou tříd. Tabulka 5.1 potom informuje o konkrétním nastavení parametrů experimentu. Následující části kapitoly se budou zabývat předpovědi zaměřenou vždy na jeden jediný čas.



Graf 5.1 - Graf s výsledky předpovědi pro 10, 20, 30, 40 a 50 minut

Velikost neuronové sítě - počet neuronů					
Vstupní v.	363	Skrytá v.	400	Výstupní v.	10 (resp.15)
Historie					
Počet snímků	3	Rozestup	10minut		
Data					
Počet okolních buněk	55	Faktor zmenšení rozlišení			11
Rozdělení intenzit(prahy mezi třídami)			0; 0.01; 60 (resp.0; 0.01; 15; 60)		

Tabulka 5.1- Nastavení parametrů pro předpověď z grafu 5.1

5.3.1 Předpověď na dobu 10 až 50 minut

Grafická podoba předpovědi se podobá svým reálným vzorům. Dokonce jsou správně předpovězeny některé drobné nesouvislosti ve výskytu srážek. Úspěšnosti vyjádřené numericky zhruba odpovídají grafu 5.1 z předchozí části s rozdílem v mírném zlepšení.

V závislosti na konkrétním čase bylo použito od 10 do 45 buněk v okolí sledovaného bodu. Experimentálně bylo dále vyzkoušeno, že počet snímků použitých v historii by měl být minimálně 3, při použití více jak tří takových snímků potom úspěšnost při testech spíše klesá. Je to nejspíše zapříčiněno tím, že snímky v krátkém rozestupu se sobě podobají. Časový rozestup by měl být zvolen takový, aby rozdíl mezi výchozím časem a časem předpovědi zhruba odpovídal rozdílu mezi prvním a posledním snímkem v historii. Nebyla vyzkoušena žádná další závislost mezi snímky v historii a úspěšností.

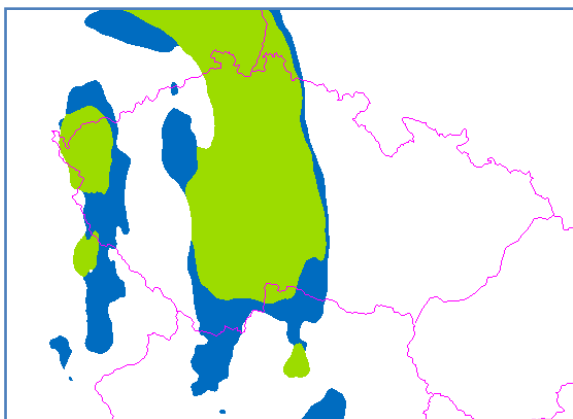
Čas předpovědi od 10ti do 50ti minut by byl vhodný pro detekci konvektivních bouří zmíněných v úvodu práce. Tato možnost nemohla být ověřena, protože v radarových snímcích, které byly k dispozici, se nenacházel dostatečný počet reprezentativních situací, které by umožnili natrénování sítě pro takovou předpověď.

5.3.2 Předpověď na 60 minut

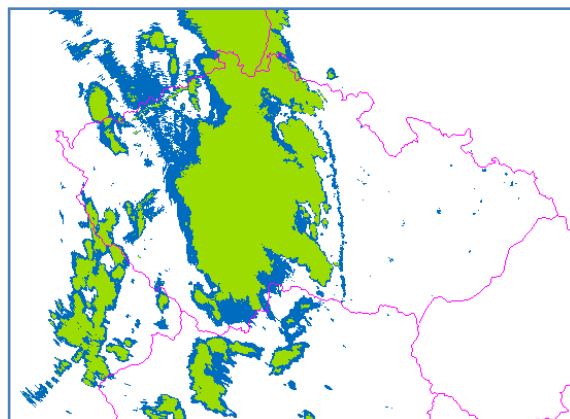
Srážkové struktury v grafické podobě předpovědi byly stále nápadně podobné reálným situacím, avšak byly již více vyhlazené tak, že na rozdíl od reálných snímků nebyly přechody mezi deštěm a místy beze srážek(nebo i mezi jednotlivými třídami srážkových intenzit) tolik rozmanité. Obrázek 5.3 a Obrázek 5.4 zobrazují předpověď na 1 hodinu a reálný radarový snímek ve stejném čase jako předpověď. Srážkové intenzity jsou rozděleny do 3 tříd. Následující tabulka obsahuje numerické zhodnocení úspěšnosti předpovědi z obr. 5.3.

Celková úspěšnost	81,63%
Úspěšnost třídy 1(bílá barva)	91,16%
Úspěšnost třídy 2(modrá barva)	28,14%
Úspěšnost třídy 3(zelená barva)	71,38%

Tabulka 5.2 - Úspěšnost předpovědi



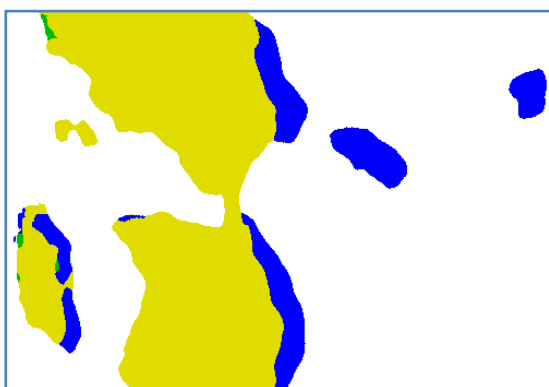
Obrázek 5.3 - Předpověď na 60 minut



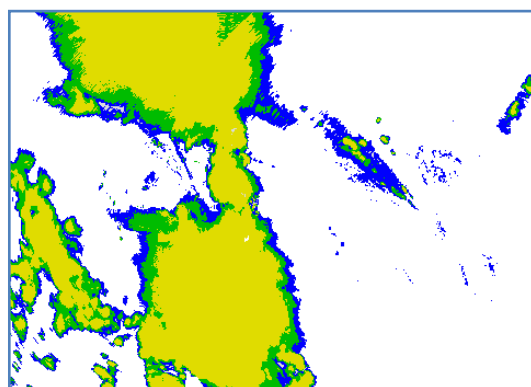
Obrázek 5.4 - Vzor předpovědi pro 60 minut

Bylo nutné použít okolí sledovaných buněk od 50 do 60 buněk. Větší či menší počet měl negativní vliv na úspěchy. V historii bylo použito 3 radarových snímků s rozestupem 30 minut. Kdy čas mezi prvním a posledním snímkem opět odpovídal době předpovědi. S žádným jiným nastavením nebyly sledovány větší rozdíly ve výsledcích a pokud byly, tak byly výsledky horší.

V případě, že v předpovědi byly srážky kvantovány do více tříd, z již zmíněného nepoměru zastoupení tříd při trénování, vykazovaly úspěšnosti některých tříd velmi špatné výsledky, proto bylo od takového kvantování upuštěno. Na obrázcích 5.5 a 5.6 je vidět předpověď na 60 minut a zároveň radarový snímek v době předpovědi. Intenzita vyznačená zelenou barvou nebyla v předpovědi zastoupena skoro vůbec, intenzita vyznačená barvou modrou zase jen ve směru pohybu srážek na čelní straně.

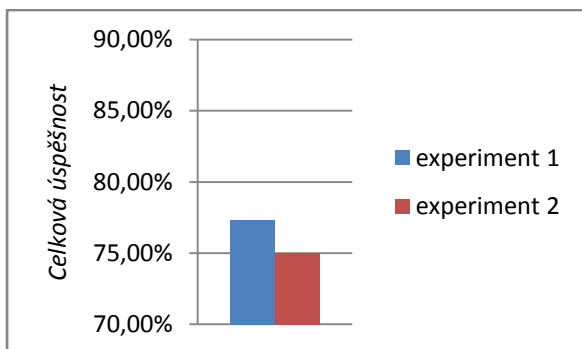


Obrázek 5.5 - 60ti minutová předpověď, 4 třídy intenzit

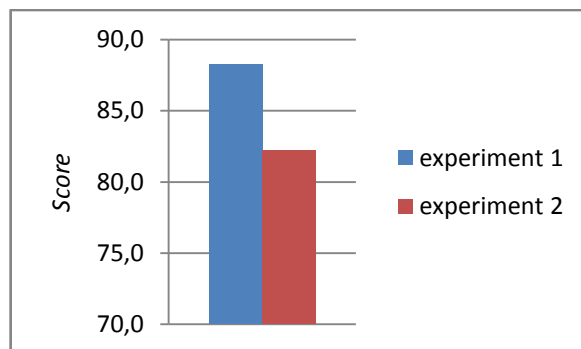


Obrázek 5.6 - Vzor 60ti minutové předpovědi, 4 třídy intenzit

Při výběru lépe natrénované sítě přispělo využití bodovaného hodnocení. Byly provedeny dva experimenty 60ti minutové předpovědi, první byla vypočítána neuronovou sítí, která byla natrénována pouze pro tuto jedinou předpověď. Naopak druhý výsledek byl vypočítán sítí, která byla natrénována pro předpověď několika okamžiků zároveň. Grafy 5.2 a 5.3 porovnávají tyto dvě sítě s využitím jiné metody hodnocení. Je vidět že metoda bodového hodnocení podává ucelenější obraz o kvalitě natrénování neuronové sítě.



Graf 5.2 - Celková úspěšnost



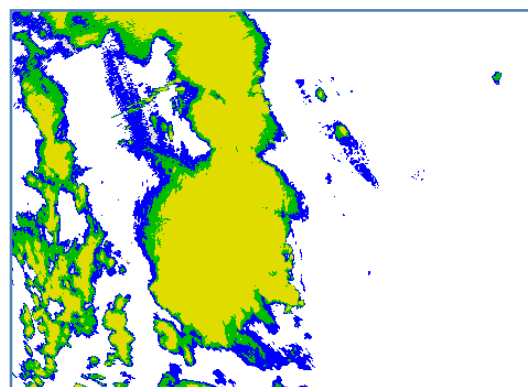
Graf 5.3 - Bodované hodnocení

5.3.3 Předpověď více jak na 60 minut

Při zvyšujícím se čase předpovědi nebylo zatím dosaženo očekávaných výsledků. V předpovědi jsou stále rozeznány srážkové struktury, ty se ale svému vzoru čím dál méně podobají. Otázkou je, kam až lze s předpovědi jen za pomoci radarových snímků dojít. Lepší předpovědi by pravděpodobně propracovanější metodou přípravy dat nebo lepším výběrem vhodných vektorů pro trénování. Obrázek 5.7 představuje předpověď na 90 minut s rozdělením srážek do čtyř tříd. Obrázek 5.8 potom představuje vzor předpovědi.



Obrázek 5.7 - 90ti minutová předpověď



Obrázek 5.8 - Vzor 90ti minutové předpovědi

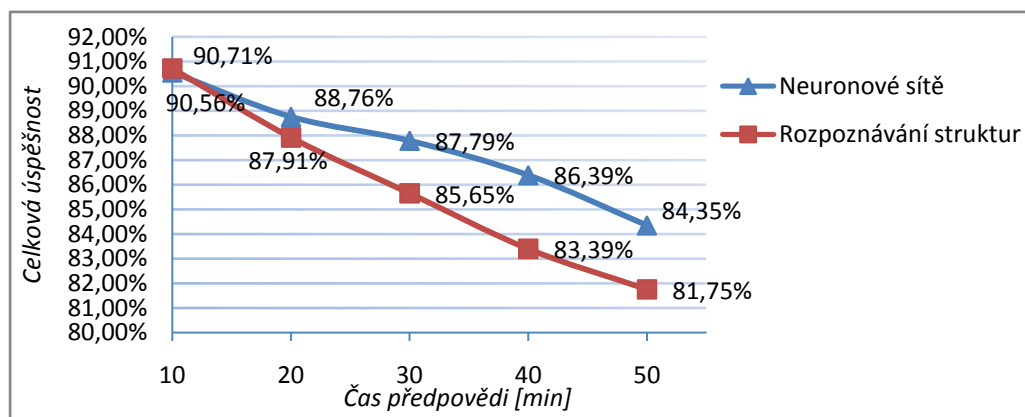
Z obrázků je vidět, že neuronová síť sice rozezná alespoň přibližně místa, kde by se srážky měli vyskytovat, není ale již schopna rozeznat jednotlivé třídy intenzit deště. Tento neúspěch je pravděpodobně způsoben nevhodně natrénovanou sítí pomocí vektorů, ve kterých je špatně rozdělen poměr mezi jednotlivými třídami. Navzdory dříve zde uvedeným úvahám, neuronová síť při této konkrétní předpovědi rozeznala na všech místech s deštěm třídu, která je nejen v konkrétním vzoru, ale i ve všech trénovacích vektorech zastoupena nejméně. Vysvětlením potom může být například to, že zelenou barvou je vyznačena třída, která se svými prahovým hodnotami nachází mezi dvěma zbylými třídami značící dešť.

5.4 Porovnání řešení s jinou metodou

Úloha řešená v této práci byla současně řešena mým kolegou Petrem Gerychem, avšak pomocí zcela jiného přístupu. Metoda je založená na rozpoznání srážkových struktur a výpočtu směru jejich

pohybu v sekvenci radarových snímků. Mraky se považují za objekty a k jejich detekci a extrakci z radarových snímků se využívá algoritmus pro semínkové vyplňování. K registraci objektů mezi statickými snímky se používají těžiště a objemy mraků. Predikce deště se provádí extrapolací kinematiky pohybu pomocí *Lagrangeova interpolačního polynomu*.

Pro účely porovnání byly dohodnuty stejné metody ohodnocení výsledků. Porovnání je na základě *Celkové úspěšnosti* (kap. 4.1) a *Úspěšnosti jednotlivých tříd* (kap. 4.2), s tím, že v obou případech byly srážkové intenzity kvantovány do *tří* tříd podle stejných prahových hodnot. Tabulka 5.3 porovnává metodu využívající neuronových sítí a metodu rozpoznávání struktur v časech 10, 20, 30, 40 a 50 minut z hlediska *celkové úspěšnosti* i *úspěšnosti jednotlivých tříd*. Graf 5.4 potom graficky shrnuje *celkové úspěšnosti* obou metod při tomto testu. Jako množina testovacích bodů byla zvolena množina všech bodů vyskytujících se na jednom konkrétním radarovém snímku v jednom čase³.



Graf 5.4 - Graf výsledků předpovědi, porovnání dvou přístupů

Z výsledků vyplývá, že řešení pomocí neuronových sítí poskytuje něco lepší výsledky, které jsou hlavně znatelné na výsledcích *úspěšnosti tříd*. *Celkové úspěšnosti* se liší o relativně malou hodnotu, která je však dána danou metodou a její náchylností ke zkreslení. Dále je z grafu vidět, že obě metody mají velice podobné výsledky při předpovědi na 10 a 20 minut, poté se rozdíly zvětší a zůstávají víceméně konstantní. Daleko znatelnější rozdíly jsou vidět v úspěšnostech jednotlivých tříd, zvláště tříd 2 a 3, jelikož průměrná úspěšnost třídy 1 je zkreslena mnohem větším množstvím výskytů, které často není obtížné rozeznat, protože v širším okolí bodu se nevyskytují žádné srážky.

³ V závislosti na čase předpovědi, tzn. jeden snímek pro jeden čas předpovědi

předpověď	Měření úspěšnosti	Neuronové sítě	Rozpoznávání struktur
10minut	<i>Celková úspěšnost</i>	90,56%	90,71%
	<i>Úspěšnost třídy 1</i>	95,31%	96,41%
	<i>Úspěšnost třídy 2</i>	55,09%	61,49%
	<i>Úspěšnost třídy 3</i>	88,91%	73,15%
20 minut	<i>Celková úspěšnost</i>	88,76%	87,91%
	<i>Úspěšnost třídy 1</i>	95,14%	94,98%
	<i>Úspěšnost třídy 2</i>	40,68%	50,41%
	<i>Úspěšnost třídy 3</i>	86,28%	66,19%
30 minut	<i>Celková úspěšnost</i>	87,79%	85,65%
	<i>Úspěšnost třídy 1</i>	95,06%	93,68%
	<i>Úspěšnost třídy 2</i>	34,53%	42,27%
	<i>Úspěšnost třídy 3</i>	84,73%	60,18%
40 minut	<i>Celková úspěšnost</i>	86,39%	83,39%
	<i>Úspěšnost třídy 1</i>	94,31%	92,69%
	<i>Úspěšnost třídy 2</i>	26,61%	35,42%
	<i>Úspěšnost třídy 3</i>	83,55%	53,10%
50 minut	<i>Celková úspěšnost</i>	84,35%	81,75%
	<i>Úspěšnost třídy 1</i>	92,93%	91,99%
	<i>Úspěšnost třídy 2</i>	26,10%	26,22%
	<i>Úspěšnost třídy 3</i>	81,94%	49,12%

Tabulka 5.3 - Výsledky různých přístupů k předpovědi

6 Závěr

Cílem této práce bylo navrhnout a vytvořit systém pro predikci deště za pomoci meteorologického radaru. Zvolil jsem možnost využití neuronových sítí, která není v praxi pro takovou úlohu běžně používána. Bylo dokázáno, že neuronové sítě určitě mají schopnost naučit se a předpovídat vývoj dešťových srážek pouze z radarových snímků.

Byl navržen postup, jak připravit data pro trénování neuronové sítě, jak postupovat při samotném trénování a jak provádět hodnocení výsledků při testech. V mnoha experimentech byly tyto postupy realizovány a byly vytvořeny předpovědi pro čas od 10ti minut do 60minut s relativně dobrou úspěšností. K některým předpovědím byla vytvořena jejich grafická podoba, takže bylo možné přímo rozpoznat nedostatky řešení. Na konci práce se nacházelo porovnání dvou odlišných metod, ze kterého vyplynulo, že řešení pomocí neuronových sítí vykazují lepší výsledky, zvláště pak při delších časech předpovědi.

Rozdíly v úspěšnosti obou metod jsou malé v porovnání s výpočetní náročností. Velkou nevýhodou v této práci popisovaného řešení je právě čas potřebný k výpočtu. I v případě, že je již k dispozici dobře natrénovaná síť a není třeba ji dále trénovat, je čas výpočtu předpovědi mnohokrát delší než při jiném postupu. Příčina je hlavně ve složité tvorbě dat, která by při dalším vývoji potřebovala optimalizovat, jak z hlediska časového, tak z hlediska paměťové náročnosti.

Bohužel se nepodařilo natrénovat síť pro detekci konvektivních bouří, která by byla jistě žádaná. I toto je otázkou dalšího vývoje, pro který by bylo potřeba větší množství radarových snímků a také propracovanější způsob výběru vhodných vektorů pro trénování. Při predikci na čas delší než 60 minut se úspěšnost předpovědi rychle zhoršovala. Otázkou je, jaké možnosti nabízí radarové snímky a jakého úspěchu lze vůbec dosáhnout. Z hlediska pokračování v práci by bylo vhodné změnit způsob přípravy dat tak, aby při použití velkého okolí bodu nedocházelo k velké ztrátě důležitých informací při zmenšování rozlišení. Řešením tohoto problému se nabízí například zmenšení rozlišení dat v závislosti na vzdálenosti od bodu. Hodnoty z buněk ve vzdálenosti, o které lze předpokládat, že srážky urazí za dobu předpovědi, budou použity v co nejvyšším rozlišení, naopak z buněk, které již na předpověď takový vliv mít pravděpodobně nebudou, použít s rozlišením menším. Tím dojde i k redukci velikosti neuronové sítě.

Další možností vývoje řešení je způsob trénování sítí. Dosud bylo výstupních neuronů využito ve smyslu, kdy každý neuron odpovídal jedné třídě intenzit srážek. Těch by mohlo být také využito tak, aby hodnota jejich výstupu přímo odpovídala požadované srážkové intenzitě v jednom čase.

V neposlední řadě by k úspěšnosti předpovědi jistě významně přispělo využití dalších měření, nejen radarových. Zakomponování satelitních snímků do systému by jistě přispělo k možnosti předpovědi na mnohem delší čas dopředu. Takový systém nejen že by vyžadoval kvalitní návrh a delší čas vývoje, ale od zadání této práce se už poměrně odchyluje.

Použitá literatura

- [1] KRÁČMAR, Jan. Meteorologické radiolokátory [online]. Vydáno: 27.11.2002, [cit. 5.5.2009]. Dostupné z WWW: <http://www.chmi.cz/meteo/rad/rad_inf.html>.
- [2] DRÁBEK, Oldřich, SEIDL, Petr, TAUFER, Ivan. UMĚLÉ NEURONOVÉ SÍTĚ– ZÁKLADY TEORIE A APLIKACE (1). *CHEMagazín* [online]. Vydáno: 2005, roč.15, č. 4, [cit. 5.5.2009], s. 23-34. Dostupné z WWW: <http://www.chemagazin.cz/Texty/CHXV_4_cl6.pdf>. ISSN: 1210-7409.
- [3] DRÁBEK, Oldřich, SEIDL, Petr, TAUFER, Ivan. UMĚLÉ NEURONOVÉ SÍTĚ– ZÁKLADY TEORIE A APLIKACE (2). *CHEMagazín* [online]. Vydáno: 2005, roč.15, č. 6, [cit. 5.5.2009], s. 10-12. Dostupné z WWW: <http://www.chemagazin.cz/Texty/CHXV_6_cl2.pdf>. ISSN: 1210-7409.
- [4] DRÁBEK, Oldřich, SEIDL, Petr, TAUFER, Ivan. UMĚLÉ NEURONOVÉ SÍTĚ– ZÁKLADY TEORIE A APLIKACE (3). *CHEMagazín* [online]. Vydáno: 2006, roč.16, č. 1, [cit. 5.5.2009], s. 12-14. Dostupné z WWW: <http://www.chemagazin.cz/Texty/CHXVI_1_cl3.pdf>. ISSN: 1210-7409.
- [5] DRÁBEK, Oldřich, SEIDL, Petr, TAUFER, Ivan. UMĚLÉ NEURONOVÉ SÍTĚ– ZÁKLADY TEORIE A APLIKACE (4). *CHEMagazín* [online]. Vydáno: 2006, roč.16, č. 2, [cit. 6.5.2009], s. 33-36. Dostupné z WWW: <http://www.chemagazin.cz/Texty/CHXVI_2_cl9.pdf>. ISSN: 1210-7409.
- [6] DRÁBEK, Oldřich, SEIDL, Petr, TAUFER, Ivan. UMĚLÉ NEURONOVÉ SÍTĚ– ZÁKLADY TEORIE A APLIKACE (5). *CHEMagazín* [online]. Vydáno: 2006, roč.16, č. 5, [cit. 6.5.2009], s. 29-31. Dostupné z WWW: <http://www.chemagazin.cz/Texty/CHXVI_5_cl6.pdf>. ISSN: 1210-7409.
- [7] HNÁTEK, Martin. Jeden perceptron – klasifikace [online]. Vydáno: 4.6.1999, [cit. 10.5.2009]. Dostupné z WWW: <<http://neuron.felk.cvut.cz/courseware/data/chapter/36nan028/s04.html>>.
- [8] MÁČEK, M., MACKOVÁ, J., RADVANSKÝ, J. Syndrom přetrénování [online]. 2003, [cit. 6.5.2009]. Dostupné z WWW: <http://ktl.lf2.cuni.cz/med_sport/med_sport_2003_vol_12/1/overtrain.pdf>.
- [9] MENDELSON, Lou. Training Neural Networks [online]. 2003, [cit. 6.5.2009]. Dostupné z WWW: <<http://www.tradertech.com/training.asp>>.
- [10] MIKOLOV, Tomáš. Language modeling of Czech using neural networks [online]. 2007, [cit. 6.5.2009]. Dostupné z WWW: <http://www.fit.vutbr.cz/research/groups/speech/publi/2007/mikolov_eiect_2007.pdf>.
- [11] Wikipedia: Mean squared error - Wikipedia, The Free Encyclopedia [online]. 3.5.2008, [cit. 3.5.2009]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Mean_squared_error>

Seznam příloh

Příloha 1: DVD obsahující veškeré zdrojové texty, jednoduchý návod na použití zdrojových textů, ukázky předpovědí a textové soubory zaznamenávající průběhy některých experimentů, ukázkové video s predikcí.