

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

FIIT-5220-52003

Bc. Michal Farkaš

Predikcia spotreby energie pomocou hlbokých neurónových sietí

Diplomová práca

Vedúci práce: Ing. Peter Lacko, PhD.

Máj 2017

Slovenská technická univerzita v Bratislave
Fakulta informatiky a informačných technológií

FIIT-5220-52003

Bc. Michal Farkaš

Predikcia spotreby energie pomocou hlbokých neurónových sietí

Diplomová práca

Študijný program: Softvérové inžinierstvo

Študijný odbor: 9.2.5 Softvérové inžinierstvo

Miesto vypracovania: Bratislava

Vedúci práce: Ing. Peter Lacko, PhD.

Máj 2017

ČESTNÉ PREHLÁSENIE

Čestne prehlasujem, že záverečnú prácu som vypracoval samostatne s použitím uvedenej literatúry a na základe svojich vedomostí a znalostí.

V Bratislave,

.....
Bc.Michal Farkaš

PodĎakovanie

Ďakujem svojmu vedúcemu Petrovi Lackovi za jeho vedenie, pomoc, neoceni-
teľné rady, svojej rodine za vytrvalú podporu, našej fakulte za kvalitné vzdelanie
a každému kto mi pomohol pri písaní tejto práce.

Anotácia

Slovenská technická univerzita v Bratislave

FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLOGIÍ

Študijný program: Softvérové inžinierstvo

Autor: Bc. Michal Farkaš

Diplomová práca: Predikcia spotreby energie pomocou hlbokých neurónových sietí

Vedúci diplomového projektu: Ing. Peter Lacko, PhD.

Máj 2017

Predikovanie spotreby energie je veľmi dôležitá súčasť plánovania firiem v energetickom priemysle. V dnešnej dobe je v mnohých krajinách energetický priemysel menej regulovaný ako v minulosti, preto na jednom trhu operuje väčšie množstvo súkromných firiem, ktoré musia byť konkurencie schopné. Práve vďaka dobrej predpovedi môže firma získať výhodu oproti konkurencií. Zvýšený záujem o alternatívne zdroje energie, decentralizované mini-elektrárne, smartgrid a mikrogrid zvyšuje nároky kladené na metódy predpovedania spotreby energie.

Problém predpovedania spotreby energie je už dlhšie objektom výskumu a existuje veľké množstvo prác, ktoré sa ho snažia riešiť. V tejto práci je predpovedania spotreby energie riešené skrz hlboké neurónové siete, ktoré od svojho spopularizovania v roku 2006, sú súčasťou mnohých úspešných riešení ako aj v akademickom tak aj v komerčnom prostredí.

Annotation

Slovak University of Technology Bratislava

FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGIES

Degree Course: Software Engineering

Author: Bc. Michal Farkaš

Master thesis: Prediction of energy load using deep neural networks

Supervisor: Ing. Peter Lacko, PhD.

Máj 2017

Prediction of energy load is very important part of planning in companies working in energy industry. Today, in many countries energy industry is more deregulated than in past, thus on one market there may be more companies, which have to be competitive. Thanks to better forecast, company can get advantage over their rivals. Increased interest of alternative sources of energy, decentralized mini-powerplants, smartgrids and microgrids cause higher demands on forecasting methods.

Problem of energy load prediction had research attention for quite some time and there are many papers, which try to come up with sufficient solution. In this paper, energy load prediction is done through deep neural networks, which since being popularized in 2006 have been part of many academic or commercial solutions.

Obsah

1	Úvod	1
2	Analýza problémovej oblasti	3
2.1	Súčasný stav v energetickom priemysle	3
2.2	Používané riešenia	5
2.2.1	Faktory zasahujúce do predpovede	5
2.2.2	Neurónové siete	6
3	Teória neurónových sietí	9
3.1	Neurónové siete	9
3.1.1	Neurón	9
3.1.2	Topológia	13
3.1.3	Učenie	15
3.1.4	Hlboké neurónové siete	16
3.1.4.1	Autoenkóder	17
3.1.4.2	Konvolučné neurónové siete	19
3.1.4.3	Hlboké reziduálne siete	19
3.1.4.4	Pred-trénované neurónové siete	20
4	Použité technológie	21
4.1	Theano	21
4.2	Lasagne	22
4.3	TensorFlow	22

5	Návrh a implementácia riešenia	25
5.1	Kauzálné konvolučné siete	25
5.2	Dilatované konvolučné siete	26
5.3	1x1 konvolúcie	28
5.4	WaveNet architektúra	28
5.4.1	WaveNet implementovaný v tejto práci	30
5.4.1.1	Predspracovanie dát	32
6	Dosiahnuté výsledky	33
6.1	Dátová sada 1: Belgicko	34
6.2	Dátová sada 2: Bratislava	38
6.3	Dátová sada 3: New York	41
7	Zhrnutie	45
7.1	Vyhodnotenie výsledkov	45
7.2	Budúca práca a návrh zlepšenia	47
	Literatúra	49
	Prílohy	52
A	Technická dokumentácia	A-1
A.1	Opis softvérovej konfigurácie	A-1
A.2	Opis tried a metód programu	A-2
B	Používateľská príručka	B-1
C	Elektronické médium	C-1
D	Vyhodnotenie plánu práce	D-1
E	Pripravený príspevok na konferenciu EANN2017	E-1

Zoznam obrázkov

3.1	Nákres biologického neurónu; prevzaté z [20].	10
3.2	Nákres umelého neurónu; prevzaté z [20].	11
3.3	Ukážky rôznych prechodových funkcií	12
3.4	Príklad feed-forward topológie	14
3.5	Príklad rekurentnej topológie	14
3.6	Proces rozloženia neurónovej siete v čase	15
3.7	Topológia neurónovej siete typu autoenkóder	17
3.8	Skladaný autoenkóder, farbami sú vyznačené vnorené autoenkóдеры	18
3.9	Ukážka hlbkej reziduálnej siete	20
5.1	Rozdiel medzi normálnou a kauzálnou konvolúciou	26
5.2	Rozdiel medzi kauzálnou a dilatovanou konvolúciou	27
5.3	WaveNet architektúra; prevzaté z [22].	29
6.1	Detail predpovede na 1 krok dopredu, Belgické dáta	35
6.2	Detail predpovede na 1 deň dopredu, Belgické dáta	35
6.3	Detail predpovede na 3 dni dopredu, Belgické dáta	36
6.4	Detail predpovede na 7 dní dopredu, Belgické dáta	36
6.5	Detail predpovede na 29 dní dopredu, Belgické dáta	37
6.6	Detail predpovede na 1 krok dopredu, Bratislavské dáta	39
6.7	Detail predpovede na 1 deň dopredu, Bratislavské dáta	39
6.8	Detail predpovede na 3 dni dopredu, Bratislavské dáta	40
6.9	Detail predpovede na 7 dní dopredu, Bratislavské dáta	40
6.10	Detail predpovede na 29 dní dopredu, Bratislavské dáta	41
6.11	Detail predpovede na 1 krok dopredu, New Yorkské dáta, august	42
6.12	Detail predpovede na 3 dni dopredu, New Yorkské dáta, november	43

6.13	Detail predpovede na 3 dni dopredu, New Yorkské dáta, august . .	43
------	--	----

Kapitola 1

Úvod

V dnešnej dobe si život bez elektrickej energie nie sme schopní ani len predstaviť. Jej použitie je tak rozšírené a všestranné, že sa dotýka takmer akejkolvek ľudskej činnosti. Okrem osobného použitia fyzickými osobami je elektrická energia extrémne dôležitým vstupom do výroby. Z tohto dôvodu je elektrická energia dôležitým ekonomickým faktorom a zároveň dôležitým strategickým zdrojom krajín.

Niet sa teda čo čudovať, že jej produkcia je samostatné, strategické priemyselné odvetvie. Takto dôležité odvetvie pre krajinu s tak veľkým vplyvom musí byť kvalitne riadené a plánované, aby bol zabezpečený dlhodobý rozvoj.

V minulosti bol bežný model jedného, štátneho dodávateľa a distribútora energie. V dnešnej dobe vďaka deregularizácii odvetvia sú vytvorené medzinárodné energetické trhy, na ktorých môže byť aj v rámci jedného štátu viacero konkurenčných firiem, ktoré vyrábajú či distribuujú energiu. Takéto prostredie je omnoho chaotickejšie a náročnejšie na manažment, kontrolu a riadenie.

Pretrvávajúcim problémom energetického priemyslu je jeho väčšia schopnosť energiu vyrábať ako skladovať. Keďže nie je k dispozícii technológia, ktorá by umožňovala účinné, dlhodobé a veľkokapacitné ukladanie je nutné vyrábať energiu navyše aby jej bolo dostatok. Lenže v dnešnej dobe kedy efektívnosť, ekonomickosť a ekologickosť sú dôležité faktory nie je takýto prístup ideálny. Navyše na deregulovanom trhu, kde producenti bojujú o zákazníka, musia byť konkurencieschopný, čo znamená, že podniky musia minimalizovať množstvo vyrobenej energie, ktorú nik nekúpi.

Komplikujúcim faktorom a zároveň problémom, ktorý treba riešiť je zvyšujúca penetrácia alternatívnych zdrojov energie. Zväčša sa jedná o zdroje, ktorých energetický výstup je závislý od vonkajších faktorov a iba ťažko sa kontroluje. Naplánovanie množstva vyrobenej energie každou elektrárnou tak aby bola pokrytá celková spotreba je už tak náročná úloha, ktorá sa iba komplikuje použitím alternatívnych zdrojov energie.

Predpovedanie spotreby energie je problém riešený už niekoľko desaťročí. V minulosti bolo navrhnutých a zavedených do praxe viacero riešení, ktoré používajú štatistické metódy, avšak v dnešnej dobe sa väčšina výskumu zamerala na neurónové siete alebo ich deriváty.

Neurónové siete sú oblasťou strojového učenia, ktorá má svoje korene v biológii. Základným kameňom takýchto sietí je umelý neurón, ktoré sú komponované do najrôznejších topológií. Dlhodobým problémom neurónových sietí bola ich zložité učenie pri vyššom počte vrstiev. v poslednom desaťročí nabrali na popularite takzvané hlboké neurónové siete, ktoré majú väčší počet vrstiev pričom ich schopnosť učiť nie je kompromitovaná. V tejto práci sa zaoberáme práve hlbokými neurónovými sieťami.

Táto práca je rozdelená na dve hlavné kapitoly, z ktorých prvá sa zaoberá teóriou neurónových sietí a známymi variantami hlbokých neurónových sietí. Druhá kapitola pojednáva o probléme predpovedania spotreby energie a použitými riešeniami z oblasti neurónových sietí.

Kapitola 2

Analýza problémovej oblasti

2.1 Súčasný stav v energetickom priemysle

Z technologického hľadiska je jednoduchšie energiu vyrábať ako skladovať. Z tohto jednoduchého dôvodu je potrebné pre producentov energie aby mali dôveryhodné odhady o budúcej spotrebe. Pre konzumenta s vysokým odberom, môže byť takýto odhad rovnako dôležitý. Takéto odhady pomáhajú dospieť k správnym krátkodobým, ale aj dlhodobým rozhodnutiam spoločností a inštitúcií.

V deregulovanom energetickom priemysle vystupuje viacero producentov a distribútorov energie, ktorý bojujú o zákazníkov. Celkové množstvo energie v takejto sieti je regulované na základe trhového mechanizmu - spoločnosti nebudú vyrábať energiu, ktorú nikto nekúpi. V takto nastavenom energetickom priemysle je predpoveď energie omnoho dôležitejšia.

V dnešnej dobe je vypúšťanie emisií CO_2 a iných skleníkových plynov vážnym ekologickým problémom kvôli skleníkovému efektu a ním spôsobenému globálnemu otepľovaniu. Z tohto dôvodu sa v posledných rokoch uprednostňujú a rozvíjajú alternatívne zdroje energie. Mnohé z týchto zdrojov, ako slnečná či veterná energia majú premenlivý produkt, ktorý je navyše závislý na vonkajších faktoroch.

Mnohé menšie solárne či veterné elektrárne sú napojené do lokálnej siete a ich výstup nie je možné preniesť mimo tejto siete. Dokonca takto generovaná energia je zvonku viditeľná iba ako pokles spotreby energie. Takéto elektrárne sa nazývajú

vložené (angl. embedded). Slúžia k zníženiu nárokov na prenosovú sústavu.

Všetky spôsoby výroby energie sa dajú rozdeliť na flexibilné, ktoré sú schopné rýchlo zmeniť množstvo nimi vyrábanej energie a tie, ktoré sa nedokážu rýchlo adaptovať na zmenu v odbere. Pre ľubovoľný okamih je potrebné dodať dostatok energie, ktorej výroba musí byť rozvrhnutá medzi flexibilné, ne-flexibilné a zdroje energie, ktorých maximálny potenciálny výstup sa nedá nastaviť, iba predpovedať (hlavne slnečná a veterná energia). Pri tomto plánovaní znovu veľmi významne vstupujú do procesu predpovede spotreby ale aj produkcie slnečnej a veternej energie.

Podľa správy o výhľade na leto 2016 v Spojenom kráľovstve [10] bude prevádzkovanie distribučnej siete náročné. Tento stav nastal hlavne kvôli zníženiu minimálnej spotreby energie, čo zvýšilo rozdiel medzi minimálnou a maximálnou spotrebou. Aby bola spotreba pokrytá počas nízkej spotreby aj počas vysokej spotreby je potrebná vyššia flexibilita zdrojov energie. Záver správy zhodnocuje, že distribučná sieť v Spojenom kráľovstve bude schopná fungovať celé leto s bezpečnostnou rezervou, avšak môže dôjsť k dočasnému odpojeniu niektorých veterných elektrární v čase nízkej spotreby. Táto správa je ideálnou ukážkou dôležitosti predpovedania spotreby energie, pretože bez nej by neboli možné.

V prípade zlyhania plánovania môžu nastať dva prípady. V prvom prípade nie je pokrytá spotreba, v takomto prípade nastávajú výpadky energie alebo takzvané brown-outy, pod týmto termínom sa rozumie zámerné alebo neúmyselné zníženie napätia v sieti. Rôzne typy elektrických zariadení môžu reagovať rôzne, od zníženia výkonu, zvýšenia poruchovosti či poškodenia zariadenia, niektoré zariadenia sa môžu správať nepredvídateľne a niektoré nemusia byť zasiahnuté vôbec. V opačnom prípade, keď je v sieti príliš veľa energie, môže dôjsť k poškodeniu rozvodnej siete.

Metódy predpovedania, bez ohľadu na oblasť do ktorej spadá riešenie, sa dajú rozdeliť na krátkodobé, stredné a dlhodobé predpovede. Krátkodobé predpovede sú v rámci jednej hodiny do jedného týždňa. Stredné predpovede sa snažia predpovedať na dobu medzi jedným týždňom a jedným rokom a dlhodobé predpovede sú na rok a viac. [8]. V niektorých prípadoch si môžu autori definovať vlastné zadelenie, napríklad predpovedania na hodinu dopredu.

2.2 Používané riešenia

Existuje veľké množstvo spôsobov ako úspešne predpokladať spotrebu energie. Je nad rámec tejto kapitoly spomenúť všetky, preto iba zbežne prejdeme niektoré z nich.

Predpovedanie spotreby energie je známy problém, na ktorý sa používa široké spektrum riešení a prístupov, ktoré sa dajú veľmi hrubo dať rozdeliť na predpovede pomocou štatistických metód a predpovedanie pomocou umelej inteligencie.

Medzi štatistickými prístupmi je pravdepodobne najznámejší ARIMA model. Takto bol použitý v [1], kde bol ARIMA model modifikovaný tak aby bral do úvahy aktuálnu teplotu. V závere autori skonštatovali, že takto upravený ARIMA model je účinnejší oproti jeho typickej variante. Zdá sa, že väčšina prác zaoberajúcich sa štatistickými modelmi, vrátane ARIMA modelu sú staršieho dátumu a už nie sú hlavným predmetom výskumu.

Okrem štatistických metód existujú aj rôzne prístupy z odboru umelej inteligencie, ktoré nespádajú čisto pod umelé neurónové siete, napríklad expertné systémy [13, 15], fuzzy logika [18] či iné [9].

Niektoré metódy predpovedania spotreby energie kombinujú viacero metód aby dosiahli lepšie výsledky.

2.2.1 Faktory zasahujúce do predpovede

Je zjavné, že do spotreby energie zasahuje obrovské množstvo faktorov, niektoré sa dajú vyjadriť a máme o nich informácie a niektoré ako napríklad kultúrne faktory sa nedajú zmysluplne zaznamenať.

Výber významných faktorov na základe, ktoré bude systém brať do úvahy je dôležitou súčasťou návrhu metódy predpovedania. V prípade štatistických metód sa jedná o faktory, ktoré budú súčasťou modelu, v prípade riešenia problému pomocou umelej inteligencie sa jedná o výber vstupov do systému.

Na zahrnuté faktory má vplyv aj zameranie na krátkodobé či dlhodobé predpovede. Niektoré faktory môžu byť irelevantné ak sa predpovedá na dlhšiu dobu.

Vo všeobecnosti platí, že niektoré metódy vyžadujú viac informácií ako iné, dokonca sa môže jednať aj o makroekonomické či poveternostné údaje, ktorých

budúce hodnoty môžeme iba predpokladať. Rovnako platí, že niektoré prístupy vyžadujú príliš ťažko získateľné údaje ako napríklad priemerné či konkrétne údaje o domácnostiach a ich obyvateľoch. Je zjavné, že takéto ukazovatele sa menia v čase a veľmi ťažko sa získavajú.

Z tohto dôvodu býva výber faktorov a aj použitej metódy, často limitovaný dostupnými dátami. V niektorých prípadoch je možné chýbajúce dáta získať z iných zdrojov a prepojiť ich pomocou dátumu a geografickej polohy.

2.2.2 Neurónové siete

Prác zaoberajúcich sa predikciou spotreby energie pomocou neurónových sietí je neúrekom, v tejto kapitole v krátkosti spomenieme niektoré práce, ktoré zachytili našu pozornosť.

Kombinácia neurónovej siete a expertného systému bola použitá v práci [16]. V tejto práci je predpovedanie rozdelené na dve fázy, v prvej fáze je použitá neurónová sieť v druhej fáze sú výstupy tejto siete upravené expertným systémom tak aby boli brané do úvahy faktory ako teplota či prázdniny.

V práci [21] je použitá kombinácia štatistickej metódy podobného dňa a jej úprava pomocou neurónovej siete. Metóda podobného dňa funguje tak, že z historických dát sa vyberá jeden či viac podobných dní k dňu pre ktorý predpovedáme, pričom sa predpokladá, že ich spotreba bude podobná. Pretože v tejto práci sa predpovedá na hodinu dopredu, nebolo by možné túto metódu použiť, avšak preto je použitá neurónová sieť, ktorá spresňuje výsledky a upravuje predpoklad na konkrétnu hodinu.

Kombinácia učenia s učiteľom a učenia bez učiteľa je použitá v práci [3]. Nejedná sa o hlboké učenie s použitím predtrénovania ale o normalizáciu a predspracovanie dát do klastrov pomocou Kohonenových samo-organizujúcich sa máp. Informácia o príslušnosti ku klastru bola súčasťou vstupov do neurónovej siete.

Architektúra jedného neurónu, pracujúceho podľa funkcie spomenutej v 3.2 bola použitá v práci [23]. V závere práce je vyhlásené, že navrhnutý neurón vykazuje lepšie výsledky ako viacvrstvové neurónové siete s nižšou výpočtovou zložitou. Tento model bol neskôr rozpracovaný [7] tak aby obsahoval rekurentné prepojenia.

Práca [2] sa zaoberala krátkodobým predpovedaním spotreby energie pomocou neurónových sietí trénovaných PSO algoritmom. V tejto práci bol tiež použitý predspracovanie pomocou vlnkovej transformácie, ktorá patrí medzi časovo-frekvenčné transformácie. Nad takto spracovanými dátami je postavená neurónová sieť, ktorá berie do úvahy aj údaje o počasí. V závere autori konštatujú, že použitie vlnkovej transformácie prispelo k presnejšej predpovedi.

V ďalšej práci [11] sú použité vlnkové neurónové siete, ktorých neuróny používajú ako aktivačné funkcie vlnkové transformácie. V tejto práci bola spotreba energie predpovedaná dopredu na jednu hodinu v intervale každých 5 minút. Výsledná predpoveď bola kombinovaná z viacerých neurónových sietí, pričom každá spracovávala jednu časť rozložených dát.

Spiking neurónová sieť bola použitá v [17] na krátkodobé predpovedanie spotreby energie. Proces predpovedania je rozdelený na dve fázy, pričom v prvej fáze sa predpovedá teplota po hodinách na deň dopredu z hodinových údajov o teplote, vlhkosti z predchádzajúcich dvoch dní a maximálnej slnečnosti z predchádzajúceho dňa. V druhej fáze sa predpovedá spotreba po pol hodine na deň dopredu z údajov získaných v prvej fáze, dátumové údaje, prázdniny a iné.

Kapitola 3

Teória neurónových sietí

3.1 Neurónové siete

Umelé neurónové siete sú aproximáciou prirodzených neurónových sietí, ktoré sa nachádzajú v telách živočíchov. Jedná sa o aproximáciu, pretože stále nepoznáme presný spôsob akým neurón, prirodzené neurónové siete a mozgy pracujú.

Prirodzené neurónové siete nemusia byť najefektívnejšou 'implementáciou' centrálného nervového systému, pretože boli vytvorené počas ľudskej evolúcie. Napriek tomu, že evolúcia je de-facto optimalizačný mechanizmus, evolúcia vyberá prvé zlepšujúce riešenie a perfektné riešenie.

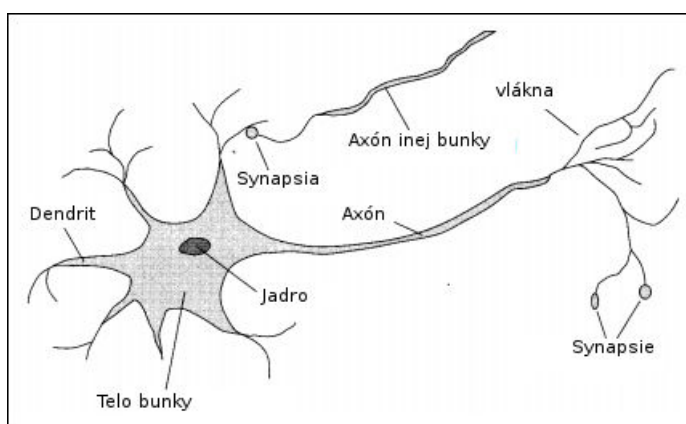
A preto aj keby sme poznali presný spôsob akým prirodzené neurónové siete fungujú, nemusia byť ideálnym riešením. Rozdiel, ktorý treba brať do úvahy je tiež fakt, že počítače sú, narozdiel od nervových sústav, digitálneho charakteru.

3.1.1 Neurón

Základným kameňom neurónových sietí je neurón, známy aj ako nervová bunka.

Nervová bunka je tvorená telom bunky, ktorá obsahuje jadro a z tela vystupuje niekoľko menších výstupkov, zvané dendrity. Tieto výstupky sa na konci rozchádzajú do väčšieho počtu vlákien. Okrem týchto krátkych výstupkov existujú aj väčšie výstupky zvané axóny, ktoré sa tiež rozchádzajú do viacerých vlákien. Tieto sú prepojené na dendrity inej bunky, takéto prepojenie sa nazýva synapsia.

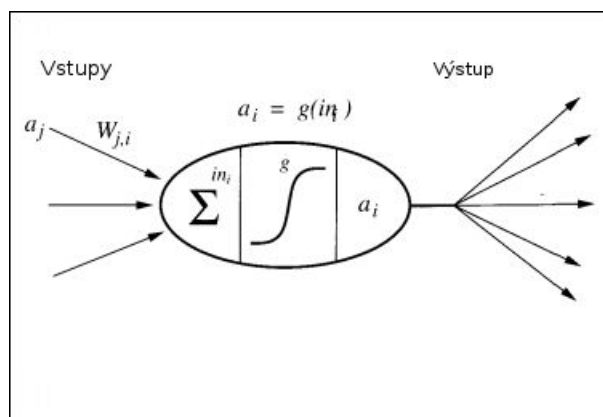
Synapsie môžu byť exhibičné, ktoré zvyšujú šancu či interval medzi výstupmi alebo inhibičné, ktoré ho znižujú. Presný popis fungovania synapsí a správania bunky je však nad rámec tejto práce a je oblasťou aktívneho výskumu. Biologický neurón sa nachádza na obrázku 3.1.



Obr. 3.1: Nákres biologického neurónu; prevzaté z [20].

Neuróny, ktoré sú súčasťou umelých neurónových sietí, sú zjednodušeným modelom prirodzeného neurónu. Namiesto komplikovaného vnútorného procesu používa takzvanú prechodovú funkciu, ktorej výstup je výstupom neurónu. Vstupy do umelého neurónu môžu byť, rovnako ako v prípade dendritov, tiež inhibičné alebo exhibičné. Umelý neurón je znázornený na obrázku 3.2. Na obrázku je vidno, že dendrity sú nahradené vstupmi a axón je nahradený výstupom neurónu. Mechanizmus synapsí biologického neurónu nie je v takomto modeli prítomný. Každý prepojenie a teda aj vstup má svoju váhu, ktorá sa väčšinou označuje ako w , tieto váhy určujú dôležitosť vstupu.

$$y = f(\sum(x_i * w_i) + b) \quad (3.1)$$



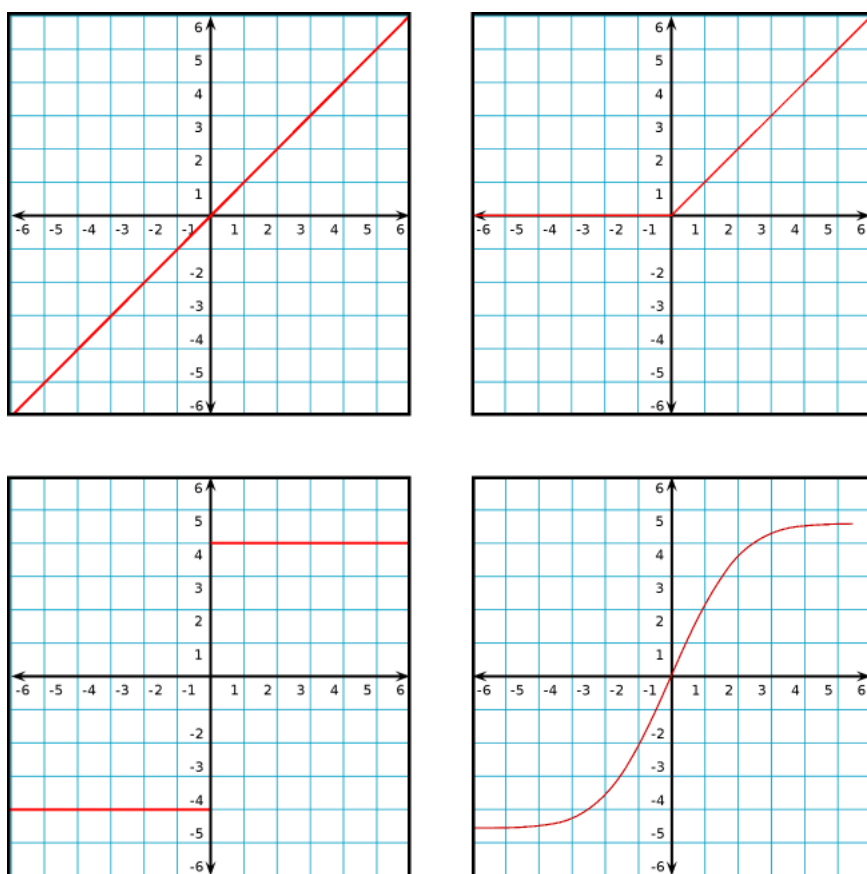
Obr. 3.2: Nákres umelého neurónu; prevzaté z [20].

Existuje viacero typov umelého neurónu, niektoré typy sú dostatočne podobné a líšia sa iba v detailoch. Matematické vyjadrenie mechanizmu fungovania typického neurónu je znázornené na funkcii 3.1.1. Vo všetkých funkciách v tejto kapitole je i -ty vstup označený ako x_i a váha pripadajúca tomuto vstupu je označená ako w_i . Hodnota bias je označená ako b a funkcia $f()$ je prechodová funkcia neurónu.

Vstupy prichádzajúce po prepojení do neurónu sú vynásobené váhou daného prepojenia a k tomuto výsledku je pripočítaná hodnota bias. Hodnoty bias sú predmetom učenia rovnako ako váhy ale sú sčítané so súčinom váh a vstupov, môžu byť implementované ako súčasť neurónu alebo pripojené cez samostatné prepojenie, ktoré má aj svoju váhu. V prípade negatívnej hodnoty váhy sa jedná o inhibičné prepojenie, v opačnom prípade sa jedná o exhibičné prepojenie. Takto transformované vstupy sú sčítané a tento produkt je použitý ako vstup do prechodovej funkcie. Výstup tejto funkcie je poslaný na výstup neurónu, ktorý môže byť súčasťou výstupu neurónovej siete alebo vstupom do ďalších neurónov.

Neuróny sa môžu líšiť aj v prechodovej funkcii, ktorá produkuje ich výstup. Často sa v jednej neurónovej sieti miešajú neuróny s rôznymi prechodovými funkciami. Niektoré bežné varianty sú znázornené na obrázku 3.3. V hornej časti ukážky je zobrazená lineárna funkcia a ReLu funkcia, čiže napravená lineárna funkcia

(angl. rectified linear unit). V dolnej časti sa nachádza kroková funkcia(angl. step function), ktorá je typická pre binárne neuróny a sigmoidálna funkcia, ktorá je z týchto asi najpoužívanejšia.



Obr. 3.3: Ukážky rôznych prechodových funkcií

I keď toto je najpoužívanejší typ neurónu existujú aj iné, príkladom môže byť neurón ktorého vstupy sú násobené a nie sčítané. Takto fungujúci neurón bol použitý v [23], schéma jeho fungovania je nasledovná.

$$y = \prod (w_i * x_i + b) \quad (3.2)$$

V inej práci [6] autor navrhol iný ešte zložitejší typ vyššieho neurónu, ktorý pracoval nasledovne.

$$y = \prod (x_i^{w_i}) \quad (3.3)$$

V oboch prípadoch bolo motiváciou autorov lepšie riešiť problémy, ktoré nemôžu byť definované ako lineárne kombinácie vstupov. V oboch prípadoch mali takéto neuróny a siete, ktoré ich obsahovali sľubné výsledky.

Okrem takto modifikovaných neurónov existujú aj takzvané spiking neuróny, ktoré sa snažia presnejšie simulovať fungovanie biologického neurónu. Takéto neuróny vysielajú svoj výstup iba ak ich vnútorná hodnota, zvaná aj potenciál, dosiahne určitú hranicu, po vyslaní sa ich vnútorná hodnota zníži. Týmto sa simulujú elektrické vlastnosti prirodzených neurónov a tak sa jedná o abstrakciu bližšiu realite.

3.1.2 Topológia

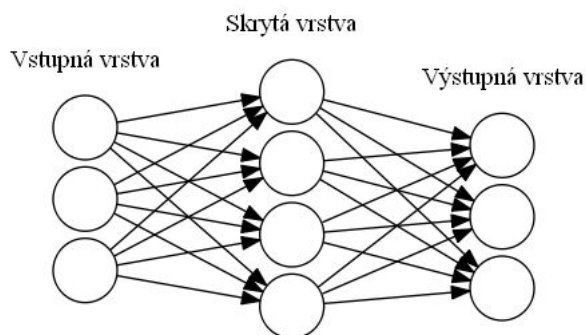
Neurón ako taký má veľmi obmedzené schopnosti, preto sa obyčajne skladá viac neurónov do jednej neurónovej siete, ktorá je obvykle roztriedená do vrstiev. Prvou vrstvou je takzvaná vstupná vrstva, ktorá obsahuje vstupné hodnoty do neurónovej siete. Výstupná vrstva je posledná a jej výstupom je výstup neurónovej siete, tie to dve vrstvy sa nazývajú aj ako viditeľné vrstvy. Medzi nimi sa nachádza ľubovoľný počet skrytých vrstiev.

Vrstvy sú definované minimálnym počtom neurónov cez, ktoré musí ich výstup prejsť aby dosiahol výstupnú vrstvu. Neurónové siete s vyšším počtom skrytých vrstiev majú pomenovanie hlboké neurónové siete a budú prebraté neskôr v tejto kapitole.

Počet neurónov a prepojení medzi nimi sa môže líšiť. Takto vznikajú najrôznejšie topológie, ktoré pri bežných prístupoch musia byť navrhnuté človekom alebo iným systémom ale ich návrh je málokedy súčasťou učiaceho algoritmu. Príkladom kedy je vytvorená aj topológia siete bez zásahu človeka sú neuroevolučné TWEANN prístupy, z ktorých najznámejší je NEAT [19].

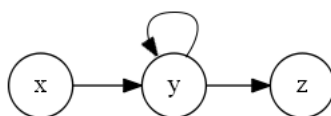
Vo všeobecnosti je možné rozdeliť neurónové siete na dopredné a rekurentné. Prvé menované siete sú jednoduchšími variantami, ktoré neobsahujú cykly a všetky prepojenia sú jednosmerné. To znamená, že výstup neurónu z jednej vrstvy môže byť vstupom do neurónu iba na jednej z vrstiev, ktorá nasleduje až za ňou. Takto

navrhnuté siete neobsahujú žiaden vnútorný stav okrem svojej topológie a svojich váh a tieto hodnoty sa počas prevádzky obvykle nemenia. Praktickým následkom takejto topológie je, že neurónová sieť nemá krátkodobú pamäť a jej schopnosti sú výrazne obmedzené. Na druhú stranu sú jednoduchšie a ľahšie sa trénujú.



Obr. 3.4: Príklad *feed-forward* topológie

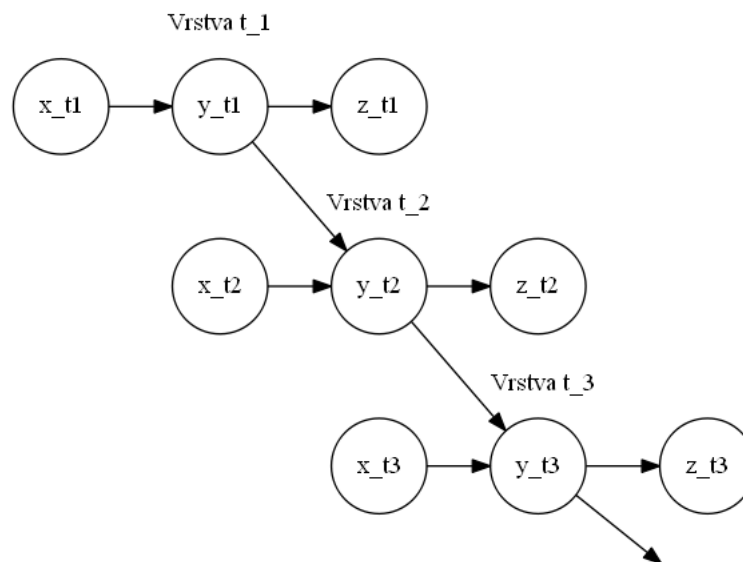
Rekurentné siete môžu obsahovať cykly a tak ich schopnosti sú výrazne vyššie. Avšak sú ťažšie na pochopenie a výpočet. Sú schopné krátkodobej pamäte, majú vnútorný stav a majú vyššiu modelovaciu schopnosť. Ukážka takejto topológie sa nachádza na obrázku.



Obr. 3.5: Príklad *rekurentnej* topológie

Problémom rekurentných sietí je, že obvykle bývajú implementované ako rekurentné siete rozložené do času tak aby z nich vznikli dopredné siete. To znamená, že hĺbka takto rozloženej siete je n -krát väčšia, kde n je počet prechodov do ktorých je sieť rozložená. Takto rozložená sieť je ukázaná na 3.6. Rozkladali sme sieť z 3.5 a to tak, že všetky rekurentné prepojenia vedú do druhého komponentu, ktorý

charakterizuje ďalší prechod siete. Vrstva t_n označuje n -tý prechod, neuróny x sú neuróny na vstupnej vrstve, neuróny y sú skrytou vrstvou a neuróny z tvoria výstupnú vrstvu.



Obr. 3.6: *Proces rozloženia neurónovej siete v čase*

Topológia siete definuje model a teda aj jej teoretické možnosti. Pokiaľ je topológia príliš jednoduchá nemusí byť schopná naučiť sa komplikovanejšie problémy. Na druhú stranu príliš komplikovaná topológia určite spomalí proces učenia a môže byť aj kontraproduktívna. Z tohto dôvodu je nutné čo najviac prispôbiť topológiu riešenému problému.

3.1.3 Učenie

Typické algoritmy učenia neurónových sietí sa dajú rozdeliť na učenie s učiteľom, učenie bez učiteľa a učenie s odmenou a trestom. Niektoré typy neurónových sietí sú špecificky postavené pre jeden typ učenia.

Učenie s učiteľom(angl. Supervised learning) sa snaží minimalizovať rozdiel

medzi dosiahnutým výstupom a požadovaným výstupom pre daný vstup. Z takejto definície vyplýva, že musia byť k dispozícii dáta a k nim patriace požadované výstupy. V prípade, že takéto dáta nie sú k dispozícii nie je možné učenie s učiteľom aplikovať.

V ideálnom prípade sa neurónová sieť naučí generalizovať a je schopná pre daný vstup vytvoriť správny výstup aj ak tento pár nebol v pôvodnom tréningovej dátovej sade.

Najznámejším algoritmom učenia s učiteľom je pravdepodobne backpropagation, ktorý kombinuje spätné šírenie informácie tak aby sa pre každý neurón dal aplikovať niektorá optimalizačná metóda, typicky gradient descent.

V niektorých prípadoch sa môže stať, že sa sieť naučí perfektne reagovať na tréningovú dátovú sadu ale pritom stratí schopnosť generalizovať, tento problém sa nazýva over-fitting.

Učenie bez učiteľa (angl. Unsupervised learning) je rodina metód, ktoré pracujú s neoznačenými dátami. To znamená, že nepotrebujeme poznať správny výstup pre každý vstup vopred. Väčšina z nich sa snaží nájsť v dátach štruktúru, model, vzory či iné dôležité vlastnosti.

Učenie s odmenou a trestom je akýsi hybrid, ktorý ako odpoveď na svoj výstup dostáva iba jednoduchú spätnú väzbu o kvalite jeho výsledku. Takáto neurónová sieť sa snaží dosiahnuť najlepších výsledkov a vyriešiť problém.

3.1.4 Hlboké neurónové siete

Hlavnou črtou hlbokých neurónových sietí je ich väčší počet vrstiev. Takto sú schopné reprezentovať komplikovanejší model s menším počtom neurónov. Tento efekt nastáva, pretože celkový počet neurónov potrebných na vytvorenie efektívneho modelu problému je nižší ak sa použije viac vrstiev [4].

Avšak hlboké neurónové siete sú veľmi náročné na tréningovanie. Väčšina metód učenia má veľké problémy s vyšším počtom vrstiev. Dôvodom prečo sa toto deje je viacero.

Jedným z nich je zvýšené riziko uviaznutia v slabom lokálnom optime. Tento problém je prepojený s náhodnou inicializáciou váh a je do určitej miery riešiteľný pred-tréningom neurónovej siete algoritmom učenia bez učiteľa. Avšak vážnosť

tohto problému sa zvyšuje s počtom vrstiev.

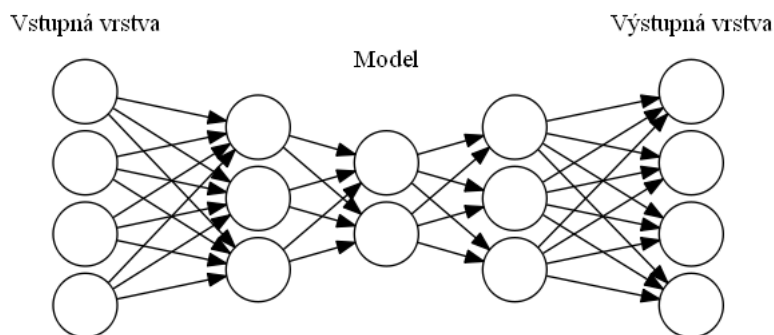
Ďalším vážnym problémom je nestabilné šírenie chybového signálu do hlbších vrstiev pri metódach využívajúcich gradient descent (napríklad algoritmus back-propagation). Signál v jednej vrstve je závislý na vrstve, ktorá sa spracovala pred ňou. Po prvých pár vrstvách môže signál narásť alebo sa zmenšiť do takých hodnôt, že učenie nie je efektívne v ďalších vrstvách. Tento problém sa nazýva Vanishing Gradient Problem.

Hlboké neurónové siete sa dajú tiež rozdeliť podľa spôsobu učenia. Autori článku [5] rozdeľujú takéto siete na hlboké siete určené pre učenie bez učiteľa, hlboké siete určené pre učenie bez učiteľa a hybridné prístupy, ktoré kombinujú tieto dva prístupy.

3.1.4.1 Autoenkóder

Tento typ neurónových sietí spadá pod unsupervised learning a ako taký nepatrí medzi hlboké neurónové siete. Avšak je veľmi častým komponentom v hlbokých neurónových sieťach.

Autoenkóder, ktorého topológia je znázornená na obrázku 3.7, sa snaží vyjadriť vstup nižším počtom hodnôt (neurónov) a následne sa pokúša takto redukovanú informáciu rekonštruovať. Takýmto spôsobom sa naučí porozumieť dátam a vyjadriť ich skrytý model. Vďaka tomu, že na výstupe očakáva rovnaké dáta ako na vstupe, nepotrebuje poznať správne výsledky pre vstupy.



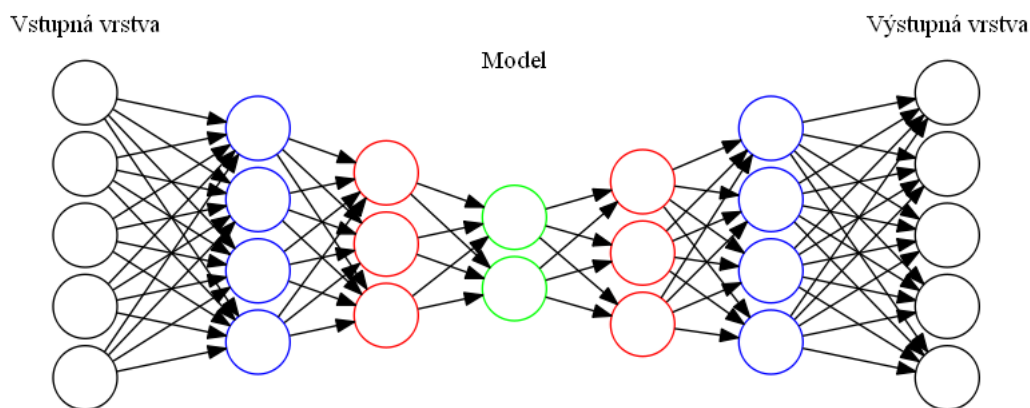
Obr. 3.7: Topológia neurónovej siete typu autoenkóder

V závislosti od charakteru vstupov treba určiť počet skrytých vrstiev a počet neurónov v nej, pretože autoenkóder musí byť schopný vyjadriť dostatočný model vstupných dát.

Aj siete, ktoré majú viac neurónov v skrytej vrstve sú použiteľné a môžu sa naučiť vhodný model. V takto postavenom autoenkóderi je však možné riziko naučenia sa identity. To znamená, že by sa vstupy bez akejkoľvek významnej transformácie dostali na výstupnú vrstvu. Existuje viacero metód ako zabrániť autoenkóderu naučiť sa identitu.

Hlbokou variantou autoenkóderov sú takzvané skladané autoenkóдеры (angl. Stacked Autoencoders). Pretože jeden autoenkóder je stále náchylný na bežné problémy súvisiace s vyšším počtom vrstiev, nie je účinné aby mal väčšie množstvo vrstiev.

Skladané autoenkóдеры, na obrázku 3.8, tento problém riešia rozdelením viacvrstvého autoenkódera na viac vnorených autoenkóderov. To znamená, že reprezentácia, ktorú sa naučí autoenkóder prvej úrovne je použitá ako vstup/výstup do autoenkódera druhej úrovne, ktorý sa znova snaží nájsť reprezentáciu vstupných dát.



Obr. 3.8: Skladaný autoenkóder, farbami sú vyznačené vnorené autoenkóдеры

Autoenkóдеры ako aj ich varianta pre hlboké učenie sú málokedy použité samostatne. Väčšinou sú použité na predspracovanie vstupov, ich transformáciu, odšumenie či zjednodušenie do jednoduchšej reprezentácie.

3.1.4.2 Konvolučné neurónové siete

Konvolučné siete sú inšpirované prirodzenými neurónovými sieťami, ktoré spracovávajú obraz. Skladajú sa z viacerých vrstiev, ktoré môžu byť konvolučné, plne prepojené, pooling vrstva, softmax a vrstva obsahujúca ReLu neuróny. Vhodnou kombináciou týchto vrstiev vznikne umelá neurónová sieť, ktorá vyniká pri spracovaní obrazu.

Konvolučná vrstva je rozdelená na viacero filtrov. Každý z týchto filtrov spracováva jeden súvislý región vstupných dát, tak že tieto regióny sa navzájom čiastočne prekrývajú. Takto sú získané vlastnosti spracovávaného obrázku.

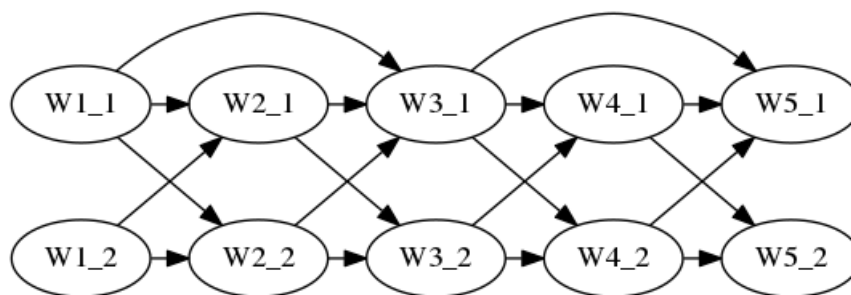
Pooling vrstvy sa snažia sa znížiť veľkosť reprezentácie a tým zároveň zvýšiť generalizáciu a rýchlosť učenia. Tieto vrstvy môžu byť aj pevne definované, aj môžu byť predmetom učenia. Existuje viacero spôsobov akým tieto vrstvy fungujú, najčastejším spôsobom býva max-pooling, ktorý rozdelí vstupnú vrstvu na regióny z ktorých vyberá tie najväčšie hodnoty.

Po sérii transformácií nasleduje posledná vrstva, ktorá je plne prepojená dopredná sieť, jej účelom je spracovať transformovaný vstup a dopracovať sa k finálnemu výsledku.

Takéto siete majú obvykle viacero vrstiev ale popritom sú jednoduchšie na učenie ako rovnako hlboké klasické dopredné siete. Existuje viacero teórií prečo lepšie zvládajú vyšší počet vrstiev.

3.1.4.3 Hlboké reziduálne siete

Tieto siete sa vyznačujú prepojeniami, ktoré neprepájajú dve po sebe nasledujúce vrstvy ale niektoré vrstvy preskočia. Takéto siete sú schopné väčších hĺbok ako normálne siete a lepšej konvergencie [12]. Pomocou týchto prepojení sú schopné udržiavať gradient a naučiť sa identitu, čo je vhodné ak je model komplikovanejší ako problém vyžaduje. Ukážka takejto siete je na obrázku 3.9. Jedná sa o 5-vrstvovú sieť s dvoma prepojeniami, ktoré preskakujú jednu vrstvu.



Obr. 3.9: Ukážka hlbkej reziduálnej siete

3.1.4.4 Pred-trénované neurónové siete

Ako bolo vyššie spomenuté, jedným riešením ako účinne učiť hlboké neurónové siete je ich pred-trénovanie, ktoré býva vykonané spravidla učením bez učiteľa. Na takto pred-trénované siete je potom možné aplikovať metódy učenie s učiteľom. Takto kombinovaný prístup je schopný učiť aj hlbokú neurónovú sieť.

Metódy bez učiteľa, ktoré sa bežne používajú sú autoenkóдеры, obmedzené boltzmannove stroje či iné. Pred-trénovanie nemusí byť vždy prospešné, v niektorých prípadoch môže takto naučená sieť podávať horšie výsledky ako bez pred-trénovania. Prečo je tomu tak nie je úplne známe a jedná sa o aktívnu oblasť výskumu.

Jednou teóriou je, že pred-trénovanie inicializuje neurónovú sieť do stavu, do ktorého by sa sieť nemohla dostať pomocou náhodnej inicializácie a následne aplikovaného učenia s učiteľom. Taktiež sa zdá, že množina stavov v, ktorých skončí neurónová sieť svoje učenie je podstatne menšia a viac sústredená. Druhou teóriou je, že niektoré vlastnosti vstupov, ktoré boli spoznané počas fázy učenia bez učiteľa, môžu byť použité vo fáze učenia s učiteľom. [14]

Pred-trénovanie síce hralo významnú úlohu v ranných hlbokých sieťach ale v dnešnej dobe sa významnejšie používa iba v úlohách, ktoré pracujú s prirodzeným jazykom.[14]

Kapitola 4

Použité technológie

V súčasnosti existuje viacero knižníc a aplikačných rámcov pre strojové učenie a neurónové siete či hlboké neurónové siete konkrétne. V dnešnej dobe sú všetky schopné využiť výkon grafických kariet či dokonca kombináciu CPU a GPU. Oboje spomenuté knižnice sú open-source a ich kód sa píše v jazyku python, tak aby bol čo najprehľadnejší. Kód v jazyku python je potom v oboch prípadoch spracovaný, skompilovaný a zoptimalizovaný tak aby bežal čo najrýchlejšie na platforme, ktorou môže byť GPU či CPU.

4.1 Theano

Knižnica Theano je primárne vyvíjaná skupinou pre strojové učenie na Montrealskej univerzite od roku 2007. Theano nie je limitovaná iba pre neurónové siete či strojové učenie ale je schopná akýchkoľvek matematických výpočtov.

Theano je založené na symbolických premenných a funkciách. To znamená, že bežná Theano premenná nemá hodnotu ale jedná sa iba o výraz pomocou, ktorého sa dá vytvoriť matematický systém. S týmito premennými sa pracuje pomocou funkcií, ktoré musia byť skompilované. Funkcia vyžaduje Theano premennú, ktorá predstavuje vstup a Theano premennú, ktorá predstavuje výstup. Vzťah medzi vstupnou a výstupnou premennou, ktorý bol pred deklarováním a skompilováním funkcie definovaný predstavuje proces výpočtu.

Tento vzťah môže byť definovaný podobne ako vzťah medzi akýmikoľvek inými

premennými, základné matematické operátory stále fungujú. Dodatočne Theano obsahuje veľké množstvo predpripravenej funkcionality a operácií. Theano poskytuje aj špeciálne zdieľané premenné, ktoré sú akýmsi hybridom medzi normálnymi a Theano premennými. Majú hodnotu, ktorá môže byť nastavená aj mimo Theano funkcie a zároveň sa táto hodnota zachováva medzi volaniami funkcie. Bežne sa používajú ako parametre modelu alebo je v nich uložená dátová sada.

Počas kompilovania je z týchto symbolických premenných a operácií vytvorený graf výpočtu, ktorý môže byť optimalizovaný. Vďaka tomuto grafu je Theano schopné automatických derivácií, čím sa uľahčuje práca používateľa ako aj výkon. Theano obsahuje viacero optimizérov a v prípade, že sú splnené niektoré požiadavky môže Theano kompilovať niektoré operácie do jazyka C alebo pracovať aj na grafickom akceleratori.

Po skompilovaní Theano funkcie môže byť volaná ako akákoľvek iná funkcia, ktorá na vstupe vyžaduje premenné kompatibilné s Theano premennými, ktoré boli pri deklarovaní použité ako vstup. Výstupom takejto funkcie bude hodnota kompatibilná s Theano premennou, ktorá bola definovaná ako výstup. Výpočet sa riadi vzťahom medzi týmito dvoma premennými.

Napriek tomu, že Theano obsahuje funkcie na výpočet niektorých typických funkcií cien, implementácia samotného optimizéra je v rukách používateľa.

4.2 Lasagne

Lasagne je vysoko úrovňová knižnica, ktorá pracuje nad Theanom. Jej úlohou je modularizovať a uľahčiť prácu s Theanom, obsahuje mnoho optimizérov a stavebných blokov, z ktorých sa dajú vybudovať najrôznejšie architektúry neurónových sietí. V tejto práci sme Lasagne použili iba ako zdroj implementácie optimizéra Adam, ktorý sme sa rozhodli použiť.

4.3 TensorFlow

TensorFlow, oficiálne vydaný v novembri 2015 je druhou generáciou systému pre strojové učenie pre projekt Google Brain. A napriek tomu, že je open-source je

stále vyvíjaný firmou Google.

Tento framework je založený na podobnej filozofii ako Theano. Tiež je založený na symbolických premenných a funkciách. Avšak obsahuje aj funkcie s vyššou abstrakciou ako v prípade Theana, kde tie isté funkcie musia byť implementované ako kombinácie používateľom alebo dodané knižnicou ako je napríklad Lasagne.

Oproti Theanu je jeho výhodou lepšia prezentácia a odlaďovanie chýb čo je aj vďaka jeho robustnému, vstavanému nástroju pre vizualizácie, ktorý sa nazýva TensorBoard. Pomocou neho je taktiež jednoduchšie nasadiť natrénovaný model. Jeho kód sa kompiluje rýchlejšie ako v prípade Theana ale zároveň sa vykonáva pomalšie, hoci to sa môže počas vývoja zmeniť.

Kapitola 5

Návrh a implementácia riešenia

Pre túto prácu sme sa rozhodli použiť architektúru WaveNet, ktorá vznikla za účelom generovania hudby, reči a ako text-to-speech systém ako výskumný projekt skupiny DeepMind [22].

Jedná sa o hlboké učenie založené na špeciálnom type konvolučných sietí, ktoré pracuje priamo nad surovými audio dátami.

V tejto kapitole popíšeme komponenty z ktorých sa skladá WaveNet sieť, menovite sa jedná o kauzálne konvolučné siete, 1x1 konvolúcie a dilatované konvolučné siete. Potom popíšeme samotnú architektúru a spôsob ako je implementovaná v tejto práci.

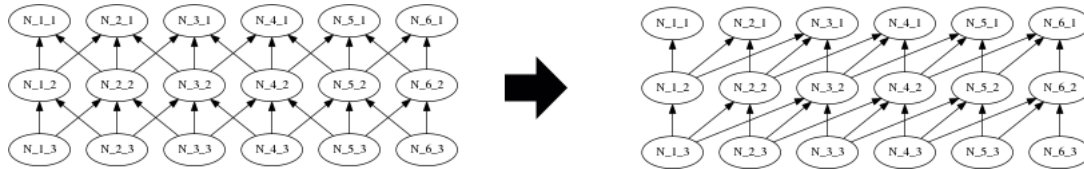
5.1 Kauzálne konvolučné siete

Kauzálne konvolučné siete boli vyvinuté z maskovaných konvolúcií, ktoré boli použité v PixelRNN. Maskovaná konvolúcia berie do úvahy iba vstupy, ktoré prejdú cez masku, týmto sa dá prinútiť sieť aby brala do úvahy iba niektoré vstupy.

PixelRNN sieť bola zameraná na prácu s obrázkami, a teda s dvoj dimenzióнными dátami či signálom. Naproti tomu WaveNet pracuje nad zvukom a teda jednorozmerným signálom, ktorý sa mení v čase. Preto kauzálne konvolúcie tiež fungujú nad signálom, ktorý sa mení v čase a v prípade WaveNetu sa jedná o jednorozmerný signál.

Kľúčový rozdiel tkvie vo výraze kauzálne, ktorý značí, že súčasná situácia

vznikla z nejakého minulého stavu. To znamená, že pre čas t_n iba stavy v časoch t_m kde m je menšie ako n majú vplyv na výsledok. Rozdiel medzi typickou konvolúciou a kauzálnou konvolúciou je znázornený na obrázku 5.1.



Obr. 5.1: Rozdiel medzi normálnou a kauzálnou konvolúciou

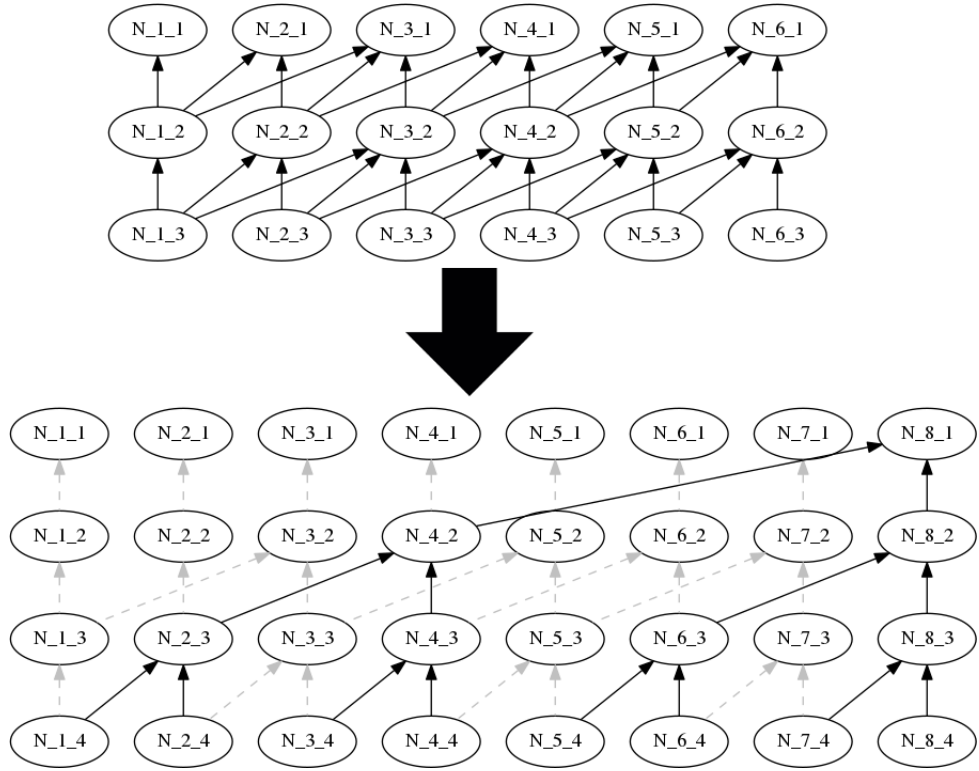
Takto štruktúrovaná sieť je veľmi výhodná práve pri predikčných úlohách. Keby sme použili typickú konvolúciu tak pre vypočítanie predpovede by sme vyžadovali dáta v čase, ktoré sa snažíme predpovedať. Kauzálna konvolúcia zabezpečí, že model je trénovaný a vyžaduje iba na dátach, ktoré má k dispozícii.

5.2 Dilatované konvolučné siete

Problémom kauzálnych konvolúcií je, že zorné pole narastá lineárne v závislosti od počtu vrstiev a veľkosti filtra. V tomto kontexte pod zorným poľom rozumieme, rozsah hodnôt z minulosti, ktoré sú brané do úvahy pri predpovedi. Je zjavné, že pre dostatočne presnú predpoveď potrebujeme dostatočne veľké zorné pole, avšak vo väčšine prípadov by ideálne zorné pole spôsobilo príliš veľký nárast vrstiev. Veľkosť zorného pola neurónovej siete je závislá podľa vzťahu 5.1, kde $v_z h$ je veľkosť zorného poľa, l je počet vrstiev a v_f je veľkosť filtra.

$$v_z h = l + v_f - 1 \quad (5.1)$$

V dilatovanej konvolúcii je iba každý n -tý element násobený s filtrom. Takže ak máme prvky n_i kde i je z intervalu $< 1, 5 >$, filter veľkosti 3 a dilatáciu veľkosti 2 tak iba prvky 1, 3, 5 budú násobené hodnotami filtra. Ukážka takejto konvolúcie je na obrázku 5.2.



Obr. 5.2: Rozdiel medzi kauzálnou a dilatovanou konvolúciou

To v praxi znamená, že niektoré hodnoty ignorujeme a jedna vrstva vie pokryť n -krát väčšie zorné pole kde n je veľkosť dilatácie. Navyše ak budeme skladať viacero vrstiev kde každá má dilatáciu veľkosti n tak veľkosť zorného poľa sa dá vypočítať podľa rovnice 5.2. Či v prípade, že dilatácie sú rôzne, zapísané v poli *dils* jedná sa o rovnicu 5.3. V oboch prípadoch, sú veličiny označené rovnako ako v 5.1.

$$v_z h = l * n * (v_f - 1) + 1 \quad (5.2)$$

$$v_z h = \sum dils_i * (v_f - 1) + 1 \quad (5.3)$$

Z týchto rovníc vyplýva, že rýchlosť rastu zorného poľa závisí aj od veľkosti dilatácií. Narozdiel od počtu vrstiev či veľkosti filtra, veľkosť dilatácie je menej problematická z hľadiska výpočtovej náročnosti a zložitosti učenia.

5.3 1x1 konvolúcie

Jedná sa o špeciálny typ konvolúcií, ktorý má filter veľkosti 1x1, to znamená, že šírka a výška sa nemení. Do vzťahu, ktorý určuje výstup na úrovni konkrétneho elementu, vstupuje iba informácia z toho istého elementu na vstupe. Inými slovami, susedné elementy na výstup nemajú vplyv.

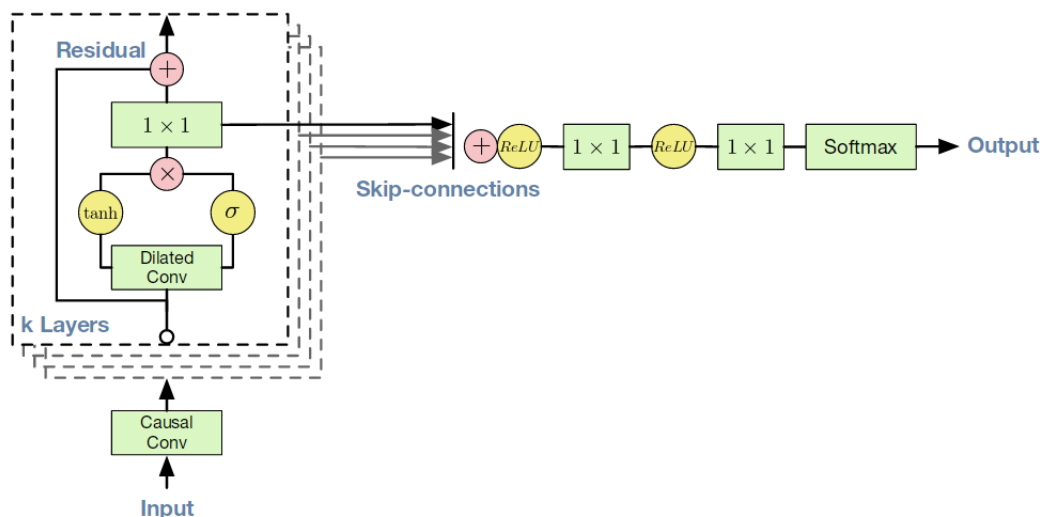
Avšak tieto konvolúcie nie sú iba násobenie maticou po elementoch. Vo väčšine prípadov na vstupe a výstupe nie je iba jeden kanál. A tak 1x1 konvolúcie fungujú ako plne prepojená vrstva nad jednotlivými elementami, kde vstupný vektor pozostáva z kanálov elementu. Parametre sú zdieľané medzi všetkými elementami.

5.4 WaveNet architektúra

Väčšina WaveNet siete pozostáva z kauzálnych, dilatovaných konvolúcií, úlohou menšej časti za konvolúciami je spracovať výstupy konvolúcií a získať konečný výstup.

Súčasťou konvolúcií sú špeciálne neuróny s bránou. Táto je zložená z dvoch konvolučných operácií, ktoré sú aplikované nad rovnakým vstupom ale s rozdielnymi filtermi. Filter navyše funguje ako brána. Na výstup konvolúcie s bránou je následne aplikovaná sigmoidálna funkcia a podobne je na konvolúciu s filtrom aplikovaný hyperbolický tangens. Výstup oboch aktivačných funkcií je spolu násobený po elementoch. Túto funkciu, v matematickom zápise je možné vidieť na výraze 5.4. Kde W_f je konvolučný filter, W_g je filter brány, operácia $\odot$ značí násobenie po elementoch a operácia $*$ značí konvolúciu.

$$output = \tanh(W_f * x) \odot \sigma(W_g * x) \quad (5.4)$$



Obr. 5.3: WaveNet architektúra; prevzaté z [22].

Na obrázku 5.4 je znázornená architektúra WaveNet. Nákres je logicky rozdelený na prvú, kauzálnu konvolúciu, viacero blokov a spracovateľskú časť. Prvá vrstva pozostáva z jednoduchkej kauzálnej konvolúcie.

Jednotlivé bloky predstavujú dilatované, kauzálne konvolučné vrstvy, kde okrem neurónov s bránou, sú aj 1×1 konvolúcie a dodatočné prepojenia.

Architektúra obsahuje dva typy preskakujúcich prepojení, ktoré sú typické pre reziduálne siete 3.1.4.3. Preskakujúce prepojenia v konvolučnej časti (na obrázku popísané ako Residual), prenášajú vstup z vrstvy l_i , ktorý sa sčíta s výstupom tej istej vrstvy a tento súčet je použitý ako vstup do vrstvy l_{i+1} . Druhý typ preskakujúcich prepojení v tejto architektúre, prenáša výstup, všetkých konvolučných vrstiev do spracovateľskej časti. Tieto prepojenia sú sčítané do jediného vektora, ktorý po aplikovaní napravenej lineárnej aktivačnej funkcie slúži ako vstup do spracovateľskej časti. Oba typy prepojení majú svoje váhy.

WaveNet obsahuje v spracovateľskej časti dve 1×1 konvolúcie, medzi ktorými je znova aplikovaná napravená lineárna aktivačná funkcia. Následne je aplikovaná softmax vrstva, ktorej výstup je výstupom neurónovej siete.

Narozdiel od zobrazenej architektúry na obrázku 5.4, kde je zobrazená jedna 1x1 konvolúcia v bloku, wavenet obsahuje dve rozdielne 1x1 konvolúcie v jednom bloku. Podľa zobrazeného diagramu ide na reziduálne a preskakujúce prepojenie tá istá hodnota, to však nie je pravda. Pretože v bloku sú dve 1x1 konvolúcie, jedna počíta výstup pre reziduálne prepojenie a jedna počíta výstup pre preskakujúce prepojenie.

Dôležitou súčasťou architektúry Wavenet je relatívne vysoký počet kanálov. Kauzálna konvolúcia na začiatku, ešte pred blokmi, prepočíta vstup tak aby mal správny počet kanálov. Ten je rozdielny v blokoch a v spracovateľskej časti, konverzia prebieha hlavne v 1x1 konvolúciách, ktorej výstup ide do preskakujúceho prepojenia.

Keďže výstupom je pred výstupom softmax vrstva, výstupom neurónovej siete je pravdepodobnostná distribúcia. Ak máme na výstupe, n neurónov tak rozsah predpovedaných hodnôt je rozdelený na n intervalov a neurónová sieť predpovedá s akou pravdepodobnosťou bude predikovaná hodnota spadať do jednotlivých intervalov. Toto je obzvlášť problematické pretože rozsah výstupnej vrstvy limituje maximálnu možnú presnosť siete.

V niektorých implementáciach Wavenetu je použitá rýchlejšia forma generovania. Keďže predikujeme iba jeden krok do budúcnosti, výsledok ktorého použijeme pre ďalší krok, nie je nutné rátať predikcie pre veľké množstvo krokov. Rýchlejšia varianta je rátať výsledok iba pre jeden krok a medzivýsledky si ukladať. Avšak túto techniku sme neimplementovali.

5.4.1 WaveNet implementovaný v tejto práci

Theano má implementovanú konvolučnú vrstvu, ktorú stačí zavolať, avšak dilatované konvolúcie boli implementované až vo verzií 0.9.0. Do rozhrania metódy *conv2d* je pridaný voliteľný parameter, ktorý určuje veľkosť dilatácie. Kauzálna konvolúcia sme implementovali pomocou bežných konvolúcií, ktorých výsledok sme orezali.

Nie všetky časti Wavenetu sme implementovali, menovite sa jedná o podmienenie a kontextové spracovanie (angl. context stacks). Tieto funkcie sú slabšie dokumentované a nie sú potrebné pre naše účely.

Pretože na vstup siete dávame aj dáta o čase a poveternostných podmienkach, ktoré sú z viacerých staníc. Umiestnili sme pred prvú kauzálnu konvolúciu, ešte jednu vrstvu, ktorá sa má naučiť zjednotiť dáta z viacerých staníc do jediného tenzoru. Tenzory obsahujúce časové dáta a poveternostné dáta sú pripojené k historickým dátam ako dodatočné kanály. Tak, že na vstupe do prvej kauzálnej konvolúcie je tenzor, ktorého jeden kanál predstavuje historické dáta, dva kanály predstavujú časové zložky a jeden kanál pre každý vybraný poveternostný atribút. Neprekvapivo, veľký počet vstupných kanálov vyžaduje dlhší čas na učenie a komplexnejší model, preto sme boli nútení niektoré poveternostné atribúty vyradiť a použiť iba dve časové zložky s periódou jedného týždňa.

Kvôli problémom s numerickou stabilitou v softmax vrstve sme boli nútení umiestniť pred ňu dodatočnú normalizačnú vrstvu, bez nej sieť nebola schopná fungovania. Zdá sa, že tento problém je súčasťou použitej verzie Theana.

Aby sme boli schopní predikovať na viac ako jeden krok dopredu, najpravdepodobnejšiu hodnotu z výstupu pripojíme na vstup v ďalšej iterácii. Tento proces je použitý iba pri generovaní a nie pri tréningu.

Aby sme utlmili nárast chyby pri takomto opakovanom reťazení, pridávame pred kauzálnou vrstvou do historickej časti dát náhodný šum. Experimentovali sme aj s gaussovým aj uniformným šumom. Veľkosť šumu je závislá od poslednej validačnej chyby. Dopadom šumu je síce o niečo menšia presnosť pri predpovedi na jeden krok dopredu ale zároveň aj väčšia pravdepodobnosť pri predpovedaní na väčší časový úsek.

Okrem týchto zmien sme experimentovali aj s predpovedaním reálnej hodnoty zmeny, kde výstupom je jediná hodnota a nie pravdepodobnostná distribúcia vytvorená softmax vrstvou. Takýto spôsob preukázal horšie výsledky a preto sme ho ďalej neskúmali.

Tiež sme skúšali znížiť vplyv veľkosti výstupnej vrstvy na presnosť výsledkov pomocou predpovedania rozdielu medzi vzorkami. Tento prístup sme po čase opustili, pretože zmena spotreby energie tvorila náročnejšie časové rady na predikciu.

5.4.1.1 Predspracovanie dát

Dáta sú načítavané z viacerých súborov .csv, po načítaní sú normalizované a kvantifikované tak aby sa ich hodnoty rovnali hodnotám, ktoré môže sieť predpovedať. Tento postup má za následok, že ignorujeme chyby spôsobené malou veľkosťou výstupu. Avšak pri experimentovaní sme neboli schopní pozorovať akékoľvek zlepšenie, takže prínos tohto procesu je pochybný.

Dáta sú rozdelené na testovacie, validačné a trénovacie. Pomery tohto rozdelenia je možné meniť podľa potreby a vo väčšine našich experimentov boli nastavené nasledovne: 80% trénovacích dát, 10% validačných dát a 10% testovacích dát.

Hoci Wavenet môže používať dávky, nie je to nutné, pretože počas tréningu sa ráta predikcia pre každý krok zvlášť, čo funguje podobne ako dávkovanie. Takže namiesto dávok rozdeľujeme dáta na akési sekcie, každá z nich má nastavenú dĺžku 12000 vzoriek.

Dáta sú najprv rozdelené na validačno-trénovacie dáta a testovacie, ktoré boli odobraté od konca a teda sú najnovšie. Validačno-trénovacie dáta sú rozdelené do sekcií s polovičným prekryvom, ktoré sú následne rovnomerne rozdelené podľa pomeru do trénovacích a validačných. Trénovacie sekcie sú premiešané a môžu byť niekoľkokrát rozmnožené kopírovaním, keby to bolo nutné. Testovacie dáta nie sú rozdelené do sekcií ale sú v tvare jedného vektora. Snažili sme sa z testovacej množiny eliminovať dobu vianočných sviatkov a nového roka, kvôli ich anomálnemu charakteru.

Všetky operácie sa vykonávajú rovnako nad všetkými zložkami dát, t.j historické, poveternostné a časové tak aby po spracovaní dát boli stále elementy z toho istého časového úseku spolu. Rozdielom je iba dimenzionalita dát, ktorá v prípade času ráta s dvoma zložkami (sínusová a kosínusová) a v prípade poveternostných dát rátame s väčším počtom staníc a atribútov.

Hoci nepoužívame dávky a pracujeme nad jednorozmernými dátami, operácie konvolúcií v Theane stále vyžadujú štvorrozmerné tenzory. Preto sú všetky dáta na vstupe transformované tak aby ich sieť prijala.

Pretože prenášanie dát medzi operačnou pamäťou počítača a pamäťou grafickej karty, je časovo náročná operácia, všetky dáta s ktorými sa pracuje sú permanentne uložené v pamäti grafickej karty.

Kapitola 6

Dosiahnuté výsledky

Testovanie prebiehalo na viacerých dátových sadách, ktoré sa líšia polohou, časom a geografickým rozsahom. Vo všetkých prípadoch sme otestovali presnosť natrénovaného modelu na jeden krok, jeden deň, tri dni, jeden týždeň a 29 dní. Presnosť je rátaná pomocou MAPE metriky, ktorá je vypočítaná pre strednú hodnotu najpravdepodobnejšieho intervalu z pravdepodobnostnej distribúcie. Pre všetky experimenty sme použili rovnaký zdroj poveternostných dát, dátová sada MERRA2, konkrétne jeho produkt s potrebnými geografickými súradnicami, dátovým rozsahom a intervalom.¹

Dosiahnutá presnosť na všetkých dátových sadách a všetkých dĺžkach predpovede je na tabuľke 6.1. Môžeme pozorovať postupné zhoršovanie presnosti pri zväčšujúcom sa predikčnom okne.

Tabuľka 6.1: *Tabuľka MAPE hodnôt*

dáta	1 krok	1 deň	3 dni	7 dní	29 dní
Belgicko	1.22	3.45	3.52	4.20	5.02
Bratislava	1.11	1.79	1.82	2.24	2.53
New York	1.7	4.14	6.97	10.29	14.75

¹<http://www.soda-pro.com/sk/web-services/meteo-data/merra>

6.1 Dátová sada 1: Belgicko

Táto dátová sada pochádza od belgického distribútora elektrickej energie² a obsahuje celkovú spotrebu v rámci jeho siete. Táto dátová sada je dostupná od roku 2005 po súčasnosť v 15 minútových intervaloch. Poveternostné dáta sú odobrané v 6 bodoch.

Testovacia časť dát je z rozsahu 11 október 2008 po 11 december toho istého roku. Dobu vianočných sviatkov a koniec roka sme náročky odstránili kvôli ich anomálnemu charakteru. Vo všetkých nasledujúcich ukážkach predstavuje modrá farba reálne hodnoty a zvyšné farby predstavujú jednotlivé predpovede. Všetky ukážky sú z toho istého obdobia.

Hyperparametre a iné vlastnosti modelu použitého v tomto experimente boli:

- Počet dilatovaných vrstiev: 10
- Veľkosť výstupnej vrstvy: 256
- Veľkosť vnútorných kanálov: 128
- Najlepší model nájdený na iterácií: 30000

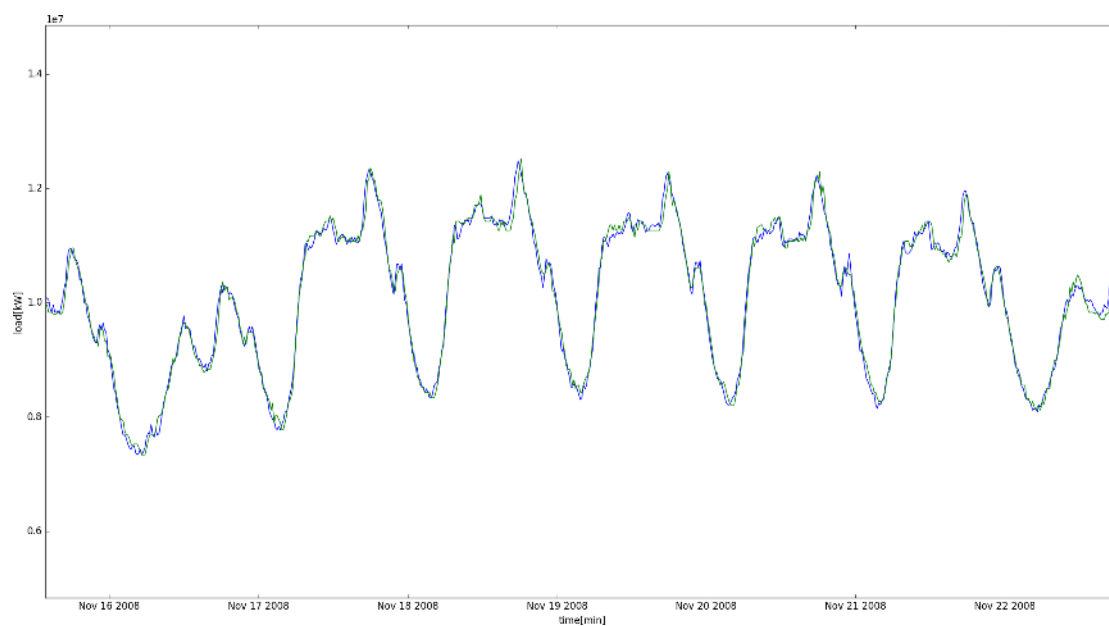
Výsledky na jeden krok dopredu, vyobrazené na obrázku 6.1, vykazujú veľmi vysokú presnosť, na mnohých miestach sa tieto krivky dokonca prekrývajú. Avšak takáto predikcia nie je príliš náročná a aj jej použitie je omedzené.

Predpovedanie na jeden deň dopredu má širšie využitie, je zobrazené na obrázku 6.2. Tu už pozorujeme jasný odklon od reálnych hodnôt. Je to zapríčinené spôsobom ako rátame predikcie na dlhšiu dobu. Najpravdepodobnejšia hodnota jedného kroku sa znovu použije ako vstup pri ďalšom kroku. Lenže už tento prvý výsledok má určitú chybu, ktorá následne rastie.

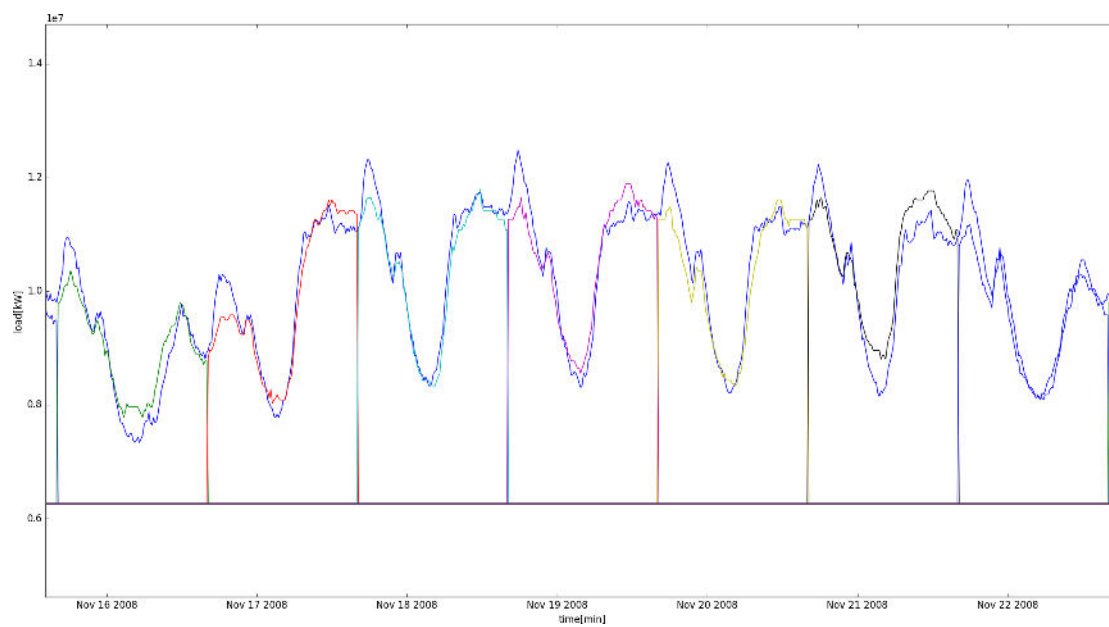
Trojdňová predpoveď je ukázaná na obrázku 6.3. Zaujímavosťou v tomto prípade je, že niektoré predpovede sú úplne rovnaké ako predpoveď na 1 deň dopredu aj keď tieto sa nenachádzajú v prvom dni trojdňovej predpovede, ktoré by mali byť rovnaké. Na grafe je vidno aj to, že v niektorých miestach sa ukazuje 3 dňová predpoveď presnejšia.

Hoci celková presnosť znova viditeľne poklesla na týždňovej predpovedi zobrazenej na obrázku 6.4, stále môžeme pozorovať jasné podobnosti s predpovedaniami

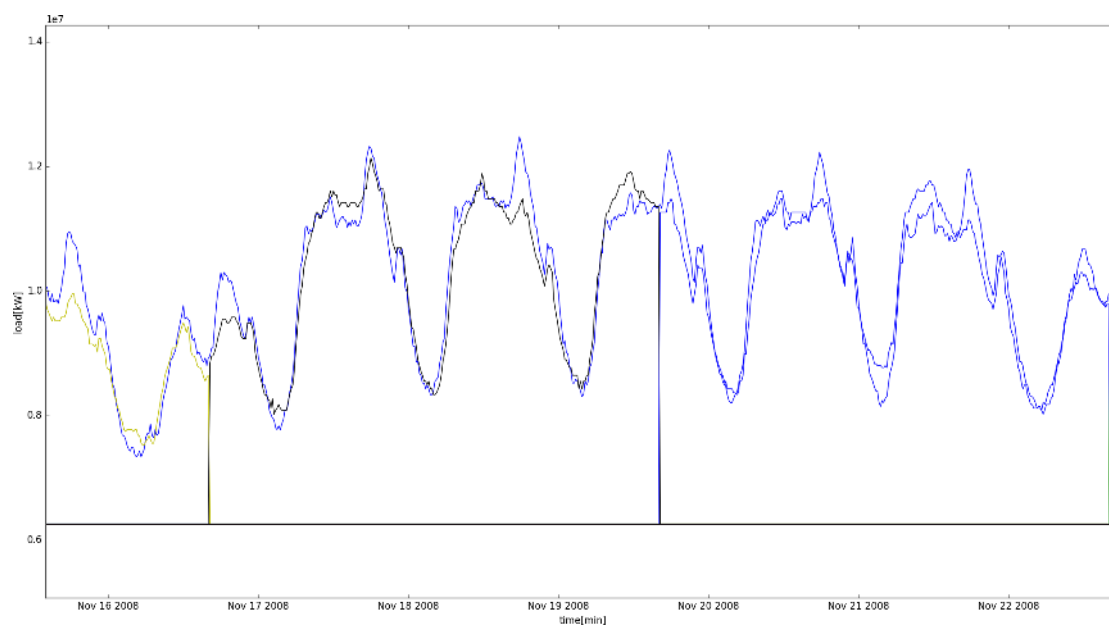
²<http://www.elia.be/en/grid-data/data-download>



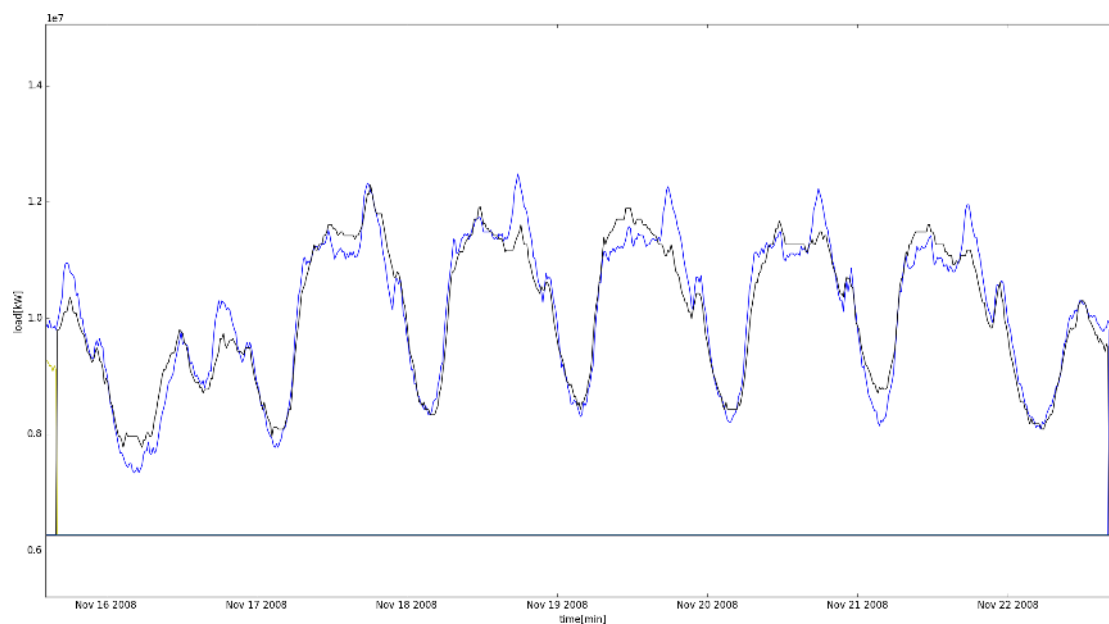
Obr. 6.1: Detail predpovede na 1 krok dopredu, Belgické dáta



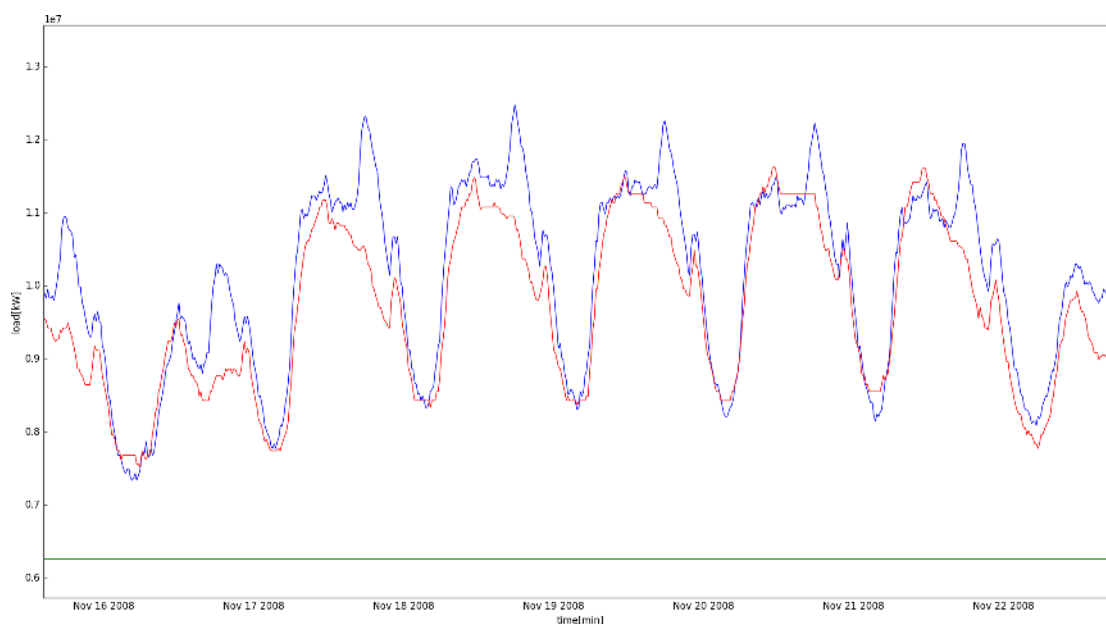
Obr. 6.2: Detail predpovede na 1 deň dopredu, Belgické dáta



Obr. 6.3: Detail predpovede na 3 dni dopredu, Belgické dáta



Obr. 6.4: Detail predpovede na 7 dní dopredu, Belgické dáta



Obr. 6.5: Detail predpovede na 29 dní dopredu, Belgické dáta

na nižších časových úsekoch.

Táto dlhodobá predpoveď, ktorú je vidno na obrázku 6.5 sa už jasne odkláňa od reálnych dát. Pravdepodobne je to zapríčinené postupným odklonom od reálnych dát. Tento detail je v strede týchto 29 dní a teda mal pred sebou niekoľko týždňov počas ktorých sa predpovedané dáta postupne odklňali od reálnych hodnôt.

Aj napriek tomuto vysvetleniu je zjavný pokles v kvalite výsledkov medzi týždňovou a 29 dňovou predikciou. Dôležitým zasahujúcim faktorom je pravdepodobne veľkosť zorného poľa siete. Keďže sme použili 10 vrstiev, veľkosť poľa je 1025 vzoriek, čo predstavuje 10 a pol dňa. Teda aj pri poslednom kroku týždňovej predpovedi, neurónová sieť vidí 3,5 dňa reálnych historických hodnôt. Pravdepodobne práve tieto reálne dáta pomáhajú ukotviť predpovede. V prípade, že predikčné okno je väčšie ako zorné pole siete, ako tomu je pri 29 dňovej predpovedi, neurónová sieť sa musí spoliehať iba na sebou predpovedané hodnoty.

6.2 Dátová sada 2: Bratislava

Jedná sa o produkt z dátovej sady, ktorou disponuje fakulta. Obsahuje spotreby zo slovenských miest medzi 1.7.2013 a 16.2.2015 v 15 minútových intervaloch. Pre tieto experimenty sme použili iba dáta s Bratislavy. Poveternostné dáta sú brané z 5 bodov, ktoré sú rovnomerne rozložené v Bratislave.

Testovacia množina je z intervalu 7.1.2015 do 16.2.2015, jej veľkosť je znížená kvôli odstráneniu vianočných sviatkov, konca a začiatku roka, pretože tieto obdobia sa vyznačujú anomálnym charakterom. Vo všetkých nasledujúcich ukážkach predstavuje modrá farba reálne hodnoty a zvyšné farby predstavujú jednotlivé predpovede. Všetky ukážky okrem poslednej sú z toho istého obdobia.

Hyperparametre a iné vlastnosti modelu použitého v tomto experimente boli:

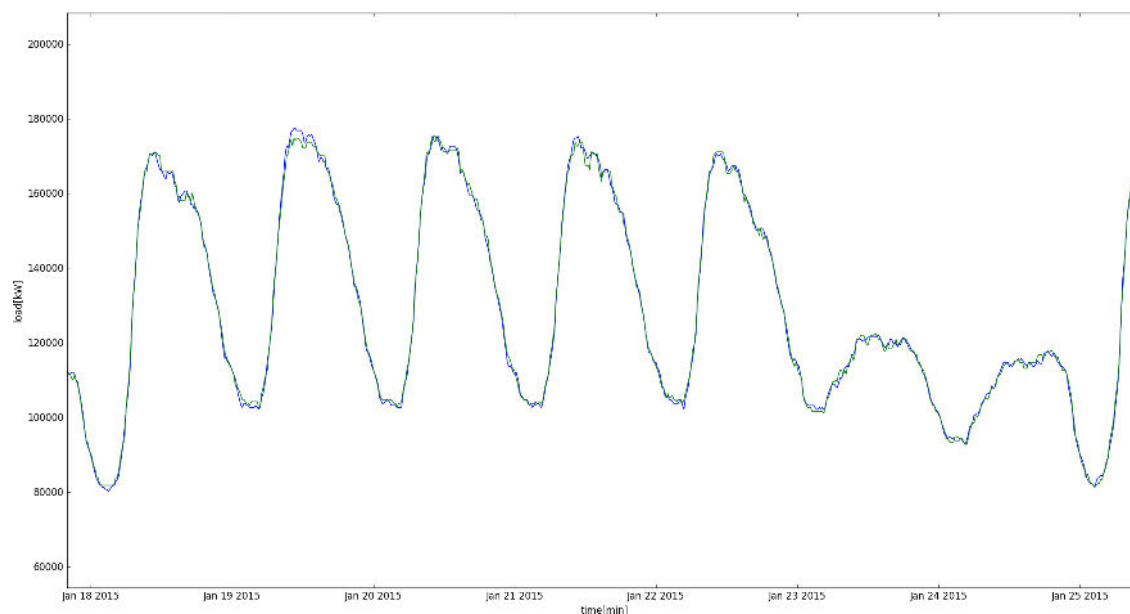
- Počet dilatovaných vrstiev: 10
- Veľkosť výstupnej vrstvy: 256
- Veľkosť vnútorných kanálov: 128
- Najlepší model nájdený na iterácií: 37000

Podobne ako v prípade predpovedí pre Belgicko aj v tomto prípade môžeme pozorovať na obrázku 6.6, že výsledky sú skoro úplne totožné s reálnymi hodnotami. Avšak znova treba podotknúť, že toto nie je príliš náročná úloha.

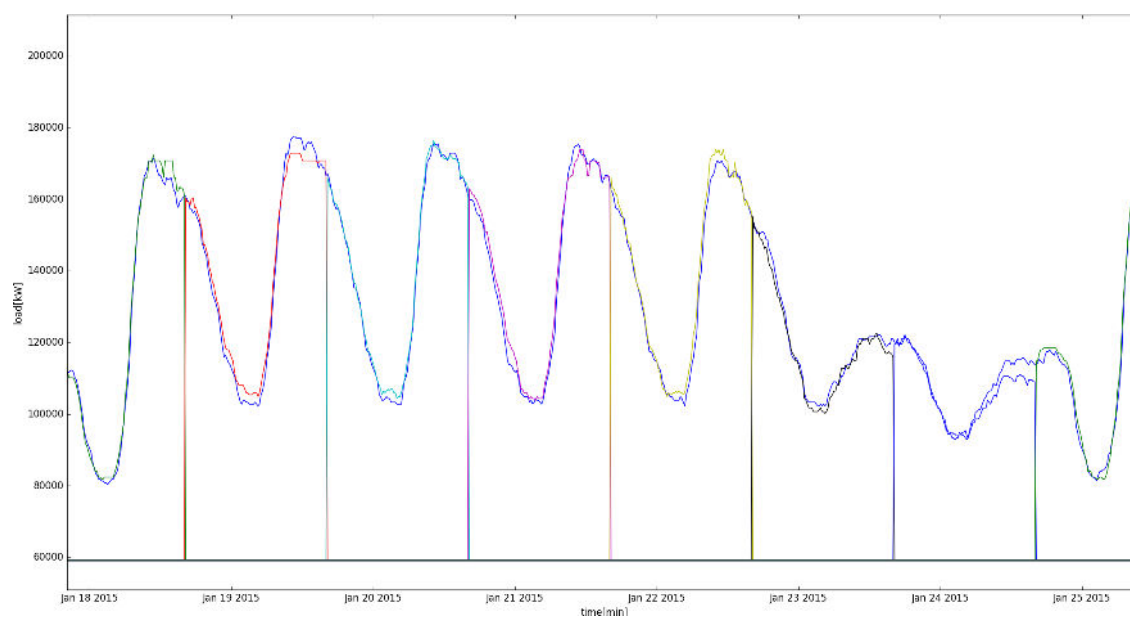
Na obrázkoch 6.7 a 6.8 vidíme výsledky pre 1 dňovú predikciu a pre 3 dňovú predikciu. Znova môžeme pozorovať isté podobnosti avšak sú menej výrazné. Zaujímavou črtou je výrazná odchýlka počas stredy, tento odklon nebol prítomný ak sme predikciu začínali na inom mieste, takže môžeme usúdiť, že presnosť predikcie je závislá od štartovnej pozície.

Týždňová predpoveď na obrázku 6.9 si zachováva svoju presnosť veľmi dobre. Znova je možné pozorovať veľkú podobnosť s predchádzajúcimi prípadmi.

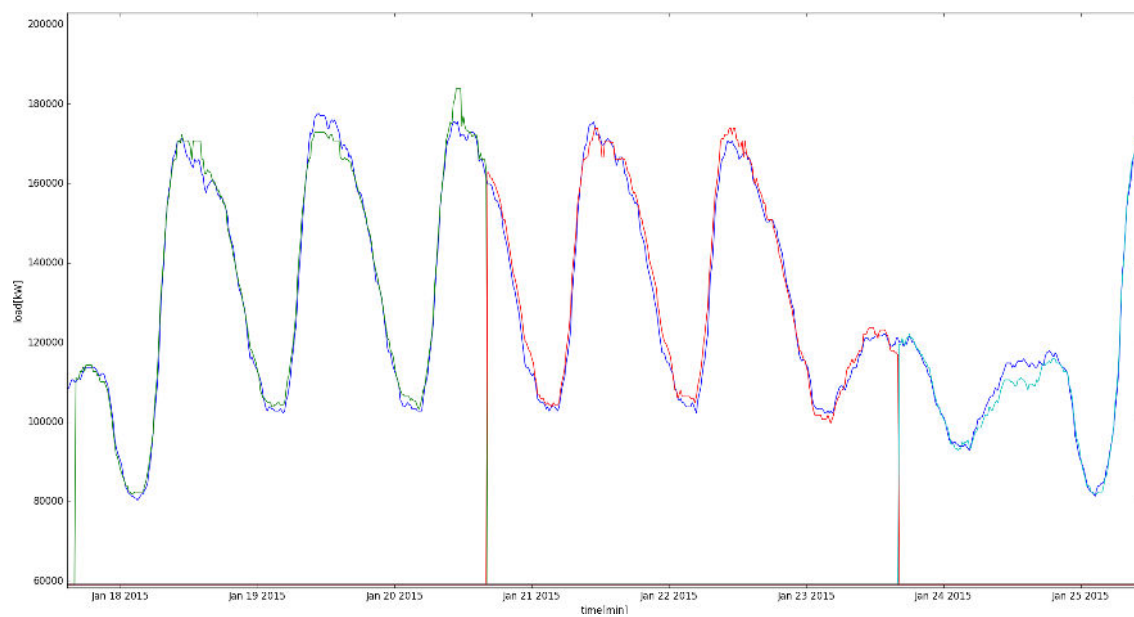
Zobrazený detail pre 29 dňovú predikciu, na obrázku 6.10, je z nasledujúceho týždňa. Pretože zobrazený týždeň v predchádzajúcich prípadoch bol prvý, jeho výsledky by boli totožné s tými na obrázku 6.9. Tento týždeň je zaujímavý práve preto že v jeho strede dojde k tomu, že celé zorné pole siete je tvorené predikovanými hodnotami. Práve v tejto dobe dochádza k výraznejšiemu odklonu.



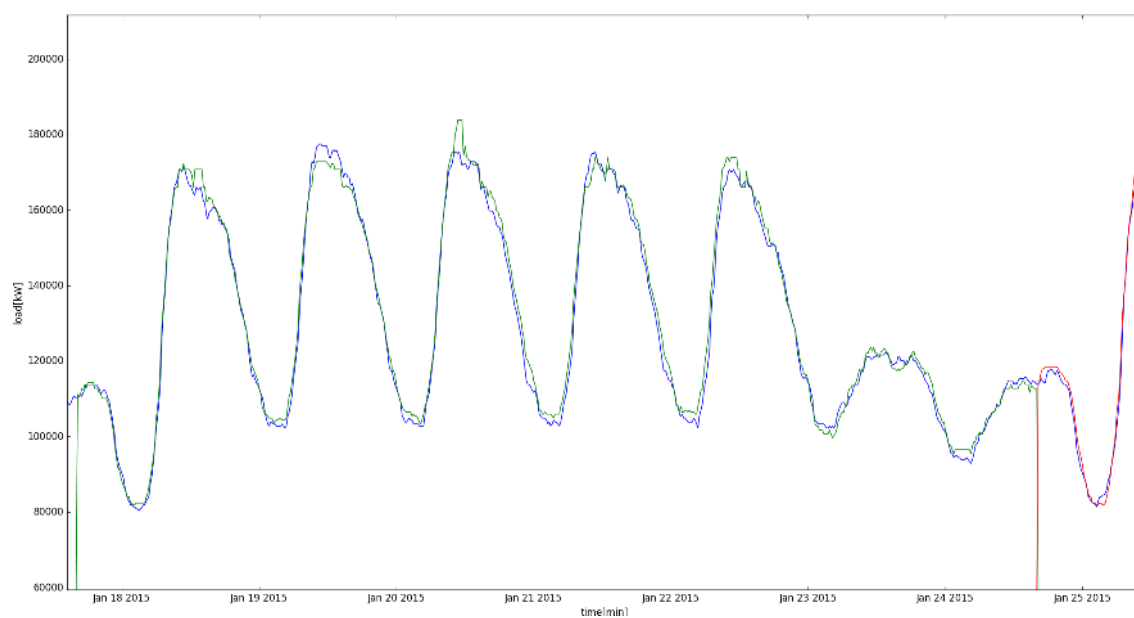
Obr. 6.6: Detail predpovede na 1 krok dopredu, Bratislavské dáta



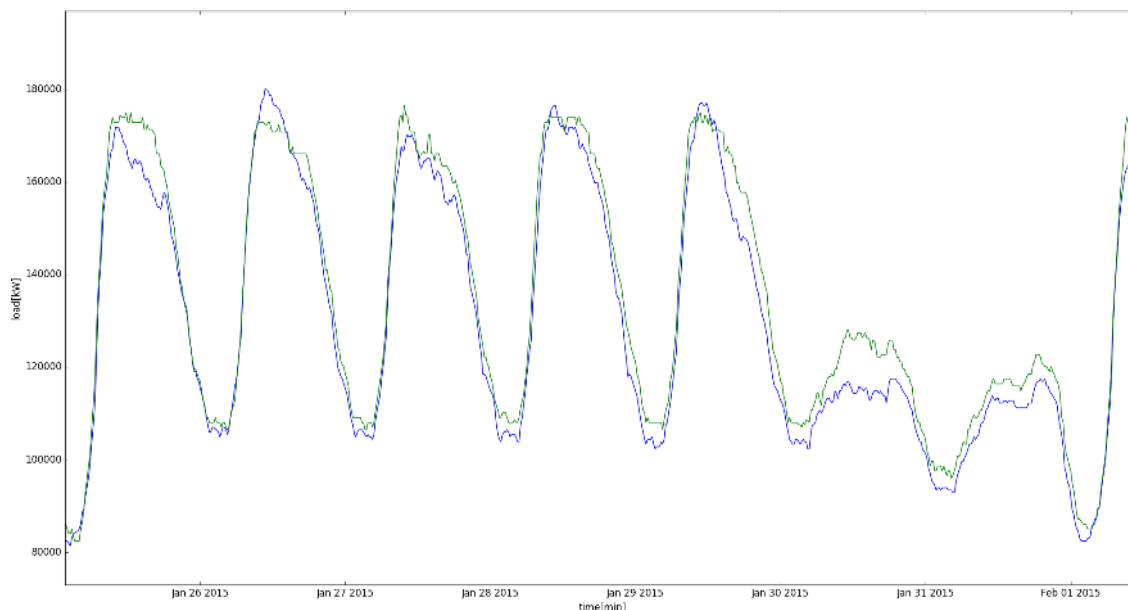
Obr. 6.7: Detail predpovede na 1 deň dopredu, Bratislavské dáta



Obr. 6.8: Detail predpovede na 3 dni dopredu, Bratislavské dáta



Obr. 6.9: Detail predpovede na 7 dní dopredu, Bratislavské dáta



Obr. 6.10: Detail predpovede na 29 dní dopredu, Bratislavské dáta

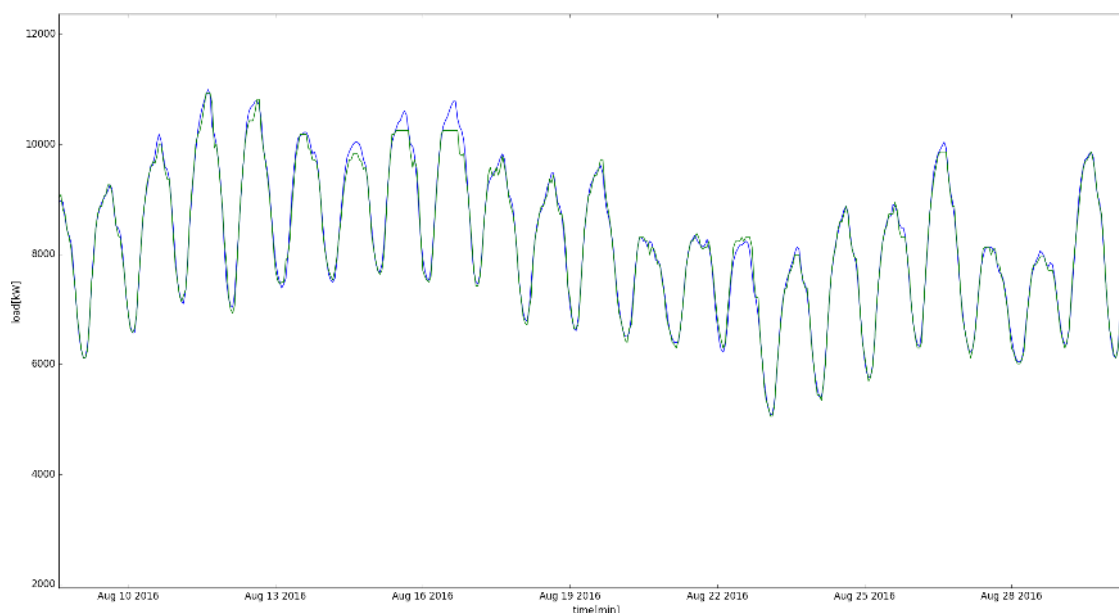
6.3 Dátová sada 3: New York

Táto dátová sada pochádza od distribútora elektrickej energie v americkom štáte New York³. Konkrétnym použitým produktom je hodinová spotreba mesta New York počas rokov 2009 až 2016.

Poveternostné dáta obsahujú 4 body, ktoré patria jednotlivým štrtiarom mesta New York.

Testovacia množina obsahuje dáta z obdobia medzi augustom a decembrom 2016, s odstránenými poslednými tromi týždňami roka kvôli vianočným sviatkom a koncu roka. Vo všetkých nasledujúcich ukážkach predstavuje modrá farba reálne hodnoty a zvyšné farby predstavujú jednotlivé predpovede. Všetky ukážky okrem poslednej sú z toho istého obdobia.

³http://www.nyiso.com/public/markets_operations/market_data/load_data/index.jsp



Obr. 6.11: Detail predpovede na 1 krok dopredu, New Yorkské dáta, august

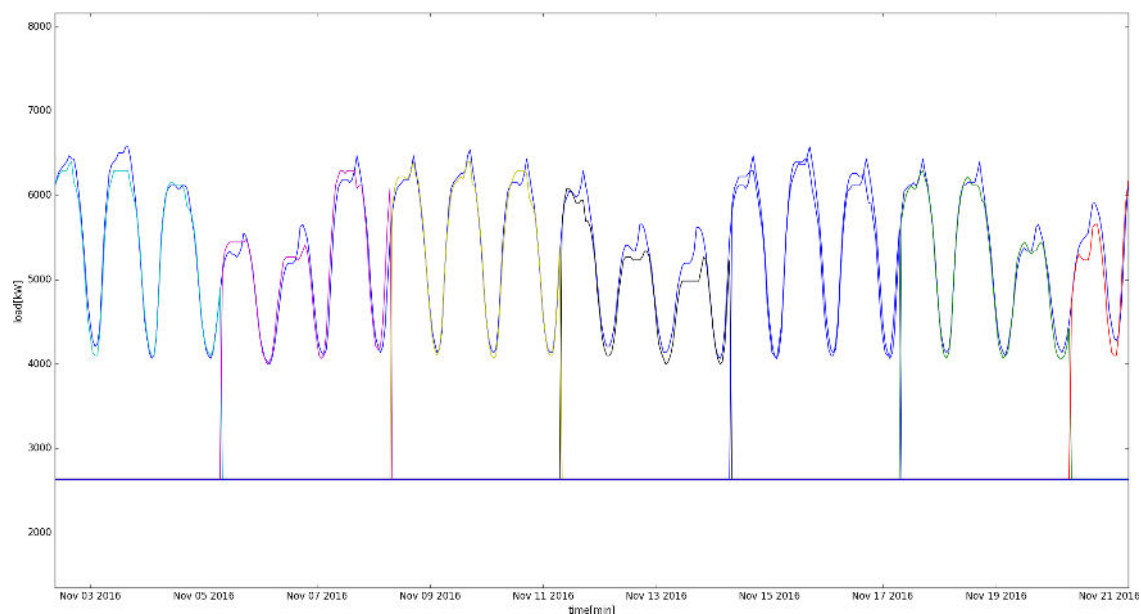
Hyperparametre a iné vlastnosti modelu použitého v tomto experimente boli:

- Počet dilatovaných vrstiev: 20
- Veľkosť výstupnej vrstvy: 256
- Veľkosť vnútorných kanálov: 512
- Najlepší model nájdený na iterácií: 60000

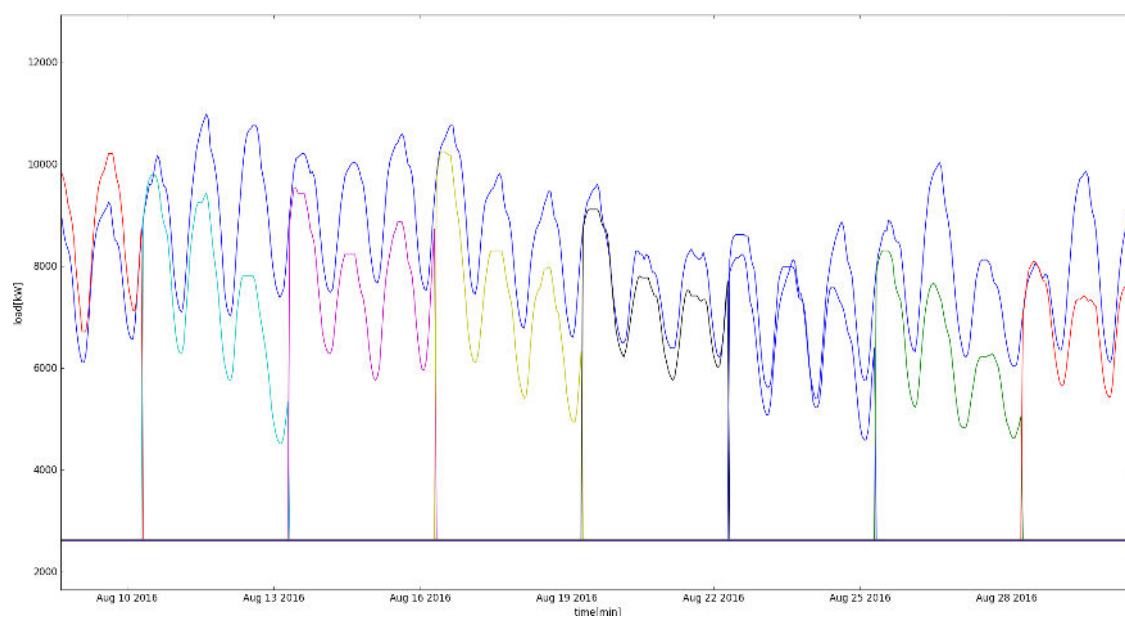
Na detaile, zobrazenom na obrázku 6.11 je možné vidieť predpovede na 1 krok dopredu. Na detaile aj mimo neho sú predpovede veľmi presné.

Na obrázkoch 6.12 a 6.13 vidíme 3 dňovú predpoveď pre dve rozdielne obdobia, august a november. Je zaujímavé sledovať veľmi odlišnú presnosť predpovede. V prípade novembrových dát, presnosť je veľmi dobrá avšak augustové dáta vykazujú silné odchýlky. Tento trend je badateľný pri všetkých predpovediach dlhších ako jedna vzorka.

Práve silné odchýlky počas letných mesiacov sú pravdepodobne dôvod horšieho MAPE skóre.



Obr. 6.12: Detail predpovede na 3 dni dopredu, New Yorkské dáta, november



Obr. 6.13: Detail predpovede na 3 dni dopredu, New Yorkské dáta, august

Kapitola 7

Zhrnutie

V tejto práci sme sa venovali predpovedaniu spotreby elektrickej energie pomocou hlbokých neurónových sietí. Rozhodli sme sa použiť architektúru WaveNet, ktorá bola vyvinutá firmou DeepMind. Jej špecializáciou je práca so zvukom a je schopná generovať hudbu či reč.

V rámci tejto práce sme sa rozhodli použiť vlastnú implementáciu tejto architektúry vo frameworku Theano.

7.1 Vyhodnotenie výsledkov

Vykonalí sme veľké množstvo experimentov, do tejto práce sme zaradili iba tie z nich, o ktorých si myslíme, že sú najvýznamnejšie. Tieto experimenty boli vykonané na 3 dátových sadách a to síce dátová sada z Belgicka a miest Bratislavy a New Yorku.

Prvý experiment bol vykonaný na belgickej dátovej sade, ktorá zaznamenáva spotrebu v celej sieti tamojšieho distribútora v 15 minútových intervaloch. Zdá sa byť chaotickejšia a náročnejšia na predikciu ako bratislavská dátová sada. Vyznačuje sa horšími výsledkami ako predikcia pre Bratislavu avšak tieto výsledky sú stále dostatočne presné. Na výsledkoch sme pozorovali hlavne odchýlky v lokálnych extrémoch. V tomto prípade bolo veľkým prínosom pridanie poveternostných dát to predpovede, čo znateľne vylepšilo výsledky.

Druhý experiment bol na bratislavskej dátovej sade, jednalo sa o predikciu

v rámci jedného mesta s 15 minútovou vzorkovacou frekvenciou. Predikcie boli výnimočne presné aj v prípade predikcií na dlhšiu dobu, kde sme zaznamenali veľmi nízku degradáciu presnosti pri narastajúcej dĺžke predpovede. Aj v tomto prípade sa väčšina odchýliek vyskytovala v lokálnych extrémoch, naproti tomu miesta kde sa spotreba prudko menila boli väčšinou veľmi presné.

V treťom experimente sme predpovedali spotrebu elektrickej energie pre mesto New York na východnom pobreží USA. V tomto prípade sa jednalo o dáta s hodinovou vzorkovacou frekvenciou, čo negatívne ovplyvnilo výsledky. Väčší interval medzi vzorkami má za následok to, že predpoveď na rovnako veľký časový úsek potrebuje menej krokov, avšak zároveň vznikajú problémy kvôli veľkosti datasetu. V prípade, že použijeme kratší časový úsek, napríklad 2 roky, máme príliš málo dát, v opačnom prípade máme v dátach príliš veľa rokov, s čím si model nie je schopný v tejto forme poradiť.

Napriek podstatne väčšiemu modelu, sme dospeli k horším výsledkom, na ktorých sme mohli pozorovať veľké rozdiely v presnostiach. Dôvodom pravdepodobne je, že kvôli väčšiemu počtu rokov v dátovej sade a nižšej vzorkovacej frekvencii má neurónová sieť problém správne predpovedať, pretože neodhadne profil dňa. Takéto vysvetlenie podporuje fakt, že predpoveď na jednu vzorku dopredu bola podobne presná ako v ostatných dvoch experimentoch a fakt, že v niektorých prípadoch, keď neurónová sieť správne odhadla profil dňa sú predpovede relatívne presné.

Zaujímavosťou sú zjavné podobnosti v predpovediach, ktoré majú rôznu dĺžku, na miestach kedy jedna predikcia sa spolieha už na predtým predikované hodnoty, zatiaľ čo tá druhá používa ešte reálne hodnoty. Je tomu tak aj napriek tomu, že predpovedané hodnoty nie sú úplne presné. Jedná sa pravdepodobne o dôsledok robustnosti predpovede dosiahnutej pomocou pridávania šumu počas tréningu.

Ako sme usúdili v predchádzajúcej kapitole, veľkosť zorného poľa neurónovej siete má pravdepodobne priamy dopad na rýchlosť akou sa degraduje presnosť pri narastajúcej dĺžke predpovedi.

7.2 Budúca práca a návrh zlepšenia

Najjednoduchšou cestou ako ďalej zlepšiť výsledky je zvýšiť zložitosť modelu. Pomocou ďalších experimentov nájsť správne kombinácie parametrov ako je veľkosť vnútorných kanálov, kanálov na výstupe alebo počet dilatačných vrstiev, tak aby bola presnosť maximalizovaná pri stále vykonateľnom modeli.

Zvýšenie počtu dilatačných vrstiev a s ním spojené zväčšenie zorného pola neurónovej siete by pravdepodobne malo za následok lepšie výsledky pri dlhodobých predpovediach.

Počas vykonaných experimentov sa doba začiatku predpovedí líšila avšak vždy bola uprostred dňa. Je možné, že iné štartovné pozície, napríklad poľnoc, by mali lepšie výsledky, pretože by ľahšie predpovedali profil dňa.

Potencionálnym vylepšením predikcií na dlhšiu dobu je zmena spôsobu reťazenia. V aktuálnej implementácii je stredná hodnota najpravdepodobnejšieho intervalu z výstupu pripojená k vstupu pre ďalší krok. V prípade, že by sme na vstupe mali historické dáta v rovnakom formáte ako výstup siete, t.j. pravdepodobnostná distribúcia, bolo by možné pripojiť k vstupu na ďalší krok celý výstup. Takto by mala neurónová sieť informácie o pravdepodobnosti pre každý interval. Avšak pri použití takéhoto postupu by sme pravdepodobne museli zmeniť spôsob akým vkladáme poveternostné a časové dáta.

Počas experimentov sa ukázali predpovede s nižšou vzorkovacou frekvenciou ako problematické. Toto je riešiteľný problém, ktorý by si zaslužil ďalšie úsilie.

V tu prezentovaných experimentoch sme použili časové dáta s periódou jedného týždňa, ktoré boli tvorené sínusovou a kosínusovou zložkou. Počas vývoja sme experimentovali aj s rôznymi inými variantami a pozorovali sme zníženie rýchlosti učenia v prípade, že týchto zložiek bolo viac. V prípade väčšieho počtu vnútorných kanálov by sme mohli rátať s viacerými zložkami časových dát a ich pomer by ostal zachovaný. Vďaka zložkám s ročnou periodicitou by sme mohli pomôcť identifikovať rôzne ročné obdobia a ich rôzne typické denné profily. Vďaka zložke, ktorá by obsahovala informáciu o roku, by bolo možné identifikovať typické profily pre konkrétne roky, čo by mohlo pomôcť pri vyššom počte rokov v dátovej sade.

Literatúra

- [1] N. Amjady. Short-term hourly load forecasting using time-series modeling with peak load estimation capability. *IEEE Transactions on Power Systems*, 16(3):498–505, Aug 2001.
- [2] Z. A. Bashir and M. E. El-Hawary. Applying wavelets to short-term load forecasting using pso-based neural networks. *IEEE Transactions on Power Systems*, 24(1):20–27, Feb 2009.
- [3] M. Beccali, M. Cellura, V. Lo Brano, and A. Marvuglia. Forecasting daily urban electric load profiles using artificial neural networks. *Energy Conversion and Management*, 45(18–19):2879 – 2900, 2004.
- [4] Yoshua Bengio. Learning deep architectures for ai. *Found. Trends Mach. Learn.*, 2(1):1–127, January 2009.
- [5] Li Deng and Dong Yu. Deep learning: Methods and applications. *Found. Trends Signal Process.*, 7(3–4):197–387, June 2014.
- [6] Richard Durbin and David E. Rumelhart. Product units: A computationally powerful and biologically plausible extension to backpropagation networks. *Neural Comput.*, 1(1):133–142, March 1989.
- [7] Erol Egrioglu, Ufuk Yolcu, Cagdas Hakan Aladag, and Eren Bas. Recurrent multiplicative neuron model artificial neural network for non-linear time series forecasting. *Neural Processing Letters*, 41(2):249–258, 2015.
- [8] Eugene A. Feinberg and Dora Genethliou. *Applied Mathematics for Restructured Electric Power Systems: Optimization, Control, and Computational*

Intelligence, chapter Load Forecasting, pages 269–285. Springer US, Boston, MA, 2005.

- [9] L. Ghelardoni, A. Ghio, and D. Anguita. Energy load forecasting using empirical mode decomposition and support vector regression. *IEEE Transactions on Smart Grid*, 4(1):549–556, March 2013.
- [10] National Grid. Summer outlook report. www2.nationalgrid.com/UK/Industry-information/Future-of-Energy/FES/summer-outlook, April 2016.
- [11] C. Guan, P. B. Luh, L. D. Michel, Y. Wang, and P. B. Friedland. Very short-term load forecasting: Wavelet neural networks with data pre-filtering. *IEEE Transactions on Power Systems*, 28(1):30–41, Feb 2013.
- [12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.
- [13] Ku-Long Ho, Yuan-Yih Hsu, Chuan-Fu Chen, Tzong-En Lee, Chih-Chien Liang, Tsau-Shin Lai, and Kung-Keng Chen. Short term load forecasting of taiwan power system using a knowledge-based expert system. *IEEE Transactions on Power Systems*, 5(4):1214–1221, Nov 1990.
- [14] Yoshua Bengio Ian Goodfellow and Aaron Courville. Deep learning. Book in preparation for MIT Press, 2016.
- [15] M. S. Kandil, S. M. El-Debeiky, and N. E. Hasanien. Long-term load forecasting for fast developing utility using a knowledge-based expert system. *IEEE Transactions on Power Systems*, 17(2):491–496, May 2002.
- [16] Kwang-Ho Kim, Jong-Keun Park, Kab-Ju Hwang, and Sung-Hak Kim. Implementation of hybrid short-term load forecasting system using artificial neural networks and fuzzy expert systems. *IEEE Transactions on Power Systems*, 10(3):1534–1539, Aug 1995.
- [17] Santosh Kulkarni, Sishaj P. Simon, and K. Sundareswaran. A spiking neural network (snn) forecast engine for short-term electrical load forecasting. *Applied Soft Computing*, 13(8):3628 – 3635, 2013.

- [18] H.C.W. Lau, E.N.M. Cheng, C.K.M. Lee, and G.T.S. Ho. A fuzzy logic approach to forecast energy consumption change in a manufacturing system. *Expert Systems with Applications*, 34(3):1813 – 1824, 2008.
- [19] Stanley K. O. ; Miikkulainen R. Evolving neural networks through augmenting topologies. *Evol. Comput.*, 10(2):99–127, 2002.
- [20] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edition, 2003.
- [21] T. Senjyu, H. Takara, K. Uezato, and T. Funabashi. One-hour-ahead load forecasting using neural network. *IEEE Transactions on Power Systems*, 17(1):113–118, Feb 2002.
- [22] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *CoRR*, abs/1609.03499, 2016.
- [23] R.N. Yadav, P.K. Kalra, and J. John. Time series prediction with single multiplicative neuron model. *Applied Soft Computing*, 7(4):1157 – 1163, 2007. Soft Computing for Time Series Prediction.

Príloha A

Technická dokumentácia

V tejto časti sa nachádza špecifikácia softvérovej konfigurácie a opis tried a metód programu.

A.1 Opis softvérovej konfigurácie

Program vytvorený v rámci tejto práce, vyžaduje Theano vo verzii aspoň 0.9.0 a všetky ním vyžadované knižnice, okrem týchto je nutná ešte knižnica Lasagne. Samotný program je písaný pre Python vo verzii 2.7.

Theano podporuje dva rôzne spôsoby, pomocou ktorých sa dajú prevádzať výpočty na grafickej karte, starší spôsob vyžadoval iba knižnicu CUDA od NVidie, novší spôsob vyžaduje knižnicu libgpuarray od tvorcov Theana. Kód bol vyvíjaný tak aby bol rátaný na grafickej karte, v prípade použitia CPU by sa mohli vyskytnúť problémy.

Konfigurácia Theana môže byť vykonaná viacerými spôsobmi v tejto práci sme použili konfiguračný súbor `theano.rc`, ktorý musí byť umiestnený na adrese `$HOME/.theanorc` pre linux a adrese `$HOME/.theanorc:$HOME/.theanorc.txt` pre windows. Použitý konfiguračný súbor je nižšie zobrazený, nie všetky nastavenia sú nutné a niektoré sú ignorované kvoli iným nastaveniam. Pre konkrétne informácie o nastaveniach odporúčam Theano dokumentáciu¹.

¹<http://deeplearning.net/software/theano/library/config.html>

```
[global]
device = cuda
floatX = float32
mode = FAST_RUN
exception_verbosity = high
openmp = True
config.openmp_elemwise_minsize = 200000
assert_no_cpu_op = warn
[nvcc]
flags= -D_FORCE_INLINES
[lib]
cnmem = .75
[cuda]
root = /usr/local/cuda-8.0
[optdb]
max_use_ratio=50
[gof]
optdb.max_use_ratio=50
```

Experimenty sme vykonávali na grafických kartách GTX1060, ktorá má veľkosť operačnej pamäte 6GB a GTX1070, ktorá má veľkosť operačnej pamäte 8GB. Práve veľkosť pamäte je dôležitá, pretože ak sa výpočtový model a dataset nezmestí do pamäte nie je možné experiment vykonať.

A.2 Opis tried a metód programu

Kód potrebný na vykonanie experimentov je rozdelený do viacerých súborov. Súbor `WaveNet_nominal_weather_A.py` obsahuje kód modelu a experimentu, slúži ako štartovný súbor. Súbor `weather.py` a `sigproc_utils.py` obsahujú metódy potrebné na načítanie a predspracovanie historických, časových a poveternostných dát, súbor `visualization.py` a `VisualizationQ.py` obsahujú metódy zaoberajúce sa vizualizáciou výsledkov, tiež môžu byť použité na načítanie a prácu s uloženými modelmi a výsledkami.

Súbor `WaveNet_nominal_weather_A.py` obsahuje 3 metódy a 2 objekty.

- metóda `validate()`

- Vytvára priečinky a štartuje experimenty, môže viackrát volať metódu `training()`, jej parametre určujú hlavné hyperparametre experimentu.
- metóda `training(layers, output, max_iter=20000,...)`
 - Ústredný kód experimentu, obsahuje hlavný cyklus. Nastavuje sa tu veľké množstvo premenných a charakter dát.
- metóda `conv1x1(unit_input, conv_filter)`
 - Proxy metóda, ktorá volá Theano operáciu pre konvolúcie. Jej argumentmi je symbolická theano premenná predstavujúca vstup a zdieľaná theano premenná, ktorá predstavuje konvolučný filter
- metóda `gau(unit_input, filter_1, filter_2, dilatation, gate_bias=None, filter_bias=None)`
 - Definuje neuróny s bránou, zobrazené na 5.4. Jej argumentmi je symbolická premenná predstavujúca vstup, veľkosť dilatácií a zdieľané premenné predstavujúce filter a bránu.
- trieda `Block(block_input, dilatation, filter_size, channels, params=None, use_bias=True)`
 - Hlavný komponent modelu, predstavuje jeden blok, ktoré je možno vidieť na obrázku 5.4. Obsahuje iba konštruktor, ktorého argumenty predstavujú symbolický vstup, veľkosť dilatácie, počet kanálov a parametre v prípade, že ich chceme inicializovať na inom mieste v programe alebo načítať zo súboru.
- trieda `WaveNet(nn_input, weather, weather_shape, time, time_shape, layers, n_steps, filter_size, channel_sizes, params=None, noise_in=None)`
 - Táto trieda definuje samotný model Wavenetu, jej konštruktor akceptuje symbolické vstupy a prípadne ich tvar, počet vrstiev, veľkosť výstupu a kanálov. Táto trieda má okrem konštruktora množstvo iných metód.
 - metóda `create_params(channel_sizes, filter_size, weather_shape, time_shape, layers, params=None)`
 - * Inicializačná metóda pre všetky parametre modelu. V prípade, že argument `params` nie je prázdny, sú parametre načítané z neho.

Dôvodom takéhoto riešenia je možnosť načítania modelu zo súboru.

- metóda *init_functions(index, x, w, t, y, r, min_max, data)*
 - * Metóda obsahuje definície všetkých symbolických funkcií, ktoré sme použili, t.j. trénovacia, validačná, evaluačná, generátor a iné. Argumentmi sú všetky symbolické vstupy do funkcií, niektoré parametre a dáta samotné.
- metóda *generate(predicted_length, sskip, n_test_subsections, testing_data, test_time_data, testing_weather_data)*
 - * Generovacia metóda, ktorá opakovane volá generátor a jeho výstupy si ukladá do ďalšieho vstupu.
- Okrem týchto metód, trieda obsahuje niekoľko metód spracúvajúce výstup, funkcie chyby(cross_entropy, MAPE). Nie všetky tieto metódy sú použité ale mohli byť použité počas vývoja a experimentovania.

Súbor `sigproc_utils.py` obsahuje veľké množstvo metód a žiadne objekty, v tomto prípade iba spomenieme niektoré dôležitejšie. Kvôli tomu, že v určitej dobe bolo viacero verzií modelu závislé od tohto súboru obsahuje viacero variant tej istej funkcie alebo funkcie, ktoré už nie sú používané. Pretože každý dataset, ktorý sme použili bol inak formátovaný, každý z nich má svoju vlastnú metódu, nebudeme ich opisovať, majú pochopiteľné mená a rozhranie.

- metóda *prepare_subsections_with_weather(data, subsection_size, overlap, distribution, weather_data=None, time_data=None)*
 - Táto metóda pripraví dáta do sekcií, bola bližšie opisovaná v [5.4.1.1](#). Existuje viacero variánt, bez poveternostných dát a časových dát a staršia varianta, ktorá pracuje s dávkami.
- metóda *repair_elia(data)*
 - Metóda pôvodne napísaná aby opravila chýbajúce hodnoty v ELIA datasete tak, že ich vyhladí na základe prvej predchádzajúcej a prvej nadchádzajúcej riadnej hodnoty. Táto metóda bola neskôr použitá aj na iné datasety.

- metóda *createTimeData(start_date, end_date, interval)*
 - Vrátí sínusovú a kosínusovú časovú zložku medzi štartovacím a konečným dátumom so špecifikovaným intervalom.

Súbor `weather.py` obsahuje načítavaciu a opravné funkcie pre Asos dáta (Automated Surface Observing System) a načítavaciu funkciu pre Merra2 dáta. Práve kvôli vysokej chybovosti a nestálymi intervalom sme neskôr nahradili Asos dáta Merra2 datasetom.

Príloha B

Používateľská príručka

Keďže program nedisponuje žiadnymi argumentami, grafickým užívateľským rozhraním alebo konfiguračnými súbormi je nutné ho manuálne nastaviť zmenou hodnôt v kóde.

V prvom rade je nutné zmeniť adresy, na ktorých sa majú hľadať datasety. Tieto sú definované priamo v kóde jednotlivých načítavacích metód v súboroch `sigproc_utils.py` a `weather.py`. Po týchto zmenách by mali byť datasety nájdené aj na iných systémoch.

V metóde `validate()` by mal používateľ nastaviť základné parametre experimentu. Je možné v jednom behu vykonať viacero experimentov.

Samotný výber datasetu prebieha na začiatku metódy `training()`, tu odporúčam postupovať podľa komentárov. Je nutné správne nastaviť dátumy aby všetky tri zložky dát(historické, časové a poveternostné) mali rovnakú veľkosť.

Počas behu programu sú používateľovi zobrazené výpisy o aktuálnej činnosti, stave, výsledkoch a iných informáciach. Medzi niektorými výpismi sa nachádzajú dlhšie intervaly, ktoré hlavne pri komplikovanejších modeloch a pomalších počítačoch môžu zabrať rádovo desiatky minút.

Po ukončení tréningu je otestovaný posledný model a model s najlepším valdiačným skóre na úlohe predpovedať na jeden krok dopredu a na úlohe predpovedať špecifikovaný počet krokov dopredu. Niektoré výsledky sú uložené ako serializovaný súbor alebo ako obrázok, iné sú priamo zobrazené a oba modely sú serializované do súbora, tak aby mohli byť neskôr znovu načítané, či už pre ďalší

tréning alebo pre ďalšie testovanie.

Okrem týchto artefaktov je vytvorený aj súbor `info.txt`, ktorý obsahuje informácie o architektúre, použitých parametroch, verzií, datasete a dosiahnutej presnosti.

Zobrazenie serializovaných výsledkov môže byť prevedené pomocou metódy `show_predictions()` v súbore `visualization.py` presné použitie je popísané v komentároch.

Súbor `VisualizationQ.py` je schopný načítať serializovaný model a prečítať údaje zo súboru `info.txt` a pomocou nich ho nanovo pripraviť na použitie. Pomocou parametru *ahead* je možné definovať pre aké úsek budú hodnoty generované.

Príloha C

Elektronické médium

Štruktúra dát na elektronickom médiu je nasledovná:

/Application

- potrebné zdrojové kódy aplikácie

/Experiments

- artefakty z experimentov prezentovaných v diplomovej práci

/Documentation

- diplomová práca spolu s anotáciami v slovenskom a anglickom jazyku

/Documentation/Latex

- latex zdrojové súbory dokumentácie

readme.txt - popis obsahu média v slovenskom a anglickom jazyku

Príloha D

Vyhodnotenie plánu práce

Plán práce bol prevažne plnený, avšak počas 3. až 4. týždňa sme prerábali veľkú časť architektúry čo sme nepredpokladali a teda to spôsobilo zameškanie, ktoré sme museli neskôr dobiehať.

1.-3. týždeň	Optimalizácia, zlepšenie evaluácie a vizualizácie
4. týždeň	Implementovanie chýbajúcich a potrebných mechaník architektúry
5. týždeň	Zpracovanie poveternostných dát do modelu
6.-10. týždeň	Vylepšovanie výsledkov
11.-12. týždeň	Vykonanie experimentov a ich evaluácia

Tabuľka D.1: *Pôvodný plán práce na tento semester*

V tomto pláne sme nepočítali s podaním príspevku na konferenciu EANN2017, ktorej termín odovzdania bol 20. apríla, neskôr posunutý až do 2. mája. Kvôli tomuto článku sme začali s vykonávaním experimentov už pred originálnym termínom.

Príloha E

Pripravený príspevok na konferenciu EANN2017

Using Advanced Audio Generating Techniques to Model Electrical Energy Load

Michal Farkas and Peter Lacko

Faculty of Informatics and Information Technologies,
Slovak University of Technology, Bratislava, Slovakia
michalfarkas1@gmail.com, peter.lacko@stuba.sk

Abstract. The prediction of electricity consumption has become an important part of managing the smart grid. Smart grid management involves energy production (from traditional and renewable sources), transportation and measurements (smart meters). Storing large amounts of electrical energy is not possible, therefore it is necessary to precisely predict energy consumption. Nowadays deep learning approaches are successfully used in different artificial intelligence areas. Deep neural network architecture called WaveNet was designed for text to speech task, improving speech quality over currently used approaches. In this paper, we present modification of the WaveNet architecture from speech (sound waves) generation to energy load prediction.

Keywords: artificial neural networks, deep learning, time series prediction

1 Introduction

Due to the characteristics of production facilities, distribution networks and free market, accurate forecasts of various values is indeed very important for modern energy industry and market, if they have to operate at peak efficiency.

Electric energy production facilities can be dependent on external factors and may not be able to change their output on moments notice. For example various alternative sources of energy such as wind, solar energy have limited output determined by wind speed and sunlight exposure. Furthermore, they are often embedded within the local distribution network, thus they are seen from the outside only as a reduction of demand.

Distribution networks can be damaged by extremely high amount of energy in network, which is not consumed. There are risks of blackouts and brownouts, which are highly damaging and disrupting to economy and society and can even result in material damage. Because of the technological inability to efficiently store vast amount of electric energy, there is little to no balancing buffer. Without this ability, produced energy has to be consumed right away, otherwise it would be wasted. Naturally, limiting amounts of wasted energy leads to more efficiency, which in turn leads to competitive advantage and subsequently less ecological impact on environment, which has proved to be growing problem in

recent years. With advent of smart-grids, deregularization of industry and other changes, energy load forecasting is even more important.

Due to how are energy markets, electric energy production and distribution industry constructed, there are various uses for load forecast. These are generally categorized by their prediction window, but they can also differ by time resolution, scope and specialization. In this paper we are considering multiple cases, which differ by time, area, resolution and scope. Artificial neural networks were successfully used for series analysis [9] and prediction. Nowadays deep learning approaches are massively used in different artificial intelligence areas providing state-of-the-art results.

2 State of the art

There are great number of various methods proposed for problem of energy load forecasting, they fall under statistical, engineering and artificial intelligence categories.

Engineering methods can, if they are complicated enough, reach quite high accuracy. These tools are well developed and used, however they require high amount of information, which is unobtainable in practice on a larger scale. Coupled with their need for domain expert, their effectiveness is severely limited. On the other hand, simplified engineering approaches don't require such vast amount of information, setting and expert work, however they are not as precise. [7]

Statistical methods try to model data by mathematical functions, in which various factors act as input. Parameters of these models are then trained on historical data, thus using this regression, relationships between input factors and output value, in our case electric energy consumption, are found. In [5] authors discriminate between two different types of models, additive and multiplicative, former models load as a sum of various components (underlying trend, weather and day variations, holidays, etc.), latter is product of multiplication of various factors and informations. Statistical models are easy to understand and maintain and they achieve average results, but it has been concluded they lack flexibility.

Artificial intelligence approaches consists mainly of artificial neural networks, but various other solutions were used, for example support vector machines, expert systems, fuzzy logic and different solutions. In [4] wavelet decomposition coupled with PSO method, similar days approach and grey model method to predict electricity consumption on various case for two distinct datasets, which did consists of rather large area. On the other hand [1, 2] predicted load on small areas, data of the former consisted of a single city and latter of a single buildings. Method for long term prediction both on single building and whole region, was proposed in [3], it was based on decomposition with various extensions of support vector regression. Incremental heterogeneous ensemble model for time series prediction was proposed in [8] and was successfully used for electricity load prediction.

3 Data characteristics

Since method used in this article, was originally developed for audio generation purposes, it is quite appropriate to describe characteristics of energy consumption time-series and how it differs from audio time-series.

In both cases there is strong periodicity of data, in case of energy it would naturally be daily, weekly and yearly periodicity. Audio data usually being composed of multiple components, which each had their own frequency, does naturally have strong periodicity.

One of the more obvious differences is sampling frequency. Energy time-series usually have sampling frequency of 15 minutes or one hour. In some special cases, dictated by data available or forecast specialization, its sampling frequency may differ. On the other hand, sampling frequency of audio data, while varying by standard, compression and recording quality, typically lie between 8kHz-44.1kHz. Even if we consider, that energy time-series cover vastly longer time-period than audio time-series, audio time-series will usually have much more information. To put it in perspective, energy time-series at 15 minutes sampling interval measured over 16 years, have same data length as 12.7 seconds of CD quality audio time-series. Direct consequences are smaller sizes of energy datasets, which can limit our prediction.

Audio data is mainly determined by its underlying information, whether it is music or speech. If we ignore content, which is far more complicated to predict in a meaningful way, there are still factors that need to be taken into consideration, such as speaker or emotion. Consumption of electric energy data lacks any meaningful content, instead it depends on socio-economical, environmental, cultural and legal factors. Most of these cannot be easily represented by a value, instead we can learn these from historical data. Factors that can be represented by a value, for example time, weather, workdays, can act as an additional input. However, even if these factors can be represented they must be available at the time of prediction. Unsurprisingly, accuracy of forecasts is highly dependent on factors taken into account.

Prime example of socio-economic or cultural factors are workdays and holidays, which have clear effect on electricity consumption. Naturally, national holidays are nation-specific and while most of the world shares the same workweeks, there are exceptions, such as Israel or some muslim countries. Weather factors, such as temperature, humidity, pressure and windspeed can affect electricity consumption in both directions. Most significant actors in this situation are HVAC systems and alternative power sources. As we mentioned in introduction chapter, some solar and wind power plants are embedded in network and their output is seen only as a reduction in local electricity consumption and since their output is dependent upon these weather factors, they have direct, inverted impact on electricity consumption. HVAC systems are mostly used in weather extremes to cool or to heat, thus generally in cold climates their impact is most pronounced in winter due to extensive heating and in tropical climates their impact is most noticeable in summer due to air conditioning.

Noticeable effect on varying scopes can have economic factors, such as economic performance, employment and others. However, impact of these factors can be marginal and they are rarely available at the time of prediction and thus are almost never used.

In data, there can be also anomalies, caused by one-time events, social unrest, conflicts, demonstrations various sporting events and competitions, list of potential disrupting events can be quite long. From their definition as anomalies these may not be predicted by model and their impact can be almost impossible to predict. In some cases, such disruptive events are announced with advance and thus human or automated system can feed some information into prediction system or different model altogether can be used.

4 Method description

Neural network architecture used in this paper is called WaveNet, it was originally proposed by DeepMind as a audio generation model [6]. Authors tested it as a text-to-speech system on english and mandarin chinese languages, as a music generation and as a speech recognition model. It operates on raw audio and its output is probability distribution. Main components of this model are 1x1 convolutions and causal dilated convolutions connected as a residual network.

Since we are predicting future values, we have to take into account only historical data, otherwise we would violate causality. This is ensured by shifting ordinary convolution by one timestep, by which we create causal convolution. Example of this type of convolution is shown in figure 3.

To enlarge receptive field of network, without changing number of layers or size of filter, convolution is modified so it takes inputs only at certain intervals. In WaveNet model, these intervals are defined as $2^{(n \bmod 10)-1}$, where n stands for depth of a layer. This variant of convolution is called dilated or convolution with holes and is shown in figure 4.

So called 1x1 convolutions have filter size of one in both dimensions, hence their name. They act as a fully connected layer on single column of input, thus they can only change number of channels. They are relatively cheap and don't interfere with causality.

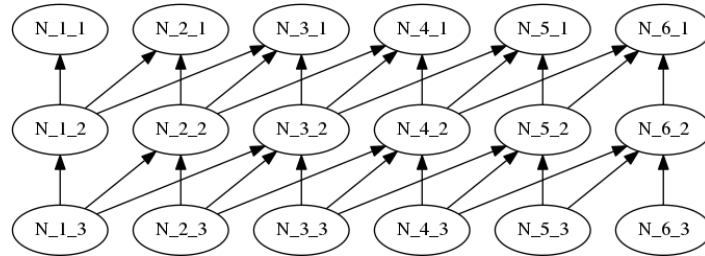


Fig. 1. Causal convolution.

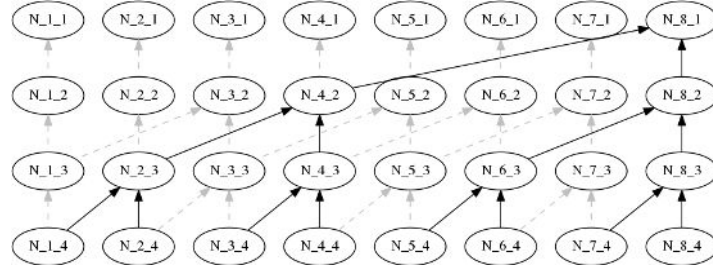


Fig. 2. Dilated causal convolution.

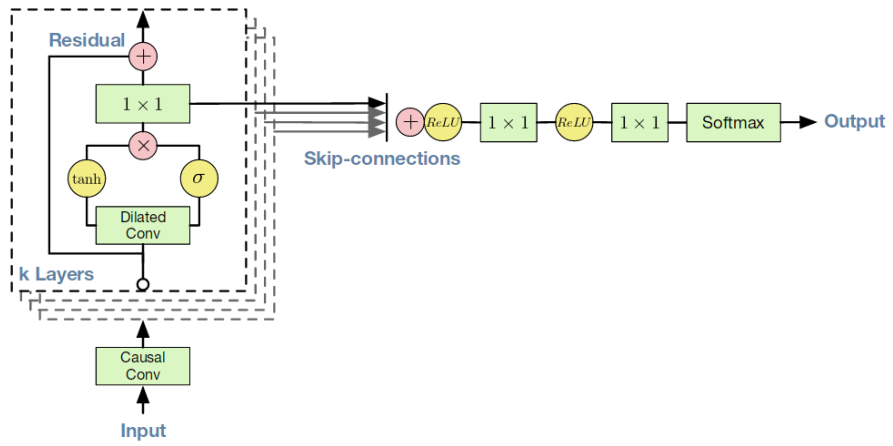


Fig. 3. WaveNet architecture; reproduced from [6].

These components are connected as shown in figure 4. Before dilatation layers, there is a single regular causal convolution layer, which changes channel size. Subsequent layers consists of two dilated convolution layers, their combination is depicted on following equation:

$$output = \tanh(W_f * x) \odot \sigma(W_g * x) \quad (1)$$

Dilatation layers are not only connected in serial fashion, but they also contain residual and skip connections, former are used to carry unchanged input to subsequent layer, latter are used to carry layer output to the end of neural network. There skip outputs are summed and processed to get final output.

Wavenet architecture implemented in this article does not have all features of original model, missing from our implementation are both local and global conditioning, context stacks and faster generation computation features.

Hovewer, since we do use weather and time data in prediction we need to feed it into neural network. We added this information as additional channels. By combining historical data, temperature, humidity, pressure, windspeed and time data, which consists of a sine and cosine components with weekly periods, our input has 7 channels. We tested different weather attributes and additional time channels with daily or yearly periods. We concluded that some weather attributes were not significant enough. Similarly daily time data did not help prediction, probably because network is able to infer time of day from historical data. We introduced yearly periodic data in hopes of identifying holidays such as christmas, however this was most likely hindered due to small data size.

Due to numerical stability of softmax function in Theano we had to add normalization function just before softmax to ensure proper results.

Output of neural network is probability distribution, size of it directly limit accuracy of prediction and more accurate prediction could be achieved by higher output resolution. To be able to predicted more than one sample ahead, we put most probable prediction at the end of input. To reduce error growth and encourage robustness, we added noise to the input during training, its size was dependent upon last evaluation error.

5 Results

In first experiment, we tested this model on ELIA¹ dataset, which consists of whole energy distribution network belonging to a company in Belgium of the same name. Weather data was extracted from NASA Merra2 reanalysis. Sampling rate for both datasets was 15 minutes. Both weather and load data were normalized and any missing values were corrected by smoothing. We ran experiments on subset ranging from years 2007 to 2008. We tested the same model on

¹ <http://www.elia.be/en/grid-data/data-download>

different prediction scopes using mean average percentage error - MAPE (Table 1). On Figure 5 are detailed examples of predictions on ELIA dataset.

In our second experiment, we used load data from city of Bratislava, between July of 2013 and February of 2015. This dataset did not consist of whole city and it was produced mainly by industry customers. This was complemented by Merra2 dataset and had interval of 15 minutes too. Normalizing and filling missing values was done in the same way. Detail is shown in Figure 5 and MAPE in Table 1.

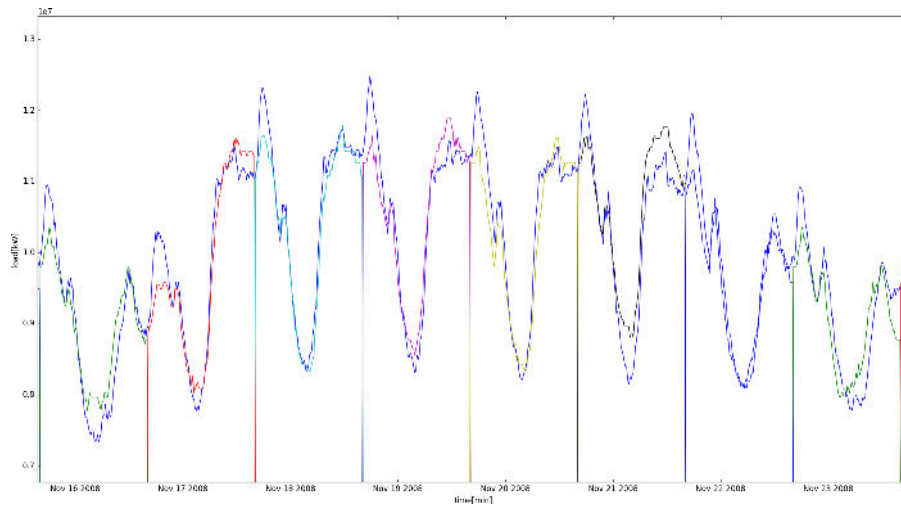


Fig. 4. Detail on one day ahead prediction on ELIA dataset, blue are real load values

Table 1. Table of MAPE values for various datasets

dataset	one sample ahead	1 day ahead	3 days ahead	7 days ahead	29 days ahead
ELIA dataset	1.22	3.45	3.52	4.20	5.02
Bratislava	1.09	1.934	2.14	2.29	3.53

6 Conclusion

In this paper, we presented a modification of the WaveNet architecture (originally used for sound generation) for electricity load forecasting. Two different datasets were used to evaluate this approach.

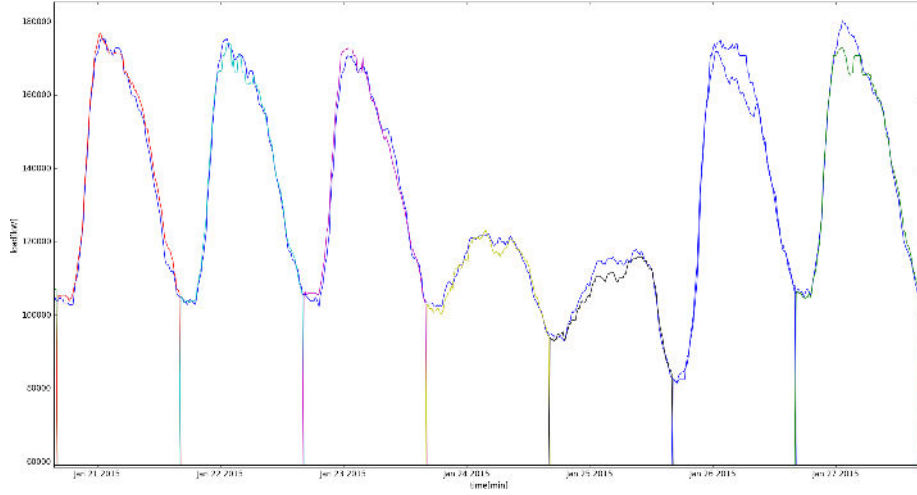


Fig. 5. Detail on day ahead prediction on Bratislava dataset, blue are real load values

Our results in both cases show that this method can be precise enough, provided that enough data with small enough sampling interval is available. In line with our expectations, growth of error with longer prediction window was dependent upon initial error size. Thus, we can conclude that with error small enough we could predict greater time frames with acceptable precision.

Forecast accuracy in this paper was severely limited by time and hardware constraints. We had to limit complexity of our models, which had only 8 layers, inner channel size of 128 and output size of 256. Higher values of these hyperparameters would almost surely result in even more accurate predictions. In our future work we would like to test this approach with more complex model with different hyperparameters and perhaps further adapt it to better suit this domain.

Acknowledgments. This work was partially supported by the Scientific Grant Agency of the Slovak Republic, grant No. VG 1/0752/14 and with the support of the Research and Development Operational Programme for the project International centre of excellence for research of intelligent and secure information-communication technologies and systems, ITMS 26240120039, co-funded by the European Regional Development Fund.

References

- [1] Hernández, L., Baladrn, C., Aguiar, J., Carro, B., Snchez-Esguevillas, A., Lloret, J.: Artificial neural networks for short-term load forecasting in microgrids environment. *Energy* 75, 252–264 (2014)

- [2] Jurado, S., Nebot, ., Mugica, F., Avellana, N.: Hybrid methodologies for electricity load forecasting: Entropy-based feature selection with machine learning and soft computing techniques. *Energy* 86, 276–291 (2015)
- [3] Ghelardoni, L., Ghio, A., Anguita, D.: Energy Load Forecasting Using Empirical Mode Decomposition and Support Vector Regression. *IEEE Transactions on Smart Grid* 4, 549–556 (2013)
- [4] Bahrami, S., Hooshmand, R-A., Parastegari, M.: Short term electric load forecasting by wavelet transform and grey model improved by PSO (particle swarm optimization) algorithm. *Energy* 72, 434–442 (2014)
- [5] Feinberg, E. A., Genethliou, D.: Load Forecasting. *App. Math. for Restruct. El. Power Sys.: Opt., Control, and Comp. Intel.*, 269–285 (2005)
- [6] Aaron van den Oord, et.al.: WaveNet: A Generative Model for Raw Audio. *CoRR* abs/1609.03499 (2016)
- [7] Zhao, H., Magouls, F.: A review on the prediction of building energy consumption *Renewable and Sustainable Energy Reviews* 16, 3586–3592 (2012)
- [8] Grmanová, G., Laurinec, P., Rozinajová, V., Ezzeddine, A. B., Lucká, M., Lacko, P., Vrablecová, P., Návrát, P.: Incremental Ensemble Learning for Electricity Load Forecasting. *Acta Polytechnica Hungarica*, vol. 13, no. 2, (2016)
- [9] Čerňanský, M., Makula, M., Trebatický, P., Lacko P.: Text Correction Using Approaches Based on Markovian Architectural Bias. In: *EANN 2007, Proceedings of the 10th International Conference on Engineering Applications of Neural Networks*, Publishing Centre Alexander Technological Educational Institute of Thessaloniki, 221-228, (2007)