



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická

Katedra kybernetiky

**Aplikace neuronových sítí pro predikci času dosažení
význačných bodů v trajektorii letadel**

**Application of neural networks for prediction of approach time to
significant points in aircraft trajectories**

Bakalářská práce

Studijní program: Kybernetika a robotika

Studijní obor: Robotika

Vedoucí práce: Mgr. Karel Hanton, CS Soft Praha

Garant za katedru: prof. Ing. Václav Hlaváč, CSc.

Peter Hroššo

Praha 2012

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Peter Hroššo

Studijní program: Kybernetika a robotika (bakalářský)

Obor: Robotika

Název tématu: Predikce budoucí polohy letadla z radarových dat

Pokyny pro vypracování:

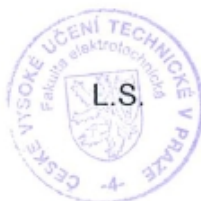
1. Seznamte se s problematikou řízení letového provozu a predikcí polohy letadel pro bezpečnost letového provozu.
2. Navažte na Vaši předchozí práci v předmětu A3B33IND-Projekt individuální, v níž jste shromáždil data mnoha přistávajících letadel.
3. Navrhněte postup predikce na principu strojového učení, navrhněte prediktor a implementujte ho. Prediktor naučte na shromážděných datech a jeho schopnost predikce ověřte na datech, která pro učení nebyla použita.
4. Vytvořený prediktor zdokumentujte a experimentálně ověřte.

Seznam odborné literatury: Dodá vedoucí práce.

Vedoucí bakalářské práce: Mgr. Karel Hanton

Platnost zadání: do konce zimního semestru 2012/2013


prof. Ing. Vladimír Mařík, DrSc.
vedoucí katedry




prof. Ing. Pavel Ripka, CSc.
děkan

V Praze dne 9. 1. 2012

Prohlášení autora práce

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

25. 5. 2012

V Praze dne

Peter Hroššo

Peter Hroššo

Poděkování

Rád bych vyjádřil svou vděčnost vedoucímu této práce, panu Mgr. Karlovi Hantonovi, za jeho ochotu a inspirativní konzultace, které mi vždy ukázaly další směr prací, když bylo třeba. Děkuji také panu prof. Ing. Václavu Hlaváčovi, Csc. za čas, který mi věnoval, a za jeho cenné připomínky, které mi pomohly vidět problém z větší perspektivy. Na závěr bych rád poděkoval Mojmíru Kittlerovi, s nímž jsem nad touto prací často vedl diskuze, které mi osvětlily danou problematiku či pomohly neztratit se ve vlastní práci.

Obsah

1. Abstrakt.....	6
2. Úvod.....	7
3. Cíle práce.....	9
4. Teoretická část.....	10
4.1 Neparametrické metody.....	10
4.2 Odhad hustoty pravděpodobnosti.....	10
4.3 Parzenova okna.....	12
4.4 Pravděpodobnostní neuronové sítě.....	13
4.5 Klasifikace.....	14
5. Data.....	16
6. Praktická část.....	18
6.1 Příprava dat pro PNN.....	18
Filtrace letů.....	18
Výběr vstupů.....	18
Úprava velikostí.....	19
6.2 Implementace.....	21
6.3 Verifikace.....	23
Úspěšnost klasifikace.....	23
Testovací data.....	24
Vliv učení na úspěšnost klasifikace.....	24
Vliv počtu tříd na klasifikaci.....	25
Časová náročnost klasifikace.....	27
Srovnání s jinými metodami.....	28
6.4 Vliv jednotlivých údajů radarového záznamu.....	30
7. Závěr.....	33
8. Seznam použité literatury.....	35
9. Seznam příloh.....	36

1. Abstrakt

Předmětem této práce je predikce budoucí polohy letadla z radarových dat. Práce se skládá z teoretické a praktické části. V teoretické části je vysvětlena metoda použitá při predikci, tj. metoda strojového učení – pravděpodobnostní neuronová síť, založená na Parzenově odhadu. V praktické části je implementován prediktor, který je na základě připravených radarových dat z reálného letového provozu schopen určit čas přistání letadla v Praze a čas průletu nad vybranými významnými body ČR. Je zde popsán způsob implementace a potřebná příprava dat. Na závěr je provedena verifikace funkčnosti programu, ze které vyplývá, jaká data jsou potřebná pro predikci a že zvolená metoda je vhodná pro tuto aplikaci.

The subject of this this paper is the prediction of future position of the aircraft based on radar data. The thesis consists of theoretical and practical part. A method used for the prediction, i.e. method of machine learning – probabilistic neural network based on Parzen estimate, is explained in the theoretical part. In the practical part a predictor is implemented, which is able to determine the time of landing in Prague and the time to approach significant points of the Czech Republic. The implementation and the preparation of the data is documented. The functionality of the program is verified at the end. It is shown which data are needed for the prediction and that the chosen method is suitable for this application.

2. Úvod

V řízení letového provozu je prvořadým požadavkem jeho bezpečnost. Jedním z prvků jsou Safety Nets, bezpečnostní sítě, které mají v poslední chvíli upozornit, že v prostoru dochází k porušení některého z bezpečnostních pravidel (přiblížení letadla k letadlu nebo k zakázané oblasti, nedodržení letové trati, nízký průlet, nedodržení nařízené letové hladiny a další). Safety Nets se zabývají i pozemním provozem, kdy kontrolují, zda letadlo na ploše nevjíždí do prostoru, kam nemá povolení a podobně. Ihned po bezpečnosti následuje požadavek na efektivitu letového provozu. Volba optimální trasy letu, minimalizace čekacích časů jak za letu (například před přistáním), tak na zemi (správně naplánovaný čas odbavení, odmražení) a další.

Abychom vyhověli oběma zmíněným požadavkům, potřebujeme co nejpřesněji odhadnout budoucí polohu a časy letadla. Nástroje (predikce trajektorie, radarová predikce), které dnes poskytují tyto informace, mají často detailně implementovány jak obecná pravidla a postupy řízení letového provozu, tak specifická pravidla a postupy lokální. S tím souvisí též detailní znalost samotného leteckého prostoru, všech jeho bodů, tratí a dalších údajů. Nezřídka musí tyto nástroje též sledovat uživatelské (řídicí) vstupy a na jejich základě upravovat odhady budoucích poloh a časů.

Hlavní myšlenkou práce je zjistit, zda by tyto nástroje mohly být nahrazeny učícím se algoritmem, který sledováním reálného (radarového) provozu bude schopen budoucí polohu a čas predikovat jen se základními znalostmi univerzálních principů létání.

Cílem této bakalářské práce je vytvořit nástroj, který tuto myšlenku ověří a který bude jako prototyp použitelný k odhadu času, kdy letadlo dosáhne předem určeného, význačného bodu (práh dráhy, koordinační bod), což je tedy jen část výše naznačené komplexní úlohy.

V teoretické části se budeme zabývat vysvětlením pojmů a principů z oblasti letectví, které souvisí s touto prací. Dále bude popsán princip získávání vstupních dat, jejich povahu a co tato data poskytují. Následně navážeme volbou vhodné neuronové sítě pro tento úkol a vysvětlíme, proč je vhodná. S tím také souvisí volba prezentace výsledku.

Praktickým úkolem je pak samotná implementace nástroje, příprava dat, provádění učení, získávání výsledků a verifikace nástroje, tedy ověření bezchybné funkčnosti. Následuje část analýzy, ve které

je třeba ověřit vliv jednotlivých parametrů vstupních dat na výsledek. Například v nočních hodinách jsou časy kratší, letová hladina / rychlost / typ letadla se vzájemně doplňují (typ určuje výkon a ten je dán i vazbou hladina-rychlost).

Po odladění vstupních parametrů bude provedena validace nástroje (tím i samotného přístupu), tedy ověření naučené neuronové sítě na testovací množině dat, případně přímo na datech reálných.

Cílem je pro libovolnou vstupní konfiguraci určit čas. Samotný časový údaj je diskrétní informace, která je pro některé aplikace dostačující, ale ve skutečnosti se jedná jen o nejpravděpodobnější čas. Pro některé aplikace (například porovnání pořadí přistání dvou letů) bude užitečné i pravděpodobnostní rozdělení časů, například formou histogramu.

Cílem do budoucna by mohlo být vytvoření nástroje, který by využíval výslednou neuronovou síť pro vyhodnocování časů a zároveň by ji v pravidelných intervalech (den, týden) průběžně učil. Takový nástroj by se pak již principiálně podobal reálnému řešení a zároveň by umožnil dlouhodobé sledování vývoje učení a dokázal by i reagovat na občasné změny struktury leteckého prostoru i pravidel.

3. Cíle práce

- 1) Vytvořit funkční prototyp prediktoru času letadla na význačném bodě ČR**
- 2) Ověřit funkčnost tohoto prediktoru a najít faktory, které na predikci mají vliv**
- 3) Zdokumentovat tuto práci a vysvětlit použité metody**

4. Teoretická část

Metody učení s učitelem (supervised learning) se obecně dělí na parametrické a neparametrické. *Parametrické* metody pracují za předpokladu, že známe pravděpodobnostní rozložení klasifikovaných vzorků předem. Navíc používají některá zjednodušení, která jsou v praxi nesplnitelná – např. se snaží aproximovat vícedimenzionální rozdělení hustoty pravděpodobnosti směsí jednodimenzionálních funkcí, nebo předpokládají, že aproximovaná rozdělení mají jedno lokální maximum, což v reálném světě také nemusí být vždy splněno.

4.1 Neparametrické metody

Metody, které na takových těžko splnitelných předpokladech nestojí a snaží se aproximovat libovolné rozdělení hustoty pravděpodobnosti, se jmenují *neparametrické*. Níže uvádíme, jak takové metody fungují, zda odhady hustot pravděpodobností konvergují ke skutečným rozdělením a popíšeme metodu, použitou v praktické části této práce.

4.2 Odhad hustoty pravděpodobnosti

Pravděpodobnost P , že vektor \mathbf{x} se nachází v regionu \mathbf{R} , je integrálem hustoty pravděpodobnosti přes tento region:

$$P = \int_{\mathbf{R}} p(\mathbf{x}') d\mathbf{x}' \quad (1)$$

Z toho vyplývá, že "... pravděpodobnost P je jakousi vyhlazenou či zprůměrovanou verzí funkce hustoty pravděpodobnosti $p(\mathbf{x})$ a tak můžeme p aproximovat odhadem pravděpodobnosti P ." (DUDA 2001, str. 162)

Tento vztah můžeme přepsat pro diskrétní systém s n vzájemně nezávislých a identicky rozložených vzorků $\mathbf{x}_1, \dots, \mathbf{x}_n$. Pak pravděpodobnost, že k z těchto n vzorků se bude nacházet v regionu \mathbf{R} je dána binomickou větou:

$$P_k = \binom{n}{k} P^k (1-P)^{n-k} \quad (2)$$

se střední hodnotou:

$$E(k) = nP \quad (3)$$

Lze předpokládat, že poměr k/n bude velice dobrým odhadem pravděpodobnosti P a tedy i hustoty pravděpodobnosti, protože binomické rozdělení pro k ostře kulminuje nad střední hodnotou. Tento odhad se navíc bude zlepšovat s rostoucím n .

Předpokládejme nyní, že $p(\mathbf{x})$ je spojitá a že velikost regionu R je tak malá, že p můžeme považovat za konstantní uvnitř tohoto regionu. Pak platí:

$$\int_R p(\mathbf{x}') d\mathbf{x}' \simeq p(\mathbf{x}) V, \quad (4)$$

kde bod \mathbf{x} se nachází v regionu R a V je objem ohraničený regionem R . Z rovnic č.1, 3 a 4 odvodíme odhad pro $p(\mathbf{x})$:

$$p(\mathbf{x}) \simeq \frac{k/n}{V} \quad (5)$$

Pokud budeme při konstantním objemu V zvyšovat množství trénovacích vzorků, poměr k/n bude konvergovat (v pravděpodobnosti), jak požadujeme. Bude to ale pouze prostorově průměrovaný odhad $p(\mathbf{x})$:

$$\frac{P}{V} = \frac{\int_R p(\mathbf{x}) d\mathbf{x}}{\int_R d\mathbf{x}} \quad (6)$$

Pokud chceme odhadnout přímo $p(\mathbf{x})$, musíme zvolit V jdoucí k nule. To ale přináší problémy při realizaci. Při omezeném počtu vzorků totiž dojdeme do situace, kdy objem V bude tak malý, že se v něm nebudou nacházet žádné vzorky a náš odhad $p(\mathbf{x}) \simeq 0$ bude nepoužitelný.

Z praxe víme, že počet vzorků je vždy omezený. Z toho vyplývá, že objem V nemůže být libovolně malý. Pokud chceme použít výše uvedený odhad, musíme připustit, že k / n bude mít určitou odchylku a že odhad $p(\mathbf{x})$ bude do určité míry průměrovaný.

4.3 Parzenova okna

Tato metoda, jinak nazývaná jádrovým odhadem hustoty, patří do skupiny neparametrických metod. Pro vysvětlení, jak tato metoda funguje, si představme, že region R je d -dimenzionální hyperkrychle s délkou hrany h_n . Její objem pak je:

$$V_n = h_n^d \quad (7)$$

Počet vzorků, které spadají do tohoto regionu, můžeme analyticky vyjádřit pomocí funkce Parzenova okna („oknová funkce“) φ :

$$\varphi(\mathbf{u}) = \begin{cases} 1 & |u_j| \leq \frac{1}{2}, \quad j = 1, \dots, d \\ 0 & \text{jinak} \end{cases} \quad (8)$$

Tato funkce vymezuje krychli se středem v počátku. Z definice φ vyplývá, že $\varphi((\mathbf{x} - \mathbf{x}_i)/h_n)$ se rovná jedné, pokud \mathbf{x}_i spadá do regionu hyperkrychle o objemu V_n , se středem v \mathbf{x} , a je rovna nule jinak. Počet vzorků v této hyperkrychli je tedy:

$$k_n = \sum_{i=1}^n \left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n} \right) \quad (9)$$

Využitím rovnice č.5 dostáváme odhad:

$$p_n(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \frac{1}{V_n} \varphi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right) \quad [2] \quad (10)$$

Takto vyjádřený odhad vybízí k myšlence, že funkce Parzenova okna nemusí být limitovaná na hyperkrychli, ale můžeme použít obecnější třídu funkcí. Rovnice č.10 pak vyjadřuje odhad hustoty pravděpodobnosti jako průměr funkcí \mathbf{x} a vzorků \mathbf{x}_i . Oknová funkce zde slouží k interpolaci – každý vzorek přispívá k odhadu v závislosti na jeho vzdálenosti od \mathbf{x} .

Pokud chceme, aby $p_n(\mathbf{x})$ byla legitimní funkcí hustoty pravděpodobnosti, musíme požadovat, aby byla nezáporná a její integrál byl roven jedné. Tento požadavek je splněn, pokud přímo oknová funkce bude funkcí hustoty pravděpodobnosti. "Pokud:

$$\varphi(\mathbf{x}) \geq 0 \quad (11)$$

a

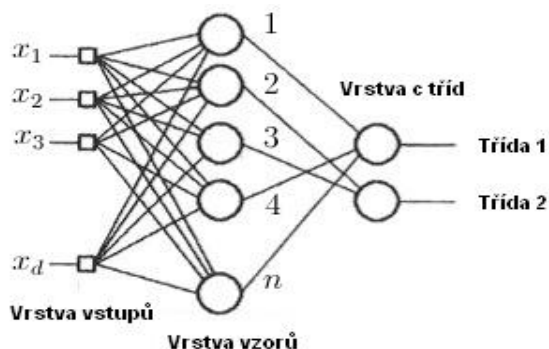
$$\int \varphi(\mathbf{u}) d\mathbf{u} = 1 \quad (12)$$

a zároveň stále platí vztah $V_n = h_n^d$, pak $p_n(\mathbf{x})$ také splňuje výše zmíněné požadavky." (DUDA 2001, str. 165)

4.4 Pravděpodobnostní neuronové sítě

Metodu Parzenových oken lze jednoduše implementovat pomocí pravděpodobnostní umělé neuronové sítě PNN (z anglického „Probabilistic Neural Network“).

Její topologie bude mít následující charakter:



Na obrázku vidíme neuronovou síť sloužící k Parzenovu odhadu, založeném na n vzorech, každý z nich má d dimenzí a je náhodně navzorkován z c tříd. PNN se v takovém případě skládá ze vstupní vrstvy o d prvcích, kde každý prvek je propojen s každým z n vzorů a každý vzor je propojen s jedním z c prvků výstupní vrstvy reprezentující třídy klasifikace.

Spojení mezi vstupní vrstvou a jednotlivými vzory zprostředkovávají váhy, které jsou nastavitelné a budou trénovány v několika jednoduchých krocích.

Nejdříve je třeba provést normalizaci vstupních vektorů, na kterých bude síť trénována. Normalizací vstupního vektoru máme na mysli úpravu velikostí jeho prvků tak, aby

$$\sum_{i=1}^d x_i^2 = 1 \quad (13)$$

První normalizovaný trénovací vektor vložíme na vstup neuronové sítě a nastavíme váhy spojující vstupní vrstvu a první prvek vrstvy vzorů tak, aby $\mathbf{w}_1 = \mathbf{x}_1$. Následně je vytvořeno spojení mezi prvním vzorem a odpovídající třídou tohoto vzoru. Celý proces se opakuje se všemi zbývajícíchmi prvky trénovací množiny tak, že

$$\mathbf{w}_k = \mathbf{x}_k \text{ pro } k = 1, \dots, n \quad (14)$$

Po takovémto tréninku máme síť, která je plně propojena mezi prvky vstupní vrstvy a vzory, a jednotlivě propojená mezi vzory a třídami.

4.5 Klasifikace

Vstupní vektor je neuronovou sítí klasifikován v těchto krocích:

- Vektor je vložen na vstupní vrstvu PNN
- Každý prvek (neuron) vrstvy vzorů spočte svou aktivaci a její hodnotu předá prvku výstupní vrstvy, s nímž je asociován
- Každý prvek výstupní vrstvy sečte příspěvky od všech prvků vrstvy vzorů
- Výstupem PNN je třída, jíž odpovídá prvek výstupní vrstvy s nejvyšším součtem

Nyní si podrobněji vysvětlíme, co přesně se v jednotlivých krocích klasifikace děje:

Aktivace neuronu

Každý neuron spočte svou aktivaci (*net*) jako součin vektoru vstupu a vektoru vah:

$$net_k = \mathbf{w}_k^T \cdot \mathbf{x} \quad (15)$$

Aktivační funkce

Máme obecnou funkci Parzenova okna φ :

$$\varphi\left(\frac{\mathbf{x} - \mathbf{w}_k}{h_n}\right) \quad (16)$$

Pro odhad hustoty pravděpodobnosti tuto obecnou funkci nahradíme Gaussovou funkcí s parametrem σ , který ovlivňuje šířku Gaussovského okna:

$$\varphi\left(\frac{\mathbf{x} - \mathbf{w}_k}{h_n}\right) = e^{-(\mathbf{x} - \mathbf{w}_k)^t (\mathbf{x} - \mathbf{w}_k) / 2\sigma^2} \quad (17)$$

Následujícími úpravami dostaneme aktivační funkci $f(x)$:

$$\begin{aligned} f(x) &= e^{-(\mathbf{x} - \mathbf{w}_k)^t (\mathbf{x} - \mathbf{w}_k) / 2\sigma^2} = \\ &= e^{-(\mathbf{x}^t \mathbf{x} - \mathbf{w}_k^t \mathbf{w}_k - 2\mathbf{x}^t \mathbf{w}_k) / 2\sigma^2} = e^{(net_k - 1) / \sigma^2} \end{aligned} \quad (18)$$

Výstup každého neuronu roven hodnotě aktivační funkce pro danou aktivaci je předán prvku výstupní vrstvy, s nímž je daný neuron asociován.

Prvky výstupní vrstvy sečtou příspěvky jednotlivých neuronů. Třída, již odpovídá prvek výstupní vrstvy s nejvyšším součtem, je požadovaným Parzenovým odhadem a výsledkem klasifikace.

5. Data

Data, která jsem pro trénink a testování neuronové sítě používal, pocházejí z ŘLP (Řízení letového provozu). Jedná se o data pocházející z radarového záznamu letového provozu vzdušného prostoru ČR, obohacená o položky z letového plánu, která jsme dostali již předzpracovaná v textovém souboru.

Máme k dispozici asi 1GB radarových dat. Tato data jsme rozdělili do 14 menších balíků. Každý z nich obsahuje záznamy maximálně 6200 letů, aby je bylo možné bez problémů načítat. Data každého letu jsme uložili do matlabovské struktury s těmito významnými položkami:

- Identifikace letadla (kombinace znaků a čísel)
- Call sign (volací kód letadla, může se opakovat např. pro pravidelné lety)
- Typ letadla
- Popis letadla (např. počet a typ motorů, atp.)
- Letiště odletu
- Letiště destinace
- Čas pořízení radarového záznamu
- Poloha letadla (zeměpisná šířka a délka ve stupních)
- Letová hladina (ve stovkách stop)
- Rychlost letadla (v metrech za sekundu)

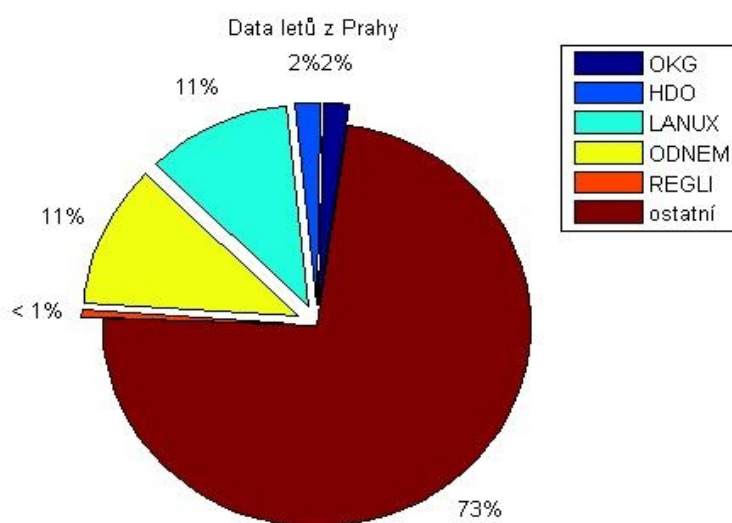
Pro praktické využití je zajímavé vědět, jaké lety jsou v datech zaznamenány. Z povahy úlohy je pro nás důležité vědět, kolik uložených letů máme k dispozici, případně kolik z nich směřuje do Prahy a kolik z Prahy.

Celkový počet letů	Letů z Prahy	Letů do Prahy
85567	1416 (1,7%)	81 (<1%)

Malý počet letů s cílovým nebo startovním letištěm v Praze v porovnání s jejich celkovým počtem je dán tím, že radar snímá všechna letadla, která se nachází ve vzdušném prostoru ČR, respektive v dosahu českých radarů (tzn. i za hranicemi). Přirozeně jen malé procento z nich letí přes Prahu.

Více zarážející je nízký počet letů z Prahy. Dalo by se předpokládat, že letadel bude v Praze přistávat stejně jako jich bude odlétat. Důvodem je způsob detekce letiště odletu. Informace o letišti, ze kterého letadlo startuje, totiž není v samotných radarových datech, ale je přidána z letového plánu. Pokud se tak nestane hned na začátku radarového záznamu, pak tuto informaci nedetekujeme. Z pohledu této práce je to metodická chyba, která ale nemá velký vliv, neboť radarových dat je i přes tento nedostatek v jejich zpracovávání postačující množství.

Důležitější než údaj o počtu letů, je pro nás počet radarových záznamů. Každý let totiž může být jinak dlouhý a při konstantní periodě 4 sekundy, se kterou se radarová data zaznamenávají, se počty záznamů jednotlivých letů liší. Podívejme se nyní na počet záznamů letů z Prahy do významných bodů v ČR (výběr těchto bodů je vysvětlen v Praktické části, Implementace, Významné body). Celkový počet radarových záznamů se startovním letištěm v Praze je 38782 a zastoupení významných bodů je zobrazeno na obrázku níže.



6. Praktická část

6.1 Příprava dat pro PNN

Data z radaru obsahují mnoho údajů, z nichž pro klasifikaci použijeme pouze některé. Než tedy přistoupíme k tréninku neuronové sítě a následné klasifikaci, musíme data profiltrovat a upravit velikosti jednotlivých vstupů. Pro neuronovou síť je totiž vhodné, aby jednotlivé vstupy (typ letadla, poloha, atd.) měly řádově stejnou velikost. Tím zaručíme, že mají všechny vstupy stejnou váhu.

Příprava dat pro potřeby neuronové sítě se skládá ze tří stupňů:

1. **Filtrace letů** zamýšlených pro klasifikaci
2. **Výběr vstupů**, na jejichž základě budeme záznamy klasifikovat
3. Závěrečná **úprava velikostí** jednotlivých vstupů

Nyní si podrobněji popíšeme postup při přípravě dat.

Filtrace letů

Neuronovou síť máme v plánu použít pro predikci času přistání letadel mířících do Prahy a pro predikci času přeletu významnými body ČR u letadel letících z Prahy.

Údaje z letů mířících do Prahy poslouží k predikci času přistání v Praze. Pro predikci času přeletu nad ostatními body filtrujeme naopak lety z Prahy. Z těchto letů pak dále vybíráme pouze ty, které se přiblíží k nějakému významnému bodu (pro účely této práce jsou to OKG, HDO, LANUX, ODNEM, REGLI, jenž jsou blíže popsány níže, v části Významné body). Třídění podle těchto bodů zachováme.

Výběr vstupů

Druhou úroveň filtrování je výběr dat z letů, o kterých již víme odkud a kam směřují, pro potřeby

klasifikace. Vliv jednotlivých vstupů je dokumentován v kapitole Verifikace. Vstupy jsme vybrali empiricky a jsou to tyto:

- Cílový bod
- Poloha letadla
 - zeměpisná šířka
 - zeměpisná délka
- Aktuální letová hladina
- Rychlost
- Typ letadla
- Čas pořízení radarového záznamu
- Denní doba
- Runway

Úprava velikostí

Na závěr je třeba filtrovaná data upravit tak, aby jednotlivé vstupy měly řádově stejnou velikost a mohly být klasifikovány pomocí PNN:

Údaj o čase

Pro účely tréninku neuronové sítě a pro testy úspěšnosti predikce potřebujeme mít informaci o čase přistání (resp. čase přeletu nad daným významným bodem). Ten detekujeme tak, že projdeme všechny radarové záznamy daného letu a najdeme čas, kdy se letadlo dostatečně přiblíží (tj. na vzdálenost menší než $0,0025^\circ$ zem. šířky a délky) k požadovanému bodu. Predikovat však chceme čas *zbývající* do přistání (resp. přeletu), ne čas přistání. Proto pro každý záznam uchováme rozdíl mezi časem přistání a časem pořízení daného záznamu. Tento údaj oddělíme od ostatních dat, jelikož pro reálnou klasifikaci nebude dostupný a uchováme ho pouze pro účely testu správnosti klasifikace a pro trénink neuronové sítě.

Denní doba

Relativizováním času ztrácíme informaci o denní době, která má značný vliv na letový provoz. Proto tuto informaci zachováme a uložíme jako rozdíl času pořízení záznamu a času 0:00 daného dne.

Poloha

Polohu letadla přepočítáme relativně k poloze cílové destinace a to prostým rozdílem, ten totiž zachová informaci o poloze na rozdíl od vzdálenosti, která by tuto informaci ztrácela.

Typ letadla

Typ letadla je v radarových datech uložen jako řetězec znaků (např. 'A320'). Znaky převedeme pomocí matlabovské funkce *double()*, která každému znaku přiřadí jedinečné číslo. Toto číslo poté upravíme podle vzorce:

$$x_{typ} = \frac{(double(a_{320}) - 100)^2}{100}$$

Kde druhá mocnina slouží ke zvýraznění rozdílů mezi jednotlivými typy a ostatní úpravy zaručí velikost výstupu v řádu jednotek.

Letová hladina a rychlost letadla

U letové hladiny a rychlosti letadla upravíme pouze jejich velikosti vydělením stem.

Runway

Runway, na kterou letadlo míří, je faktor, který může značně ovlivnit délku letu. Pokud letadlo přistává např. kvůli větru v jiném směru (tzn. na jinou runway), než odkud letadlo letí, musí k letišti přilétat obloukem, který let prodlouží. Volba runwaye je dána aktuálními podmínkami na letišti (provoz, směr větru, apod.) a je známa dopředu, proto ji můžeme brát jako vstup pro klasifikaci.

Detekce runwaye u letů směřujících do Prahy probíhá zároveň s výše popsanou detekcí času přistání. U letadel startujících z Prahy ukládáme informaci o směru jejich pohybu, když poprvé dosáhnou 100 km/h.

Normalizace

Takto upravené radarové záznamy nakonec normalizujeme ve smyslu:

$$x_{norm} = \frac{x}{\sqrt{\sum_i x_i^2}}$$

6.2 Implementace

Máme připravená data pro trénink a testování neuronové sítě, jakým způsobem však tuto síť použijeme k určení času přistání, respektive přeletu? Čas je spojitá veličina. Výstup neuronové sítě je z principu diskretní. Pokud chceme k predikci použít PNN, musíme tedy čas diskretizovat. Časový úsek mezi přistáním (přeletem) a záznamem s nejvyšším údajem času z trénovacích dat (ten, který odpovídá letadlu s nejdelším trváním letu) rozdělíme na c časových úseků, - v rámci názvosloví neuronových sítí – tříd. Na tyto třídy pak budeme naši síť učit a tyto třídy bude schopna klasifikovat.

Příklad vzorkování času

V použitých radarových datech se vyskytují lety o maximální délce cca 1,5 hodiny. Pokud tento časový úsek navzorkujeme do 150 tříd, budeme po klasifikátoru požadovat přesnost cca půl minuty ($1,5 \cdot 3600 / 150 = 36 \text{ sekund}$). Přesnost predikce bude tedy přímo úměrná počtu tříd, které budeme rozlišovat. Zároveň je ale jasné, že s počtem rozlišovaných tříd bude klesat úspěšnost predikce. Proto je počet tříd velice důležitým parametrem a jeho vlivu se budeme důkladně věnovat v kapitole Verifikace.

Volba vývojového prostředí

Neuronová síť implementující Parzenovy odhady pracuje s velkým množstvím dat, se kterými provádí mnoho jednoduchých operací, většinou sčítání a násobení. Ostatně základní myšlenkou neuronových sítí je distribuovat časovou náročnost výpočtů mezi velký počet neuronů a provádět je paralelně.

Na paralelní výpočty v tomto rozsahu jsme neměli hardwarové prostředky a tak jsme se rozhodli neuronovou síť implementovat v Mathworks Matlabu a maximum výpočtů jsme se snažili zapsat maticově. Matlab totiž provádí operace nad maticemi velice rychle (násobení, prohledávání, logické indexování) a pro účely této práce je vyhovující. Časová náročnost výpočtů je dále popsána v kapitole Verifikace.

Významné body

Z hlediska implementace může naše neuronová síť pracovat pro jakýkoliv bod, pro který budeme

mít dostatek radarových dat. Pro účely této práce jsme však připravili data a otestovali predikci pouze pro několik vybraných bodů ČR. Pro letiště v Praze (souřadnice 50 05 44,80 N 014 15 55,81 E ^[4]) jsme předpovídali čas přistání a pro následující koordinační body jsme určovali čas přeletu:

Koordinační bod	Souřadnice ^[4]	Relativní poloha
OKG	50 03 54,53 N 012 24 20,66 E	Západ
HDO	50 55 41,34 N 014 22 07,68 E	Sever
LANUX	48 53 17,18 N 015 36 56,84 E	Jih
ODNEM	48 51 11,52 N 017 10 04,24 E	Jihovýchod
REGLI	50 01 49,00 N 017 59 44,00 E	Severovýchod

Koordinační bod je místo poblíž státní hranice, ve kterém si dva řídící letového provozu sousedních států předávají dohled nad letadlem. Proto je důležité mít odhad za jak dlouho k němu letadlo dorazí. Koordinačních bodů je po obvodu ČR mnoho. Těchto pět výše uvedených bodů jsme vybrali ze dvou důvodů: provoz kolem nich je relativně hustý, takže máme dostatek radarových dat, a jsou rozmístěné přibližně rovnoměrně po obvodu státu (viz „relativní poloha“ v tabulce výše).

Implementace učení

Radarová data upravená tak, jak bylo popsáno v části Příprava dat pro PNN, už jsou připravená pro použití neuronovou sítí. Protože učení je velice jednoduché (viz rovnice č.14 v kapitole Teoretická část, Pravděpodobnostní neuronové sítě), stačí data pouze načíst do matice vah, případně pro účely testování data před tím rozdělit na testovací a trénovací množinu, čímž je trénink PNN hotov.

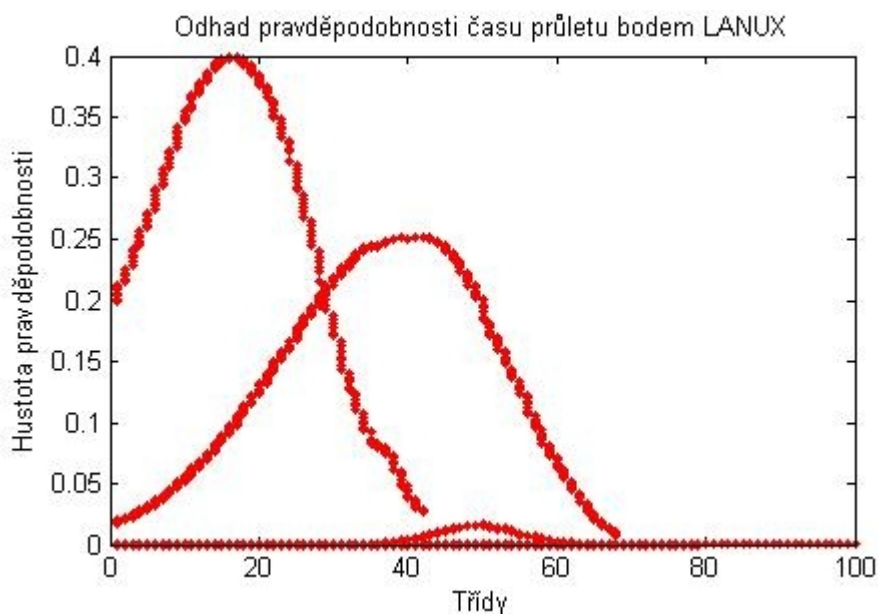
Implementace klasifikace

Postup při klasifikaci radarového záznamu je následující: Spočteme matici aktivací všech neuronů (rovnice č.15 v kapitole Teoretická část, Pravděpodobnostní neuronové sítě) jako součin matice všech vah naší PNN a vektoru radarového vstupu. Pro rychlejší klasifikaci většího počtu vstupů jejich vektory sloučíme do jedné matice všech vstupů, což urychlí výpočty.

Pro tyto aktivace vypočítáme hodnotu aktivační funkce (opět pro všechny najednou). Následně projdeme všechny rozlišované třídy a sečteme příspěvky jednotlivých neuronů. Výsledkem klasifikace je třída s jejich nejvyšším součtem.

6.3 Verifikace

Nejdříve si ukážeme, jak náš prediktor funguje. Na grafu níže vidíme odhad hustoty pravděpodobnosti každé třídy. Názornější je vykreslit výstupy každého neuronu vrstvy vzorů místo součtů pro každou třídu (na nichž je predikce založena). Vidíme tak například, že odhadovaná hustota je směsí hustot pravděpodobností. Důvodem je pravděpodobně přítomnost nějakého neznámého parametru ve vstupních datech, který při klasifikaci nerozlišujeme.



Na tomto obrázku můžeme vidět, že neuronová síť rozlišuje 3 různé funkce hustoty, tzn. výsledná hustota pravděpodobnosti je směsí tří dílčích hustot.

Úspěšnost klasifikace

Ověřit, zda náš program pracuje správně lze dvěma základními způsoby. První možností je vyjádřit úspěšnost klasifikace v procentech jako poměr mezi počtem úspěšně klasifikovaných vzorků a celkovým počtem vzorků v testovací množině dat.

Takto vyjádřená úspěšnost nám pomůže vytvořit si představu, jak náš program funguje. V praxi nám ale o funkčnosti programu více řekne druhá metoda - tou je výpočet rozptylu a směrodatné odchylky.

Výpočet směrodatné odchylky:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2}$$

Výpočet rozptylu, zkráceně MSE (z anglického Mean Squared Error):

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2$$

Směrodatnou odchylku lze brát jako průměrnou odchylku celé testovací množiny, respektive časů určených klasifikací a skutečných časů. Její význam je jednoduše představitelný, protože její jednotkou jsou vteřiny [s].

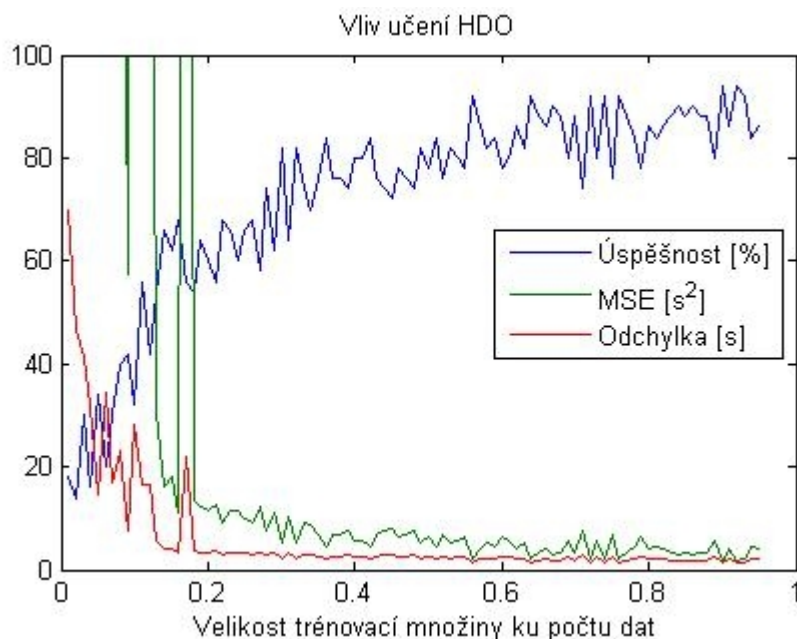
Jednotkou rozptylu je druhá mocnina vteřiny [s²], což nemá žádný fyzikální - tzn. lehce představitelný – význam. Údaj je to ale důležitý, protože díky druhé mocnině znatelně zvýrazňuje velkou chybu klasifikace a takové chyby nás zajímají nejvíce.

Testovací data

Při všech testech v této kapitole jsme postupovali stejně. Všechna dostupná data pro danou destinaci jsme náhodně promíchali a poté rozdělili na trénovací a testovací množinu. Data trénovací množiny byla použita pro váhy neuronové sítě, jejíž úspěšnost klasifikace jsme ověřovali na testovacích datech.

Vliv učení na úspěšnost klasifikace

Úkolem prvního testu, který jsme na našem prediktoru provedli, bylo prověřit, jaký má učení vliv na úspěšnost klasifikace. Predikovali jsme čas průletu koordinačním bodem HDO na 50 vzorcích. Vliv učení jsme zkoumali tak, že jsme vykreslovali úspěšnost klasifikace pro různé poměry velikostí trénovací množiny ku počtu všech dostupných dat pro tento bod.



Z grafu je vidět, že procentuální úspěšnost klasifikace je nízká a odchylka vysoká pro malou trénovací množinu. Při poměru 1:1 již neuronová síť začíná pracovat celkem dobře. Její úspěšnost se pohybuje kolem 75% a odchylka v řádu sekund. Při zvětšování trénovací množiny k 95% dostupných dat se úspěšnost klasifikace vyšplhá až za hranici 90%.

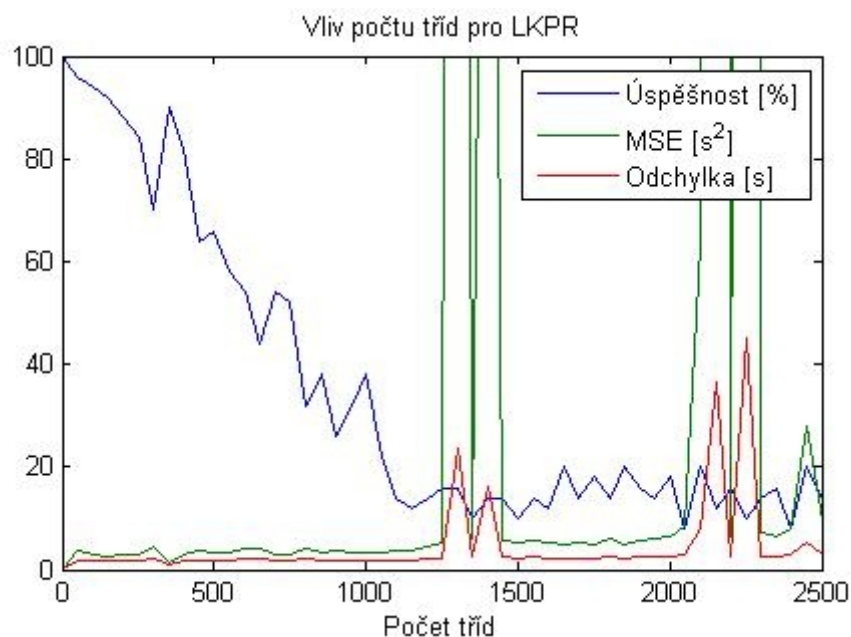
Vliv počtu tříd na klasifikaci

Faktor, který má velký vliv na úspěšnost klasifikace, je počet klasifikovaných tříd. Čím více tříd rozlišujeme, tím přesnější odhad od neuronové sítě požadujeme. Pokud požadujeme málo přesný odhad (malý počet tříd), úspěšnost klasifikace ve smyslu správně určené třídy je vysoká. S rostoucím počtem tříd pak přirozeně klesá, protože tím klademe na predikci vyšší požadavky, které jsou splněny s nižší úspěšností.

Provedli jsme dva testy 50 radarových záznamů: první pro predikci času přistání v Praze (LKPR), pro které máme velký počet trénovacích dat (řádově stovky tisíc), a druhý pro predikci času přeletu nad koordinačním bodem HDO, pro který máme znatelně menší počet dat (řádově tisíce).

LKPR

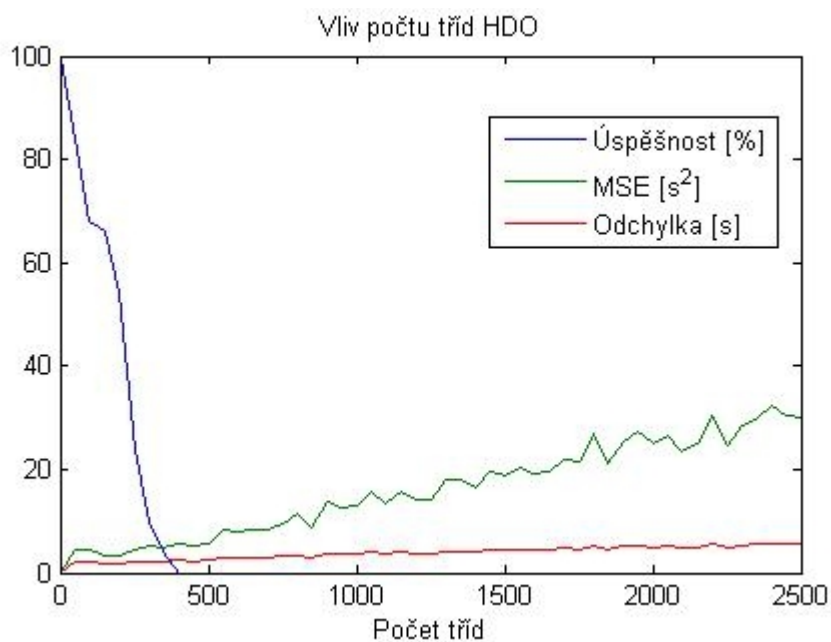
Velikost trénovací množiny: 668991 vzorků (99% dostupných dat)



Z grafu je vidět, že úspěšnost klasifikace prudce klesá při počtu tříd rostoucím od 1 do 1000, ale odchylka predikce je stále v řádu sekund, což je přijatelné. Při cca 1300 rozlišovaných třídách již nastávají problémy s vysokou odchylkou a úspěšnost se pohybuje okolo pouhých 17%. Predikce pro takový počet tříd přestává mít smysl.

HDO

Velikost trénovací množiny HDO: 1577 vzorků (85% dostupných dat)



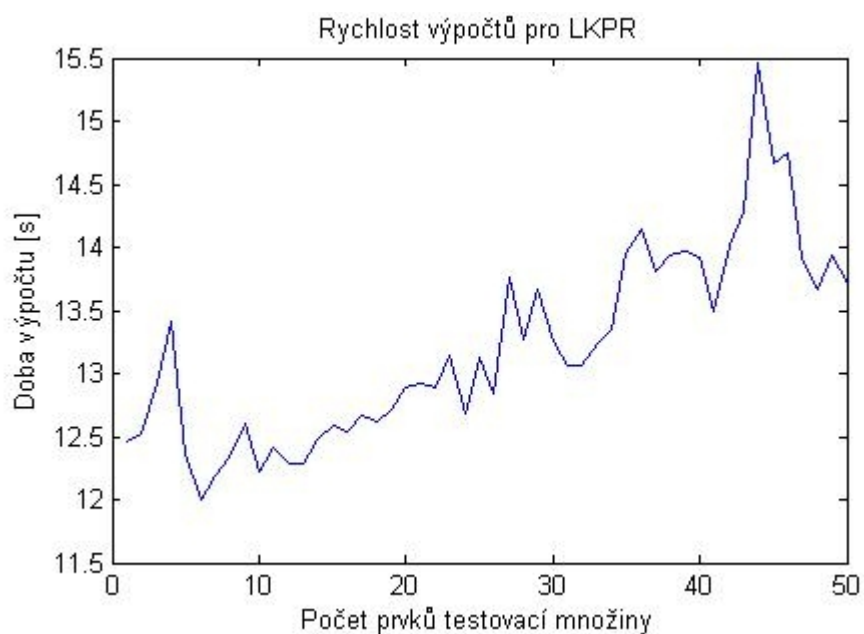
Vidíme, že predikce založená na nižším počtu trénovacích dat se zhoršuje znatelně rychleji při rostoucím počtu tříd, než v předchozím případě (LKPR). Při 400 rozlišovaných třídách je již úspěšnost klasifikace nulová. Odchylka roste lineárně s rostoucím počtem tříd a drží se na přípustné hodnotě jednotek vteřin.

Časová náročnost klasifikace

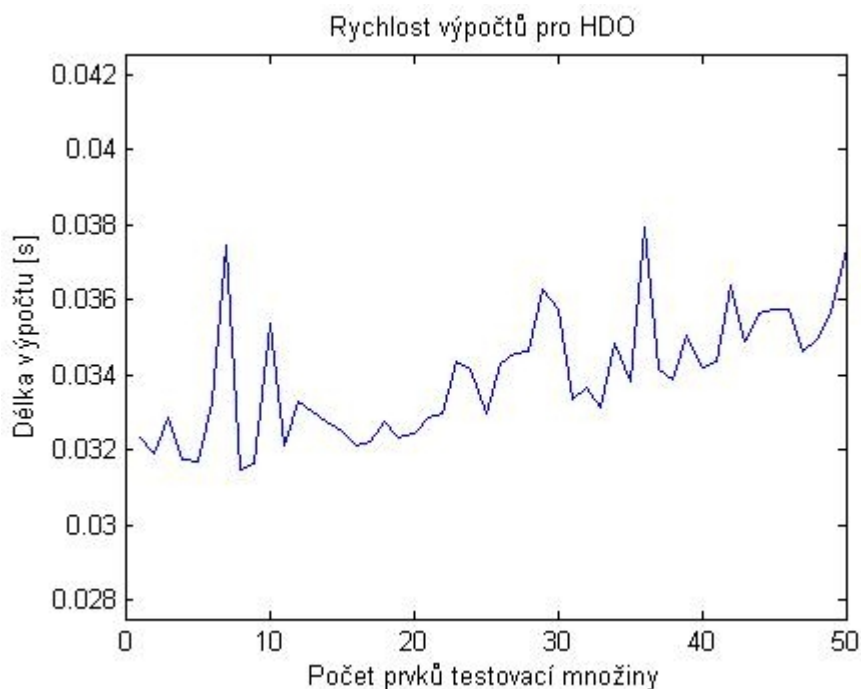
Dalším aspektem, který jsme hodnotili při testování našeho programu, je rychlost výpočtů při klasifikaci vzorků. Ta obecně závisí hlavně na výkonu použité techniky. Proto jsme se raději zaměřili na zjištění závislosti rychlosti výpočtů na počtu vstupních dat.

Provedli jsme 2 testy rychlosti, pro LKPR a pro HDO, jakožto zástupce s nejvyšším a nejnižším počtem trénovacích dat.

Velikost trénovací množiny LKPR: 668991 vzorků (99% dostupných dat)



Velikost trénovací množiny HDO: 1577 vzorků (85% dostupných dat).



V testech byla měřena celková rychlost výpočtů při klasifikaci včetně vzorkování času do tříd. V praxi stačí vzorkování časů trénovací množiny provést pouze jednou při tréninku neuronové sítě a tím klasifikaci urychlit. Protože jsme však volili konstantní počet tříd (1000) pro oba rychlostní testy, tento fakt výsledky testů neovlivní.

Zjistili jsme také, že rychlost výpočtů při vzorkování tříd a klasifikaci je lineárně závislá na počtu dat. Tuto závislost jsme pozorovali při zvyšování počtu vstupních dat (viz graf pro LKPR a HDO) a porovnáním dvou různě velkých trénovacích množin (LKPR má cca 420 krát větší trénovací množinu než HDO a délka výpočtu je přibližně 420 krát větší).

Srovnání s jinými metodami

Posledním použitým způsobem, jak verifikovat funkčnost našeho prediktoru, je porovnání s jinou metodou odhadu a srovnání s reálnou hodnotou času letadla na daném bodě.

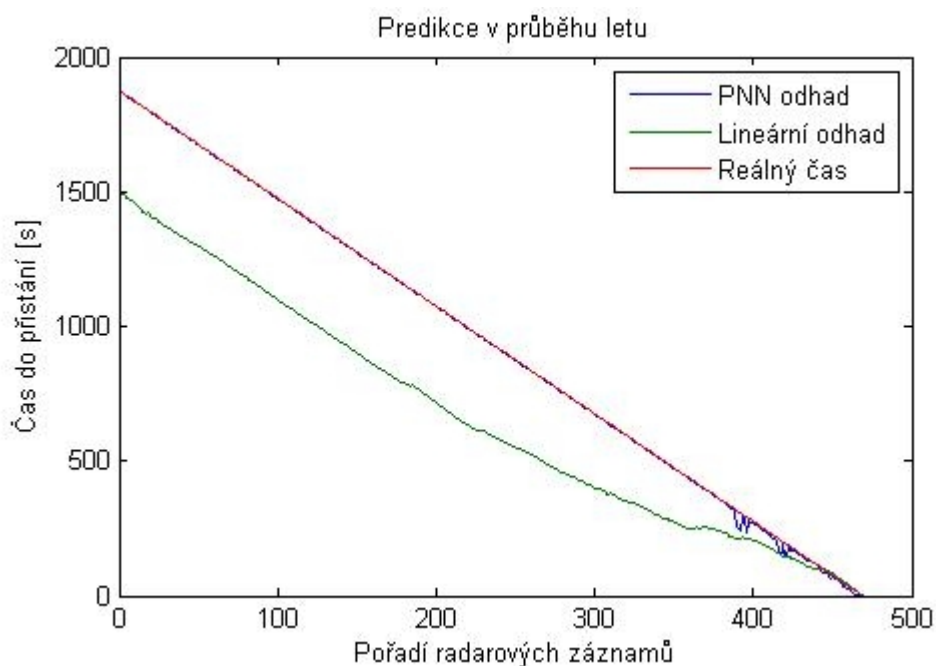
Srovnání jsme pro větší názornost neprováděli na náhodně vybraných vzorcích letů, ale na souvislém radarovém záznamu jednoho letu.

Na prvním grafu vidíme čas zbývajících do přistání v Praze (osa y). Časy jsou v grafu vyneseny ve stejném pořadí, jako byly pořízeny radarové záznamy (osa x) a jsou odlišeny červenou barvou. Zeleně jsou vykresleny časy predikované lineárně z radarových dat o poloze a rychlosti dle vzorce:

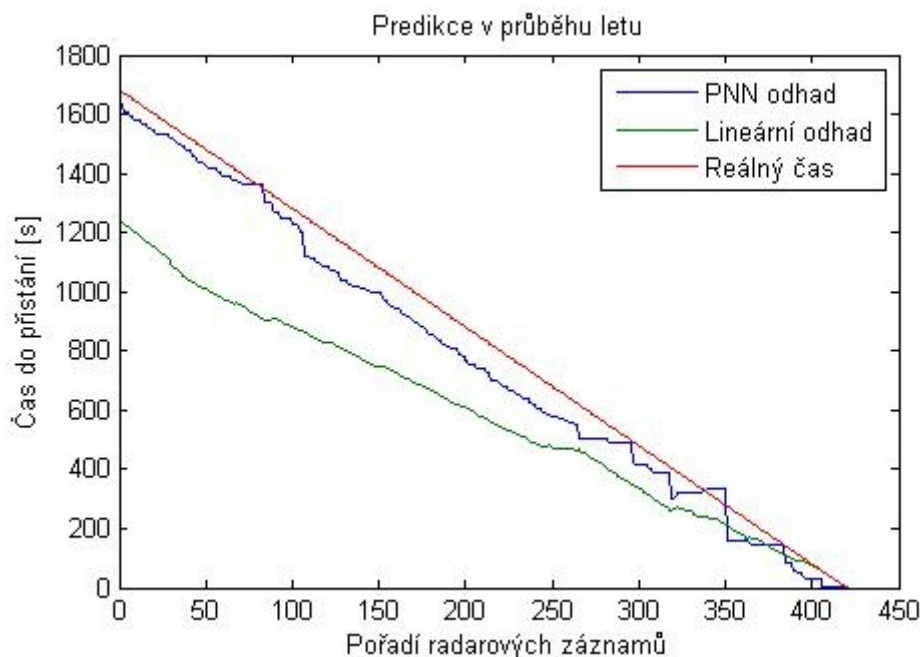
$$t = \frac{s}{v} \text{ [s]}$$

Tento odhad je základem odhadů budoucí polohy letadla, používaných při řízení letového provozu v praxi (v reálném provozu jsou do predikce zakomponovány také znalosti řízení letového provozu).

Modře je zobrazena predikce neuronovou sítí (PNN).



Predikce je provedena na datech letu, která se značně podobají datům v trénovací množině. Jedná se o téměř ideální predikci, se kterou se v praxi nesetkáváme často. Proto jsme provedli ještě jeden stejný test ale pro záznamy speciálně vybrané tak, aby se více lišily od těch v trénovací množině. Takto by mohla vypadat predikce použitá v praxi.



Odchylka času predikovaného PNN od reálného času se drží v mezích do 100 sekund. V grafu můžeme také vidět, že náš odhad je mnohem přesnější než lineární.

6.4 Vliv jednotlivých údajů radarového záznamu

Funkčnost programu byla ověřena a tak můžeme přistoupit ke zkoumání vlivu každého z údajů radarového záznamu. Provedeme testy, jejichž cílem bude zjistit, které vstupy jsou nezbytné pro uspokojivou predikci a které případně můžeme vynechat. Testy provádíme na datech pro LKPR, při 50 testovacích vzorcích a trénovací množině, založené na 95% dostupných dat, přičemž rozlišujeme 100 tříd. Výsledky prezentujeme v přehledných tabulkách, v nichž uvádíme testované vstupy (označené křížkem „X“) a vstupy, které nejsou pro daný test použity (označené „-“). Pro každý test zaznamenáváme úspěšnost klasifikace a MSE.

Vycházíme z předpokladu, že největší vliv na predikci má vstup polohy (zeměpisná šířka a délka). Tento předpoklad ověříme predikcí pouze na základě polohy a predikcí založenou na všech ostatních vstupech mimo polohy:

Úspěšnost [%]	MSE [s ²]	Poloha	Ostatní
22	9,09E+003	X	-
64	1,32E+003	-	X

Vidíme, že samotná poloha k odhadu nestačí, ale bez ní má odhad velkou chybu. Ve všech dalších testech tedy polohu zahrneme do vstupů (a nebudeme ji dále uvádět).

Dalšími důležitými údaji o letadle jsou letová hladina, rychlost a jeho typ. Predikce založená na těchto vstupech má ucházející úspěšnost a mnohem nižší chybu než v předchozím testu:

Úspěšnost [%]	MSE [s ²]	Denní doba	Letová hladina	Rychlost	Typ letadla	Runway
84	54,42	-	X	X	X	-

Dá se předpokládat, že tyto tři vstupy jsou do určité míry provázané. Maximální rychlost letadla je dána jeho typem. Letová hladina je přes hustotu vzduchu, která je závislá na nadmořské výšce, propojena s aktuální rychlostí letadla. Podíváme se tedy, jaký mají tyto vstupy jednotlivě vliv na predikci, abychom zjistili případnou závislost.

Úspěšnost [%]	MSE [s ²]	Denní doba	Letová hladina	Rychlost	Typ letadla	Runway
26	2,51E+003	-	X	-	-	-
46	1,15E+003	-	-	X	-	-
38	2,22E+003	-	-	-	X	-

Ani jeden z těchto vstupů ve spojení s polohou nezajistí dostatečnou úspěšnost predikce. Jejich vzájemnou provázanost se nyní pokusíme ověřit tak, že vynecháme vždy jen jeden z těchto tří vstupů:

Úspěšnost [%]	MSE [s ²]	Denní doba	Letová hladina	Rychlost	Typ letadla	Runway
68	520,23	-	X	X	-	-
70	208,46	-	-	X	X	-
68	282,25	-	X	-	X	-

Úspěšnost klasifikace je značně vyšší, přesto je ale zřejmé, že každý z těchto údajů nese nějakou novou informaci, která predkci zlepšuje, a proto nemůže být žádný z nich vynechán.

Poslední dva vstupy, které by mohly mít vliv na predikci, jsou denní doba a runway, na kterou letadlo přistává, nebo z níž startuje. Po jednom je tedy připojíme k ostatním vstupům pozorujeme, že úspěšnost klasifikace se značně zvýšila a chyba je malá.

Úspěšnost [%]	MSE [s ²]	Denní doba	Letová hladina	Rychlost	Typ letadla	Runway
92	3,69	X	X	X	X	-
90	4,61	-	X	X	X	X

Na závěr se podíváme, jak by vypadala predikce založená pouze na poloze, runwayi a denní době. Pro porovnání uvádíme i predikci se všemi vstupy:

Úspěšnost [%]	MSE [s ²]	Denní doba	Letová hladina	Rychlost	Typ letadla	Runway
62	279,49	X	-	-	-	X
98	0,92	X	X	X	X	X

Provedli jsme také testy všech zbývajících kombinací vstupů, ale žádnou nečekanou závislost mezi nimi jsme neobjevili, z čehož vyplývá, že žádný ze vstupů nemůžeme vynechat. Nejlepší predikce dosáhneme pouze při použití všech vstupů.

Největší vliv na predikci má podle očekávání poloha (bez ní klesne úspěšnost predikce na 64%). Další skupinou vstupů, která výrazně ovlivňuje úspěšnost klasifikace, je letová hladina, rychlost a typ letadla. Bez těchto údajů klasifikujeme správně maximálně 62% záznamů.

Denní doba a runway mají menší vliv (při vyřazení pokles úspěšnosti na 90%, resp. 92%), přesto jejich vliv nemůžeme zanedbat.

7. Závěr

V teoretické části jsme popsali problematiku strojového učení, založeného na neuronových sítích. Po zvážení několika metod jsme pro řešení úlohy predikce času letadla na významných bodech ČR zvolili neparametrickou metodu klasifikace, založenou na Parzenově odhadu. Prediktor jsme implementovali pomocí pravděpodobnostní neuronové sítě.

Provedli jsme trénink této neuronové sítě na radarových datech reálného letového provozu ČR. Připravili jsme data pro predikci týkající se těchto významných (koordinálních) bodů ČR: HDO, LANUX, LKPR, ODNEM, OKG, REGLI. Důkladně jsme zdokumentovali postup při přípravě dat a jejich strukturu. Popsali jsme jednotlivé kroky implementace prediktoru a důkladně jsme se věnovali verifikaci jeho funkčnosti.

Za tímto účelem jsme provedli testy, při kterých jsme ověřili vliv učení na úspěšnost predikce. Uspokojivou úspěšnost klasifikace jsme pozorovali již při poměru velikostí trénovací a testovací množiny jedna ku jedné (úspěšnost $\sim 85\%$) a ta se dále zvyšovala s rostoucí trénovací množinou až k 95% .

Dále jsme zkoumali vliv počtu rozlišovaných tříd na úspěšnost klasifikace a shledali jsme, že ideální počet je 150 tříd. Čas v takovém případě predikujeme s cca 30 vteřinovou přesností, úspěšnost klasifikace je vysoká ($\sim 94\%$) a směrodatná odchylka je v řádu sekund.

Testovali jsme také časovou náročnost predikce. Protože je tento údaj závislý na použité technice, stačilo nám obecně platné zjištění, že doba klasifikace roste lineárně s počtem dat trénovací a klasifikované množiny.

Navržený prediktor jsme také porovnali s naivním lineárním odhadem. U letů podobných letům v trénovací množině se predikovaný čas téměř nelišil od reálného času. Při pokusu o ztížení podmínek testu jsme predikovali čas pro let, který se podstatně lišil od letů v trénovací množině. I v tomto testu prediktor obstál dobře – největší chyba predikce se pohybovala kolem 100 sekund.

Na závěr jsme testovali vliv jednotlivých vstupních údajů radarových dat. Nejvyšší váhu při predikci má poloha letadla, následovaná skupinou vstupů: letová hladina, rychlost a typ letadla. Vstupy s nižší váhou jsou denní doba a (startovací / přistávací) runway. Jejich vliv přesto nelze zanedbat, protože zlepšují úspěšnost predikce asi o 8%.

Použitou metodu a způsob implementace považujeme za vhodné pro danou úlohu a predikci při popsanych podmínkách shledáváme dostatečně přesnou.

8. Seznam použité literatury

- [1] PRÍNCIPE, José C. , Neil R. EULIANO , W. Curt LEFEBVRE. *Neural and Adaptive Systems: Fundamentals through Simulations*. New York: Wiley, c2000, xiii, 656 s. ISBN 04-713-5167-9.
- [2] DUDA, Richard O., Peter E. HART and David G. STORK. *Pattern Classification*. 2nd ed. New York: J. Wiley, 2001, xx, 654 s. ISBN 04-710-5669-3.
- [3] http://lis.rlp.cz/ais_data/www_main_control/frm_cz_aip.htm
- [4] http://lis.rlp.cz/ais_data/aip/data/valid/e3-3.pdf

9. Seznam příloh

Na přiloženém CD jsou tyto přílohy:

1. Text bakalářské práce v elektronické podobě

2. Prediktor (praktická část) v podobě matlabovských souborů:

<code>approachTime.m</code>	<code>detectRunway.m</code>
<code>classify.m</code>	<code>extractData.m</code>
<code>dataLKPR.m</code>	<code>main.m</code>
<code>dataOKX.m</code>	<code>train.m</code>

3. Připravená data, na kterých prediktor pracuje:

<code>weightsHDO.mat</code>	<code>weightsODNEM.mat</code>
<code>weightsLANUX.mat</code>	<code>weightsOKG.mat</code>
<code>weightsLKPR.mat</code>	<code>weightsREGLI.mat</code>

4. Skripty použité při verifikaci prediktoru a vizualizaci výsledků:

<code>graphData.m</code>	<code>test2c.m</code>
<code>plotMATs.m</code>	<code>test3inputs.m</code>
<code>test1.m</code>	<code>testFlightLengths.m</code>
<code>test2a.m</code>	<code>testGraphs.m</code>
<code>test2b.m</code>	