

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2018

Bc. Jakub Hladík



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

ANALÝZA ČASOVÝCH ŘAD S VYUŽITÍM HLUBOKÉHO UČENÍ

TIME SERIES ANALYSIS USING DEEP LEARNING

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Jakub Hladík

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Václav Uher

BRNO 2018



Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

Student: Bc. Jakub Hladík

ID: 164281

Ročník: 2

Akademický rok: 2017/18

NÁZEV TÉMATU:

Analýza časových řad s využitím hlubokého učení

POKyny PRO VYPRACOVÁNÍ:

Vytvořte nástroj pro predikci vývoje časových řad. Na vstupu nástroje bude časová řada s jedním nebo více atributy. Tato vstupní časová řada bude sloužit pro vytvoření modelu s využitím hlubokého učení, který bude následně použit pro predikci budoucího vývoje řady.

DOPORUČENÁ LITERATURA:

[1] LÄNGKVIST, Martin; KARLSSON, Lars; LOUTFI, Amy. A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recognition Letters*, 2014, 42: 11-24.

[2] WANG, Xinying; HAN, Min. Online sequential extreme learning machine with kernels for nonstationary time series prediction. *Neurocomputing*, 2014, 145: 90-97.

Termín zadání: 5.2.2018

Termín odevzdání: 21.5.2018

Vedoucí práce: Ing. Václav Uher

Konzultant:

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Cieľom práce bolo vytvoriť nástroj pre predikciu časových radov s pomocou hlbokého učenia. V prvej časti práce je stručný popis hlbokého učenia a jeho porovnanie s klasickým strojovým učením. V ďalšej časti sa nachádza stručný rozbor niektorých, už v praxi využívaných nástrojov pre predikciu. Posledná časť je zameraná na rozbor problému ako aj na samotnú tvorbu programu.

KĽÚČOVÉ SLOVÁ

Umelá inteligencia, hlboké učenie, predikcia časových radov

ABSTRACT

The aim of the thesis was to create a tool for time-series prediction based on deep learning. The first part of the work is a brief description of deep learning and its comparison to classical machine learning. In the next section contains brief analysis of some tools, that are already used for time-series forecasting. The last part is focused on the analysis of the problem as well as on the actual creation of the program.

KEYWORDS

Artificial Intelligence, deep learning, time series forecasting

HLADÍK, Jakub. *Analýza časových řad s využitím hlubokého učení*. Brno, 2017, 49 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedúci práce: Ing. Václav Uher

VYHLÁSENIE

Vyhlasujem, že som svoju diplomovú prácu na tému „Analýza časových řad s využitím hlubokého učení“ vypracoval(a) samostatne pod vedením vedúceho diplomovej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor(ka) uvedenej diplomovej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil(a) autorské práva tretích osôb, najmä som nezasiahol(-la) nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý(-á) následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákoníka Českej republiky č. 40/2009 Sb.

Brno

.....

podpis autora(-ky)

POĎAKOVANIE

Rád bych poděkoval vedoucímu diplomové práce panu Ing. Uhrovi za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno

.....

podpis autora(-ky)



Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic
<http://www.six.feec.vutbr.cz>

POĎAKOVANIE

Výzkum popsaný v tejto diplomovej práci bol realizovaný v laboratóriách podporených projektom SIX; registračné číslo CZ.1.05/2.1.00/03.0072, operačný program Výzkum a vývoj pro inovace.

Brno

.....
podpis autora(-ky)



EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI



OBSAH

Úvod	10
1 Teoretická časť studentskej práce	11
1.1 Hlboké učenie, čo to je, ako funguje	11
1.2 RNN	13
1.2.1 Long short-term memory	14
1.2.2 Rozbor LSTM	15
1.3 Hlboké učenie v praxi	19
1.3.1 Predpoveď počasia	19
1.3.2 Predikcia elektrickej záťaže	20
1.3.3 Diagnostikovanie v zdravotníctve pomocou LSTM	21
1.3.4 Predikcia výkonu solarnej elektrárne v Hae-Nam [14]	22
1.3.5 Hlboké učenie pre predikciu solárnej energie [5]	23
2 Výsledky studentskej práce	25
2.1 Dáta	25
2.1.1 Spracovanie dát	25
2.2 Model	28
2.3 Dosiahnuté výsledky	29
2.3.1 Prvotný model	29
2.3.2 Model s upravenými vstupnými dátami	30
2.3.3 Upravený model	33
2.3.4 Pridanie polohy slnka a skrytých vrstiev	35
2.3.5 Predpoveď pomocou lineárnej regresie	37
2.3.6 XGBoost predikcia	39
2.4 Výsledný model	40
2.5 Zhrnutie výsledkov	43
3 Závěr	44
Literatúra	45
Zoznam symbolov, veličín a skratiek	47
Zoznam príloh	48
A Obsah priloženého CD	49

ZOZNAM OBRÁZKOV

1.1	Porovnanie neurónu a umelého neurónu	11
1.2	Výkonnosť sietí v závislosti na veľkosti dát [3]	12
1.3	Model RNNLSTM [2]	13
1.4	Model LSTM [2]	14
1.5	Priechod stavu bunky [2]	15
1.6	Brána v LSTM [2]	15
1.7	Prvý krok, kde sa rozhoduje, ktoré informácie sa ponechajú a ktoré sa naopak zahodia. [2]	16
1.8	Vytvorenie aktualizácie pre stav bunky. <i>Sigmoid</i> funkcia pre vybratie, ktoré informácia a ako veľmi treba upraviť a hyperbolický tangens pre vytvorenie vhodného vektoru pre úpravy [2]	17
1.9	Spojenie jednotlivých krokov pre úpravu stavu bunky	17
1.10	Upravenie výstupu podľa našich potrieb. Tak aby sa na výstup dostalo len to, čo naozaj chceme a zároveň vďaka <i>tanh</i> funkcii v rozmedzí -1 a 1 [2]	18
1.11	Model siete [11]	22
2.1	Ukážka dát upravených na učenie pod dohľadom. S desiatimi vstupnými premennými. Pri čom var1 až var5 reprezentujú výkon, teplotu, osvit, azimut a zenit, menovite.	27
2.2	Grafy vykreslené programom pre prvotný model . Prvý ukazuje ako sa vyvíjala chyba pri učení a testovaní. Druhý graf zobrazuje porovnanie predikovaného a očakávaného výstupu na rozsahu 1000 vzoriek.	30
2.3	Grafy vykreslené programom pre model s upravenými vstupnými dátami	31
2.4	Grafy vykreslené programom pre model s dátami z dvoch rokov	32
2.5	Grafy vykreslené programom pre upravený model	33
2.6	Grafy vykreslené programom pre model s dvoma skrytými vrstvami	34
2.7	Grafy vykreslené programom pre model s tromi skrytými vrstvami	35
2.8	Grafy vykreslené programom pre model s údajmi o polohe slnka . Prvý ukazuje ako sa vyvíjala chyba pri učení a testovaní. Druhý graf zobrazuje porovnanie predikovaného a očakávaného výstupu na 1000 vzorkoch.	36
2.9	Graf zobrazujúci predikciu pomocou lineárnej regresie pri škálovaných vstupných hodnotách.	37
2.10	Graf zobrazujúci predikciu pomocou lineárnej regresie s väčším detailom, pre lepšie pozorovanie záporných hodnôt.	38

2.11 Graf zobrazujúci predikciu pomocou lineárnej regresie pri škálovaných vstupných hodnotách. Hodnoty predikcie výkonu boli orezané od θ po <i>nekonečno</i>	39
2.12 Obrázok zobrazujúci konečnú podobu modelu.	41
2.13 Graf zobrazujúci vývoj chyby a detailnejší pohľad na porovnanie predikovaného a očakávaného výstupu na 1000 vzorkoch	42

ÚVOD

Strojové učenie sa v praxi používa už dlhú dobu. No len nedávno toto odvetvie zažilo obrovský rozkvet. Je to spôsobené množstvom dát, ktoré má ľudstvo k dispozícii, ako aj samotným časom, investovaným do výskumu umelej inteligencie a odvetviami s ňou spojenými. Každý deň pribúdajú nové a nové dáta. So správnym pohľadom a so správnym prístupom sa dá toto kvantum dát využiť práve pri učení neurónových sietí. A toto isté kvantum dát, práve podporuje nedávny rozkvet istého druhu učenia, ktoré sa volá hlboké učenie. Neurónová sieť postavená na hlbokom učení potrebuje práve veľký obnos dát pri procese učenia. V dnešnej dobe sa takéto siete využívajú už aj v praxi. V práci sa neskôr rozoberajú isté projekty založené na tomto druhu učenia. Najčastejšie sa využíva pri tvorení modelov sietí, ktoré majú za úlohu predikovať časové rady.

Zadaním práce je vytvoriť nástroj pre predikciu časových radov na základe vstupných dát. Dáta, ktoré boli k dispozícii sú merania záťaže, osvitú a teploty, po dobu dvoch rokov, v intervaloch 15 minút. To predstavuje veľké množstvo dát. V práci sa rozoberá predspracovanie dát a následné použitie dát pri učení. Pre vypracovanie bolo zvolené práve hlboké učenie, ktoré, ako je opísane neskôr, je vhodné pre riešenie takéhoto typu problému. Pri učení tieto dáta postupne prechádzajú sieťou, vrstvu po vrstve, až kým sa nedopracuje k výsledku, ktorý potom porovnáva s očakávanou hodnotou. Takýmto spôsobom sa vypočíta chyba predikcie, ktorá sa následne šíri sieťou späť a na základe nej, sú upravované váhy tak, aby sa pri ďalších prechodoch sieťou táto chyba minimalizovala. Preto model navrhnutý v projekte slúži len na predikciu výkonu.

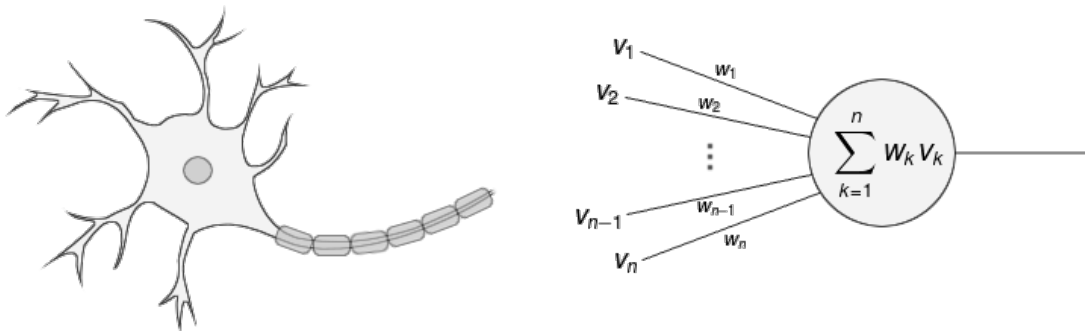
1 TEORETICKÁ ČÁST STUDENTSKÉ PRÁCE

Táto kapitola obsahuje stručný popis neurónových sietí a použitie podobných neurónových sietí v praxi.

1.1 Hlboké učenie, čo to je, ako funguje

Skôr ako sa dostaneme k samotnému hlbokému učeniu, spomenieme si zopár vecí, ktoré je dobré vedieť, než prejdeme na danú tému.

Pojem hlboké učenie, sa spája s umelými neurónovými sieťami. Umelá neurónová sieť – Artificial Neural Network (ANN) je inšpirovaný biologickými neurónovými sieťami. Takto navrhnutý systém sa dokáže učiť a zdokonalovať svoje výsledky na základe predložených príkladov, bez nutnosti špecifikovať konkrétny problém. [2] [3]



Obr. 1.1: Porovnanie neurónu a umelého neurónu

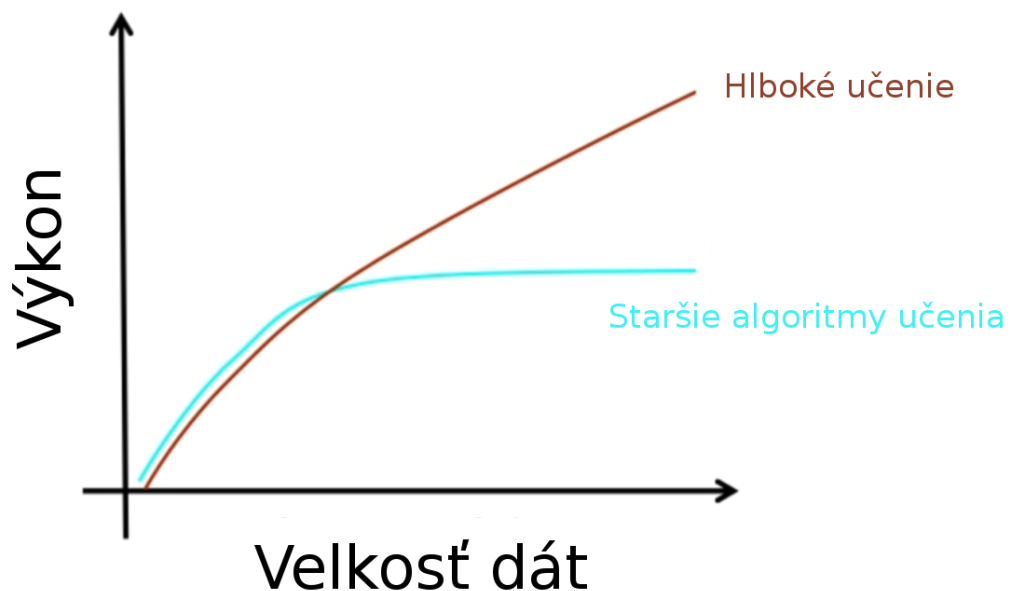
Aby ANN fungovala, musí sa najskôr naučiť na príkladoch, ktoré jej predložíme. Toto učenie môže byť buď:

1. **Učenie s učiteľom** - učenie sa na vopred označených dátach
2. **Učenie bez učiteľa** - dáta nie sú vopred označené, ide o hľadanie skrytých vzorov v dátach

Hlboké učenie je zvláštny druh strojového učenia, ktoré dosahuje veľkú silu a flexibilitu tým, že sa naučí predstaviť svet ako vnorenú hierarchiu konceptov, pričom každý koncept je definovaný vo vzťahu k jednoduchším konceptom a abstraktnejšie reprezentácie sa vypočítavajú z hľadiska menej abstraktných. [3] Klasické neurónové

siete sa vyznačujú jednou vstupnou, jednou výstupnou a väčším počtom skrytých vrstiev medzi nimi. Ako náhle sa v sieti nachádza viac vrstiev, hovoríme o hlbokoj neurónovej sieti. V hlbokoj neurónovej sieti, sa uzly každej vrstvy trénujú na odlišnej sade funkcií, ktoré sú výstupom predchádzajúcich vrstiev. Čím hlbšie sa v sieti dostávame, tým zložitejšie funkcie sme schopný rozpoznávať.

Najväčší rozdiel medzi klasickým strojovým a hlbokým učením je v účinnosti na veľkosti dát. Hlboké neurónové siete vynikajú pri veľkých dátach, ktoré potrebujú na pochopenie súvislostí medzi nimi. Hlboké učenie podáva vynikajúce výsledky pri práci s neštrukturovanými dátami (napr. obrázky, videá, zvuk, text). [3]



Obr. 1.2: Výkonnosť sietí v závislosti na veľkosti dát [3]

Najčastejšie využitie hlbokého učenia je pri spracovávaní neštrukturovaných dát ako napríklad:

- **Počítačové videnie** - rozpoznávanie tvárí, alebo predmetov z fotografií. [13]
- **Vyhľadávanie informácií** - pre aplikácie ako sú vyhľadávacie nástroje. [12]
- **Marketing** - cieleňá reklama, automatické e-maily. [10]
- **Medicína** - detekcia rakoviny, alebo rôznych anomálií [11]

- **Spracovanie reči** - rozpoznanie emócií z hovorenej reči, alebo preklad cudzieho jazyka. [7]

Samozrejme, toto je len zlomok odvetví, kde sa dajú neurónové siete využiť. No vždy treba pri práci s nimi myslieť na to, že to je blackbox, do ktorého nevidíme. Vieme pomocou rôznych matematických operácií zistiť, ktoré uzly boli aktivované, no nikdy si nemôžeme byť istý, čo jednotlivé uzly, alebo vrstvy predstavovali v konečnom výsledku. [1] Preto si treba dávať veľký pozor pri výbere tréningovej množiny. Ďalšia vec na ktorú sa musí človek sústrediť, je správny výber neurónovej siete. Existuje mnoho druhov neurónových sietí a každá sa hodí na niečo iné. Napríklad:

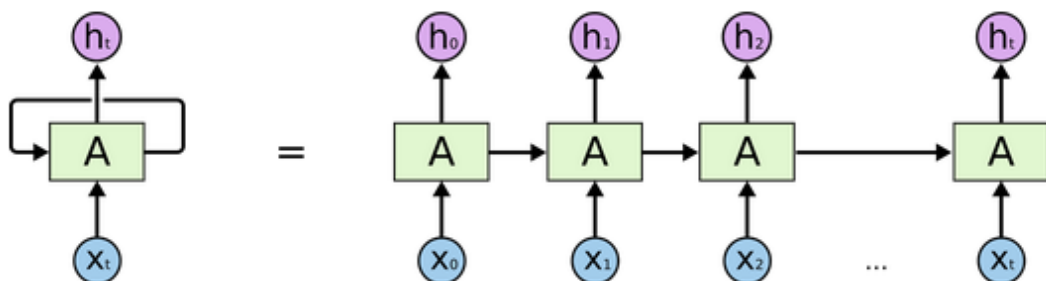
- Konvolučné (CNN) - rozpoznanie objektov (obrázky, videá)
- Pravdepodobnostné (PNN) - klasifikácia a rozpoznanie vzorov
- Rekurentné (RNN) - rozpoznanie rukopisu, rozpoznanie reči

V ďalšej kapitole sa budeme podrobnejšie venovať práve RNN, pretože svojou vnútornou štruktúrou sú vhodnejšie pre predikciu časových radov.

1.2 RNN

Rekurentná neurónová sieť – Recurrent Neural Network (RNN) je sieť, ktorá má v sebe cykly, ktoré umožňujú informácií, aby pretrvala, pretože ľudský mozog tiež nezačína každú sekundu myšlienkové procesy od začiatku, ale myšlienky pretrvávajú. [6] Tradičné neurónové siete túto vlastnosť nemajú, preto sa nedávno začali RNN používať vo veľkom, napríklad pri rozpoznaní reči, prekladaní do cudzích jazykov a podobne.

Cykly u neurónových sietí môžu pôsobiť trochu mätúco, no keď sa nad tým zamyslíme, nejde o nič iné, ako o viacero kópií tej istej neurónovej siete. Hlavnou výhodou

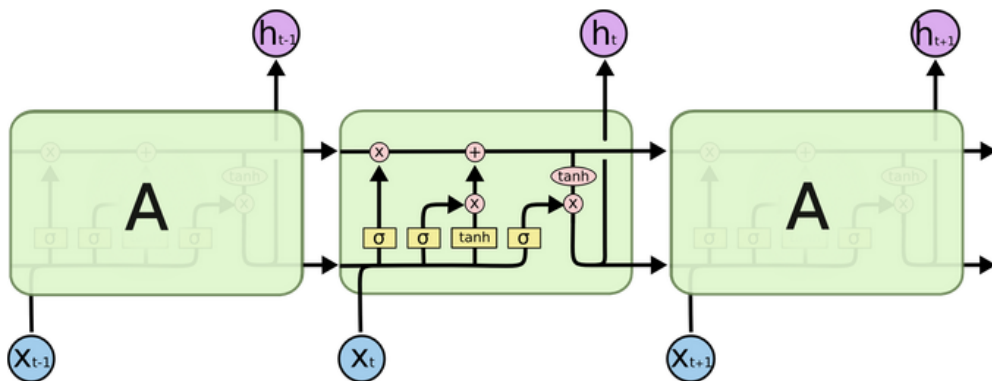


Obr. 1.3: Model RNNLSTM [2]

RNN je, že dokážu použiť staršiu informáciu pri riešení aktuálneho problému. No aj tento koncept má svoje medzery. Napríklad pri dopĺňovaní viet sa spolieha na malú medzeru medzi relevantným slovom a miestom, kam má nové slovo doplniť. No čo ak nastane situácia, kde je medzera veľká a treba sa dívať a kontext celej vety? Žiaľ čím je medzera medzi relevantným slovom a miestom, kam treba doplniť nové slovo väčšia, RNN prestáva byť účinná, pri spájaní informácií. Teoreticky, by mali byť RNN schopné spracovať takéto vzdialené závislosti, no v praxi tomu tak žiaľ nie je. V praxi sa používa veľa modelov RNN. Medzi najviac používané a najviac rozšírené patrí LSTM [4]. No okrem týchto poznáme aj Konvolučná, dlhá krátkodobá sieť – Convolutional Long Short-Term Memory (CLSTM2D) [4], alebo Uzavretá rekurentná jednotka – Gated Recurrent Unit (GRU) [4].

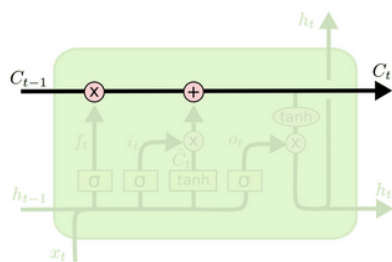
1.2.1 Long short-term memory

Dlhá krátkodobá pamäť – Long short-term memory (LSTM) je špeciálny druh RNN, ktorý navrhli páni Hochreiter a Schmidhuber. [2] LSTM siete sú navrhnuté tak, aby sa vyhli problémom s dlhodobou pamäťou. Uchovanie informácií po dlhú dobu je ich základným prejavom, nie niečo, čo sa učia. Rovnako ako klasické RNN majú aj LSTM zretazenú štruktúru, len s tým rozdielom, že pri druhých menovaných je o niečo zložitejšia. Namiesto jednej vrstvy neurónovej siete, sa skladajú zo štyroch, ktoré medzi sebou spolupracujú. [8]

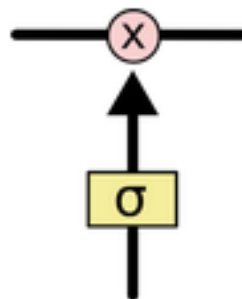


Obr. 1.4: Model LSTM [2]

Kľúčovou časťou LSTM je stav bunky. Tento stav prúdi celou bunkou len s minimom lineárnych interakcií a kludne je možné, že prejde celým reťazcom bez akýchkoľvek zmien. No tento stav sa dá v priebehu cesty meniť, či už pridávať, alebo odoberať informácie, pomocou štruktúr, ktoré nazývame brány. Tieto brány sú založené na *sigmoid* funkcií a násobení. *Sigmoid* funkcia má rozsah medzi 0 a 1. Pri



Obr. 1.5: Priechod stavu bunky [2]



Obr. 1.6: Brána v LSTM [2]

čom 0 znamená, že neprepustí nič a naopak 1 prepustí všetko. LSTM obsahuje tri takéto brány, kvôli ochrane a ovládaniu bunky.

1.2.2 Rozbor LSTM

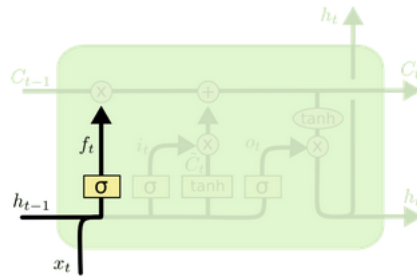
Skôr ako sa dostaneme k samotnému rozboru LSTM, treba podotknúť, že nie všetky LSTM siete, na ktoré môže človek naraziť, sú rovnaké. Takmer v každom projekte, ktorý pracuje s LSTM, uvádza dokumentácia iný model LSTM. Väčšinou sa jedná len o drobné modifikácie modelu, ktorý si ukážeme nižšie. Spomínaný model je veľmi jednoduchý, no pre demonštráciu funkcie LSTM postačujúci.

Prvým krokom v našej LSTM je rozhodnutie, aké informácie chceme ponechať a akých sa chceme pre zmenu zbaviť. Na to slúži už spomínaná brána so *sigmoid* funkciou. Brána na základe vstupov h_{t-1} a x_t pošle na výstup hodnotu medzi 0 a 1 pre každé C_{t-1} . [2]

Druhý krok je trochu členitejší. Môžeme ho rozdeliť na tri časti. V prvej, *sigmoid* vrstva určí hodnoty, ktoré budú v stave bunky upravené. V druhej časti sa funkcia *hyperbolický tangens* stará o vytvorenie vektoru hodnôt \tilde{C}_t , ktorými sa potom môže upraviť stav bunky. Poslednou časťou je už len spojenie predchádzajúcich častí. [2]

Už sme si povedali ako sa postupuje pri úpravách stavu bunky, preto je treba dané kroky už len zrealizovať. Takže ak chceme upraviť stav bunky C_{t-1} na C_t , musíme vynásobiť starý stav bunky hodnotou f_t , čím sa zabudnú vopred určené informácie. Potom k stavu bunky pripočítame $i_t * \tilde{C}_t$. Toto sú nové hodnoty na základe toho, ako sme sa rozhodli jednotlivé vstupy upraviť. [2]

Výstup samozrejme závisí na upravenej hodnote stavu bunky. No pred tým ako sa dostaneme k nejakému výsledku, musí stav bunky podstúpiť ešte pár úprav. Pri tých

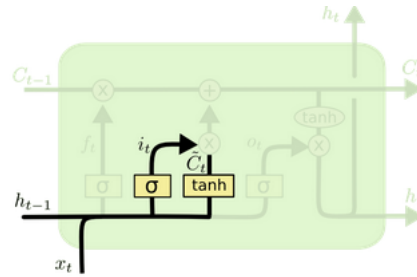


Obr. 1.7: Prvý krok, kde sa rozhoduje, ktoré informácie sa ponechajú a ktoré sa naopak zahodia. [2]

$$f_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_f) \quad (1.1)$$

sigmoid funkcia rozhodne, čo prejde na výstup. Ďalej stav bunky prejde funkciou *hyperbolický tangens*, ktorý sa stará o úpravu výstupov do rozsahu **-1** a **1**. Výstupy týchto dvoch funkcií treba vynásobiť medzi sebou, aby sa na výstup dostali naozaj len tie hodnoty, ktoré sme vybrali a v našom rozmedzí. [2]

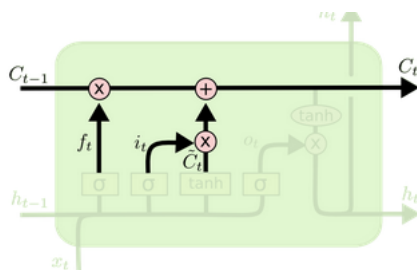
Ako už bolo raz spomenuté, RNN preukazujú vynikajúce výsledky. Faktom je, že za to vďaka práve LSTM, ktoré fungujú naozaj perfektne pri riešení niektorých úloh.



Obr. 1.8: Vytvorenie aktualizácie pre stav bunky. *Sigmoid* funkcia pre vybratie, ktoré informácia a ako veľmi treba upraviť a hyperbolický tangens pre vytvorenie vhodného vektoru pre úpravy [2]

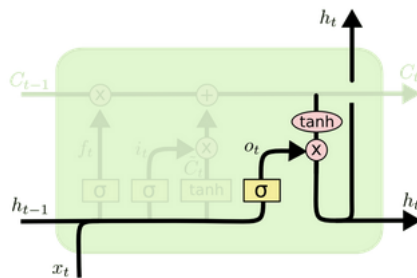
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (1.2)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (1.3)$$



Obr. 1.9: Spojenie jednotlivých krokov pre úpravu stavu bunky

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (1.4)$$



Obr. 1.10: Upravenie výstupu podľa našich potrieb. Tak aby sa na výstup dostalo len to, čo naozaj chceme a zároveň vďaka \tanh funkcii v rozmedzí -1 a 1 [2]

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (1.5)$$

$$h_t = o_t * \tanh(C_t) \quad (1.6)$$

1.3 Hlboké učenie v praxi

V nasledujúcich podkapitolách sa nachádzajú stručné analýzy niektorých projektov, ktoré taktiež využívajú hlboké učenie a ešte konkrétnejšie LSTM. Jedná sa o využitie LSTM pri predpovedi počasia [15], pri predikcii elektrickej záťaže [9] a v medicíne [11]. Konkrétnejšie je venovaná pozornosť predspracovaniu dát, modelu siete a na záver každej podkapitoly sa nachádza stručné zhrnutie článku.

1.3.1 Predpoveď počasia

V článku [15] je popísaný proces krátkodobej predpovede intenzity zrážok v istej oblasti. Systém sa využíva pre včasné výstrahy týkajúce sa zrážok, napríklad pre letiská, ako aj pre implementáciu v dlhodobej predpovedi počasia. [15]. Pre vytvorenie modelu použili konvolučnú LSTM, pri tvorbe ktorej vychádzali z Plne prepojená LSTM – Fully Connected LSTM (FC-LSTM).

Pri pokusoch použiť klasické strojové učenie, sa stretli s niekoľkými problémami. [15] No tieto problémy im pomohol obísť práve model založený na LSTM. V tomto prípade bolo hlboké učenie skvelým postupom, nakoľko získať množstvo dát o počasi z konkrétnej oblasti je jednoduché, ako aj zabezpečiť prísun stále nových dát. Ako bolo spomenuté v predchádzajúcich kapitolách, veľké množstvo tréningových dát je potrebné, pre správnu funkciu modelu, ako aj jeho učenie a vytvorenie.

Predspracovanie dát

V článku [15] sú ako dáta pre tvorbu modelu použité satelitné snímky z mesta Hong-Kong v rozmedzí rokov 2011-2013. No tieto dáta boli pretriedené, kvôli potrebe intenzity zrážok na 97 naupršanějších dní. Najskôr boli hodnoty intenzity upravené do odstieňov šedej a následne sa jednotlivé snímky sústredili na centrálnu časť v rozlíšení 330×330 pixelov. Tieto subsnímky, boli po následovnej aplikácii *disk filtru* pomocou *Matlabu*, zmenšené na rozmery 100×100 pixelov. Ďalej boli aplikované rôzne filtre pre potlačenie šumu.

Snímky boli zhotovované každých 6 minút, čím na jeden deň pripadá 240 snímok. [15] Aby sa predišlo nevhodnému výberu tréningovej, testovacej a overovacej množiny snímok, boli každodenné sekvencie rozdelené do 40, neprekrývajúcich sa blokov [15], z ktorých boli náhodne vybrané práve 4, ktoré sa použili pre učenie. Tieto bloky boli rozdelené pomocou kľzavého okna o veľkosti 20 snímok. Takýmto spôsobom vzniklo 8148 tréningových sekvencií a po 2037 testovacích a overovacích sekvencií, pri čom každá sa skladá z 20 snímok. [15] Z týchto 20 snímok je použitých 5 na vstup a 15 pre predikciu.

Model siete

V práci sa spomína a bolo odskúšaných viacero modelov. My si spomenieme len ten najúspešnejší. Jednalo sa práve o nimi vytvorený model konvolučnej LSTM. Sieť mala 2 vrstvy, pri čom každá obsahovala 64 uzlov a jadrá o veľkosti 3×3 . Veľkosť políčka bol nastavený na 2. [15]

Zhrnutie

Tvorcom práce sa podarilo vytvoriť model pre predpoveď zrážok, založený na hlbokom učení, ktorý podával lepšie výsledky, ako do vtedy používané modely, ktoré nepoužívali túto technológiu. Nimi vytvorený model konvolučnej LSTM vrstvy, nie len že zachováva výhody FC-LSTM, ale je aj vhodný pre spracovanie dát, závislých na čase a priestore, pre jeho konvolučnú štruktúru. [15] Ďalej by sa chceli tvorcovia modelu venovať jeho úprave a aplikácii na video sekvencie.

1.3.2 Predikcia elektrickej záťaže

V článku [9] je problém predikcie elektrickej záťaže rozdelený do 3 skupín:

1. Krátkodobá predikcia - v rozmedzí pár hodín, až týždňov
2. Stredne dlhá predikcia - v rozmedzí pár týždňov, až mesiacov
3. Dlhodobá predikcia - v rozmedzí pár mesiacov, až celý rok

Takáto predikcia je možným pomocníkom energetických spoločností, pri plánovaní dodávok, alebo cien a mnohé iné. Taktiež sa v [9] zaoberajú porovnaním tradičných metód predikcie (klzavé priemery, regresívne stromy, vektorová regresia) a rôznych architektúr umelej inteligencie (zhluky autokodérov, CNN, RNN a LSTM).

Predspracovanie dát

Dáta použité v [9] sú merania z inteligentných elektromerov umiestených v domácnostiach. Merania trvali po dobu jedného roka, pri čom domácnosti sa nachádzajú v tropickom prostredí. Kvôli nerovnomernosti využívania energie počas týždňa a víkendov, boli tieto dáta neskôr rozdelené na dva rôzne zdroje.

Použité dáta pozostávajú z hodinových meraní po dobu jedného roka, pri čom obsahujú až 18 prvkov. Pomer rozdelenia dát pre učenie, testovanie a overovanie bol menovite: 65%, 20%, 15% [9]. Databáza obsahovala napríklad: merania z predchádzajúcej hodiny, 2 hodiny dozadu, predchádzajúci deň v tú istú hodinu, priemer za posledný týždeň a podobne. Sledované hodnoty boli vyberané tak, aby spĺňali normu krátkodobej predikcie elektrickej záťaže.

Predbežné spracovanie dát pozostávalo z prečistenia a prispôsobenie nulovej priemernej a jednotkovej odchylky.

Použité modely

Pri práci bolo využitých viacero modelov neurónových sietí. Všetky sú spomenuté v článku [9]. Pre nás najzaujímavejšie sú RNN, LSTM, CNN a konvolučná LSTM.

Zhrnutie

Z obsahu [9] vyplíva, že niektoré metódy predikcie elektrickej záťaže, založené na hlbokom učení, dosahujú lepšie výsledky ako klasické metódy. Jedinou nevýhodou je dĺžka učenia, no tieto algoritmy sú schopné podávať lepšie výkony ako klasické metódy, aj pri omedzenom počte krokov.

1.3.3 Diagnostikovanie v zdravotníctve pomocou LSTM

V článku [11] sa pojednáva o využití ANN pri diagnostikovaní v medicíne. Konkrétne sa jednalo o využitie na jednotke intenzívnej starostlivosti, kde sa pri prijatí pacienta, vykonáva množstvo vyšetrení a meraní. Tieto merania sú zdrojom veľkého množstva dát, ktoré sú potrebné pre aplikáciu hlbokého učenia. Množstvo dát, chybné merania a chýbajúce hodnoty tvoria ideálny stav pre aplikáciu LSTM. Tie sú vhodné aj pre dlhodobú závislosť medzi dátami. Tak isto je v článku porovnaná metóda s LSTM s metódou viacvrstvého perceptronu.

Predspracovanie dát

Dáta použité v článku pochádzajú z detskej nemocnice v Los Angeles. Jedná sa o viac ako 10-tisíc prípadov. V každom z prípadov sa sledovali napríklad: krvný tlak, hodnota CO₂, cukor, telesná teplota. Celkovo sa jednalo o 13 sledovaných hodnôt. [11]

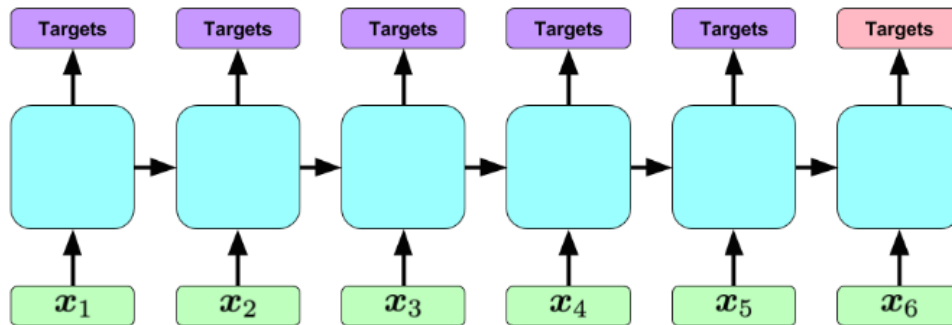
Každý záznam pozostáva z nepravidľne meraných dát. Záznamom chývajú údaje, alebo celé premenné. Pri chýbajúcich údajoch sa postupovalo dopredným, alebo spätným doplnením hodnôt. V prípade chýbajúcej celej premennej, sa doplnila bežná hodnota, podľa expertov. [11] Ak niektorá premenná chýba, zvyčajne je to pre to, že zdravotníci veria, že sa pohybuje v bežných hodnotách. Všetky merania sú prevzorkované na hodiné intervaly. Takéto spracovanie dát je vhodné pre klasifikáciu, no nie však pre predikciu, vzhľadom na spätné dopĺňovanie dát.

Každý prípad je označený žiadnou, alebo viacerými značkami, na základe pravidiel nemocnice. Celkovo má nemocnica pre choroby vyčlenených 429 kódov, no keďže sú medzi nimi aj vzácne choroby, program používa len 128 kódov. Každý z kódov sa

vyskytuje v dátach minimálne 50-krát. [11]

Pri učení bolo využitých 80% dát, 10% pri testovaní a zvyšné dáta sa použili pri overovaní funkčnosti. [11]

Model siete



Obr. 1.11: Model siete [11]

V článku sú spomenuté dva modely. Prvý, jednoduchší, je model LSTM, kedy musia informácie prejsť celým modelom, aby ovplyvnili výstup. Druhý model, sa oproti prvému testovanému líši v tom, že v každom kroku je replikovaný výstup, čím sa zaistí, že aj najmenšia odchylka má dopad na konečný výsledok. Táto replikácia sa využíva aj pre získanie výstupu na konci každej sekvencie. [11] Model sa sústreďuje na 128 kódov z celkového počtu 429. Zvyšné sa pri učení používajú na zaistenie, že sa sieť nepretrénuje. [11]

Zhrnutie

Model je schopný diagnostikovať choroby na základe nameraných údajov. No do budúcnosti sa od modelu očakáva predpoveď postupu choroby, alebo reakcia na liečbu, ako aj pravdepodobnosť úmrtia a iné. Ako sa sieti darilo pri diagnostike sa dá nájsť v článku [11].

1.3.4 Predikcia výkonu solarnej elektrárne v Hae-Nam [14]

V článku sa rozoberá problém predikcie výkonu solarnej elektrárne v Hae-Nam. Pretože tvorba elektriny pomocou fotovoltaických článkov je závislá na viacerých faktoroch, ktoré sú prírodného charakteru, je úspešná predikcia dôležitá, kvôli plánovaniu dodávok do elektrickej siete. Momentálne sa využívajú viaceré druhy predikcie. V článku je porovnaná úspešnosť LSTM modelu a Metóda pohyblivého priemeru – Moving average method (MA).

Prespracovanie dát

K prespracovaniu dát je v texte spomenuté len toľko, že vstupné dáta boli rozdelené na 90% tréningových a 10% testovacích dát.

Model siete

Model obsahoval jednu vstupnú vrstvu, dve skryté vrstvy LSTM a jednu výstupnú vrstvu. Na výstupe bola použitá funkcia lineárnej regresie. Počet epochov (hodnota, koľkokrát sa pri tréningu prejde celým datasetom) bola nastavená na 500.

Zhrnutie

Tvorcom modelu sa podarilo dosiahnuť relatívnu chybu 0.199. Dokázali, že pre krátkodobú predikciu je metóda s použitím LSTM presnejšia ako metóda MA.

1.3.5 Hlboké učenie pre predikciu solárnej energie [5]

Článok pojednáva o dôležitosti predikcie výroby energie pomocou obnoviteľných zdrojov, kvôli ich dopadu na celkovú energiu v sieti. Je to najmä preto, že ich nedokážeme nijakým spôsobom ovplyvniť. Tento fakt môže spôsobiť prebytok energie v sieti, ako aj nedostatok. Oba prípady by mohli mať katastrofálne následky. V článku su rozoberané rôzne spôsoby predikcie ako aj ich využitie pre rôzne časové horizonty.

Predspracovanie dát

Dáta z článku pochádzajú z 21 fotovoltaických elektrární v Nemecku. Pre každú elektráreň obsahuje vyprodukovanú energiu v rozmedzí 3 hodín po dobu 990 dní. Tieto dáta boli škálované v rozmedzí hodnôt 0 a 1. Primárne pre to, aby sa zamedzilo prejavaniu rozdielných veľkostí jednotlivých elektrární.

Dataset bol rozdelený na 490 tréningových dní, 250 pre kontrolu a 250 pre samotné testovanie. Predikčný horizont bol 24 až 48 hodín.

Model siete

V článku je spomenuté, že sa jednalo o 4 vrstevú topológiu. To znamená, že sa jednalo o 2 skryté vrstvy LSTM. V texte sa však neuvádza, koľko neurónov mali jednotlivé vrstvy. Ako výstupná funkcia je použitá funkcia *ReLU*.

Zhrnutie

Predikované dáta už nepodstúpili žiadnu úpravu. Priemerne sa Stredná kvadratická chyba – Root Mean Square Error (RMSE) pre sieť LSTM pohybovalo okolo hodnoty *0.0675*. Relatívna chyba sa v texte neuvádza.

Záver práce obsahuje prehľadné zhrnutie všetkých výsledkov a ich porovnanie s inými metódami.

2 VÝSLEDKY STUDENTSKÉ PRÁCE

Cielom práce bolo vytvoriť model, ktorý by bol schopný predikovať výkon fotovoltaických článkov v časovom horizonte 24 hodín. Existuje viacero možností ako k danej problematike pristúpiť. V tejto práci sa primárne využíva hlboké učenie s LSTM vrstvami. Pre porovnanie sa použila aj metóda lineárnej regresie.

Táto kapitola sa skladá z viacerých podkapitol, ktoré sa venujú:

1. **Dáta** - stručný popis dát a ich predspracovanie
2. **Model** - stručný popis prvotného modelu a vysvetlenie niektorých dôležitých pojmov.
3. **Dosiahnuté výsledky** - Popis jednotlivých modifikácií modelu a ovplyvnenie presnosti modelu týmito úpravami.
4. **Výsledný model** - Popis nastavenia výsledného modelu, s najlepšou dosiahnutou presnosťou.

Pre písanie kódu bol použitý *Python3.5* s pridanými knižnicami:

- *numpy*- knižnica vhodná pre prácu s n-rozmernými radmi a pre vedecké výpočty
- *pandas*- knižnica vhodná na spracovanie označených aj neoznačených dát a ich analýzu
- *matplotlib*- knižnica vhodná pre tvorbu 2D grafov
- *tensorflow*- knižnica vhodná pre aplikáciu strojového učenia a umelej inteligencie
- *sickit-learn*- knižnica vhodná pre spracovanie, analýzu dát a strojové učenie
- *keras* - knižnica pre rýchlu a jednoduchú implementáciu neurónových sietí

2.1 Dáta

Databáza, ktorú máme k dispozícii, pozostáva z 35041 záznamov. Každý záznam obsahuje časovú pečiatku, výkon, teplotu a osvit. Merania boli uskutočňované od 1.1.2015 do 31.12.2015. Záznamy boli zhotovované v intervale 15 minút.

2.1.1 Spracovanie dát

Vzhľadom na to, že poskytnuté dáta sú vo formáte *.xlsx*, ktorý je pri spracovávaní pomalý, je treba dáta najskôr preformátovať do niektorého z rýchlejších dátových formátov. V našom prípade, použijeme formát *.csv*, ktorý sa pri testovaní osvedčil ako vhodnejší. Preto po načítaní originálnych dát, prekonvertujeme dáta do formátu

.csv a uložíme ich, aby sme pri prípadných zásahoch do dát, neprišli o originál. Následne načítame tieto prekonvertované dáta.

Pri načítaní sú $3/4$ hodnôt nahradených hodnotou *NaN*, čo v preklade znamená, že sa nejedná o číslo. Zväčša sa jedná o hodnoty, ktoré v originálnom datasete chýbajú. To nemusí nevyhnutne znamenať chybu pri meraní. Ale ako už bolo spomenuté skôr a tak isto v článku [11] ide o hodnotu, ktorá sa nemenila. Tieto chýbajúce hodnoty treba nejakým spôsobom upraviť. K dispozícii je viacero možností:

1. Riadky s chýbajúcimi hodnotami zmazať. No tým by sa zapríčinilo zredukovanie dát, a vznikli by výpadky medzi meraniami. To je niečo čo by malo neagatívny dopad.
2. Chýbajúce dáta doplniť poslednou nameranou hodnotou. Toto je postup vhodný skôr pre klasifikáciu. [15]
3. Chýbajúce dáta doplniť najbližšou nasledujúcou hodnotou. Tento postup sa využíva práve pri tvorbe datasetu pre predikciu. [15]

Tento model je zameraný na predikciu, preto sa dáta nahradili najbližšou ďalšou hodnotou. Fakt, že je tolko hodnôt potrebné nahradiť sa výrazne prejaví na presnosti výsledného modelu. Pred samotným učením ešte dáta prechádzajú niekoľkými úpravami. Treba dáta upraviť pre učenie pod dohľadom. Táto úprava spočíva vo formovaní párov vstup a očakávaný výstup, aby sa sieť naučila, ako predikovať výstup na základe vstupných dát. Ako vstup sa v našom prípade využíva osvit, teplota a záťaž v predchádzajúcom období, podľa veľkosti predikčného horizontu. Ešte skôr ako sa dáta posunú do funkcie, ktorá ich upraví pre učenie, sú dáta normalizované aby sa pohybovali v rozsahu 0 až 1. Normalizáciou dát sa zamedzuje prílišnému vplyvu niektorých veličín na konečný výsledok. O normalizácii sa dá v podstate povedať, že zabezpečuje všetkým veličinám rovnakú dôležitosť a vplyv na váhy. Na normalizáciu je využitá funkcia *MinMaxScaler* knižnice *sckit-learn*. Pre ďalšiu úpravu datasetu sa dá použiť funkcia knižnice *pandas* - *shift()*. Táto funkcia vytvorí kópiu stĺpca a pridá podľa výberu, buď na začiatok, alebo na koniec hodnoty *NaN*.

Funkcia programu *series_to_supervised* prijíma na vstupe 4 argumenty:

- *data* - Dáta pre spracovanie vo formáte 2D *numpy array*
- *n_in* - Počet vstupov. Predvolená hodnota je 1
- *n_out* - Počet výstupov. Predvolená hodnota je 1.
- *dropnan* - *Boolean* hodnota, či sa majú riadky s hodnotou *NaN* zahodiť, alebo ponechať. Predvolená hodnota je *True*, takže takéto riadky sa zahodia.

Následne sa dáta upraví do podoby, aby hodnota, ktorú chceme predikovať bola

na konci, pre lepšiu manipuláciu. Pred procesom učenia, sa tento posledný stĺpec odstráni a všetky ostatné hodnoty sa použijú na predikciu. V prípade, že niektoré parametre pri predikcii použiť nechceme, môžeme ich jednoducho odstrániť. Na obrázku môžeme vidieť 5 premenných, pri čom každá je zastúpená hodnotou v čase $t-1$ a hodnotou t . Jedná sa o predchádzajúce hodnoty a o hodnoty v aktuálnom čase, respektíve v čase, pre ktorý chceme predikciu uskutočniť. V poslednom stĺpci sú hodnoty $var1(t)$, čo predstavuje reálne hodnoty výkonu, ktoré chceme pri predikcii dosiahnuť, preto sa pred učením zo vstupných dát tento stĺpec odstráni.

	var1(t-1)	var2(t-1)	var3(t-1)	var4(t-1)	var5(t-1)	var2(t)	var3(t)	\
96	0.006814	0.237175	0.0	0.076031	0.980179	0.27024	0.0	
97	0.006814	0.237162	0.0	0.094550	0.970402	0.26824	0.0	
98	0.006814	0.237162	0.0	0.111756	0.958846	0.26824	0.0	
99	0.006814	0.237162	0.0	0.127655	0.945771	0.26824	0.0	
100	0.006814	0.237162	0.0	0.142318	0.931420	0.26824	0.0	
	var4(t)	var5(t)	var1(t)					
96	0.074518	0.979451	0.006814					
97	0.093059	0.969828	0.006814					
98	0.110305	0.958412	0.006814					
99	0.126256	0.945462	0.006814					
100	0.140976	0.931219	0.006814					

Obr. 2.1: Ukážka dát upravených na učenie pod dohľadom. S desiatimi vstupnými premennými. Pri čom var1 až var5 reprezentujú výkon, teplotu, osvit, azimut a zenit, menovite.

Výstupom funkcie sú dáta pripravené na učenie s učiteľom, vo formáte *Pandas-DataFrame*. Následne sú dáta rozdelené na tréningovú a testovaciu množinu. Tréningová množina má veľkosť 250 dní, zvyšok dní je využitých na testovanie. Dáta sa neprehadzujú, čo znamená, že vstupy reprezentujúce prvých 250 dní sú určené na tréningovanie, zvyšok na učenie. V skutočnosti to však predstavuje 24000 hodnôt pre učenie a 10848 hodnôt pre testovanie.

Nakoniec musia byť vstupy tréningovej aj testovacej množiny pretvorené do 3D. Aby spĺňali podmienky pre použitie LSTM, musia byť vo formáte [vzorky, časové intervaly, funkcia].

Po naučení modelu a následnom vytvorení predikcie prichádza ešte k dodatočným úpravám dát, skôr, ako sa dostaneme k samotnému vyhodnoteniu. Pred učením, boli hodnoty normalizované, aby sa pohybovali v rozsahu hodnôt 0 až 1. Pôvodne boli inverzné hodnoty škaloované, no kvôli problémom, ktoré vznikli v neskorších modeloch sa z tohto kroku upustilo. Ďalej sa ešte predikcia upravila tak, aby neobsahovala záporné hodnoty. Na to sa využila funkcia *clip* knižnice *numpy*. Parametre predané funkcii sú *numpy.array* a následne treba špecifikovať rozsah, do akého má predané hodnoty orezať. V našom prípade je *a_min=0* a *a_max=None*. Týmito parametrami sa zaistí, že všetky záporné hodnoty sa prepíšu na 0 a hodnotou *None* sa zaistí, že nejestvuje žiadne obmedzenie maxima.

2.2 Model

Model obsahuje 50 neurónov v skrytej vrstve a na výstupe jeden neurón, ktorý slúži ako výstup predikcie. Jedná sa o predikciu záťaže na základe časového intervalu a 3 veličín. Ako vstup slúži pole príznakov vo formáte 3D. Pôvodný formát 2D, musel byť pre potreby LSTM prerobený. Pre výpočet odchylky je použitá Stredná absolútna chyba – Mean Absolute Error (MAE) a ako metóda poklesu je použitá funkcia *Adam*.

$$MAE = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{n} \quad (2.1)$$

y_i = očakávaná hodnota

\hat{y}_i = predikovaná hodnota

n = počet vzorkov

Model sa zastaví po 30 opakovaníach. To znamená, že kým sa model zastaví, prejde 30-krát celým datasetom. Parameter *batch_size* je nastavený na hodnotu 97. Tento parameter predstavuje veľkosť časového úseku známych dát, ktoré do siete posielame. V našom prípade to znamená, že sieť vezme hodnoty od 0 do 97 (v ďalšom

cykle hodnoty 98-194) a trénuje sa na nich, až kým neprejde celým datasetom. Po vyčerpaní datasetu spomínaný proces opakuje, až dokým celým datasetom neprejde už spomenutých 30-krát. Keď celý *batch* prejde sieťou, je pri použití knižnice *Keras* vnútorný stav LSTM premazaný.

Počas učenia a predikcie boli sledované dve hodnoty. RMSE a *relatívna chyba*.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n}} \quad (2.2)$$

y_i = očakávaná hodnota

\hat{y}_i = predikovaná hodnota

n = počet vzorkov

$$\delta = \frac{\sum_{i=1}^n |\hat{y}_i - y_i|}{\sum_{i=1}^n y_i} \quad (2.3)$$

\hat{y}_i = predikovaná hodnota

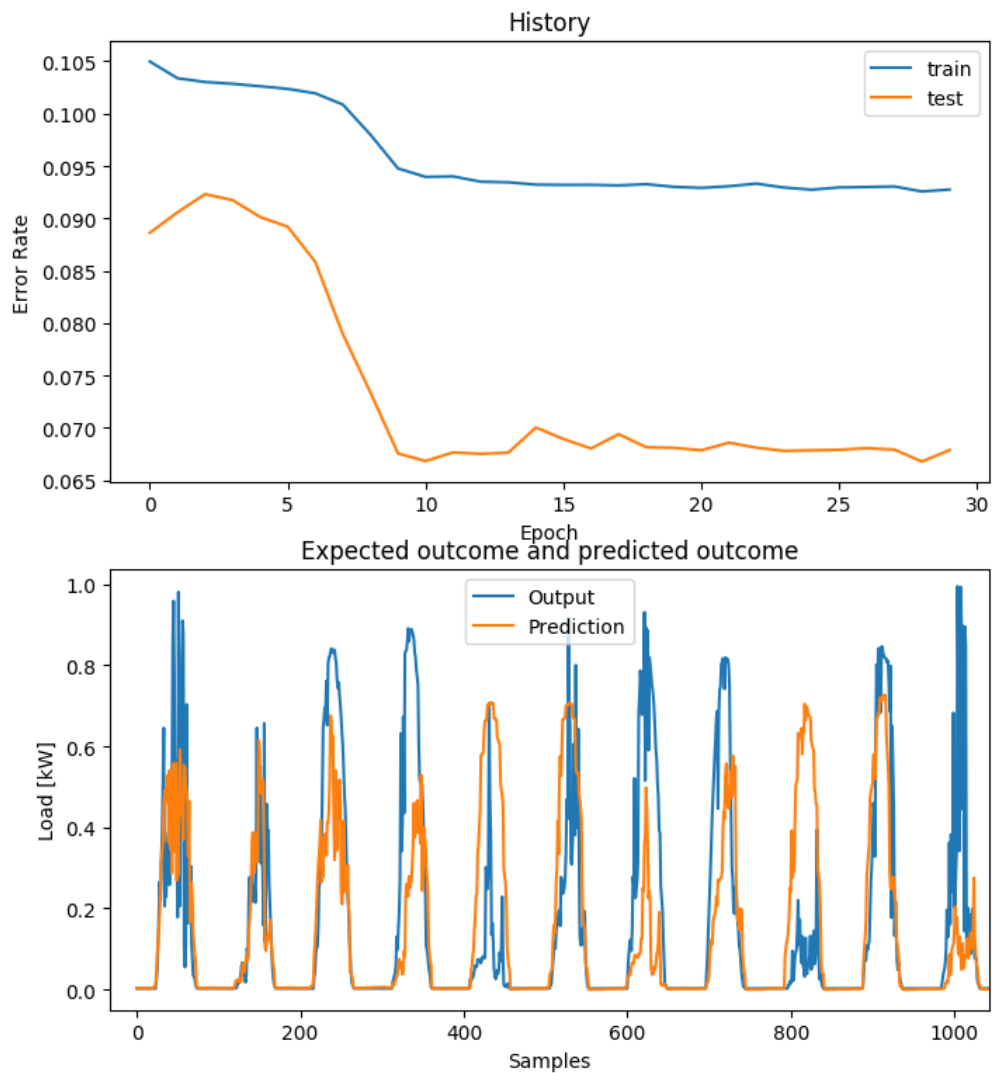
y_i = očakávaná hodnota

2.3 Dosiahnuté výsledky

V tejto stati sú zhrnuté výsledky jednotlivých testovaných modelov.

2.3.1 Prvotný model

Pri tomto modeli bolo použité rovnaké predspracovanie dát, ako je spomenuté v predchádzajúcej kapitole. Neurónová sieť, sa v tomto prípade, pri trénovaní na 24000 vzorkoch a následnom testovaní na 10845 vzorkoch dopracovala na chybu RMSE - *0.150*, čo predstavuje relatívnu chybu *0.787*. Za behu programu boli sledované dva parametre. Prvým bol vývoj chyby pri učení a testovaní a druhým bol samotný rozdiel medzi predikciou a očakávaným výstupom. Program obe závislosti vyniesol do grafu. Pri podrobnejšom preskúmaní grafu a jeho priblížení, môžeme vidieť, že sieť je schopná úspešne predikovať zmeny záťaže, no nedokáže si poradiť s predikcou presnej hodnoty záťaže.



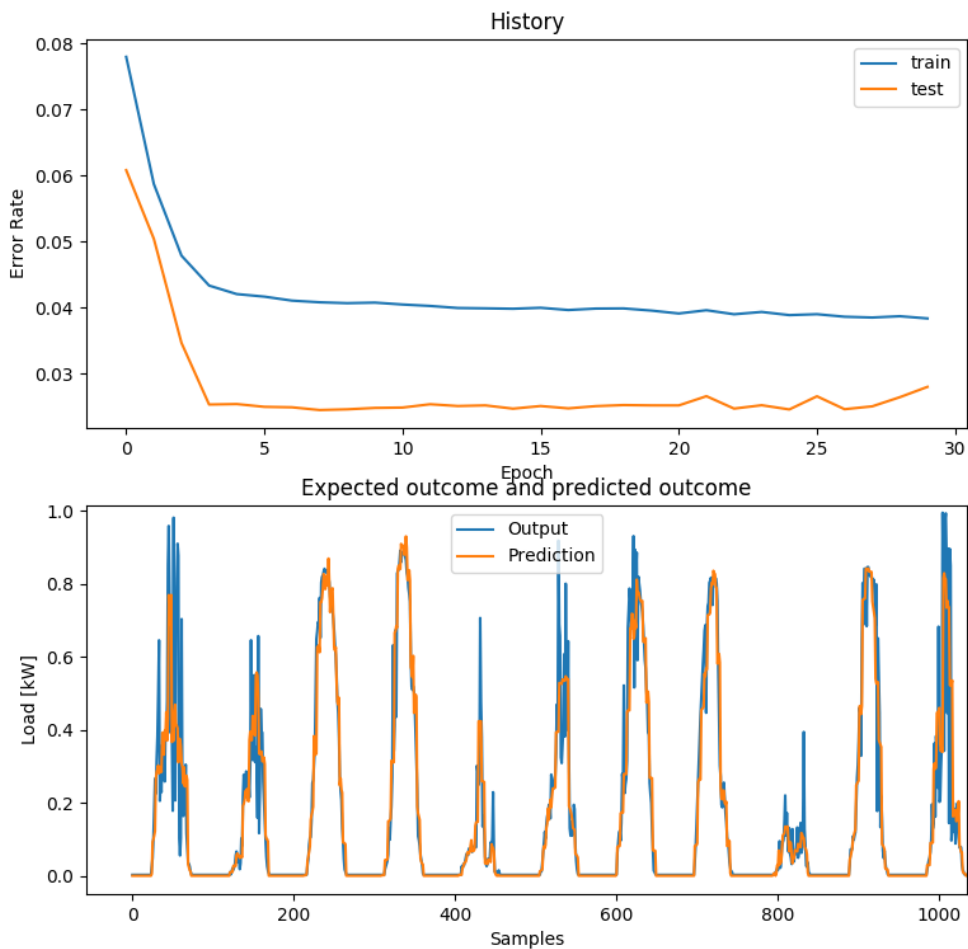
Obr. 2.2: Grafy vykreslené programom pre **prvotný model**. Prvý ukazuje ako sa vyvíjala chyba pri učení a testovaní. Druhý graf zobrazuje porovnanie predikovaného a očakávaného výstupu na rozsahu 1000 vzoriek.

2.3.2 Model s upravenými vstupnými dátami

Pri snahe zlepšiť presnosť modelu, sa pristúpilo k úprave vstupných dát. To spočívalo v rozšírení vstupných dát o sprievodné veličiny (teplota, osvit) v čase t . Následne sa veličina, ktorú chceme predpovedať v čase t posunula na koniec a z datasetu odstránila. Z týchto odstránených veličín bol následne vytvorený dataset očakávaných

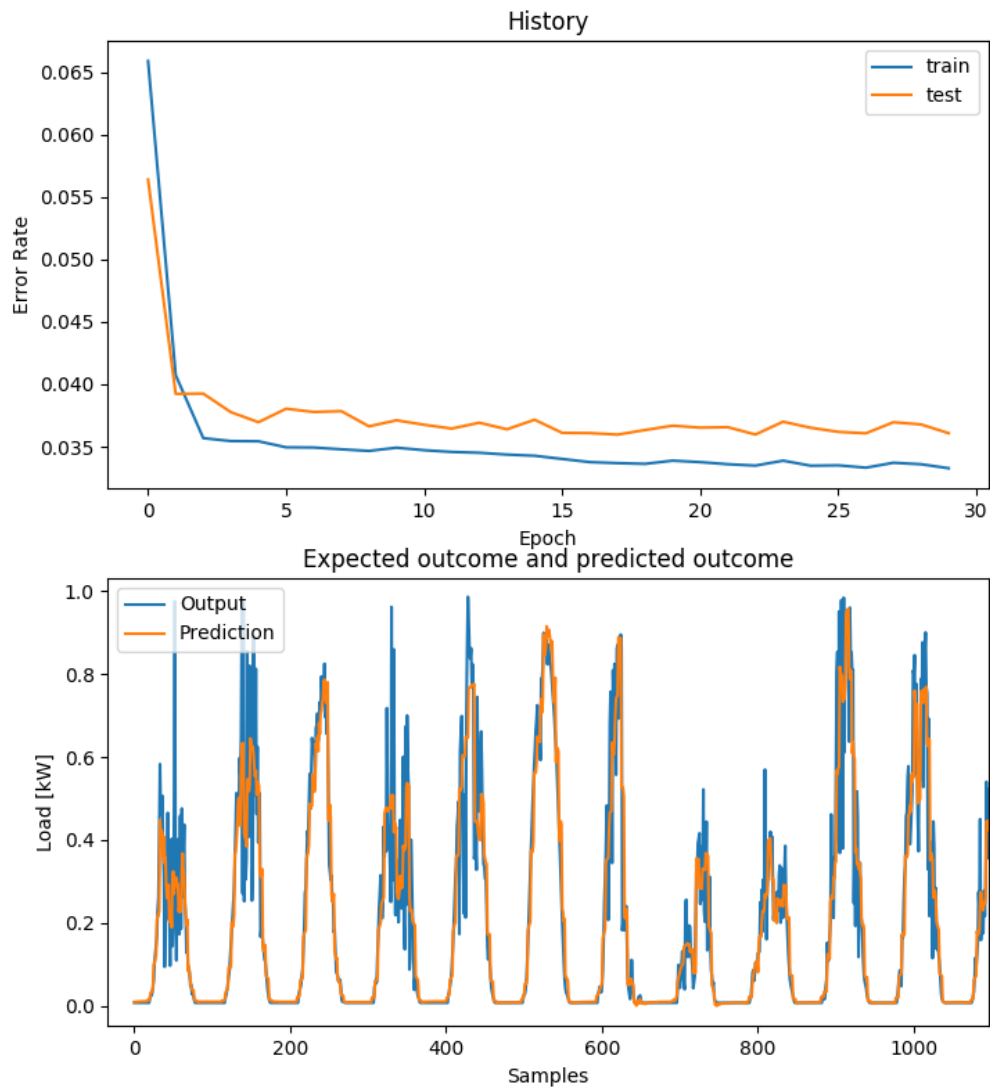
výstupov. Následne sa dataset rozdelil na 24000 tréningových a 10845 testovacích vzorkov.

S takto upraveným datasetom bol model schopný eliminovať podstatnú časť chyby pri učení a následnej predikcii. Bol schopný dopracovať sa k chybe RMSE - 0.067 , čo predstavuje relatívnu chybu 0.294 . Za behu programu boli sledované dva parametre. Prvým bol vývoj chyby pri učení a testovaní a druhým bol samotný rozdiel medzi predikciou a očakávaným výstupom. Program obe závislosti vyniesol do grafu. Pri podrobnejšom preskúmaní grafu môžeme vidieť, že sieť už úspešne nepredikuje len čas, kedy nastane maximálna odchyľka, ale v mnohých prípadoch aj úspešne túto maximálnu odchyľku predikuje.



Obr. 2.3: Grafy vykreslené programom pre **model s upravenými vstupnými dátami**.

Pri obohatení vstupných dát, o záznamy namerané v priebehu ďalšieho roka, sa schopnosti modelu predikovať výkon opäť zlepšila. Dataset bol rozdelený do 48000 tréningových vzorkov a 21981 testovacích vzorkov. RMSE sa zmenila na 0.082 čo predstavuje relatívnu chybu 0.250.



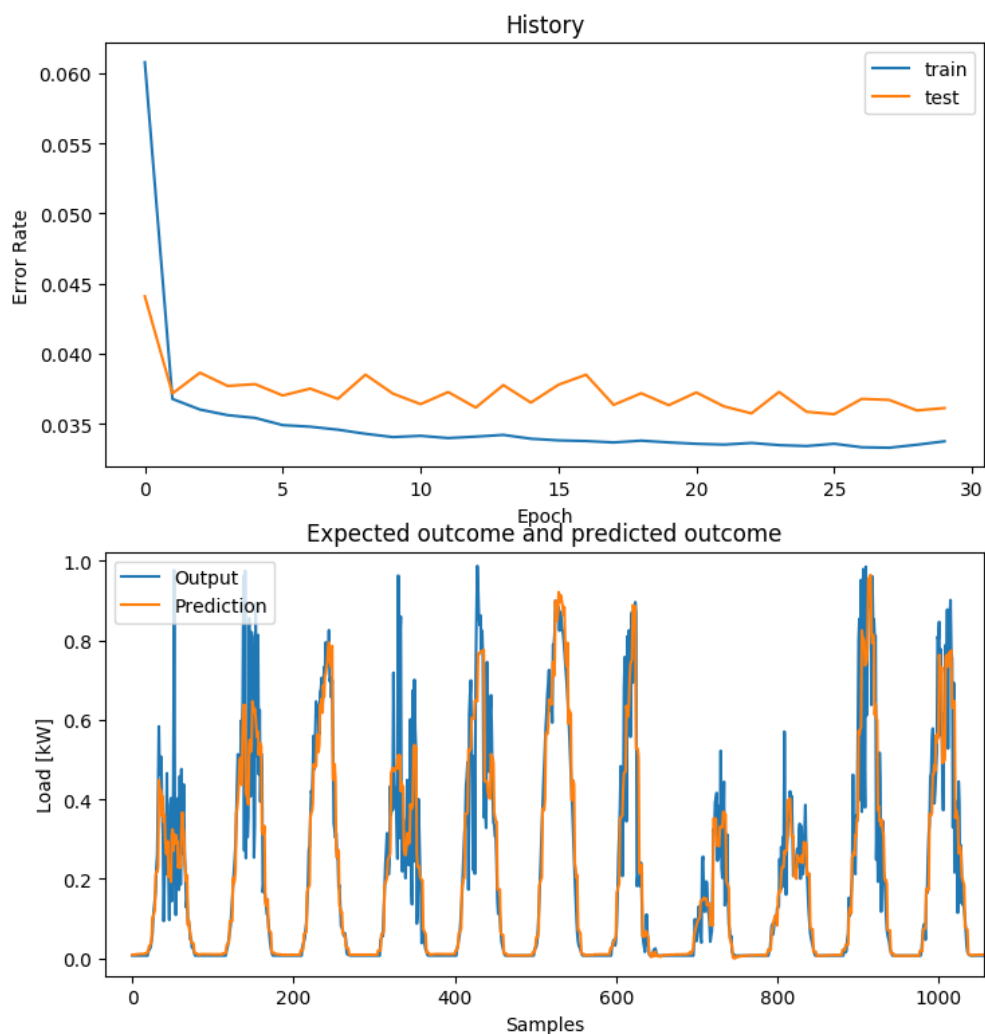
Obr. 2.4: Grafy vykreslené programom pre **model s dátami z dvoch rokov**.

Pri pokuse navýšiť počet epochov (hodnota udávajúca, koľkokrát prejde sieť celým datasetom) z pôvodných 30 na 100 sa presnosť nijako výrazne nezvýšila,

no výrazne narástol čas učenia siete. Tento výsledok bol pozorovaný pri viacerých opakovaníach.

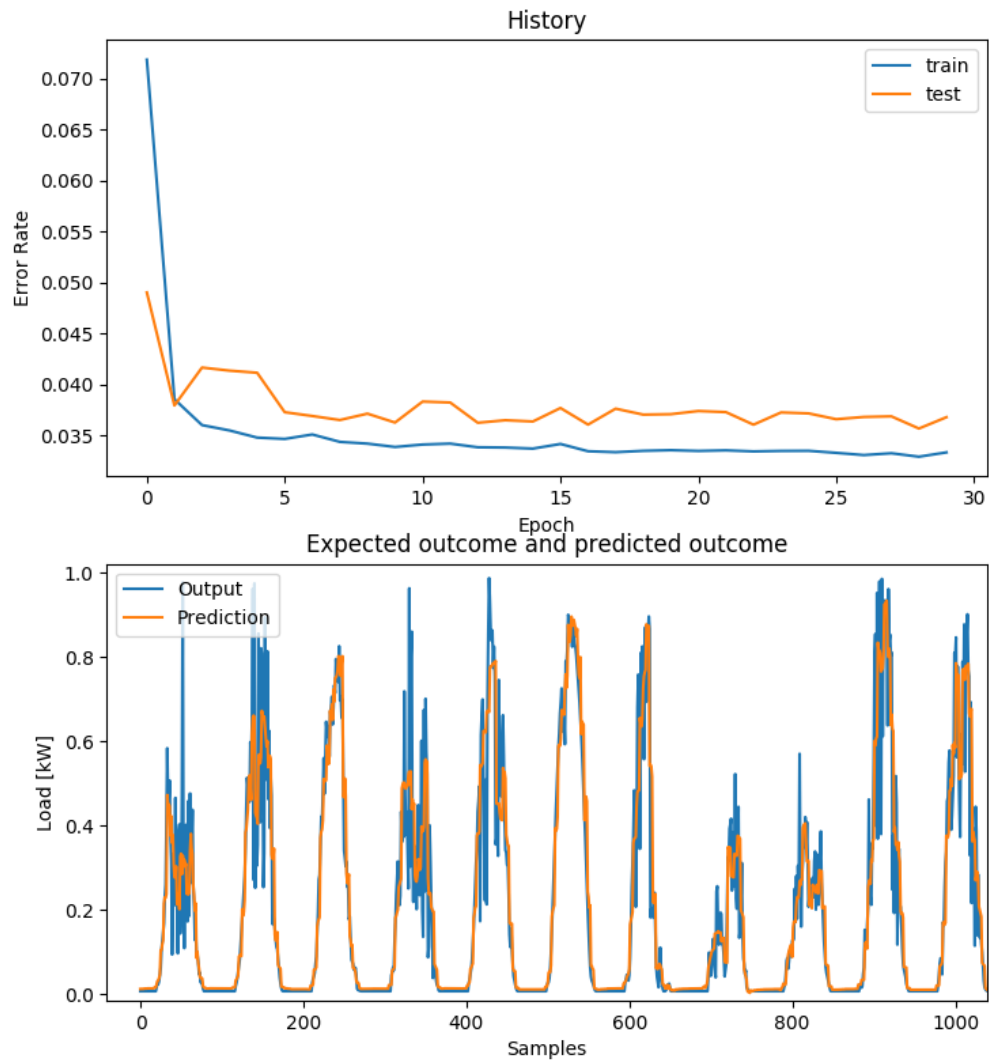
2.3.3 Upravený model

Pri zvýšení počtu neurónov v skrytej vrstve z pôvodného počtu 50 na 100 sa relatívne chyba vôbec nezmenila. Pri tejto konfigurácii bola hodnota RMSE - 0.082 čo zodpovedá relatívnej chybe 0.250.



Obr. 2.5: Grafy vykreslené programom pre **upravený model**.

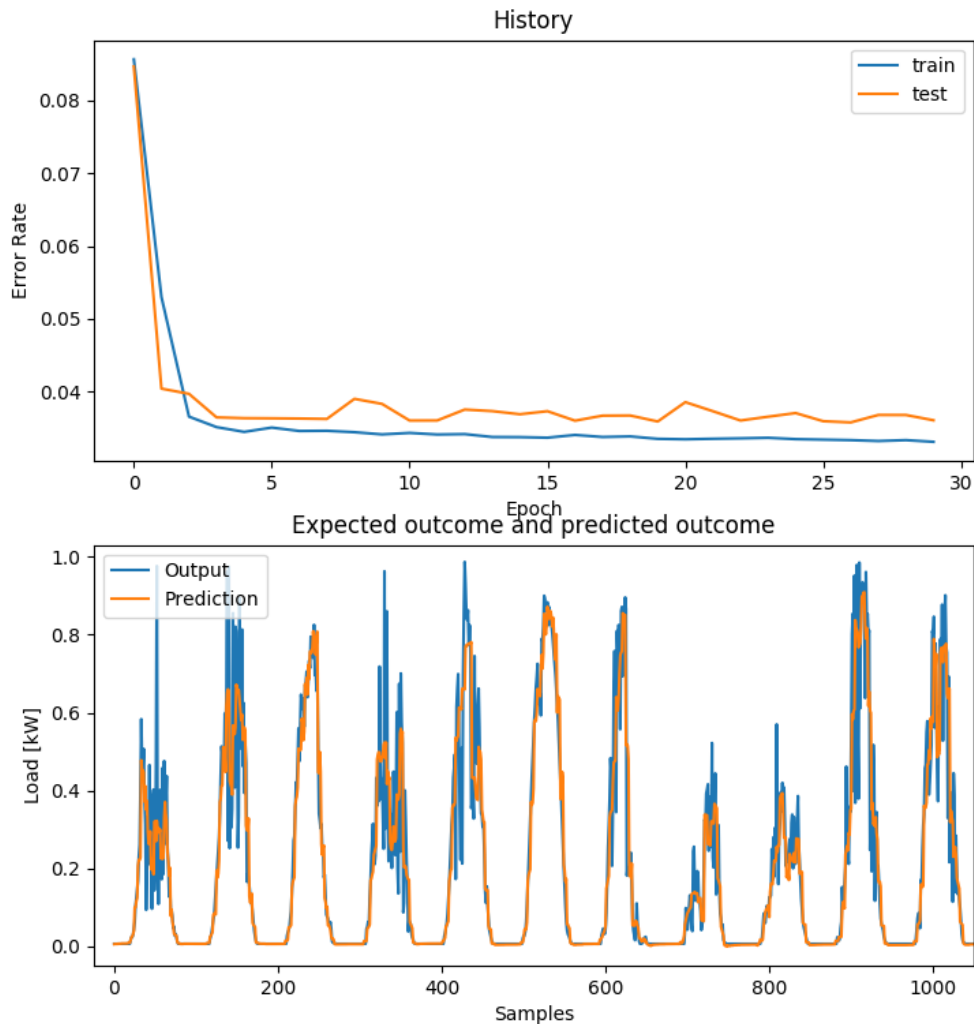
Ďalší zásah do modelu predstavovalo pridanie ďalšej skrytej vrstvy. Jednalo sa o vrstvu LSTM, ktorá obsahovala 50 neurónov. Bola pozorovaná drobná zmena presnosti k horšiemu. Relatívna chyba sa pohybovala v okolí 0.251. Tento zásah sa negatívne odrazil na rýchlosti učenia siete.



Obr. 2.6: Grafy vykreslené programom pre **model s dvoma skrytými vrstvami**.

Pri pridaní tretej skrytej vrstvy typu LSTM, ktorá obsahovala 50 neurónov sa proces učenia ešte viac spomalil. Presnosť predikcie sa zhoršila. Relatívna chyba v tomto prípade bola 0.255. V takomto prípade pripadá na každý príznak (teplota,

výkon, osvit), práve jedna vrstva. Fakt, že sa pri pridaní vrstiev predikcia zhoršila, môže byť zapríčinený nedostatkom tréningových dát.

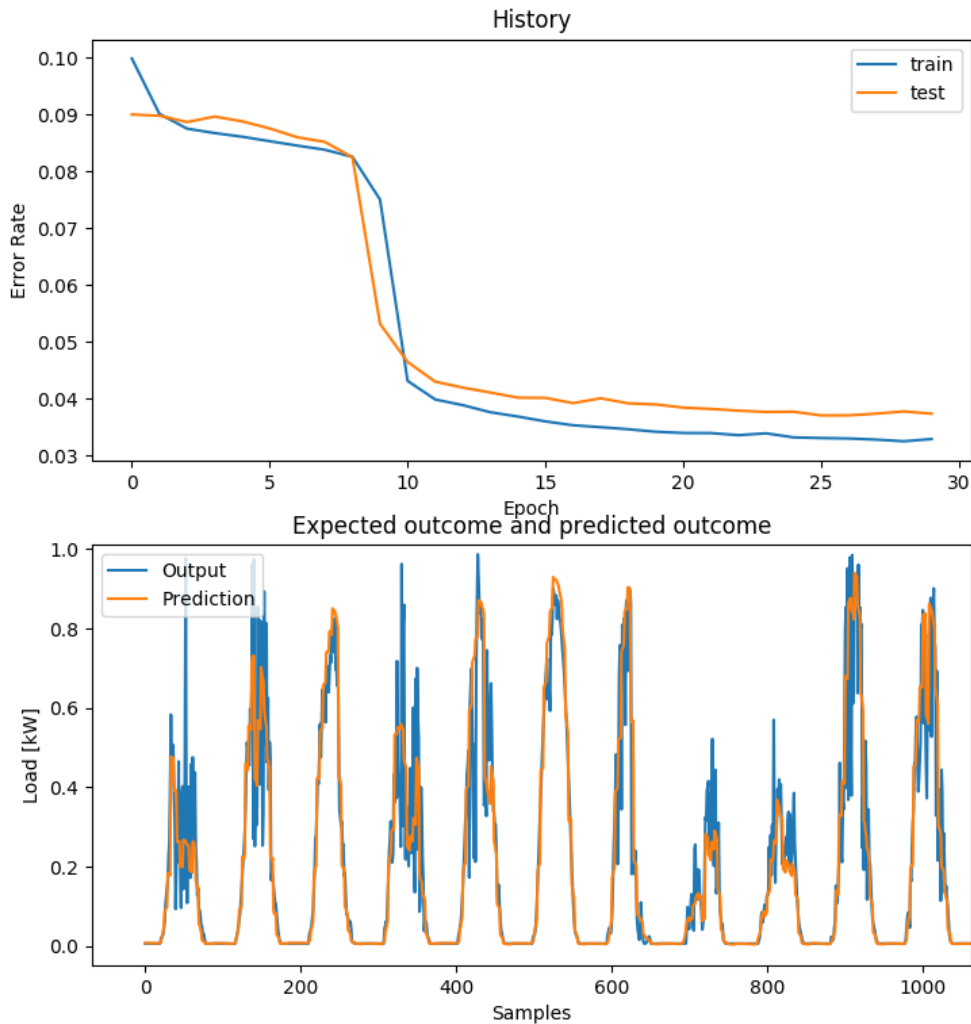


Obr. 2.7: Grafy vykreslené programom pre **model s tromi skrytými vrstvami**

2.3.4 Pridanie polohy slnka a skrytých vrstiev

Pri snahe o zlepšenie predikcie sa pristúpilo k rozšíreniu datasetu o polohu slnka na oblohe. Konkrétne sa jednalo o hodnoty *azimut* a *zenit*. Výpočet polohy slnka sme spravili pre geografickú polohu *Českých Budějovic*. Následne sme model rozšírili o ďalšie skryté vrstvy. Konkrétne boli pridané 2 LSTM vrstvy s 50 neurónami.

Predikcia sa zhoršila na hodnotu RMSE - 0.085 , čo zodpovedá relatívnej odchyľke 0.259 . To by mohlo mať viacero dôvodov.



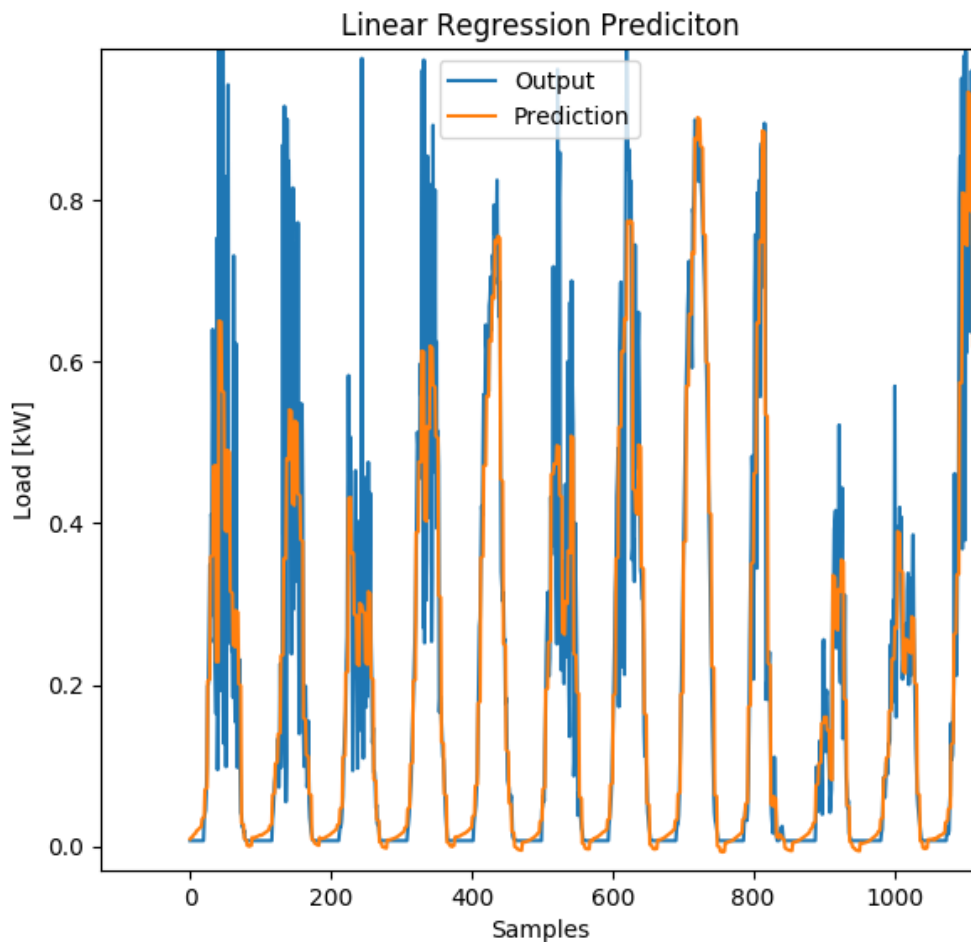
Obr. 2.8: Grafy vykreslené programom pre **model s údajmi o polohe slnka**. Prvý ukazuje ako sa vyvíjala chyba pri učení a testovaní. Druhý graf zobrazuje porovnanie predikovaného a očakávaného výstupu na 1000 vzorkoch.

1. Pridané dáta (poloha slnka) sú s výkonom nesúvisiace a teda do výslednej predikcie zanesli chybu.
2. Pridané dáta súvisia s daným problémom a fakt, že sa predikcia zhoršila môže súvisieť s nedostatkom dát.

Vzhľadom na to, že sa predikcia po pridaní vrstiev zhoršila už pri predchádzajúcom modeli a v tomto modeli sa počet vrstiev ešte zväčšil, bol vydedukovaný záver, že počet meraní je nedostatočný pre takto komplexný problém.

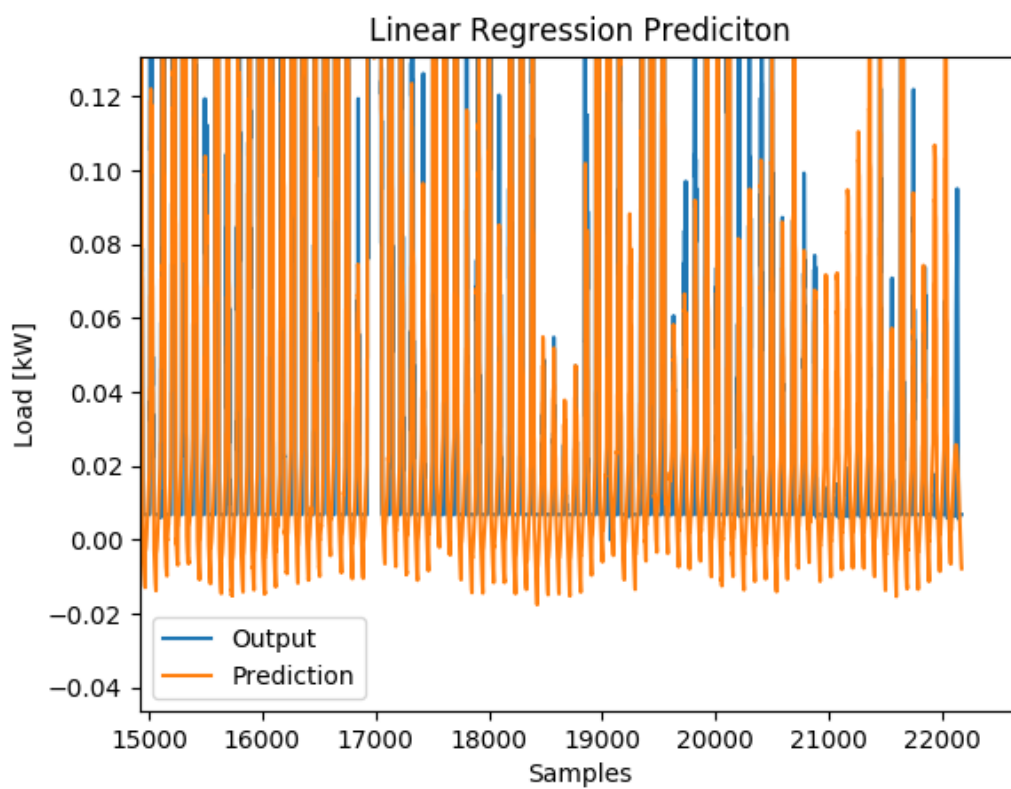
2.3.5 Predpoveď pomocou lineárnej regresie

Lineárna regresia bola použitá na neupravené dáta. Predpoveď takýmto spôsobom mala relatívnu chybu 0.504 . Po škálovaní dát do rozsahu medzi hodnotami 0 a 1 sa relatívna chyba rovnala 0.275 .



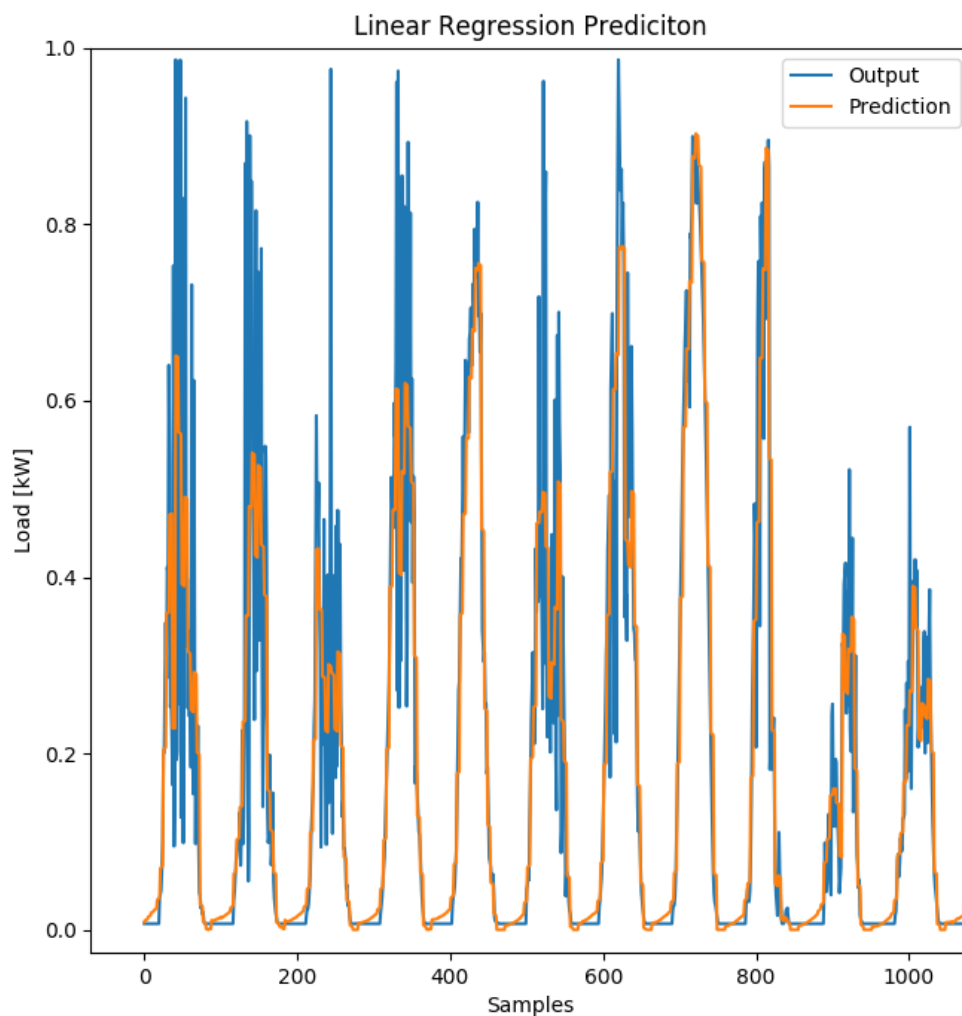
Obr. 2.9: Graf zobrazujúci predikciu pomocou lineárnej regresie pri škálovaných vstupných hodnotách.

Predpovedané hodnoty sa pohybovali v záporných hodnotách. Čo nezodpovedá realite, lebo články energiu nespotrebovávajú.



Obr. 2.10: Graf zobrazujúci predikciu pomocou lineárnej regresie s väčším detailom, pre lepšie pozorovanie záporných hodnôt.

Po orezaní predikovaných dát tak, aby sa pohybovali len v rozsahu 0 až *nekonečno* bola relatívna chyba 0.268 .



Obr. 2.11: Graf zobrazujúci predikciu pomocou lineárnej regresie pri škálovaných vstupných hodnotách. Hodnoty predikcie výkonu boli orezané od 0 po nekonečno.

2.3.6 XGBoost predikcia

XGBoost je metóda, ktorá implementuje gradientom posilnený rozhodovací strom. V podstate sa jedná o súťaživé strojové učenie. Táto metóda však neposkytla žiadne výsledky. Dôvodom bola nedostatočná pamäť RAM.

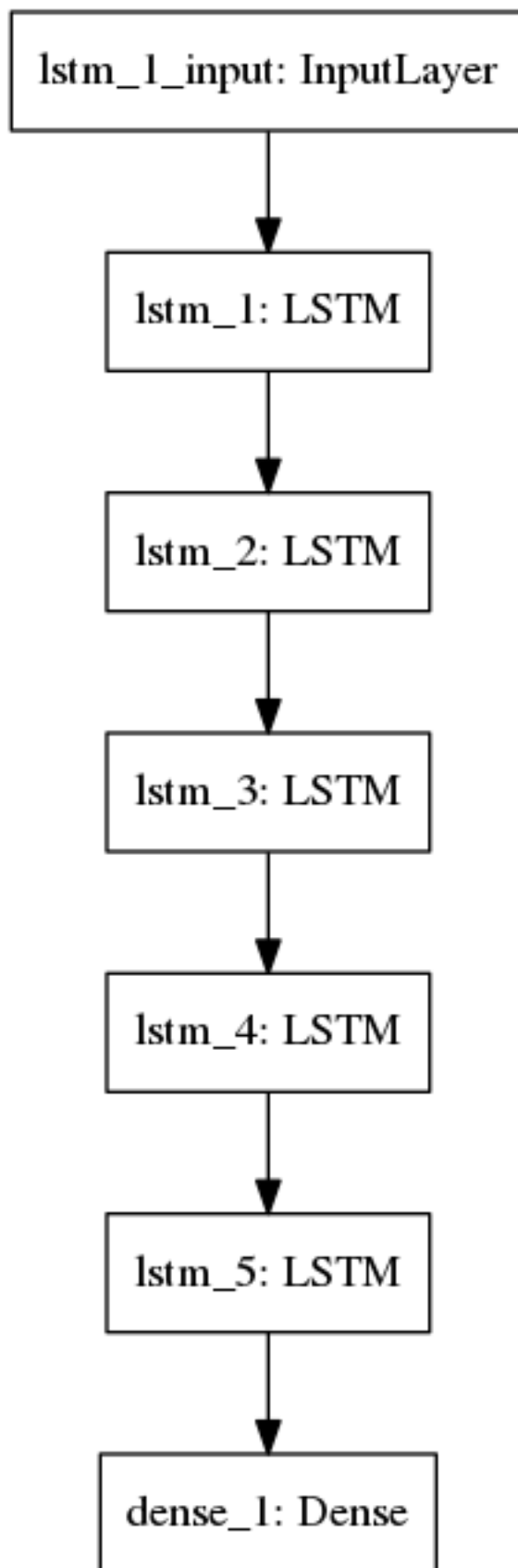
2.4 Výsledný model

Model, ktorý na vstup dostal dáta rozšírené o polohu slnka pre každé časové razítko. Dáta boli merania uskutočňované počas dvoch rokov v 15 minútových intervaloch síce nedosahoval najlepšie výsledky, no predpokladá sa, že na rozsiahlejšom datasete by mal najlepšie výsledky. Trénovacia množina obsahovala 48000 vzorkov a testovacia množina obsahovala 21981. Model sa skladá z 5 skrytých LSTM vrstiev, pričom každá obsahuje 50 neurónov.

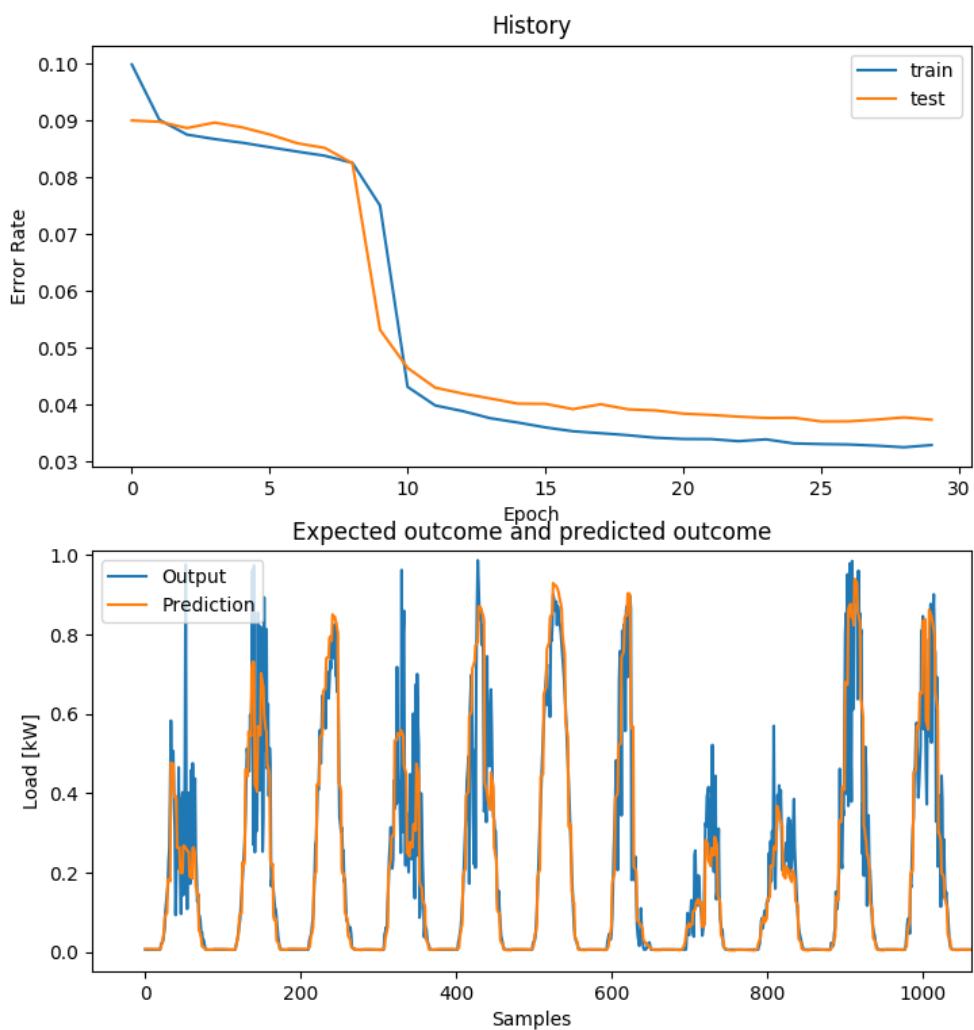
Pre výpočet odchylky je použitá MAE a ako metóda poklesu je použitá funkcia *Adam*. Model sa zastaví po 30 opakovaníach.

Takto navrhnutý model dosahuje predikciu s relatívnou odchylkou 0.259. Pri viacerých testovaniach sa hodnota relatívnej chyby menila len o tisíciný. Program po naučení a testovaní vykreslil priebeh vývoja chyby a porovnanie predikovanej a očakávanej hodnoty do grafu.

Na grafe možno vidieť, že nedokáže úplne presne odhadnúť tie najväčšie maximá. Možné riešenia tejto situácie sú rozšírenie datasetu o veličinu, na ktorej je výkon závislý a zväčšenie počtu vzorkov v datasete. Zväčšenie by sa malo mať najväčší pozitívny dopad na tento výsledný model.



Obr. 2.12: Obrázok zobrazujúci konečnú podobu modelu.



Obr. 2.13: Graf zobrazujúci vývoj chyby a detailnejší pohľad na porovnanie predikovaného a očakávaného výstupu na 1000 vzorkoch .

2.5 Zhrnutie výsledkov

V tejto kapitole sa nachádza tabuľka s prehľadom jednotlivých modelov a ich dosiahnutých výsledkov. Ako aj porovnanie s výsledkami, ktoré boli dosiahnuté inými modelami, ktoré boli spomenuté v teoretickej časti.

Model	RMSE	Relative Error
Prvotný Model	0.150	0.787
Model s upravenými vstup. dátami	0.067	0.294
Model s 2 roč. dátami	0.082	0.250
Upravený model 100 neurónov	0.082	0.250
Upravený model 2 skryté vrstvy	-	0.251
Upravený model 3 skryté vrstvy	-	0.255
Model s polohou slnka	0.085	0.259
Lineárna regresia	-	0.268

Relatívna chyba predikcie modelu 0.259 je horšia, ako predikcia, ktorú získali v článku [14], čo bolo 0.199 . Z článku však nie je jasné, o konkrétne aké dáta sa jednalo, akú mali štruktúru a ako boli predspracované. Taktiež sa tam nepojednáva o ďalšom spracovaní výsledných dát. Čo sa týka modelu sú spomenuté len počty vrstiev, ale nie koľko neurónov obsahujú. Z článku nie je ani jasné aký veľký je predikčný horizont.

Pri porovnaní siete so sieťou z článku [5] môžeme vidieť, že RMSE z článku má hodnotu 0.0675 , pri čom hodnota nášho modelu je 0.085 . Na základe číselných údajov RMSE však nemôžeme s určitosťou povedať, že sieť z článku [5], je presnejšia. Rozdiel v hodnotách ako aj neurčitost porovnania vzniká rozdielnym datasetom.

3 ZÁVĚR

Cielom práce bolo vytvoriť nástroj na predikciu vývoja časových radov s jedným alebo viacerými vstupnými atribútmi.

V teoretickej časti je stručne popísané hlboké učenie. Čo to vlastne je a ako funguje, stručný popis druhov učenia a porovnanie hlbokého učenia a klasického strojového učenia. Podrobnejšie sú rozobraté RNN a predovšetkým LSTM, ktoré sú vhodné práve pre predikciu časových radov. Nachádza sa tam rozbor funkcie sietí LSTM.

V ďalšej časti teoretického úvodu sú rozobrané dokumentácie rôznych projektov zaoberajúcich sa predikciou časových radov, ktoré sú použité v praxi. Jedná sa o:

1. Predikciu zrážok v určitej oblasti.
2. Predikciu elektrickej záťaže
3. Diagnostiku v zdravotníctve
4. Predikciu výkonu solárnej elektrárne v Hae-Nam
5. Hlboké učenie pre predikciu solárnej energie

Každý problém je rozdelený do podkapitol, ktoré sa podrobnejšie venujú použitým dátam, ich spracovaniu a použitému modelu. Na záver obsahujú kapitolu, ktorá je stručným zhrnutím nastoleného problému a výsledného riešenia.

V praktickej časti sú stručne popísané použité knižnice a dáta, ktoré boli k dispozícii pre učenie a testovanie siete. Podrobnejšie je opísané spracovanie dát. Od uloženia dát vo formáte vhodnejšom pre spracovanie a úpravu, až po samotnú úpravu dát, aby spĺňala podmienky potrebné pre učenie pod dohľadom. Pozornosť tu je venovaná aj modelu a prvotnému nastaveniu parametrov pre proces učenia. V závere kapitoli sú popísané a vykreslené výsledky.

V praktickej časti sa na začiatku podarilo vytvoriť model, ktorý zo vstupných dát, predikuje výkon siete v predikčnom horizonte 24 hodín. Kým sa model dostal do výslednej podoby, prešiel mnohými úpravami. Prvotný model mal relatívnu odchylku 0.787 . Finálny model, ktorý bol rozšírený o vrstvy a dostal rozsiahlejší dataset, predikuje záťaž v horizonte 24 hodín s relatívnou chybou 0.259 .

Predikcia by sa dala spresniť rozšírením dát o ďalšiu premennú, na ktorej je výkon článkou závislý, alebo rozšírením datasetu o počet vzorkov. Popríklad kombináciou oboch týchto možností.

Pre predikciu boli skúšané aj iné metódy, ale buď nedosahovali ani z ďaleka takú presnosť, alebo z technických dôvodov vôbec nefungovali.

LITERATÚRA

- [1] Neural Networks. 2000.
URL <https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/index.html>
- [2] Understanding LSTM Networks. 2015.
URL <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>
- [3] Deep Learning vs. Machine Learning – the essential differences you need to know! 2017.
URL <https://www.analyticsvidhya.com/blog/2017/04/comparison-between-deep-learning-machine-learning/>
- [4] Keras Documentation. 2017.
URL <https://keras.io/layers/recurrent/>
- [5] Gensler, A.; Henze, J.; Sick, B.: Deep Learning for Solar Power Forecasting – An Approach Using Autoencoder and LSTM Neural Networks. 2016.
URL <https://ieeexplore.ieee.org/document/7844673/>
- [6] Goodfellow, I.; Bengio, Y.; Courville, A.: *Deep Learning*. MIT Press, 2016,
<http://www.deeplearningbook.org>.
- [7] Hinton, G.; Deng, l.; Yu, D.; aj.: Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups. -, rok 29, 11 2012: s. 82–97.
- [8] Hochreiter, S.; Schmidhuber, J.: LONG SHORT-TERM MEMORY. 1997.
URL <http://www.bioinf.jku.at/publications/older/2604.pdf>
- [9] Hosein, S.; Hosein, P.: Load Forecasting using Deep Neural Networks. 2016.
URL http://www.rndrepository.com/smartgriduwi/files/smartgridcomm_2016.pdf
- [10] Lacerda, A.; Cristo, M.; Gonçalves, M. A.; aj.: Learning to Advertise. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, New York, NY, USA: ACM, 2006, ISBN 1-59593-369-7, s. 549–556, doi:10.1145/1148170.1148265.
URL <http://doi.acm.org/10.1145/1148170.1148265>
- [11] Lipton, Z. C.; Elkan, C.; Kale, D. C.; aj.: Learning to diagnose with LSTM recurrent neural networks. *ICLR*, rok 2016, -, 2016: str. 18.
URL <https://arxiv.org/pdf/1511.03677.pdf>

- [12] McCallum, A.; Nigam, K.; Seymore, K.; aj.: Building Domain-Specific Search Engines with Machine Learning Techniques. 1999.
- [13] Pinheiro, P. H. O.; Collobert, R.; Dollár, P.: Learning to Segment Object Candidates. *CoRR*, rok abs/1506.06204, 2015, 1506.06204.
URL <http://arxiv.org/abs/1506.06204>
- [14] Poundel, P.; Jank, B.: Solar Power Prediction Using Deep Learning Technique. *Advanced Science and Technology Letters*, rok 26, 46, 2017: str. 4.
URL <http://dx.doi.org/10.14257/astl.2017.146.26>
- [15] SHI, X.; Chen, Z.; Wang, H.; aj.: Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In *Advances in Neural Information Processing Systems 28*, editace C. Cortes; N. D. Lawrence; D. D. Lee; M. Sugiyama; R. Garnett, Curran Associates, Inc., 2015, s. 802–810.
URL <http://papers.nips.cc/paper/5955-convolutional-lstm-network-a-machine-learning-approach-for-precipitation-nowcasting.pdf>

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

ANN	Umelá neurónová sieť – Artificial Neural Network
CNN	Konvolučná neurónová sieť – Convolutional Neural Network
PNN	Pravdepodobnostná neurónová sieť – Probabilistic Neural Network
RNN	Rekurentná neurónová sieť – Recurrent Neural Network
LSTM	Dlhá krátkodobá pamäť – Long short-term memory
GRU	Uzavretá rekurentná jednotka – Gated Recurrent Unit
FC-LSTM	Plne prepojená LSTM – Fully Connected LSTM
CLSTM2D	Konvolučná, dlhá krátkodobá sieť – Convolutional Long Short-Term Memory
2D	Dvojrozmerná – Two Dimensional
3D	Troj rozmerná – Three Dimensional
MAE	Stredná absolútna chyba – Mean Absolute Error
RMSE	Stredná kvadratická chyba – Root Mean Square Error
RAM	Pamäť s náhodným prístupom – Random-Access Memory
MA	Metóda pohyblivého priemeru – Moving average method

ZOZNAM PRÍLOH

A Obsah priloženého CD

49

A OBSAH PRILOŽENÉHO CD

CD obsahuje súbor *Data2.py* v ktorom sa nachádza konfigurácia LSTM modelu. Taktiež sa na CD nachádza súbor *excel_to_csv*, ktorý sa stará o načítanie a prekonvertovanie súboru *data.xlsx* do formátu *csv*, v ktorom sa nachádzajú originálne dáta a taktiež je súčasťou priloženého CD. Ďalej sa na CD nachádza súbor *LinearRegression2.py*, ktorý obsahuje kód pre model s lineárnou regresiou. Ďalší súbor *XboostPred.py* obsahuje kód, pre metódu pomocou *XGBoost*. Súbor *SUN2.py* obsahuje kód, ktorý pre každé časové razítko v datasete, pridá údaj o polohe Slnka. Taktiež obsahuje súbor *sunposition.py*, ktorý slúži ako knižnica práve pre súbor *SUN2.py*. Súbor *datasetwithsun.csv* obsahuje dataset, ktorý je použitý pri učení a testovaní finálneho modelu. Na CD sa taktiež nachádza elektronická verzia diplomovej práce.