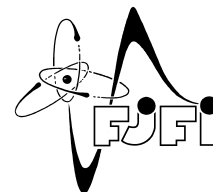


ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
Fakulta jaderná a fyzikálně inženýrská



# **Rozpoznávání vzorů v časových řadách pomocí hlubokých neuronových sítí**

## **Pattern recognition in time series by means of deep neural network**

Bakalářská práce

Autor: **Tereza Vorlová**  
Vedoucí práce: **Ing. Pavel Strachota, Ph.D.**  
Akademický rok: 2020/2021

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student:	Tereza Vorlová
Studijní program:	Aplikace přírodních věd
Studijní obor:	Matematické inženýrství
Studijní zaměření:	Aplikované matematicko-stochastické metody
Název práce (česky):	Rozpoznávání vzorů v časových řadách pomocí hlubokých neuronových sítí
Název práce (anglicky):	Pattern recognition in time series by means of deep neural networks

### Pokyny pro vypracování:

- 1) Prostudujte zadanou literaturu a seznámte se s principy hlubokých neuronových sítí, jejich různými typy, algoritmy pro jejich učení a metodikou pro vyhodnocování jejich úspěšnosti.
- 2) Seznámte se se softwarovými nástroji Keras, Tensorflow a přidruženým ekosystémem jazyka Python pro implementaci hlubokých neuronových sítí.
- 3) Seznámte se s pojmy a mechanismy svázanými s finančními trhy a jejich obdobou na trhu s kryptoměnami.
- 4) Vyzkoušejte si návrh, implementaci, použití a vyhodnocení úspěšnosti vybrané architektury neuronové sítě v prostředí Keras/TensorFlow na některé volně dostupné datové sadě.
- 5) Prozkoumejte pokročilé architektury neuronových sítí (např. ResNet) a možnost jejich využití k analýze vzorů v poskytnutých historických datech z kryptoměnové burzy.

Doporučená literatura:

- 1) C. C. Aggarwal, Neural Networks and Deep Learning. Springer, 2018.
- 2) F. Chollet, Deep Learning with Python. Manning Publications Co., 2018.
- 3) G. Zaccane, R. Karim, Deep Learning with TensorFlow. Packt Publishing, 2018.
- 4) S. Pattanayak, Pro Deep Learning with TensorFlow. Apress, 2017.
- 5) K. Söze, Bitcoin and Cryptocurrency Technologies. Sabi Shepherd Ltd., 2019.

Jméno a pracoviště vedoucího bakalářské práce:

Ing. Pavel Strachota, Ph.D.

KM FJFI ČVUT v Praze, Trojanova 13, 120 00 Praha 2


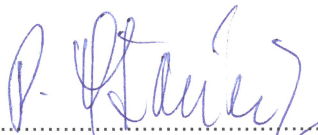
Jméno a pracoviště konzultanta:

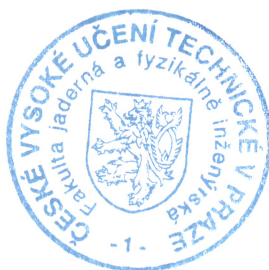
Datum zadání bakalářské práce: 31.10.2020

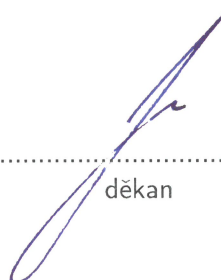
Datum odevzdání bakalářské práce: 7.7.2021

Doba platnosti zadání je dva roky od data zadání.

V Praze dne 30.10.2020

  
.....  
garant oboru  
  
  
.....  
vedoucí katedry



.....  
  
děkan

### *Poděkování:*

Na tomto místě bych ráda poděkovala především svému školiteli Ing. Pavlu Strachotovi, Ph.D., a to nejen za poskytnutí velmi kvalitního odborného zázemí, ale také jeho velké trpělivosti, vstřícnosti a ochotě po celou dobu naší spolupráce. Vážím si toho, že jsem měla tu čest pracovat na této bakalářské práci s někým tak profesně schopným a lidským zároveň. Díky patří i celé mojí rodině, především rodičům Mgr. Jiřině Vorlové a Ing. Jiřímu Vorlovi, za psychickou, ale i materiální podporu během celého bakalářského studia.

### *Čestné prohlášení:*

Prohlašuji, že jsem tuto práci vypracovala samostatně a uvedla jsem všechnu použitou literaturu.

V Praze dne 7. července 2021

Tereza Vorlová

*Název práce:*

**Rozpoznávání vzorů v časových řadách pomocí hlubokých neuronových sítí**

*Autor:* Tereza Vorlová

*Obor:* Matematické inženýrství

*Zaměření:* Aplikované matematicko-stochastické metody

*Druh práce:* Bakalářská práce

*Vedoucí práce:* Ing. Pavel Strachota, Ph.D., KM FJFI ČVUT v Praze, Trojanova 13, 120 00 Praha 2.

*Abstrakt:* Tato bakalářská práce se zabývá rozpoznáváním vzorů v časových řadách obsahující historická data z kryptoměnové burzy. Jako vhodný nástroj tohoto rozpoznávání byly zvoleny konvoluční neuronové sítě (CNN). Na jedné straně tak práce vysvětluje pojmy a porovnává mechanismy, na jejichž principu fungují klasické a kryptoměnové spotové a termínové finanční trhy. Zvláštní důraz je přitom kladen na téma samotného obchodování. Na straně druhé pak tato práce seznamuje s principy fungování neuronových sítí, převážně CNN, popisuje proces jejich aplikace, včetně důkladného rozebrání problematiky předzpracování dat, a vliv různých architektur při řešení úlohy predikce budoucího vývoje ceny aktiv na kryptoměnové burze. Jednotlivé architektury se od sebe přitom liší pouze různým nastavením hyper-parametrů. Konkrétně se jedná o vliv počtu neuronů, vrstev, množství vstupních dat, využití dropoutu, dávkové normalizace a dropoutu společně s MaxNorm regularizací na úspěšnost fungování sítě na formulované úloze. K trénování modelů byla použita jak reálná historická data, tak i data syntetická.

*Klíčová slova:* finanční trhy, konvoluční neuronové sítě, kryptoměny, strojové učení

*Title:*

**Pattern recognition in time series by means of deep neural network**

*Author:* Tereza Vorlová

*Abstract:* The aim of this bachelor thesis is the study of pattern recognition in time series, which contain historical data from the cryptocurrency exchange. As a suitable tool for this study, convolutional neural networks (CNN) have been chosen. Firstly, the thesis explains the concepts and compares the mechanisms behind the classical and cryptocurrency spot and derivatives markets. Special emphasis is put on trading itself. Secondly, the thesis introduces the principles of neural networks, predominantly CNN, describes the process of their application including a thorough analysis of data preprocessing, and the influence of various architectures on solving the task of predicting future asset price developments on the cryptocurrency exchange. The architectures differ from each other only by different hyper-parameter settings. In particular, the effect of the number of neurons, layers, the amount of input data, the use of dropout, batch normalization and dropout with MaxNorm regularization is studied. Both real historical and synthetic data were used to train the models.

*Key words:* convolutional neural network, cryptocurrencies, financial markets, machine learning

# Obsah

<b>Úvod</b>	<b>8</b>
<b>1 Kryptoměny</b>	<b>9</b>
1.1 Blockchain . . . . .	9
1.2 Těžba Bitcoinů . . . . .	9
1.3 Potvrzování transakcí . . . . .	10
1.4 Uložení Bitcoinů . . . . .	11
1.5 Výhody a nevýhody kryptoměn . . . . .	12
1.6 Budoucnost kryptoměn . . . . .	12
<b>2 Finanční trhy</b>	<b>13</b>
2.1 Býčí a medvědí trhy . . . . .	13
2.2 Technická a fundamentální analýza . . . . .	13
2.3 Obchodování . . . . .	14
2.4 Spotové trhy . . . . .	15
2.5 Termínové trhy . . . . .	16
2.5.1 Finanční deriváty . . . . .	16
2.5.2 Forward kontrakty . . . . .	16
2.5.3 Klasické fututres kontrakty . . . . .	16
2.5.4 Kryptoměnové fututres kontrakty . . . . .	19
<b>3 Umělé neuronové sítě</b>	<b>23</b>
3.1 Neuron . . . . .	24
3.1.1 Aktivační funkce . . . . .	24
3.2 Vícevrstvá neuronová síť . . . . .	25
3.2.1 Učení vícevrstvé neuronové sítě . . . . .	26
3.3 Konvoluční neuronová síť . . . . .	28
3.3.1 Struktura konvoluční neuronové sítě . . . . .	28
3.3.2 Učení konvoluční neuronové sítě . . . . .	32
<b>4 Experimenty</b>	<b>34</b>
4.1 Hlavní myšlenka . . . . .	34
4.2 Formulace úlohy a výběr dat . . . . .	34
4.3 Programovací prostředí . . . . .	36
4.4 Vizualizace sítě . . . . .	36
4.5 Seznámení s konvoluční neuronovou sítí . . . . .	37
4.6 Aplikace konvolučních neuronových sítí na formulovanou úlohu . . . . .	40

4.6.1	Zpracování datového souboru . . . . .	40
4.6.2	Sestavení modelu konvoluční neuronové sítě . . . . .	42
4.6.3	Vyhodnocení úspěšnosti sítě . . . . .	43
4.6.4	Výsledky . . . . .	43
<b>Závěr</b>		<b>58</b>
<b>Literatura</b>		<b>58</b>
<b>Prílohy</b>		<b>65</b>

# Úvod

Pojem neuronových sítí je v současné době čím dál tím více slýchaný nejen v odborné, nýbrž i laické společnosti. Jejich využívání v rozličných oblastech života se stává téměř dennodenní realitou. Konkrétně s konvolučními neuronovými sítěmi je přitom možné se setkat např. ve zdravotnictví (segmentace obrazu, či vývoj nových léků), v aplikacích pracujících na principu rozpoznávání řeči (chytrí asistenti - Siri, Alexa, či Google Assistant), či tváří z kamerových záznamů. Potenciál konvolučních neuronových sítí byl ale před nedávnem objeven také v oblasti dlouhodobého předpovídání [1], [2]. Výzkum v této oblasti je zatím na úplném počátku, stál však za nápadem využít konvoluční neuronové sítě pro řešení konkrétních úloh s daty pocházejícími z kryptoměnové burzy. Chování kryptoměnových trhů je mnohem hůře předvídatelné pomocí běžných nástrojů analýzy a predikce tržního trendu používaných na trzích klasických, proto se zde nabízí možnost využít algoritmy strojového učení. Finanční trhy navíc každý den produkují velké množství dat, což je hlavním a klíčovým předpokladem pro aplikaci těchto algoritmů.

Celá práce je koncipována tak, aby byly veškeré teoretické poznatky demonstrovány na konkrétních příkladech. Hlavním cílem je proniknout do problematiky finančních trhů a zjistit, co je od nich možné očekávat, vysvětlit teoretický princip fungování neuronových sítí a následně ho aplikovat na data z kryptoměnové burzy.

Úvodní část práce představuje sice pouze základní vhled do principu fungování kryptoměn, avšak detailní přehled pojmů a mechanismů svázaných s finančními trhy a jejich obdobou na trzích kryptoměnových. Hlavní důraz je přitom kladen na popis samotného procesu a objasnění hlavních principů obchodování, a to jak na spotových, tak termínových trzích. V případě termínových trhů je diskutováno hlavně obchodování s futures a forward kontrakty. Následuje část zaměřená na problematiku neuronových sítí. Zde je možné se seznámit s principy jednoduchých, jednovrstvých, či složitějších, vícevrstvých, neuronových sítí. Velmi detailně je zde přitom popsána teorie konkrétně konvolučních neuronových sítí, jež jsou speciálním typem sítí vícevrstvých. Závěrečná část je pak věnována celému procesu aplikace konvolučních neuronových sítí na úlohy strojového učení – od výběru a zpracování dat, přes návrh modelu, implementaci až po vyhodnocení úspěšnosti fungování vybraných architektur. Závěrem jsou pak představeny konkrétní kroky, které by mohly vést ke zvýšení úspěšnosti předpovědí.



# Kapitola 1

## Kryptoměny

Kryptoměny jsou elektronicky tvořené digitální měny. Finanční systém kryptoměn je tvořen sítí rovnocenných uzlů („spolků“ kryptoměnových těžařů) rozmístěných po celém světě, která je zabezpečena pomocí kryptografie (šifrování). V současné době existuje již obrovské množství kryptoměn jako například Ethereum, Ripple, BNB, či nejrozšířenější Bitcoin. Princip jejich fungování bude představen pouze pro Bitcoin, avšak u ostatních kryptoměn se nijak výrazně neliší [3].

### 1.1 Blockchain

Hlavní princip fungování finančního systému Bitcoinu je založen na tzv. *blockchainu*, což je speciální druh neustále se rozšiřující databáze, který je součástí každého uzlu. V blockchainech lze uchovávat různé druhy dat, avšak ve spojení s Bitcoinem se tento pojem používá pro jakousi účetní knihu, kde jsou zaznamenány veškeré provedené transakce. Blockchain se od klasické databáze obecně liší ve způsobu, jakým jsou zde informace ukládány, v kontextu Bitcoinů pak ještě i ve způsobu, jakým je celý systém spravován. Zatímco data v databázi jsou obvykle strukturována ve formě tabulky, blockchain shromažďuje informace po tzv. blocích, které uchovávají celou sadu dat. Jakmile se objeví nová data k uložení, vzniká nový blok, kam jsou data zaznamenána. Po naplnění kapacity uložení bloku dochází k jeho zřetězení za již dříve vyplněný blok. Tímto způsobem tedy vzniká jakýsi řetězec dat neboli blockchain. Všechny blockchainya jsou tedy databáze, avšak ne všechny databáze jsou blockchainya. V kontextu Bitcoinů je pak blockchain navíc plně decentralizovaný, takže nad ním žádná entita nemá kontrolu [4].

### 1.2 Těžba Bitcoinů

Vznik nového bloku vlastně představuje *těžbu* Bitcoinů. Každý blok je určen 6 hodnotami: číslem verze softwaru Bitcoin, hashem hlavičky předešlého bloku ( $\text{SHA} - 256^2$ ), hashem všech transakcí zahrnutých v bloku (angl. Merkle Root), časovou stopou (angl. timestamp), obtížností (angl. target) a proměnnou zvanou Nonce. Všechny hodnoty, až na proměnnou Nonce, jsou v daném okamžiku jasně definovány. Právě hodnota Nonce je získávána těžbou. Těžba tedy spočívá ve snaze najít na základě vstupních dat předdefinovanou proměnnou Nonce, přičemž obtížnost tohoto hledání se neustále mění, tak aby se průměrný čas těžby bloků držel na 10 minutách. Úkolem těžaře je pak vzít vstupy (prvních 5 výše uvedených hodnot), přidat proměnnou Nonce a vše poslat do tzv. *hash funkce* Bitcoinu. Pokud tzv. *hash* bloku vygenerovaný hashovací funkcí splňuje požadavky v síti (tj. byla přidána správná proměnná Nonce), dochází k zápisu do blockchainu. Hash je přitom jakýsi kód o fixní délce, který vzniká působením hashovací funkce na vstupní údaje. Hashovací funkce je jednosměrná, tzn. že z hashe prakticky nelze rekonstruovat obsah

vstupních dat, a zároveň prostá, tzn. že pro dvojice různá vstupní data vznikají dva různé hashe. Dokonce již nepatrná změna vstupních dat navíc vede ke vzniku diametrálně odlišných hashů. Za takto vytěžený blok získává těžař odměnu ve formě Bitcoinů, přibližně 6,25 BTC. Tato odměna přitom těžaři slouží jako jakási kompenzace za výpočetní výkon, resp. propálenou elektřinu, který musel být poskytnut jeho strojem pro nalezení správné proměnné Nonce, resp. hashe bloku. Bitcoinový rozhodně nemá smysl těžit prostřednictvím běžných počítačů, neboť výpočetní výkon potřebný pro nalezení správné proměnné Nonce je tak velký, že by to těmto prostředkům trvalo přibližně celý lidský život, či ještě déle. Na úplných počátcích těžby to však možné bylo. Dlouhou dobu pak probíhala těžba na grafických kartách, např. GTX 1080<sup>ti</sup>, či tzv. rigách, které představují spojení několika grafických karet dohromady. Ani tento způsob těžby však v současné době již není efektivní, nyní tak těžba probíhá pomocí speciálního hardwaru. Jednotliví těžaři se navíc shlukují do jakýchsi těžařských spolků, tzv. minig poolů. Každý minig pool přitom představuje jeden z těžařských uzlů sítě (angl. miner node). Celá síť aktuálně čítá 15 těžařských uzlů, jmenovitě BTC.com, F2Pool, AntPool, Poolin, BTC.TOP, SlushPool, ViaBTC, unknown, BitFury, Bixin, BitClub, Huobi.pool, okpool.top, Bitcoin.com, WAYI.CN, KanoPool, tigerpool.net. Pokud některý z minig poolů vytěží blok, dělí se odměna mezi jednotlivé těžaře podle toho, kolik výpočetního výkonu na vytěžení poskytli [5].

### 1.3 Potvrzování transakcí

Jak už bylo řečeno, blockchain je plně decentralizovaný, tudíž ve finančním systému kryptoměn neexistuje žádná centrální autorita, která by potvrzovala transakce, které přirozeně jednotliví uživatelé sítě provádí, tak jak je tomu např. u běžných bankovních systémů. Toto potvrzení je tak nutné zajistit prostřednictvím jiných mechanismů a sice tzv. algoritmů konsenzu. Jedná se o algoritmy, které na základě shody velkého množství těžařských a úplných uzlů (angl. full node) poskytují jakýsi matematický důkaz, jimž je transakce potvrzena. V dnešní době již existuje nepřeberné množství konceptů, které se snaží o decentralizovaný konsenzus, mezi ty nejpoužívanější však patří Proof of Work, Proof of Stake, či Proof of Space and Time.

- *Proof of Work* je systém využívaný Bitcoinem. Jedná se o způsob validace, který vzniká nalezením správného hashe pro vytvoření bloku, tj. vytěžení Bitcoinu. Jak bylo uvedeno, tvorba takového důkazu, tj. těžba Bitcoinu, je velmi náročná a vyžaduje obrovský výpočetní výkon, ale je velmi snadné ji ověřit. Pokud by se totiž některý z těžařů rozhodl „podvádět“ a podstrčil do systému nějakou transakci, celý algoritmus by ostatním těžařům okamžitě přestal fungovat, na základě čehož by byl viník velmi rychle odhalen. Proof of Work je tedy mechanismus založený na shodě těžařských uzlů.
- *Proof of Stake* je alternativa systému Proof of Work, kterou využívá např. kryptoměna Ethereum. Tento systém funguje na základě shody nikoliv těžařských, nýbrž úplných uzlů. Úplné uzly tvoří tzv. validátoři, kteří uzamčením jistého množství svých coinů získávají právo tvořit nové bloky, resp. ověřovat transakce. Za tuto práci přitom, stejně jako těžaři, získávají transakční poplatky. Šance validátora na potvrzení dané transakce, je však tentokrát dána čistě množstvím uzamčených peněz. Platí, že, čím více svých coinů validátor uzamkne, tím větší šanci má, že bude vybrán pro potvrzení dané transakce. Kromě toho základního konceptu existují i různé další variace Proof of Stake mezi něž patří např. Byzantine Fault Tolerance, či tzv. Delegated Proof of Stake a mnohé další [6].
- *Proof of Space and Time* je v současné době nejnovější systém využívaný kryptoměnou Chia. Jedná se o techniku složenou ze dvou konceptů - *Proof of Space* a *Proof of Time*. Proof of Space je

systém, kde důkaz vzniká na základě uzlů tvořených tzv. farmáři. Při vyslání signálu ke tvorbě nového bloku v blockchainu je přitom pravděpodobnost potvrzení transakce daným farmářem přímo úměrná procentu úložného prostoru, které má farmář k dispozici na svém pevném disku vzhledem k celé síti. Aby se však skutečně jednalo o algoritmus konsenzu musí být Proof of Space svázaný s Proof of Time, který zajišťuje, že těžba je časově konzistentní. Výhody systému spočívají v jeho ekologičnosti, neboť v porovnání s Proof of Work, či Proof of Stake spotřebovává mnohem méně energie. Zároveň však jeho prudký rozvoj vyvolal přehnaný zájem o pevné disky SSD a s tím i jejich aktuální nedostatek na trhu [7].

## 1.4 Uložení Bitcoinů

Bitcoin lze, obdobně jako klasické měny, uchovávat různými způsoby, všechny jsou však jistou formou digitální peněženky. *Kryptoměnová digitální peněženka* je softwarový program, který pomáhá usnadnit aktualizace hlavní knihy blockchainu při provádění transakcí. Tento program je zabezpečený sadou veřejných a soukromých klíčů. Veřejné klíče identifikují peněženku jako takovou, umožňují přijímání coinů bez toho, aniž by musela být odhalena identita příjemce. Jedná se o jistou analogii k uživatelským jménům, resp. číslům účtu. Soukromé klíče pak umožňují uživateli přístup k peněženke, iniciují transakce a další. Tedy jsou jistou alternativou přihlašovacích údajů do internetového bankovníctví nebo PINu ke kartě. Při ztrátě soukromého klíče, již není možné se k uloženým coinům dostat. Obecně lze rozlišovat mezi dvěma základními typy digitálních peněženek – tzv. horkou a studenou.

- *Studená peněženka* (angl. cold storage) je offline digitální peněženka, která je umístěna na hardwaru počítače. Veřejné i soukromé klíče jsou tak uloženy v bezpečí mimo internet. Jedinou možností, jak ztratit klíč, je fyzická ztráta počítače, či selhání pevného disku. Studená peněženka je proto vhodná pro skladování velkých finančních objemů [8].

Speciálním typem studené peněženky je pak peněženka papírová. Soukromé klíče a odpovídající QR kódy má každý uživatel fyzicky u sebe - vytištěné na kusu papíru. Pro generování soukromých klíčů je však nutno navštívit speciální web. Před samotným generováním se tedy doporučuje na daném počítači zkontrolovat malware a během generování pak odpojit přístup k internetu [9].

- *Horká peněženka* (angl. hot storage) je naopak online digitální peněženka, která je umístěna na internetu. Umožňuje tak sice rychlou manipulaci s coinem, avšak je zároveň zranitelná vůči napadení hackerů a krádeži coinů. Veřejné a soukromé klíče jsou totiž umístěny přímo na internetu. Horká peněženka se proto obvykle používá pouze na uchování malého množství kryptoměn.

Za speciální typ horké peněženky lze považovat i uložení coinů na některé z kryptoměnových burz mezi které např. patří Binance, Bitstamp, Coinbase Pro, Kraken, Bittrex a jiné. Tyto společnosti přitom ukládají coinu uživatelů do své online infrastruktury, takže napadením některého z těchto serverů hackery může opět velmi snadno dojít k jejich ztrátě. I zde se tedy doporučuje uchovávat pouze malé finanční objemy.

Uložení Bitcoinů pomocí výše zmíněných peněženek nepodléhá tradičnímu systému bankovních poplatků a limitů. To znamená, že nejsou účtovány žádné poplatky za založení peněženky, vklad či samotné uložení coinů, neexistují žádné minimální či maximální částky, která zde mohou být uloženy, či s jakými může být nakládáno. Náklady na samotné transakce jsou potom trvale udržovány velmi nízko, což je umožněno právě díky tomu, že do oběhu není zapojena žádná zprostředkující instituce (banka, resp. vláda) [10].

## 1.5 Výhody a nevýhody kryptoměn

Hlavní výhoda kryptoměn jako takových tedy spočívá v jejich autonomii, diskrétnosti, odolnosti vůči padělání, manipulaci a cenzuře. Vlastníci kryptoměn jsou schopni kontrolovat, jak utrácejí své peníze bez nutnosti kontaktu se zprostředkujícím orgánem, což kryptoměny činí autonomní. Pokud uživatel dobrovolně nezveřejní své kryptoměnové transakce, pak nejsou tyto transakce nikdy spojeny s jeho osobní identitou, tzn. nákupy a prodeje kryptoměn jsou diskrétní. Vzhledem ke způsobu, jakým jsou kryptoměny zabezpečeny, je prakticky nemožné kryptoměny padělat nebo s nimi platit více než jedenkrát. Díky decentralizaci je navíc celý systém odolný vůči manipulaci a cenzuře. Tyto a mnohé další výhody kryptoměn jsou hlavní motivací, proč do kryptoměn investovat [11].

Kryptoměny mají však i své nevýhody. Jak bylo již dříve zmíněno, blockchain kryptoměn je sice vysoce zabezpečený, to však již neplatí o jiných aspektech celého ekosystému, a sice horkých peněženkách a burzách, které nejsou imunní vůči hackerům. Zvláště v poslední době je také široce diskutována poměrně velká volatilita kryptoměn, obzvláště pak Bitcoinu. S tím souvisí i obavy o rychlosti, s jakou lze kryptoměny vyměnit za jinou měnu, která tím pádem také velmi kolísá. V neposlední řadě jsou kryptoměny vzhledem ke své poloautonomní povaze vhodnými kandidáty např. legalizaci výnosů z trestné činnosti („praní špinavých peněz“), či pro financování kriminálních aktivit [3].

## 1.6 Budoucnost kryptoměn

Nabízí se otázka, zda by se kryptoměny v budoucnu mohly stát plnohodnotným platidlem, či zcela mohly nahradit běžné fiatové měny? Pokud by se tak stalo, vyžadovalo by to vyvinutí úplně nových infrastruktur. Hotovost by se navíc poměrně rychle stala nekompatibilní a mnoho lidí by tak přišlo o svá aktiva. Finanční instituce by pravděpodobně z velké části zanikly, či by musely rapidně změnit své dosavadní způsoby. V neposlední řadě by pak zavedením kryptoměn jakožto plnohodnotných platidel ztratily vlády příslušných států téměř veškeré pravomoci zasahovat do finančního dění. Snaha všech těchto orgánů tak míří zcela proti myšlence zavedení kryptoměn jako reálných platidel a vzhledem k jejich vlivu, je opravdu otázkou, zda někdy v tomto ohledu dojde ke změně [12].

## Kapitola 2

# Finanční trhy

Obchodování je základní ekonomický koncept označující nákup a prodej aktiv. Termín denní obchodování se běžně používá k označení procesu, kdy obchodníci aktivně nakupují a prodávají aktiva v relativně krátkých časových horizontech, a to na základě krátkodobých výkyvů v jejich cenách. Ve skutečnosti však termín obchodování může odkazovat na širokou škálu různých strategií. Termín investování naproti tomu označuje obchodování v delších časových horizontech, kdy investoři provádějí své nákupy a prodeje bez ohledu na krátkodobé výkyvy v jejich cenách.

V kontextu finančních trhů se aktiva, s nimiž je obchodováno, nazývají *finančními instrumenty*. Finanční instrumenty lze dělit na peněžité (měny, komodity, cenné papíry, akcie, dluhopisy), s nimiž se obchoduje na tzv. spotových trzích, a derivátové (finanční deriváty), obchodované na tzv. termínových trzích [13].

### 2.1 Býčí a medvědí trhy

Dle vývoje ceny finančního instrumentu rozlišujeme mezi trhy *býčími* a *medvědími*. Býčí trh je rostoucí trh, tj. cena daného aktiva má vzestupný trend. Tento vzestup je přitom způsoben postupně se zvyšující důvěrou obchodníků v dané aktivu - jejich "býčí" náladou. Naproti tomu medvědí trh je trhem klesajícím, kdy cena daného aktiva stabilně klesá, investoři jsou vůči danému aktivu nedůvěřiví. U obou trhů však může docházet ke krátkodobým pohybům cen opačným směrem, tzn. býčí trh může obsahovat píky spojené s poklesem ceny, a naopak medvědí trh píky spojené s růstem ceny. Uvedená klasifikace trhu je otázkou perspektivy, cenový trend závisí na časovém rámci, který obchodník sleduje [14] [15].

### 2.2 Technická a fundamentální analýza

Na dění na finančních trzích a jejich budoucí vývoj lze nahlížet dvěma možnými způsoby a sice prostřednictvím *základní* nebo *technické* analýzy trhu. Základní analýza představuje posouzení ceny finančního instrumentu na základě makroekonomických okolností, jako je např. stav širší ekonomiky, či podmínky daného odvětví. V případě kryptoměn může fundamentální analýza představovat analýzu transakcí, sledování rychlosti sítě, nejvýznamnějších držitelů, či počtu aktivních adres. Tyto základní faktory však mají, vzhledem ke spekulativní povaze kryptoměn, na jejich cenu pouze zanedbatelný vliv. Proto se základní analýza pro posouzení ceny kryptoměn zatím příliš nepoužívá. Technická analýza se zabývá historií obchodování s daným finančním instrumentem, což zahrnuje analýzu grafů popisujících historický vývoj jejich cen, analýzu obchodovaných cen a objemů, či použití jiných technických indikátorů. Kromě předpovídání budoucího vývoje trhu může technická analýza sloužit i jako užitečný nástroj pro zajištění a řízení rizik. Tento druh analýzy je hojně využíván na kryptoměnových trzích [13].

## 2.3 Obchodování

Pojem otevření pozice na trhu vyjadřuje nákup nebo prodej finančního instrumentu. Velikost pozice je přitom přímo úměrná velikosti obchodu. Seznam všech nevyřízených obchodů tvoří tzv. *knihu objednávek* (angl. order book), která se skládá ze dvou částí – *poptávkové* (angl. ask) a *nabídkové* (angl. bid). V případě, že obchodník požaduje určité množství aktiva za cenu, která není aktuálně na trhu dostupná, je jeho objednávka zařazena do knihy poptávek. Naopak pokud obchodník nabízí dané aktivum za cenu, za kterou ho momentálně nikdo nepožaduje, je objednávka přidána do knihy nabídek. Rozdíl mezi nejnížší cenou nabídky a nevyšší cenou poptávky se nazývá *spread*. Na základě velikosti spreadu je měřena míra likvidity trhu, přičemž platí, že čím je spread menší, tím je trh likvidnější a naopak. Zlikvidováním pozice je pak myšleno automatické odstranění pozice z trhu a to na základě nedostatečného zajištění (kolaterálu) [13].

*Likvidita* trhu je jednou z důležitých charakteristik finančních trhů a vyjadřuje rozsah, v jakém trh umožňuje nákup a prodej aktiv za spravedlivé ceny, tj. ceny blízké vnitřní hodnotě aktiva. Vysoce likviditní trh tedy umožňuje snadný nákup a prodej aktiva, tzn. jedná se o trh s velkou mírou nabídky a poptávky (typicky měnové trhy). Trh s nízkou likviditou se naproti tomu vyznačuje obtížným nákupem a prodejem daného aktiva [16]. S pojmem likvidita je úzce spjatý pojem hloubka trhu vyjadřující objem aktiva, se kterým je možné obchodovat, aniž by došlo ke zhodnocení ceny. Jinými slovy čím hlubší trh, tím více nevyřízených pokynů na straně nabídky i poptávky [17]. Mezi další důležité charakteristiky finančních trhů patří tzv. *volatilita*. Volatilita označuje kolísavost, tzn. rychlost a velikost změn, cen daných aktiv [18].

Obchodníci jsou na základě svého tržního chování označováni buďto jako *příjemci* (angl. taker) nebo *tvůrci* (angl. maker) likvidity. Zadáním objednávky s požadavkem na okamžité vyplnění se obchodník staví do pozice tzv. příjemce likvidity. Svým pokynem snižuje tržní likviditu, neboť odebírá objednávky z knihy objednávek. Naopak zadáním objednávky, která není vyplněna okamžitě, se obchodník stává tzv. tvůrcem likvidity. Na základě svého pokynu zvyšuje likviditu trhu, neboť zvětšuje objem knihy objednávek [19].

Na klasických i kryptoměnových trzích lze provádět nejrůznější druhy objednávek, jmenovitě se jedná o tržní, limitní a tzv. stop-loss objednávky.

- *Tržní objednávka* (angl. market order) je typ objednávky, při které dochází k okamžitému nákupu, resp. prodeji aktiva za nejlepší aktuálně dostupnou cenu. S tržními objednávkami je spojené riziko tzv. skluzu, neboli možného rozdílu mezi cenou objednávky, kterou obchodník očekává a cenou, za kterou je objednávka vyplněna. Pokud totiž obchodník požaduje větší množství aktiva, než je při dané ceně aktuálně dostupné množství, je zbytek objednávky vyplněn z nabídek umístěných výše v knize objednávek, tj. za vyšší cenu. Tržní objednávky snižují likviditu daného trhu.
- *Limitní objednávka* (angl. limit order) je naopak druh objednávky, kdy obchod není proveden okamžitě. Je stanovena limitní cena obchodu a příkaz k nákupu, resp. prodeji, aktiva umístěn do knihy objednávek. Limitní příkaz zůstává otevřený až do chvíle, kdy tržní cena dosáhne požadované limitní ceny (nebo lepší), pak je vykonán. Limitní objednávka tak na jednu stranu zaručuje, že obchod nebude proveden za horší cenu než obchodník požaduje, na stranu druhou však může zůstat nevyplněná, neboť trh nikdy nemusí dosáhnout stanovené ceny. Limitní objednávky obvykle zvyšují likviditu trhu.
- *Stop-loss objednávka* je druh objednávky, k jejíž aktivaci dochází ve chvíli, kdy cena aktiva dosáhne tzv. stop ceny. Po dosažení této ceny je provedena buďto tržní nebo limitní objednávka. V obdobích extrémních tržních podmínek (prudkých růstů/poklesů cen aktiv) může nastat situace,

kdy i přes dosažení stop ceny, zůstává stop-limitní objednávka nevyplněná. Pro zajištění, že po dosažení stop ceny dojde k vyplnění objednávky, je vhodné využít stop-tržní objednávky [21].

Pokud chce obchodník provést objednávku větší, než na kterou stačí jeho dosavadní kapitál, lze využít tzv. obchodování s marží (angl. margin trading). Jedná se o metodu obchodování s aktivy založenou na využití finančních prostředků poskytnutých třetí stranou. *Marže* označuje množství kapitálu, se kterým obchodník otevírá pozici a *páka* je částka, kterou svou pozici zesiluje. Tzn. použití dvojnásobné páky znamená otevření 2x větší pozice než na kterou stačí obchodníkův kapitál. Zatímco na klasických trzích jsou vypůjčené prostředky poskytovány finančním makléřem, v případě kryptoměnových trhů jsou prostředky poskytovány jinými obchodníky, kteří za ně získávají finanční kompenzace ve formě úroků, případně samotnou burzou [13].

## 2.4 Spotové trhy

Spotový trh je trh, kde dochází k okamžitému vypořádání sjednaného obchodu, tzn. finanční instrumenty jsou dodány okamžitě po jejich nákupu. Tržní cena aktiv, tj. cena za kterou je aktivum nakupováno, resp. prodáváno, je v daném okamžiku určována na základě rovnováhy mezi jeho nabídkou a jeho poptávkou [20].

Kromě již dříve uvedených druhů objednávek lze na spotových trzích provádět ještě tzv. *OCO objednávky* (angl. One cancels the other). Jedná se o druh objednávky, která kombinuje limitní a stop-limitní objednávku, přičemž jakmile dochází k částečnému nebo úplnému vyplnění jedné z nich, druhá je automaticky zrušena. OCO objednávky mohou být zadány buďto jako prodejní nebo nákupní objednávky a nejčastěji se využívají pro zajištění zisků za současné minimalizace potenciálních ztrát.

**Příklad 1** Předpokládejme, že obchodník nakoupí 1 BTC za cenu 50 000 USDT a očekává, že cena BTC poroste. Aby si v případě dobré predikce zajistil zisk, nastaví limitní prodejní objednávku např. na 52 000 USDT, naopak aby se vyvaroval případným ztrátám, nastaví stop-limitní objednávku na 49 000 USDT. Pokud cena vzroste, jak předpokládal, provede se příkaz k prodeji BTC, který zároveň zruší stop-limitní příkaz. Naopak pokud cena poklesne, stop-limitní příkaz zajistí, že BTC nebude prodán za cenu nižší, než je nastavená limitní cena stop-limitní objednávky [21].

Obchodování na klasických a kryptoměnových spotových trzích se v zásadě nijak neliší, řídí se stejnými pravidly a zásadami.

Na Binance spotovém trhu se obchoduje s nejrůznějšími kryptoměnami jejichž cenový pohyb je vztažen k ceně kryptoměny BNB, Bitcoinu, Ethereum, USDT či BUSD. Kromě tradičního spotového obchodování je k dispozici také obchodování s marží poskytované ve dvou režimech – izolovaném (isolated margin mode) a křížovém (cross margin mode). Obchodování s izolovanou marží znamená, že každému měnovému páru, se kterým je obchodováno přísluší vlastní účet marže. Na konkrétním účtu lze pak držet a půjčovat pouze kryptoměny, kterých se účet týká. Je možné otevřít i několik účtů izolované marže najednou, přičemž jednotlivé pozice jsou na sobě nezávislé. To znamená, že v případná likvidace některé z otevřených pozic nebude mít vliv na ostatní otevřené pozice. Obchodování s křížovou marží naproti tomu znamená držení pouze jednoho účtu marže, který přísluší všem dostupným měnovým párům. Kryptoměny na účtu křížové marže jsou sdíleny všemi pozicemi, pozice jsou tak na sobě závislé. V případě likvidace dochází ke zlikvidování všech otevřených pozic [22].

## 2.5 Termínové trhy

Termínový trh je trh, kde probíhají tzv. termínové obchody, tj. obchody u kterých dochází ke zpoždění mezi jejich sjednáním a plněním. Na termínových trzích se obchoduje s nejrůznějšími druhy finančních derivátů (kontraktů).

### 2.5.1 Finanční deriváty

Finančními deriváty (kontrakty) nazýváme finanční instrumenty, jejichž cena je odvozena od hodnoty *podkladového aktiva*. Mezi podkladová aktiva patří typicky akcie, měny, nejrůznější druhy komodit (např. ropa, zlato, rýže), úrokové platby nebo dokonce dluhopisy. Obecně rozlišujeme dva základní typy derivátů - podmíněné (opční) a nepodmíněné (pevné). Podmíněné finanční deriváty, opce, se vyznačují tím, že uskutečnění obchodu závisí na vůli pouze jedné strany. U nepodmíněných finančních derivátů, futures a forward kontraktů, vzniká povinnost uskutečnit daný obchod pro obě strany. Vypořádání obchodů probíhá fyzickou, či finanční směnou podkladového aktiva nebo pouze vyrovnáním rozdílu mezi aktuální a sjednanou cenou [23].

### 2.5.2 Forward kontrakty

Forward je finanční kontrakt, smlouva, mezi dvěma stranami o nákupu, resp. prodeji, podkladového aktiva. Smlouva předem určuje množství a kvalitu aktiva, celkovou cenu a datum, ke kterému bude aktivum dodáno, tzv. *expirační dobu*. Podmínky smlouvy nejsou u forward kontraktů nijak standardizovány, vše záleží na domluvě mezi prodávajícím a kupujícím. Forward kontrakty jsou obchodovány na OTC trzích (angl. Over the counter market) neboli přímo „za přepážkou“. Jejich likvidita je tak poměrně nízká a zároveň není nijak zajištěno tzv. kreditní riziko, tj. nemožnost naplnění závazku v důsledku zhoršení finanční situace jedné ze zúčastněných stran. K vypořádání forward kontraktů dochází právě jednou a to k datu uvedenému ve smlouvě [23].

Historie termínového obchodování sahá až do 17. století. Tou dobou se v Japonsku oběhuje Dōjima Rice Exchange, trh, který vzniká za účelem zajištění rizik spojených s pěstováním a prodejem rýže. Na jedné straně zde stojí japonský pěstitel rýže a na straně druhé vlastník sítě japonských obchodů. Zatímco pěstitel potřebuje určitý kapitál, ze kterého by financoval náklady spojené se sazbou, ošetřováním a sklizní rýže, obchodník potřebuje mít v budoucnu k dispozici určité množství rýže, které bude moci dále prodat. Za těmito účely tak mezi oběma stranami vzniká dohoda o prodeji, resp. nákupu patřičného množství rýže za předem domluvenou cenu, která je zapečetěna smlouvou (dnešními slovy forward kontraktem). Tato smlouva sloužila pěstiteli jakožto jistina. Na jejím základě získal v bance půjčku, díky níž byl následně schopen financovat náklady spojené s pěstováním rýže. Naopak obchodníkovi smlouva poskytovala jistotu, že v budoucnu obdrží potřebné množství rýže, a to právě za smlouvenou cenu.

Od prvních termínových obchodů uplynulo mnoho let, avšak jejich princip zůstal zachován až do dnešních dob. Hlavní funkcí forward kontraktů, stejně tak jako prvních obchodů s rýží, je tedy zajištění rizik neboli tzv. *hedging* [24].

### 2.5.3 Klasické futures kontrakty

Stejně jako forward kontrakty jsou i futures kontrakty (jistým) závazkem mezi dvěma stranami o nákupu, resp. prodeji, podkladového aktiva a to v předem stanoveném množství, kvalitě, ceně a s příslušnou dobou expirace. K obchodování s futures kontrakty však dochází na organizovaných burzách, které pevně stanovují a hlídají podmínky těchto obchodů. Futures kontrakty jsou tak, narozdíl od forward kontraktů,



tzv. standardizovány. Burzovní obchodování zároveň zajišťuje jejich vysokou likviditu a absenci kreditního rizika, neboť v případě platební neschopnosti jedné ze zúčastněných stran dopláci rozdíl sama burza [23].

### 2.5.3.1 Funkce

Vraťme se nyní zpět k příkladu japonského pěstitele rýže a japonského obchodníka. Postupem času, se ukázalo, že termínový obchod nemusí být vždy výhodný pro obě strany. Po uzavření smlouvy totiž mohly nastat mj. 2 situace. Buď byla v daném roce úroda rýže malá, což vedlo k nárustu ceny, nebo naopak velká, což způsobilo pokles ceny. Aby se znevýhodněná strana vyvarovala případné ztrátě začalo se obchodovat se samotnými kontrakty, nikoliv přímo podkladovými aktivy.

Předpokládejme, že v daném roce nebyly podmínky pro pěstování rýže příliš příznivé, a tak se očekávalo, že její úroda bude nízká. V důsledku toho byla hodnota kontraktu v den jeho expirace výrazně odlišná oproti hodnotě, za kterou byl uzavřen. Co tedy mohl s uzavřeným kontraktem udělat pěstitel, aby netratil, resp. obchodník, aby naopak vydělal? Pěstitel mohl uzavřený kontrakt prodat jinému farmáři a tím se zbavit svého závazku. To mu v konečném důsledku umožnilo prodávat v budoucnu vlastní rýži za aktuální, vysokou, cenu. Naproti tomu obchodník mohl nasmlouvat obrovské množství kontraktů, čímž si v budoucnu zajistil výrazně nižší cenu rýže oproti ceně aktuální.

Pokud byly naopak podmínky příznivé a předpokládala se velká úroda rýže, pak, co mohl s uzavřeným kontraktem udělat pěstitel, aby vydělal, resp. obchodník, aby netratil? Pěstitel mohl nasmlouvat velké množství kontraktů, čímž si zajistil, že rýži prodá za vyšší cenu než bude její cena v budoucnu. Obchodník naopak mohl všechny uzavřené kontrakty prodat jiným obchodníkům a tím se vyvarovat nákupu rýže za vyšší cenu [24].

Výše uvedený příklad ilustruje jednu z hlavních funkcí futures kontraktů a sice možnost spekulace nad budoucí cenou příslušného aktiva. Spekulace se přitom realizuje pomocí tzv. *pákového efektu*, tzn. možnosti složit při otevření obchodu pouze jistou část z celkové ceny futures kontraktu. Tato částka je pak po celou dobu obchodování blokována. Obchodování s futures kontrakty tedy umožňuje ovládat mnohonásobně větší objem podkladového aktiva, než na který stačí obchodníkův kapitál. Mezi další funkce patří, stejně jako u forward kontraktů, zajištění a řízení rizik, tj. nákup futures kontraktu za účelem zmírnění konkrétního rizika, typicky změny měnových kurzů. Futures kontakty dále umožňují nepřímé obchodování s aktivy. Poslední a zároveň jednou z nejdůležitějších funkcí futures kontraktů je stanovování cen aktiv. Trhy s futures kontrakty totiž splňují požadavky nabídky a poptávky po nejrozličnějších aktivech. Velikost nabídky, resp. poptávky, je pak tím, co stanovuje výslendou cenu aktiv [25].

### 2.5.3.2 Obchodování s klasickými futures kontrakty

Jak už bylo řečeno, specifika obchodovaných futures kontraktů nastavuje burza. Každý futures kontrakt má jasně danou velikost, vždy dochází k nákupu celého futures kontraktu nebo případně jeho násobku. Např. jeden futures kontrakt na ropu typicky představuje 1000 barelů ropy, což je v přepočtu asi 120 000 litrů. U každého futures kontraktu je navíc jasně definována velikost páky, se kterou lze obchod provést. Pevně nastaven je i poslední možný den obchodování s daným kontraktem neboli expirační doba. Standardní doba trvání jednoho futures kontraktu je měsíc, čtvrt nebo půl roku, kontrakty je však možné nakoupit i na několik let dopředu. Cena futures kontraktu, tj. cena bez využití pákového efektu, je pak dána aktuální cenou podkladového aktiva v den otevření obchodu s jemu příslušným futures kontraktem. Burza dále určuje peněžité vyjádření cenové kótace, tj. v jakých jednotkách, případně centech, je vyjádřena cena příslušné části kontraktu na grafu. Graf zachycující cenu ropy tak například vyjadřuje cenu jednoho barelu v jednotkách amerických dolarů a centech. Mezi další specifika kontraktů patří i jejich minimální cenová fluktuace, tj. velikost změny v ceně na základě které dochází k přičtení, resp.

odečtení, částky na účtech účastníků trhu. V neposlední řadě burza stanovuje symbol kontraktu pod kterým je veden jemu příslušný trh (např. pro ropu CL), hlavní obchodní hodiny, premarket a aftermarket. Premarket a aftermarket jsou trhy, na kterých je sice možné obchodovat i mimo hlavní obchodní hodiny, avšak s mnohem menšími objemy. Obchodování dle stanovených pravidel garantují v každé zemi, kde k burzovnímu obchodování dochází, příslušné komise.

Obchodování s klasickými futures kontrakty vždy zajišťuje *finanční makléř* (angl. broker), který si za tuto službu účtuje patřičný poplatek, tzv. *brokerskou komisi* (angl. brokerage fee). Samotný obchodník tak k trhu nemá přímý přístup, pouze na základě vlastních analýz vysílá příkazy k nákupu, resp. prodeji, daného aktiva brokerovy a ten za něj obchod vyjedná. Vzhledem k povaze (hlavní funkci) futures kontraktů je jasné, že podkladové aktivum nemusí v době uzavření obchodu ani existovat, nebo-li prodávající nemusí aktivum vůbec vlastnit a nakupující ho nemusí vůbec potřebovat. Nákupem, resp. prodejem, futures kontraktu na dané aktivum pouze spekulují nad jeho budoucí cenou [25].

Při obchodování s futures kontrakty je možné zaujmout jednu ze dvou pozic - *krátkou* (angl. short) nebo *dlouhou* (angl. long). Otevření pozice short znamená prodej futures kontraktu na příslušné podkladové aktivum, obchodník spekuluje na pokles jeho ceny, tj. následuje taktiku prodat futures kontrakt dříve a nakoupit ho levně. Naopak otevření pozice long označuje nákup futures kontraktu na dané podkladové aktivum, obchodník spekuluje na růst jeho ceny, tj. řídí se principem nakoupit futures kontrakt levně a prodat dříve. Obě pozice lze následně uzavřít a sice otevřením pozice opačné. Nová pozice však musí být otevřena pro stejné podkladové aktivum a se shodnou expirační dobou jako pozice původní. K uzavření pozice short tak dochází nákupem futures kontraktu neboli otevřením pozice long a naopak. Na trhu s futures kontrakty je samozřejmě možné držet i více pozic najednou [26].

Futures pozice jsou do jisté míry kryty systémem marží. Pro otevření obchodu (libovolné pozice) a jeho držení je totiž zapotřebí složit na burzovní účet zálohu, tzv. *initial margin*, závislou na definované velikosti páky pro daný obchod. Tato záloha slouží jako zajištění, aby obě strany byly schopny dostát svým závazkům. Výše initial marginu (resp. možnosti páky) vychází z maximální částky, kterou může obchodník u daného futures kontraktu s určitou mírou pravděpodobnosti za den prodělat a je pevně stanovena burzou nebo finančním makléřem. Obecně platí, že čím rizikovější trh je, tím je tato částka (resp. možnost páky) vyšší. Initial margin tak mj. závisí na aktuálních politických či mikroekonomických podmínkách, které rizikovost trhu ovlivňují. Pokud účet neobsahuje dostatečný obnos sloužící pro pokrytí initial marginu, finanční makléř pozici nedovolí otevřít. Během obchodování je pak důležité výši této zálohy hlídat, tak, aby neklesla pod marži udržovací, tzv. *maintenance margin*. Výše maintenance marginu se mění v závislosti na aktuální tržní ceně a zůstatku na účtu. Případný pokles účetního zůstatku pod hranici maintenance marginu totiž vede k automatické likvidaci pozice na trhu a další sankcím [27].

K obchodování s futures kontrakty dochází na denní bázi. Na základě vývoje ceny daného aktiva, nastavené minimální cenové fluktuaci a počtu otevřených pozic jsou na konci každého obchodního dne připsány zisky, resp. odepsány ztráty na účty všech obchodníků. Při růstu ceny daného aktiva přitom vydělávají obchodníci na pozici long a při poklesu ceny naopak obchodníci na pozici short. Výše celkového denního zisku nebo ztráty je dána velikostí tzv. ticku, který představuje změnu ceny v desítkách dolarů a to v závislosti na počtu otevřených pozic. Např. pokud dojde ke změně ceny ropy, kde je nejmenší cenový pohyb nastaven na \$0,01 za barel, a to z 49,49 dolarů na 49,50 dolarů, pak jeden tick představuje, při držení jednoho kontraktu, zisk 10 dolarů. Na konci každého obchodního dne zároveň dochází ke kontrole účetních zůstatků a případné likvidaci pozic.

V den expirace daného futures kontraktu, někdy i dříve, jsou všechny obchody uzavřeny. K tomu dochází nákupem kompenzujících kontraktů všech účastníků trhu. To znamená, že obchodníci, kteří podkladové aktivum nabízeli ho nakoupí a ti, kteří ho poptávali, ho prodají. V praxi tak ke skutečnému vypořádání futures kontraktů nikdy nedochází.

### 2.5.3.3 Druhy klasických futures kontraktů

Na klasických trzích lze obchodovat s nepřeberným množstvím futures kontraktů, jednotlivé druhy tak dělíme dle typu podkladového aktiva do následujících kategorií [25]:

1. Komoditní futures kontrakty (podkladovým aktivem ropa, zlato, rýže apod.)
2. Akciové futures (podkladovým aktivem akcie nebo akciové indexy)
3. Úrokové futures (podkladovým aktivem různé druhy dluhových instrumentů)
4. Měnové futures (podkladovým aktivem měna jejíž cenový pohyb je vztažen k pohybu amerického dolaru)

### 2.5.4 Kryptoměnové futures kontrakty

Kryptoměnové futures kontrakty se řadí mezi měnové kontrakty, avšak obchodování s nimi se v některých aspektech od obchodování s klasickými měnovými kontrakty výrazně liší. Kryptoměnové futures kontrakty představují reprezentaci smluvního obchodu, kde podkladová aktiva tvoří nejrozličnější druhy kryptoměn (např. Bitcoin, Ethereum, Dogecoin). Cenový pohyb podkladových aktiv je přitom vztažen buď k pohybu theteru USDT, či samotného amerického dolaru USD.

Kryptoměnové futures kontrakty lze dělit buď to na základě jejich doby expirace - *čtvrtletní futures kontrakty* (angl. *Quartely futures contracts*) a *pevné futures kontrakty* (angl. *Perpetual futures contracts*) nebo dle aktiva v němž jsou kótovány a vypořádávány - *USDT-margined futures kontrakty* a *COIN-margined futures kontrakty* [13].

#### 2.5.4.1 Trvalé futures kontrakty (angl. *Perpetual futures contracts*)

Trvalé futures kontrakty jsou typem kryptoměnových futures kontraktů, které nemají datum expirace. Ačkoliv obchodníkům umožňují držet pozice „věčně“, využívají se spíše pro krátkodobé obchodování. V principu je totiž obchodování s trvalými futures kontrakty podobné obchodování na trhu spotovém. Díky možnosti pákového efektu je však možná realizace velkých obchodů, které nejsou plně kryty počátečním kapitálem, což obchodování na spotových trzích v zásadě neumožňuje.

Cena futures kontraktů na danou kryptoměnu navíc nemusí být obecně stejná jako její cena spotová. Je však žádoucí, aby k sobě tyto dvě ceny dlouhodobě konvergovaly a obchodování na trzích s trvalými futures kontrakty tak, co nejvíce napodobovalo obchodování na trhu spotovém. Za tímto účelem používají kryptoměnové trhy mechanismus zvaný financování sazeb. Financováním nazýváme pravidelné platby přímo mezi obchodníky, kteří mají v daných časech na trhu otevřené pozice. Platby probíhají v osmi-hodinových intervalech (00:00, 08:00 a 16:00 UTC), přičemž samotná burza si neúčtuje žádné poplatky za jejich převod. Míra toho financování (angl. *funding rate*) je dána výší paušální úrokové sazby (pevně nastavena na 0,03% za den, tj. 0,01% na interval financování) a tzv. premiovým indexem (angl. *premium index*), závislém na rozdílu mezi cenami na trhu s trvalými futures kontrakty a cenami spotovými. Pozitivní míra financování značí, že ceny futures kontraktů jsou vyšší než spotové ceny příslušných kryptoměn. Obchodníci na pozici long (nakupující) musí platit obchodníkům na pozici short (prodávajícím), což vede uzavírání nevýhodných pozic long a otevírání pozic short. V konečném důsledku pak díky tomu dochází k poklesu cen futures kontraktů. Negativní míra financování naopak vyjadřuje nižší cenu futures kontraktů oproti cenám spotovým, obchodníci na pozici short (prodávající) pak platí obchodníkům na pozici long (nakupující), to následně vede k uzavírání nevýhodných pozic short a otevírání pozic long a tím i růstu cen futures kontraktů [28].

Kromě samotné ceny kryptoměnového futures kontraktu je na kryptoměnových trzích zavedena ještě tzv. *Mark price*. Mark price představuje odhad reálné hodnoty futures kontraktu a její výpočet je založen na *cenovém indexu* (angl. price index) a *míře financování*. Mezi cenou futures kontraktu a Mark price přitom může být výrazný rozdíl, konvergenci těchto dvou cen vynucuje prémiový index. Mark price se oproti samotné ceně futures kontraktu vyznačuje mnohem vyšší mírou volatility [29].

Cenový index je vážený průměr cen dané kryptoměny na hlavních burzách (tj. Binance, Bitstamp, Coinbase Pro, Kraken a Bittrex) k reálnému objemu jejího obchodování. V jistém slova smyslu, lze tedy cenový index považovat za reprezentaci ceny spotové [30]. Vzorec pro výpočet tohoto cenového indexu, např. BTC/USDT na trhu Binance, vypadá následovně:

$$\frac{\text{BTC}}{\text{USDT}}_{\text{index}} = \frac{\left(\left(\frac{\text{BTC}}{\text{USDT}} \text{ z Bitstamp}\right) * \text{váha 1}\right) + \left(\left(\frac{\text{BTC}}{\text{USDT}} \text{ z Coinbase Pro}\right) * \text{váha 2}\right)}{\text{celková váha}} + \frac{\left(\left(\frac{\text{BTC}}{\text{USDT}} \text{ z Kraken}\right) * \text{váha 3}\right) + \left(\left(\frac{\text{BTC}}{\text{USDT}} \text{ z Bittrex}\right) * \text{váha 4}\right)}{\text{celková váha}} \quad (2.1)$$

Výpočet cenového indexu a s tím související výkon trhu s futures kontrakty je silně závislý na datech z příslušných burz. Při výpadech na spotových trzích nebo problémech s připojením je tudíž nutné volit speciální postup pro výpočet cenového indexu a tím zabránit špatnému výkonu trhu s futures kontrakty. Pokud se tedy například cena dané kryptoměny na některé z burz liší o více jak 5% od střední ceny všech cenových zdrojů, je její váha nastavena na nulu. Pakliže k této odlišnosti dochází dokonce u více než jedné burzy je dán cenový index střední hodnotou všech dostupných cenových zdrojů, nikoliv váženým průměrem. Při problémech s připojením je pak možné místo aktuálních dat využít poslední dostupná data, která však nesmí být starší než 10 sekund. Pokud nejsou k dispozici data z posledních 10 sekund, je cenový index opět nastaven na nulu [31]. Výpočet mark price pro trvalé futures kontrakty opět na trhu Binance pak probíhá na základě formule

$$\text{MARK PRICE} = \text{median}(\text{cena 1}, \text{cena 2}, \text{cena kontraktu}), \quad (2.2)$$

přičemž

$$\text{cena 1} = \text{cenový index} + \left(1 + \text{poslední míra financování} * \left(\frac{\text{čas do financování}}{8}\right)\right), \quad (2.3)$$

$$\text{cena 2} = \text{cenový index} + \text{pohybující se průměr}, \quad (2.4)$$

$$\text{pohybující se průměr} = \text{pohybující se průměr} \left(\frac{\text{Bid1} + \text{Ask1}}{2} - \text{cenový index}\right). \quad (2.5)$$

kde *Bid1* označuje první cenu v knize nabídky a *Ask1* první cenu v knize poptávky. Pohybující se průměr je tedy dán průměrem rozdílu střední ceny v knize objednávek a cenového indexu, které jsou vypočítávány každou první sekundu v každých 30 minutách [32].

#### 2.5.4.2 Čtvrtletní futures kontrakty (angl. Quartely futures kontrakt)

Čtvrtletní futures kontrakty mají oproti pevným futures kontraktům datum expirace a jsou vypořádávány v hotovosti. Po vypršení doby expirace jsou tedy nakupující, resp. prodávající povinni dostát svým závazkům a dochází k vyrovnání příslušného rozdílu mezi průměrnou cenou kryptoměny za poslední hodinu

a cenou v době uzavření kontraktu, nikoliv přímé směně kryptoměny. Čtvrtletní futures kontrakty se využívají převážně pro spekulaci nad budoucí cenou kryptoměn, v menší míře pak pro zajištění měnových kurzů mezi obchodovanými kryptoměnami, resp. kryptoměnami a dolary [33].

Ceny čtvrtletních futures kontraktů konvergují s blížícím se datem expirace k cenám spotovým zcela přirozeně, není proto nutné využívat mechanismu financování. Stejně tak i mark price odráží ceny spotové. Vzorec pro její výpočet se tak v případě čtvrtletních futures kontraktů liší od vzorce 2.1. Výpočet je totiž mj. závislý na čase, ve kterém probíhá. Před datem vypořádání je mark price dána vzorcem

$$\text{MARK PRICE} = \text{cenový index} + \text{pohybující se průměr}, \quad (2.6)$$

kde cenový index i pohybující se průměr jsou dány vzorcí (2.1) a (2.5), pouze s tím rozdílem, že výpočet cenového indexu je založen na trhu BTC/USD, nikoliv BTC/USDT. V den vypořádání je pak Mark price v libovolném čase  $t$  dána jako průměr cenových indexů vypočítávaných každou sekundu až do času  $t$  [34].

#### 2.5.4.3 USDT-margined futures kontrakty vs. COIN-marigned futures kontrakty

USDT-margined futures kontrakty jsou vedeny a vypořádávány v USDT a v zásadě nemají datum expirace, tj. jedná se o trvalé futures kontrakty. Kontrakty vypořádávané v USDT jsou intuitivní (snadný výpočet celkových výnosů), flexibilní (možnost držení pozic na trzích s různými měnovými páry bez nutnosti nákupu příslušných podkladových mincí) a pomáhají snižovat riziko velkých cenových výkyvů v období vysoké volatility.

COIN-margined futures kontrakty jsou naproti tomu kótovány a vypořádávány v kryptoměně a mohou ale nemusí mít datum expirace, tj. jedná se o čtvrtletní nebo pevné futures kontrakty. Kontrakty vypořádávané v kryptoměnách jsou vhodné pro zajištění pozic na trhu s futures kontrakty, bez nutnosti převodu podílů na USDT, nebo pro zvětšení kryptoměnových zásob obchodníků (zisky vypořádávány v kryptoměnách) [35].

#### 2.5.4.4 Obchodování s kryptoměnovými futures kontrakty

Obchodování s kryptoměnovými futures kontrakty se vyznačuje větší mírou svobody, než obchodování s klasickými měnovými futures kontrakty. Zatímco klasické kontrakty mají jasně danou velikost a pevně nastavenou páku, u kryptoměnových futures kontraktů určuje velikost pákového efektu sám obchodník. Maximální dostupná páka je však ovlivněna velikostí obchodu (tj. hodnotou pozice), kterou chce obchodník uzavřít a také konkrétní kryptoměnou, se kterou obchoduje. Platí, že čím větší je pozice obchodníka, tím nižší je pákový efekt a naopak čím nižší je pozice obchodníka, tím vyšší je pákový efekt.

Dalším rozdílem oproti klasickým futures kontraktům je, že v obchodování s kryptoměnovými futures kontrakty vystupují kromě samotné ceny futures kontraktu ještě další dvě ceny a sice *Last price* a *Mark price* futures kontraktu. Důvodem zavedení dvou dalších cen kontraktů je, že kryptoměny jsou vzhledem ke své digitální povaze, nízké úrovni regulace a menšímu trhu mnohem více volatilní, tj. dochází k poměrně velkým a rychlým změnám v jejich cenách. V obdobích vysoké volatility však může docházet ke zbytečným likvidačním otevřených pozic na trhu. Tím, že je výpočet příslušných parametrů určujících likvidaci závislý na těchto dvou cenách, je možné těmto zbytečným likvidacím zabránit. Last price je jednoduše poslední cena, za kterou bylo s daným kontraktem obchodováno a používá se pro výpočet *realizovaného PnL* (angl. Profits Losses), tj. celková výše zisků/ztrát po uzavření pozice. Mark Price, jak již bylo uvedeno dříve, představuje odhad reálné ceny futures kontraktu a její výpočet probíhá pomocí vzorce (2.2) nebo (2.6). Na základě Mark Price dochází k výpočtu potřebného initial marginu a *nerealizovaného PnL*, tj. výše zisků/ztrát při otevřených pozicích na trhu [36]. K likvidaci otevřených pozic,

pak dochází právě tehdy, když:

$$\text{kolaterál} = \text{initial margin} + \text{realizované PnL} + \text{nerealizované PnL} < \text{maintenance margin}$$

Výše initial marginu přitom souvisí se zvolenou velikostí pákového efektu, tzn. čím nižší je pákový efekt, tím vyšší je initial margin a naopak. Velikost maintenance marginu je naproti tomu dána velikostí pozice, kterou má obchodník na trhu otevřenou a její výpočet tak probíhá stále stejným způsobem. Obecně platí, že se rostoucí velikostí pozice se snižuje riziko likvidace a naopak. Riziku automatické likvidace lze předejít včasným ručním uzavřením pozice a to právě tehdy, když se obchodníkův aktuální poměr marží (Margin Ratio) blíží 100%. Ručním uzavřením pozice se obchodník vyvaruje peněžním sankcím spojeným s automatickou likvidací [37].

Obchodování na kryptoměnových trzích probíhá přímo mezi obchodníky, nikoliv přes finanční makléře. Stejně jako při obchodování s klasickými futures kontrakty, lze i při obchodování s kryptoměnovými futures kontrakty zaujmout jednu nebo i více pozic long nebo short a to pro každý obchodní pár. Naproti klasickým trhům je však navíc možné otevření obou pozic najednou a to prostřednictvím tzv. *hedge mode*. Tento druh obchodování je vhodný v případě, že obchodník hodlá držet jednu z pozic dlouhodobě a druhou zaujímat krátkodobě. Burza s kryptoměnovými futures kontrakty dále umožňuje otevírat pozice napříč jednotlivými obchodními páry a to buď to v *cross* nebo *isolated* margin režimu.

Na trhu s kryptoměnovými futures kontrakty lze, kromě již dříve zmíněných tržních, limitních a stop objednávek provádět ještě objednávky s názvem take profit a trailing.

- Princip a způsob využití *limitních objednávek* se na kryptoměnových trzích s futures kontrakty neliší od principů popsaných v úvodu kapitoly 2.
- *Stop objednávky* na kryptoměnových trzích slouží převážně pro zajištění a řízení rizik, tj. lze pomocí nich zajistit, aby při pohybu trhu opačným směrem, než obchodník předpokládal, nedošlo ke zbytečným ztrátám. Pokud má tedy obchodník např. otevřenou pozici long, tj. očekává růst ceny, avšak cena začne náhle klesat, dochází při určitém, obchodníkem předem definovaném, poklesu, tj. dosažení stop (aktivační) ceny, k vykonání stop objednávky a otevření opačné pozice, pozice short. Objednávky typu *stop* jsou dvojího druhu a sice *stop tržní objednávky* a *stop limitní objednávky*. Narozdíl od stop tržních objednávek lze stop limitní objednávky využít i jako nástroj umožňující automatický vstup na trhy za požadovanou cenu. Stop cena a limitní cena mohou být stejné, avšak není to příliš žádoucí. Pokud je stop a limitní cena nastavena stejně, pak po dosažení stop ceny, automaticky dojde k vykonání obchodu, tj. limitní objednávka se stává tržní objednávkou. Při prodeji je proto vhodné nastavit stop cenu o něco vyšší než je limitní cena, resp. při nákupu nižší. Za stop cenu může obchodník zvolit buď Last nebo Mark price.
- *Take profit objednávka* je druh objednávky, která je v jistých ohledech podobná objednávce stop, avšak její použití je zcela opačné. Take profit objednávka se totiž používá pro zajištění zisků a uzavření pozic. Pokud má tedy obchodník např. otevřenou pozici long, tj. očekává, že cena poroste, lze při určitém, obchodníkem předem definovaném, vzrůstu ceny vykonat take profit objednávku, čímž dochází k uzavření obchodu. Objednávky typu *take profit* jsou opět dvojího druhu a sice *take profit tržní* a *take profit limitní objednávky*. Jediný rozdílem je však opět okamžité provedení objednávky při dosažení požadované tržní ceny, vs. zadání příkazu k nákupu/prodeji se stanovenou limitní cenou do knihy objednávek.
- *Trailing objednávka* je robustní druh objednávky, který slouží jak pro zajištění rizik tak i zisků. Po otevření pozice na trhu lze nastavit velikost zarážky, pod resp. nad aktuální tržní cenou, při které dochází k otevírání opačných pozic, a tak zajištění rizik, resp. zisků [38] [39].

## Kapitola 3

# Umělé neuronové sítě

Umělé neuronové sítě se řadí mezi podoblasti strojového učení, jenž je jistým pododvětvím umělé inteligence. Umělá inteligence je věda, která se zabývá vývojem strojů, zejména pak inteligentních počítačových programů, využívaných pro řešení komplexních úloh z oblasti logistiky, robotiky, či zpracování velkého objemu dat [49]. Strojové učení se pak konkrétně zaměřuje na vývoj takových modelů umělé inteligence, které jsou trénovány speciálním procesem napodobujícím lidské učení. Učení v tomto kontextu označuje optimalizaci vhodného kritéria vzhledem k dostupným datům a probíhá na množině tzv. trénovacích dat. Tento proces přitom vede k modelům, které jsou, resp. jejich parametry jsou nastaveny tak, aby odpovídaly nalezení co nejvýstižnější transformace vstupních dat, ozn.  $x$ , na data výstupní, ozn.  $y$ , a to za dodatečného předpokladu na obecnost této transformace. Celý proces učení modelů pak obstarává tzv. adaptační algoritmus, na základě kterého jsou jednotlivé parametry modelu neustále přenastavovány. Podle způsobu učení lze adaptační algoritmy rozdělit do následujících kategorií:

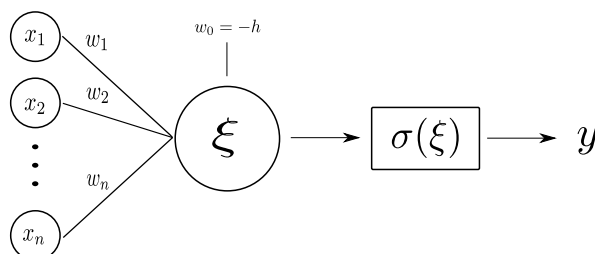
- *Učení s učitelem* (angl. supervised learning) je metoda, při které je model trénován pomocí spočetné množiny dvojic vstupních dat  $x$  a jim odpovídajících dat výstupních  $y$  (tzv. štítků). Adaptační algoritmus má tedy k dispozici příklady správné transformace vstupních dat  $x$  na data výstupní  $y$  a jeho úlohou je provést vhodné zobecnění. Učení s učitelem je metoda, na níž je založeno trénování nejen většiny neuronových sítí, ale např. také modelů lineární, či logistické regrese, modelu náhodného lesa a jiných.
- *Učení bez učitele* (angl. unsupervised learning) je naopak metoda, během níž je model trénován na základě množiny obsahující pouze vstupní data  $x$ . Adaptační algoritmus tedy nemá k dispozici žádné kritérium správnosti hledané transformace těchto dat. Celé učení závisí pouze na informacích obsažených ve vstupních datech. Učení bez dozoru je metoda, která se hojně využívá např. pro trénování samoorganizujících se map, které jsou jistým druhem neuronových sítí, modelů určených k průzkumné analýze dat, strategii křížového prodeje, či segmentaci zákazníků.
- *Zpětnovazebné učení* (angl. reinforcement learning) je poslední metoda, která je kombinací dvou výše zmíněných metod. Pro část množiny vstupních dat  $x$  jsou známa skutečná výstupní data  $y$ , typicky větší část množiny vstupních dat  $x$  je však poskytnuta bez skutečných výstupních dat  $y$  [50], [51].

Jak již bylo řečeno, trénování umělých neuronových sítí je založeno na metodě učení s, ve zvláštních případech i bez, učitele. Hlavní myšlenkou neuronových sítí je přitom využít pro řešení daných úloh konkrétně principy fungování lidského mozku. Stejně jako mozek přijímá vstup, zpracovává ho a generuje určitý výstup, na podobném principu fungují i umělé neuronové sítě. Příjem vstupu, zpracování

informace a generování výstupu jsou zde přitom reprezentovány ve formě vrstev – vstupní, jedné nebo více skrytých a výstupní. Zatímco vstupní vrstva je tvořena pouze vstupními daty, skryté a výstupní vrstvy jsou vrstvami výpočetními. Výstupní vrstva pak musí být navržena tak, aby poskytovala výsledky ve vhodném tvaru. Úloh, které se prostřednictvím neuronových sítí dají zpracovávat existuje celá řada. Převážně se přitom jedná o klasifikaci dat (tzv. klasifikační modely), či predikci konkrétních hodnot (tzv. regresní modely), v menší míře se pak neuronové sítě používají pro analýzu a shlukování dat. Příkladem klasifikačního modelu neuronové sítě je model určený k rozpoznávání obrázků psů a koček, tj. segmentaci do příslušných skupin. Za regresní model neuronové sítě lze pak považovat například takový model, jenž je schopen na základě údajů o spotřebě domácnosti v uplynulém roce predikovat výši spotřeby na následující měsíc [40].

### 3.1 Neuron

Neuron je základní stavební jednotka neuronových sítí. Základní schéma neuronu zachycuje obrázek 3.1.



Obrázek 3.1: Základní schéma neuronu

Vstupní vrstva je tvořena vektorem o  $n$  prvcích, které jsou obecně reálné, tzn.  $\mathbf{x} \in \mathbb{R}^n$ . Každý vstup do neuronu je přitom se samotným tělem neuronu spojen prostřednictvím váhy  $w_i$ , která může nabývat kladných i záporných hodnot. V těle neuronu je získáván jeho vnitřní potenciál  $\xi$ , který vzniká jako rozdílová váženého součtu jednotlivých vstupů  $\mathbf{x}$  přes všechny váhy  $w_i$  a prahu  $h$ ,

$$\xi = \sum_{i=1}^n w_i x_i - h. \quad (3.1)$$

Vnitřní potenciál se následně stává argumentem tzv. *aktivační funkce*  $\sigma$ , která provádí jeho nelineární transformaci za vzniku jedné, obecně kladné, hodnoty, jež je výstupem neuronu. Akci neuronu je tedy možné popsat jako zobrazení  $(\mathbb{R}^n) \rightarrow \mathbb{R}$ . Z praktických důvodů se práh obvykle modeluje jako jedna z vah a sice tak, že je vstupní vektor  $\mathbf{x}$  i vektor vah  $\mathbf{w}$  rozšířen o nultý prvek,  $x_0 = 1$  resp.  $w_0 = -h$ . Všechny váhy a práh pak souhrnně označujeme jako parametry modelu neuronu [41]. Odezvu neuronu na daný vstup  $\mathbf{x}$  pak můžeme vyjádřit vzorcem

$$y = \sigma(\xi) = \sigma\left(\sum_{i=0}^n w_i x_i\right). \quad (3.2)$$

#### 3.1.1 Aktivační funkce

Aktivační funkce, jak název napovídá, slouží k aktivaci, resp. deaktivaci konkrétních hodnot vnitřního potenciálu neuronu a to za účelem klasifikace, resp. predikce těchto hodnot. Při volbě aktivační funkce je obvykle kladen požadavek na její diferencovatelnost [42]. Mezi nejpoužívanější aktivační funkce patří



následující nelineární funkce:

**ReLu**

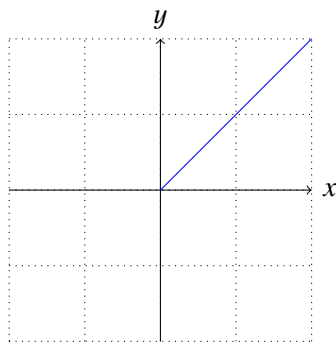
$$f(x) = \max(0, x) \quad (3.3)$$

**Sigmoida**

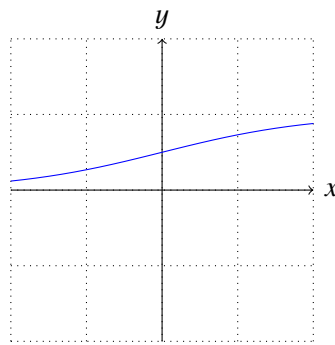
$$f(x) = \frac{1}{1 + e^{-x}} \quad (3.4)$$

**Hyperbolický tangens**

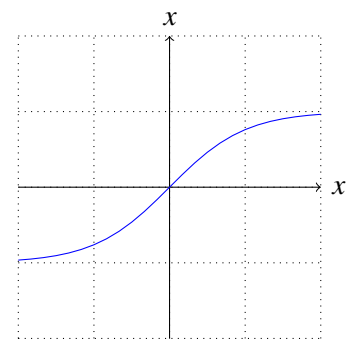
$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.5)$$



(a) ReLu



(b) Sigmoida



(c) Hyperbolický tangens

Obrázek 3.2: Aktivační funkce

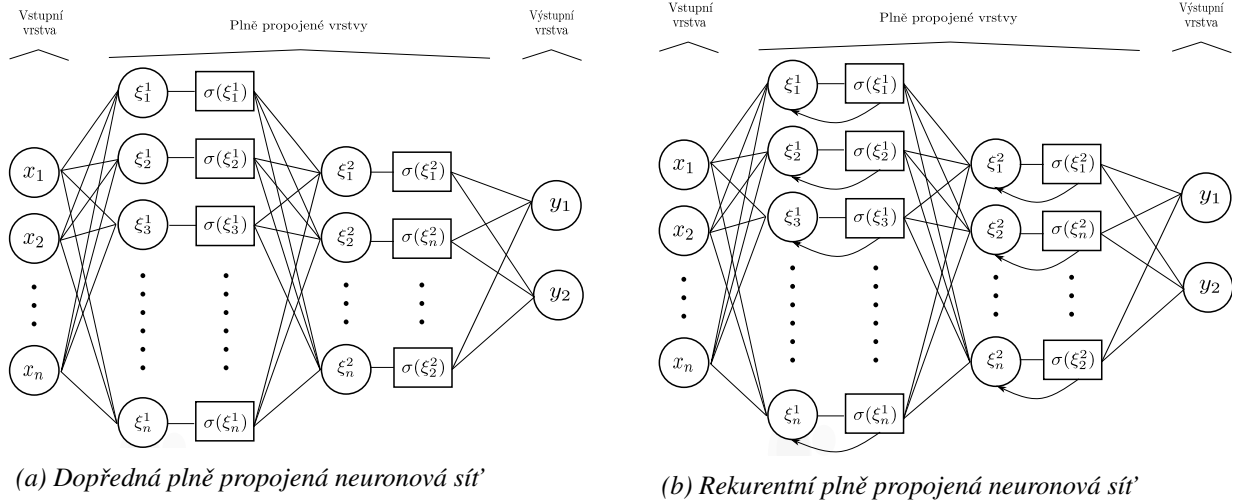
## 3.2 Vícevrstvá neuronová síť

Vícevrstvá neuronová síť je síť tvořená velkým množstvím neuronů, které jsou uspořádány do tzv. skrytých vrstev. Obecně rozlišujeme mezi dvěma druhy vícevrstevných neuronových sítí a sice *dopřednými* a *rekurentními*. Zároveň však existuje mnoho typů skrytých vrstev na základě jejichž přítomnosti dopředné, resp. rekurentní vícevrstvé neuronové sítě dále klasifikujeme. Nejzákladnějším typem skrytých vrstev jsou přitom tzv. *plně propojené vrstvy*. Zatímco v dopředných plně propojených neuronových sítích jsou prostřednictvím vah spojeny pouze všechny neurony dané vrstvy se všemi neurony vrstvy následující, v případě rekurentních plně propojených neuronových sítí jsou k těmto spojmům přidány ještě spoje každého neuronu se sebou samým. Oba druhy vícevrstevných neuronových sítí tvořených plně propojenými vrstvami jsou schematicky zachyceny na obrázcích 3.3a a 3.3b. Počet neuronů tvořících plně propojené vrstvy může být obecně různý, avšak obvykle nabývá hodnot v rozmezí mezi počtem vstupních a výstupních neuronů.

Akci jedné plně propojené vrstvy tvořící dopřednou vícevrstvou neuronovou síť je možné chápat jako zobrazení  $(\mathbb{R}^n) \rightarrow \mathbb{R}^m$ . Výstup  $j$ -tého neuronu v  $q$ -té plně propojené vrstvě je přitom dán vztahem

$$y_j^q = \sigma(\xi_j^q) = \sigma\left(\sum_{i=0}^n w_{ji}^q \xi_i^k\right), \quad \text{kde} \quad (3.6)$$

$$\xi_i^k = \begin{cases} x_i & \text{pro: } k = 1. \\ y_i^{k-1} & \text{pro: } k > 1. \end{cases}$$



Obrázek 3.3: Vícevrstvé neuronové sítě

kde  $\sigma$  značí zvolenou aktivační funkci a  $\xi_j^q$  vnitřní potenciál  $j$ -tého neuronu tvořící  $q$ -tou plně propojenou vrstvu. Výstupy jednotlivých neuronů  $q$ -té vrstvy tvoří vstupní hodnoty  $(q + 1)$ -ní vrstvy se kterými se celý proces opakuje [41].

Akce jedné rekurentní plně propojené vrstvy je pak možné chápat jako zobrazení  $(\mathbb{R}^{n+1}) \rightarrow \mathbb{R}^m$ , neboť vstup do každého neuronu je tvořen vektorem o  $n$  prvcích plus výstupem daného neuronu v předchozím kroku [43].

### 3.2.1 Učení vícevrstvé neuronové sítě

Učení vícevrstvých neuronových sítí probíhá na trénovacích datech a to striktně metodou učení s učitelem, jenž byla představena již na počátku kapitoly 3. Využívaný adaptačním algoritmem je zde přitom algoritmus tzv. *gradientního sestupu*, mezi jehož možné alternativy patří např. optimalizátor Adam, RMSprop, AdaDelta a další. Důležitou roli v celém algoritmu hraje tzv. *ztrátová funkce*.

#### 3.2.1.1 Ztrátová funkce

Hlavní úlohou ztrátové funkce, ozn.  $L$ , je zahrnout všechny aspekty modelu do hodnoty tzv. *ztráty* (angl. loss), a to tak, aby nízká hodnota výsledné ztráty byla známkou dobře fungujícího modelu. Ztrátová funkce přitom u regresních modelů porovnává výstupy neuronové sítě (modelu)  $\mathbf{y}$  s výstupy skutečnými  $\mathbf{y}^p$ , resp. u klasifikačních modelů výstupní a skutečnou pravděpodobnost výskytu vstupu v dané třídě. Toto porovnání vždy probíhá na jisté dávce množiny trénovacích dat, tzv. *batchy*. Velikost ztráty lze tedy také interpretovat jako velikost chyby, které se neuronová síť během predikce dopustila [44]. Volba ztrátové funkce je klíčová a silně závisí na formě výstupu z neuronové sítě, tj. typu modelu. Jiná ztrátová funkce bude použita pro úlohu regrese a jiná zase pro úlohu klasifikace.

V regresních modelech lze využít například následující ztrátové funkce:

#### Mean Squared Error (MSE)

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n |y_i - y_i^p|^2 \quad (3.7)$$

kde  $y_i$  je skutečný výstup pro  $i$ -tý vstup,  $y_i^p$  je výstup predikovaný sítí a  $n$  velikost datového vzorku trénovacích dat. Mean Squared Error lze tedy interpretovat jako průměr čtverců vzdáleností mezi skutečnými hodnotami  $y_i$  a hodnotami predikovanými sítí  $y_i^p$ .

#### Mean Absolute Error (MAE)

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - y_i^p| \quad (3.8)$$

kde  $y_i$  je opět skutečný výstup pro  $i$ -tý vstup,  $y_i^p$  je výstup predikovaný sítí a  $n$  velikost příslušného datového vzorku. Mean Absolute Error tedy vyjadřuje průměrnou vzdálenost mezi skutečnými výstupy  $y_i$  a predikovanými výstupy  $y_i^p$  [47].

Klíčový rozdíl mezi MSE a MAE lze vysvětlit na příkladu, kdy se v předpovězených datech vyskytují velké odchylky od dat skutečných. Kvadrát v předpisu MSE způsobí, že hodnota ztráty bude v takovém případě mnohem vyšší než hodnota ztráty při použití MAE. MAE je tedy vůči velmi odlehlým předpovědím robustnější a je vhodné ji využít, pokud jsou tyto chyby způsobeny poškozenými daty, nikoliv anomáliemi, které by měl model zachytit. Mohou však nastat i případy, kdy ani jedna z funkcí neposkytuje žádoucí předpovědi. Například pokud se 90% predikovaných hodnot velmi blíží hodnotám skutečným, avšak zbylých 10% se výrazně odlišuje. Model s MAE jakožto ztrátovou funkcí v takovém případě zanedbá 10% výrazně odchýlených případů a svou predikci směřuje k hodnotám zbylých 90%. Zatímco model s MSE se chová zcela opačně. V mnoha regresních úlohách, týkajících se převážně oblasti obchodování na finančních trzích, je tak vhodné využít tzv. Huberovu funkci, která vlastnosti MSE a MAE vhodně kombinuje [46].

#### Huberova ztrátová funkce

$$L_\delta = \sum_{i=1}^n \begin{cases} \frac{1}{2} |y_i - y_i^p|^2 & \text{pro: } |y_i - y_i^p| \leq \delta \\ \delta |y_i - y_i^p| - \frac{1}{2}\delta^2 & \text{jinak.} \end{cases} \quad (3.9)$$

kde  $y_i$  označuje skutečný výstup pro  $i$ -tý vstup,  $y_i^p$  je výstup predikovaný sítí a  $\delta$  je volitelný parametr.

V klasifikačních modelech lze využít například následující ztrátová funkce:

#### Binary Cross Entropy

Binary Cross Entropy je ztrátová funkce určená pro úlohy tzv. binární klasifikace, tj. úlohy, kde jsou vstupní data rozdělena do dvou různých tříd, jímž je přidělena číselná hodnota 0 nebo 1. Výstupem sítě je v takovém případě pravděpodobnost výskytu daného vstupu ve skupině 1. Funkce Binary Cross Entropy proto určuje průměrný rozdíl mezi skutečným a předpovězeným rozdělením pravděpodobnosti výskytu ve skupině 1.

#### Multi-Class Cross Entropy

Multi-Class Cross Entropy je alternativou ztrátové funkce Binary Cross Entropy pro úlohy klasifikace vstupních dat do obecně  $n$  různých tříd. Výstupem této funkce je proto průměrný rozdíl mezi skutečným a předpovězeným rozdělením pravděpodobnosti pro všechny třídy [48].

### 3.2.1.2 Algoritmus gradientního sestupu

Algoritmus gradientního sestupu je optimalizační proces, při kterém dochází k iterativní úpravě parametrů sítě za účelem minimalizace výsledné ztráty a to vždy pro určitou dávku trénovací množiny (batch). Celý algoritmus se sestává z celkem 6 kroků:

1. Krok: Inicializace všech parametrů náhodnou hodnotou.  
Před tím, než sítí projde první dávka trénovací množiny (batch) nejsou k dispozici žádné hodnoty parametrů, proto je nutné je nejprve tzv. inicializovat.
2. Krok: Neuronové síti předložena dvojice vstupní vektor a jemu příslušný výstup.  
Sítí postupně prochází jeden vektor dané části trénovací množiny po druhém.
3. Krok: Výpočet výstupu.  
Výpočet výstupu probíhá pro každou dvojici zvlášť, avšak ztrátová funkce je vyhodnocována vždy po batchy.
4. Krok: Zpětné šíření (angl. backpropagation).  
Jedná se o metodu výpočtu gradientu ztrátové funkce podle jednotlivých parametrů. Tento výpočet přitom probíhá postupně od zadních vrstev až po vrstvy přední, neboť výsledky vypočtené v pozdějších vrstvách lze s výhodou využít pro výpočty ve vrstvách dřívějších (odtud také název metody). K výpočtu gradientu tedy dochází v jednotlivých vrstvách neuronové sítě zvlášť.
5. Krok: Úprava parametrů.  
K úpravě parametrů dochází vždy po průchodu celého batche neuronovou sítí a je dána předpisem
$$w_i^{k+1} = w_i^k - \alpha * \frac{\partial L}{\partial w_i}, \quad (3.10)$$
kde  $w_i^{k+1}$  značí  $i$ -tou složku vektoru vah  $\mathbf{w}$  v  $(k+1)$ -ní vrstvě neuronové sítě,  $w_i^k$  značí  $i$ -tou složku vektoru vah  $\mathbf{w}$  v  $k$ -té vrstvě,  $\alpha$  rychlost učení a  $L$  použitou ztrátovou funkci.
6. Krok: Opakování procesu učení od kroku 2 dokud není velikost ztráty pro všechny dávky trénovací množiny menší než  $\epsilon$  [45].

### 3.3 Konvoluční neuronová síť

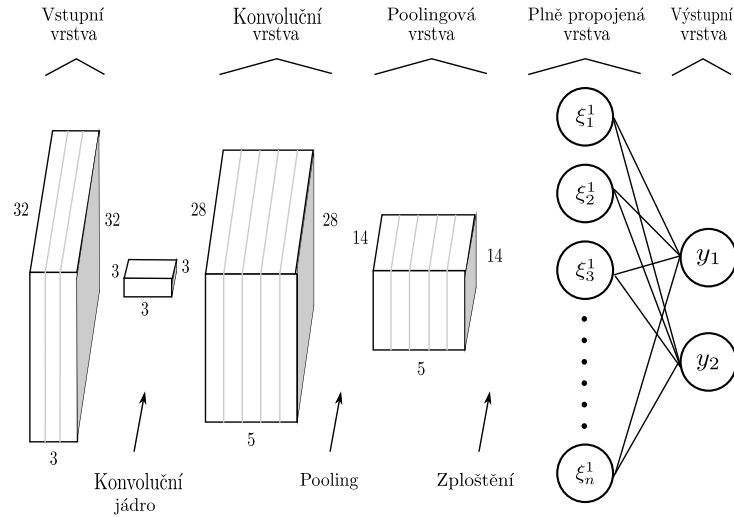
Konvoluční neuronová síť je dopředná neuronová síť, která obsahuje alespoň jednu skrytou vrstvu ve formě vrstvy konvoluční. Tyto sítě byly navrženy pro práci s geometricky uspořádanými vstupy. Mezi takové patří například obrázky, kde buňky prostorové mřížky tvoří pixely, či časové řady, kde buňky představují hodnotu v daném časovém kroku. Jednotlivé buňky prostorové mřížky jsou na sobě přitom silně prostorově závislé, přičemž porozumění těmto závislostem je hlavním cílem architektury. Právě touto skutečností se konvoluční neuronové sítě odlišují od dopředných plněpropojených sítí, v nichž se informace o uspořádání jednotlivých prvků ztrácí.

Konvoluční neuronové sítě se využívají převážně v klasifikačních úlohách, stále častěji své místo však nacházejí i v úlohách regresivních [52].

#### 3.3.1 Struktura konvoluční neuronové sítě

Nejjednodušší konvoluční neuronová síť se skládá ze vstupní, konvoluční, aktivační, poolingové, zplošťující a jedné plně propojené vrstvy, viz obrázek 3.4. Vstupní vrstva je tvořena pouze vstupními daty, které mohou být uspořádány nejen do podoby 3D tenzorů (př. obrázky), jak je znázorněno na obrázku 3.4, ale také matic, či vektorů (př. časové řady). Vhodnou formu výstupu pak zprostředkovává plně propojená vrstva. Složitější architektury konvolučních neuronových sítí je možné utvářet přidáním, či jiným přeuspořádáním konvolučních, aktivačních a poolingových vrstev. Typicky se může jednat např. o umístění dvou konvolučních vrstev za sebe a až poté aplikaci vrstvy aktivační a poolingové, či přidání ještě jedné

maximálně dvou plně propojených vrstev.



Obrázek 3.4: Základní schéma konvoluční neuronové sítě

V celém následujícím textu budeme uvažovat konvoluční neuronovou síť s  $n$  konvolučními vrstvami, přičemž jak vstupní a tak i za ní následující vrstvy mají strukturu prostorové mřížky, která je matematicky popsána 3D tenzory. Prostorové vztahy mezi jednotlivými buňkami prostorové mřížky se přenáší z jedné vrstvy na druhou, tzn. hodnota prvku 3D tenzoru v dané vrstvě je založena na hodnotách ve vrstvě předcházející.  $Q$ -tá vrstva konvoluční neuronové sítě je pak charakterizována výškou  $L_q$ , šířkou  $B_q$  a hloubkou  $d_q$  prostorové mřížky, kterou je tvořena. Zatímco výška  $L_q$  a šířka  $B_q$  mapují pouze prostorové vlastnosti, hloubka  $d_q$  vyjadřuje počet sledovaných, vzájemně nezávislých, charakteristik. Například u vstupní vrstvy tvořené obrázky určuje hloubka  $d_q$  počet barevných kanálů, z nichž jsou tyto obrázky složeny (R, G, B). U konvolučních a poolingových vrstev pak hloubka  $d_q$  zachycuje počet tzv. *aktivačních map*, tj. vlastností, které byly ze vstupní nebo předcházející vrstvy extrahovány [52].

### 3.3.1.1 Konvoluční vrstva

Data ze vstupní nebo předcházející vrstvy jsou nejprve zpracována konvoluční vrstvou, kde probíhá operace zvaná, jak i název samotné vrstvy napovídá, konvoluce. Konvoluci zprostředkovává konvoluční jádro, jehož velikost v  $q$ -té vrstvě je obecně  $F_q \times F_q \times d_q$ , tzn. je čtvercového tvaru. Zatímco výška  $F_q$  a šířka  $F_q$  jádra jsou typicky mnohem menší než výška  $L_q$  a šířka  $B_q$   $q$ -té vrstvy, hloubka  $d_q$  jádra se vždy s hloubkou  $d_q$   $q$ -té vrstvy shoduje. V praxi nejvíce používanými konvolučními jádry jsou jádra o rozměrech  $(2, 2, d_q)$ ,  $(3, 3, d_q)$  nebo  $(5, 5, d_q)$ .

Konvoluce v  $q$ -té vrstvě je operace, během níž se konvoluční jádro postupně posouvá po celé ploše  $q$ -té vrstvy, přičemž velikost jednoho posunutí, tj. velikost kroku, je nejčastěji nastavena na 1 prvek. V každém kroku pak dochází ke skalárnímu součinu mezi 3D tenzorem konvolučního jádra a 3D tenzorem příslušné oblasti  $q$ -té vrstvy, který je následován součtem. Matematicky, lze operaci konvoluce z  $(q + 1)_n$

do  $q$ -té vrstvy vyjádřit pomocí vzorce

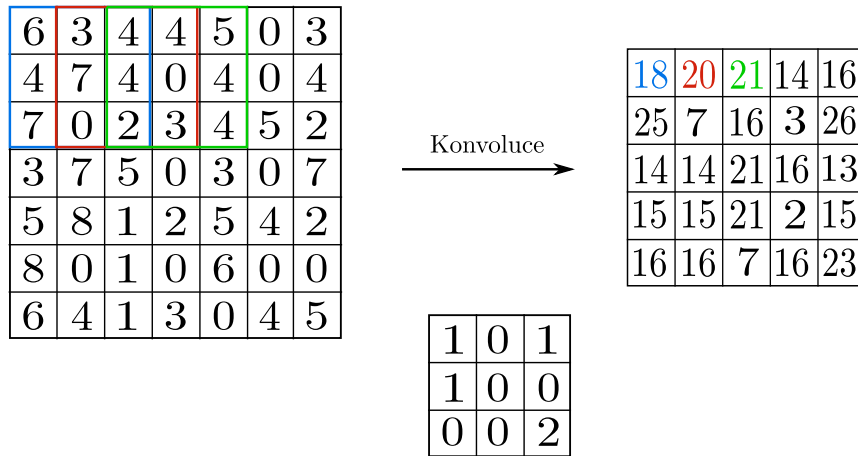
$$h_{ijp}^{(q+1)} = \sum_{r=1}^{F_q} \sum_{s=1}^{F_q} \sum_{k=1}^{d_q} w_{rsk}^{(p,q)} h_{i+r-1,j+s-1,k}^{(q)} \quad (3.11)$$

$$\forall i \in \{1 \dots, L_q - F_q + 1\}$$

$$\forall j \in \{1 \dots, B_q - F_q + 1\}$$

$$\forall p \in \{1 \dots, d_q\}$$

kde  $w_{ijk}^{(p,q)}$  jsou prvky 3D tenzoru  $W^{(p,q)}$  konvolučního jádra a  $h_{ijk}^{(q)}$  jsou prvky 3D tenzoru  $H^{(q)}$  reprezentujícího  $q$ -tou vrstvu. Indexy  $i, j, k$ , pak označují polohy podél výšky, šířky a hloubky konvolučního jádra. Výsledky skalárního součinu přes všechna umístění filtrů jsou uspořádány do matice, zvané aktivační mapa, která tvoří dílčí část  $(q+1)_n$  vrstvy. Vzhledem k tomu, že hloubka  $q$ -té vrstvy je shodná s hloubkou filtru, je výsledná hodnota skalárního součinu v daném bodě aktivační mapy  $(q+1)_n$  vrstvy příspěvkem od každé aktivační mapy  $q$ -té vrstvy. Příklad kovoluce na vrstvě o velikosti  $7 \times 7 \times 1$  s konvolučním jádrem  $3 \times 3 \times 1$  zachycuje obrázek 3.5.



Obrázek 3.5: Kovoluce na vrstvě o velikosti  $7 \times 7 \times 1$  s konvolučním jádrem  $3 \times 3 \times 1$

Počet pozic, na které lze konvoluční jádro na  $q$ -té vrstvě zarovnat, určuje výšku  $L_{q+1}$  a šířku  $B_{q+1}$   $(q+1)_n$  vrstvy, přičemž platí:

$$L_{q+1} = L_q - F_q + 1 \quad (3.12a)$$

$$B_{q+1} = B_q - F_q + 1 \quad (3.12b)$$

Hloubka  $(q+1)_n$  vrstvy je potom dána počtem aktivačních map, které vznikly aplikací různých konvolučních jader, kódujících různé vlastnosti, na  $q$ -tou vrstvu. Čím větší je tedy počet jader, tím více je aktivačních map a tím větší je hloubka  $(q+1)_n$  vrstvy. Hloubka  $(q+1)_n$  vrstvy tak žádným způsobem nezávisí na hloubce vrstvy předcházející.

### Polstrování (padding)

Jak je zřejmé z rovností (3.12a) a (3.12b), aplikací operace konvoluce dochází ke zmenšení rozměrů vrstvy,  $(q+1)_n$  vrstva je o  $(F_q + 1)$  v prostorových dimenzích menší než vrstva  $q$ -tá. Tento typ zmenšení však obecně není žádoucí, neboť se díky němu mohou ztratit informace, které jsou obsaženy na okrajích

$q$ -té vrstvy. Aby byla prostorová stopa zachována, tj. nedošlo ke zmenšení rozměrů, lze využít tzv. *polovičního* nebo *plného* polstrování.

Poloviční polstrování spočívá v přidání  $\frac{1}{2}(F_q - 1)$  nulových prvků podél okraje každé aktivační mapy, která tvoří  $q$ -tou vrstvu. V důsledku toho se výška  $L_q$  a šířka  $B_q$   $q$ -té vrstvy zvětší o  $(F_q - 1)$ , což je přesně velikost o kterou se následnou aplikací konvoluce rozměry vrstvy zmenší. Samotná výplň  $q$ -té vrstvy do skalárního součinu, daného vzorcem (3.11), nijak nepřispívá, neboť její hodnoty jsou nastaveny na nulu. Jediným účelem polstrování je tedy umožnit konvolučnímu jádru zafixovat informace obsažené na krajích vrstvy. Obrázek 3.6 zachycuje příklad polovičního polstrování jedné aktivační mapy o velikosti  $7 \times 7 \times d_q$ , kdy je na danou vrstvu aplikováno konvoluční jádro o rozměrech  $5 \times 5 \times d_q$ ,

0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	6	3	4	4	5	0	3	0	0
0	0	4	7	4	0	4	0	4	0	0
0	0	7	0	2	3	4	5	2	0	0
0	0	3	7	5	0	3	0	7	0	0
0	0	5	8	1	2	5	4	2	0	0
0	0	8	0	1	0	6	0	0	0	0
0	0	6	4	1	3	0	4	5	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

Obrázek 3.6: Poloviční polstrování jedné z aktivačních map vrstvy o velikosti  $7 \times 7 \times d_q$

Plné polstrování pak spočívá v přidání  $(F_q - 1)$  nulových prvků na každou stranu každé aktivační mapy tvořící  $q$ -tou vrstvu. Tím dochází ke zvětšení prostorové dimenze  $q$ -té vrstvy o  $2(F_q - 1)$ . Následnou aplikací konvoluce se však rozměry vrstvy sníží opět pouze o  $(F_q - 1)$ . Velikost výsledné  $(q + 1)_n$  vrstvy je tak při použití plného polstrování oproti  $q$ -té vrstvě dokonce zvětšena [52].

### 3.3.1.2 Aktivační vrstva

Po každé konvoluční vrstvě následuje vrstva aktivační, jejíž aplikace je stejná, jako tomu bylo v případě vícevrstvých neuronových sítí. Na každou skalární hodnotu v  $(q + 1)_n$  vrstvě je aplikována aktivační funkce a vzniká  $L_{q+1} \times B_{q+1} \times d_{q+1}$  prahových hodnot, které jsou předány další vrstvě. Velikost vrstvy přitom opět zůstává zachována. Hlavním účelem aktivační vrstvy u konvolučních neuronových sítí je však nejen aktivace hodnot, které jsou pro predikci klíčové, nýbrž také odstranění lineárních vztahů, které vznikly aplikací konvoluce. Hojně využívanou aktivační funkcí je pak nelineární aktivační funkce ReLu daná vzorcem (3.3) [52].

### 3.3.1.3 Poolingová vrstva

Za aktivační vrstvou následuje vrstva poolingová, kde je výstup předchozí vrstvy zpracován nelineární transformací zvanou max-pooling. Max-pooling je operace, během níž je z každé čtvercové oblasti  $(q + 1)_n$  vrstvy o velikosti  $P_{q+1} \times P_{q+1}$  vypočteno maximum, a to pro každou aktivační mapu. Výška  $L_{q+2}$  a šířka  $B_{q+2}$   $(q + 2)_n$  vrstvy vzniklé aplikací poolingů je tak zmenšena, zatímco hloubka  $d_{q+2}$ , tj. počet aktivačních map, zůstává zachována. Tím se operace max-pooling výrazně odlišuje od konvoluce, která

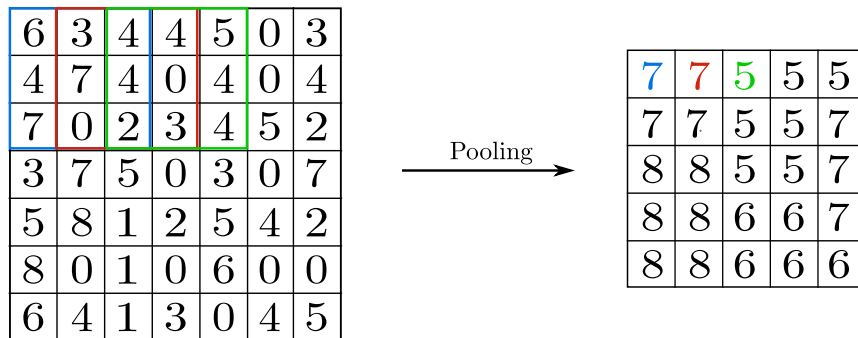
pracuje s příslušnou částí všech aktivačních map najednou a vrací jedinou hodnotu. Pro rozměry  $(q+1)_n$  vrstvy tedy platí:

$$L_{q+2} = L_{q+1} - P_q + 1 \quad (3.13a)$$

$$B_{q+2} = B_{q+1} - P_q + 1 \quad (3.13b)$$

$$d_{q+2} = d_{q+1} \quad (3.13c)$$

Hlavním účelem této vrstvy je redukce dat, která nejsou k predikci potřebná [52]. Příklad poolingů na čtvercové oblasti  $3 \times 3$  aktivační mapy o velikosti  $7 \times 7 \times 1$  zachycuje obrázek 3.7.



Obrázek 3.7: Poolingu na čtvercové oblasti  $3 \times 3$  aktivační mapy o velikosti  $7 \times 7 \times 1$

### 3.3.1.4 Vrstva zploštění

Ještě před vstupem do plně propojené vrstvy, je výsledný výstup tzv. zploštěn (angl. flatten). Během této operace je 3D tenzor hodnot jednoduše převeden na jeden dlouhý vektor prvků. Hlavní účelem této vrstvy je získat výsledné hodnoty v takové podobě, aby na ně následně bylo možné aplikovat plně propojenou vrstvu [53].

### 3.3.1.5 Plně propojená vrstva

Poslední vrstvou je vrstva plně propojená, jejíž princip fungování je totožný s jednovrstvou dopředné plně propojenou sítí, viz sekce 3.2. S vektorem, jenž je výstupem vrstvy zploštění zde probíhají naprosto stejné matematické operace. V závislosti na řešeném problému pak, při dané aktivační funkci, dochází ke klasifikaci nebo regresi [54].

## 3.3.2 Učení konvoluční neuronové sítě

Jak již bylo řečeno na počátku sekce 3.3, konvoluční neuronová síť je dopředná neuronová síť, přičemž tento druh sítě spadá do obecné skupiny vícevrstevných neuronových sítí. Princip jejich učení byl přitom detailně rozebrán v sekci 3.2.1. Na tomto místě tak budou vysvětlena pouze specifika algoritmu gradientního sestupu pro konvoluční neuronovou síť, koncept však zůstává v zásadě stejný.

### 3.3.2.1 Algoritmus gradientního sestupu

Zatímco v případě plně propojených vrstev jsou jednotlivé kroky algoritmu zcela analogické jako u vícevrstevných neuronových sítí, na vrstvách poolingových tento algoritmus vůbec neprobíhá, u konvolučních



vrstev je pak lehce modifikovaný.

Poolingová vrstva neobsahuje žádné parametry, které by bylo třeba upravovat. V této vrstvě tak dochází pouze k distribuci gradientů [55]. Necht'  $P_1, \dots, P_r$  jsou čtvercové oblasti, které pokrývají oblast  $q$ -té vrstvy s hodnotou  $h$ . Pro každou takovou oblast pak dostáváme maxima  $h_1, \dots, h_r$  v  $(q + 1)_n$  vrstvě. Necht' dále  $h$  je maximální hodnota pro čtvercovou oblast  $P_i$ , tzn.  $h = h_i$ , potom gradient ztrátové funkce dle váhy příslušící tomuto spojení míří opět do  $h$ . Z čehož vyplývá, že gradient vypočtený při zpětném šíření nemá žádný vliv na prvky, které nebyly zahrnuty do výpočtů při šíření dat směrem od vstupní vrstvy k výstupní.

Počet parametrů v konvoluční vrstvě, které jsou v rámci algoritmu gradientního sestupu upravovány, je dán počtem aplikovaných konvolučních jader a počtem jejich zarovnání na  $q$ -té vrstvě. Hodnota  $c$  v  $q$ -té vrstvě přispívá k výpočtu několika hodnot v  $(q + 1)_n$  vrstvě, přičemž počet prvků, ke kterým hodnota  $c$  přispívá je dán velikostí kroku a hloubkou  $(q + 1)_n$  vrstvy. Necht'  $S_c$  je množina prvků, k jejichž výpočtu přispívá hodnota  $c$ . Vzhledem k tomu, že prvky množiny  $S_c$  vznikají násobením prvku  $c$  s prvkem konvolučního jádra, je pro nalezení celkové ztráty nutné vynásobit gradienty všech prvků množiny  $S_c$  dle vah zprostředkovávající spojení mezi filtrem a daným prvkem a váhou prvku jádra použitým pro výpočet daného prvku, neboli

$$\delta_s = \sum_{r \in S_c} \delta_r w_r \quad (3.14)$$

kde  $\delta_r$  je ztráta u prvku  $r \in S_c$  a  $w_r$  váha prvku filtru použitého pro výpočet prvku  $r$  z prvku  $c$  [52].

## Kapitola 4

# Experimenty

### 4.1 Hlavní myšlenka

Analýza jak klasických, tak i kryptoměnových trhů je velmi častou a zajímavou úlohou. Zatímco dění na klasických trzích je silně ovlivňováno mnoha vysoce korelovanými ekonomickými, politickými, či dokonce psychologickými faktory, na trzích kryptoměnových to tak úplně platit nemusí. Nabízí se proto hned několik otázek: Existují na kryptoměnových trzích nějaká psychologie transakcí? Pokud ano, jak případné závislosti mezi transakcemi odhalit? Na základě čeho rozhodnout, zda nakoupit nebo naopak prodat kryptoměnu, či nějaký kryptoměnový kontrakt?

Jak bylo zmíněno v sekci 2.2, jednou z možností, jak pohlížet na dění nejen na kryptoměnových, nýbrž i klasických trzích je prostřednictvím technické analýzy. Na funkčnost této metody však existuje řada protichůdných názorů. Zatímco někteří tvrdí, že je zavádějící a postrádá teoretický základ, jiní ji označují za „svatý grál“ obchodování. Článek *Efficient Capital Markets: A Review of Theory and Empirical Work* od ekonoma Eugene Fama, viz [56], hovoří jasně proti funkčnosti technické analýzy. Kniha *Market Wizards: Interviews With Top Traders* od Jacka D. Schwagera, viz [57], pak naopak uvádí řadu obchodníků obchodujících na klasických trzích, kteří profitují pouze na základě technické analýzy. Technická analýza je pro obchodování na klasických trzích hojně využívána nejen jednotlivci, nýbrž také zajišťovacími fondy, či investičními bankami. Na technických konceptech je silně závislé i tzv. vysokofrekvenční obchodování (angl. High-frequency trading), viz [58], které zahrnuje značné množství obchodovaného objemu na klasických burzách. To vše vedlo k myšlence použití technické analýzy i na trzích kryptoměnových. Trvalý úspěch v obchodování však vyžaduje nejen důkladnou znalost veškerých principů a metod technické analýzy, ale také dlouhodobou praxi, pozornost a disciplínu [59]. V současné době existuje již celá řada softwarů, které analýzu klasických i kryptoměnových trhů výrazně usnadňují jako např. Tradingview, NinjaTrader, Ticker Tocker, aj., viz [60]. Tyto softwary poskytují veškeré potřebné nástroje pro analýzu dat, včetně dat samotných, avšak správná interpretace trendů, vzorů a dat je vždy na obchodníkovi. Zobecnění technické analýzy společně s vývojem plně automatizovaného systému pro rozpoznávání vzorů a odhalení případných závislostí v datech z kryptoměnových burz je hlavní myšlenkou následujících sekcí. Mohla by to dokázat umělá inteligence, resp. hluboké neuronové sítě?

### 4.2 Formulace úlohy a výběr dat

Úspěšnost hlubokých neuronových sítí při řešení konkrétních úloh je silně podmíněna, ale rozhodně ne zaručena, vhodným výběrem dat, které je založeno na důkladných znalostech cílové domény. Nejprve bylo tedy nutné se s kryptoměnovými trhy a jejich fungováním pečlivě seznámit, následně formulovat konkrétní úlohu, pro ni nakonec zvolit vhodná data a příslušnou architekturu neuronové sítě. Zde je však

nutno poznamenat, že vhodnost dat závisí vždy pouze na subjektivním názoru, tj. neexistuje návod, podle kterého by se dalo při výběru dat řídit.

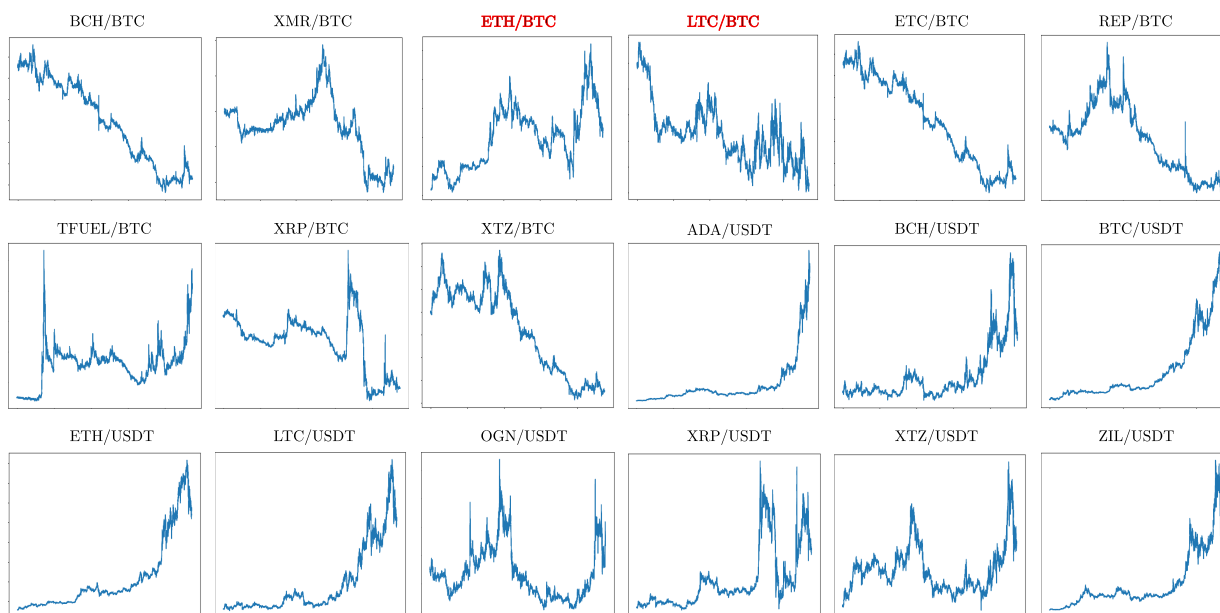
V oblasti kryptoměnových trhů, existuje nepřehledné množství úloh, které se prostřednictvím hlubokých neuronových sítí dají řešit. Otázka je, zda toto řešení bude využitelné v praxi. Jedná se o úlohy jak klasifikačního, tak regresního charakteru. Příkladem klasifikační úlohy může být např. rozhodnout, zda cena kryptoměny v budoucím období poroste nebo bude klesat. Mezi regresní úlohy pak patří např. předpověď budoucích cen, či jejich relativních změn, dále také předpověď obchodovaných objemů nebo klouzavých průměrů. Řešení všech těchto úloh přitom probíhá na základě vhodného počtu historických dat. Jako první pokus byla z této množiny úloh vybrána úloha predikce relativní změny v ceně dané kryptoměny v následujících 10 minutách.

Tato úloha přitom byla řešena za pomoci konvolučních neuronových sítí, jejichž hlavní přednosti a princip fungování byl nastíněn v sekci 3.3. K dispozici přitom byla datová sada obsahující ceny 18 kryptoměnových párů na trhu Binance, které byly zaznamenávány vždy po jedné minutě v období od 4. 4. 2020 do 28. 2. 2021. Tato data jsou veřejně dostupná přes aplikační programovací prostředí REST API, které používá dotazy HTTP pro přístup k datům. Výsledná datová sada má tedy formát časové řady obsahující zhruba 470 000 vzorků pro každý z 18 kryptoměnových párů, jejichž přehled je znázorněn pomocí tabulky 4.1. Vzhledem k tomuto formátu dat pak byla pro řešení úlohy zvolena 1D konvoluční síť.

BCH/BTC	XMR/BTC	ETH/LTC	BCH/USDT	ADA/USDT	ETH/USDT
ETC/BTC	XRP/BTC	XTZ/BTC	BTC/USDT	XRP/USDT	XTZ/USDT
REP/BTC	TFUEL/BTC	LTC/BTC	OGN/USDT	ZIL/USDT	LTC/USDT

*Tabulka 4.1: Přehled kryptoměnových párů*

Průběh cen mezi jednotlivými kryptoměnovými páry byl následně vykreslen, viz obrázek 4.1, a na základě vizuální analýzy těchto grafů, pro účely dalšího zkoumání, vybrán pouze jeden kryptoměnový pár, a sice Ethereum (ETH) vůči Bitcoinu (BTC). Tento pár byl vybrán proto, že jeho průběh je velmi rozmanitý, tj. vyskytuje se zde více různých period vzestupu i poklesu ceny, které se neuronová síť bude muset naučit. Jinak řečeno, data nemají jasně definovaný celkový trend, který se vyskytuje např. u párů BCH/BTC, ETC/BTC, či BTC/USDT. Ethereum je navíc v současné době pro mnoho obchodníků zajímavá kryptoměna, na trhu ETH/BTC lze tak zaznamenat velké množství transakcí, což je žádoucí. Podobné vlastnosti vykazuje např. kryptoměna Litecoin (LTC) jejíž cena je vztažena k ceně Bitcoinu (BTC). Nabízí se však otázka, zda je volba kryptoměnového páru ETH/BTC na základě analýzy grafů pouze minutových cen oprávněná, tj. zda bychom při znalosti cen např. v každých 10 sekundách nedospěli k jinému závěru. V rámci této práce tato otázka nebude již blíže rozpracována, avšak je možným předmětem dalšího zkoumání. Dále stojí za zmínku ještě fakt, že výběr pouze 1 kryptoměnového páru byl inspirován technickou analýzou, neuronové sítě by však potenciálně měly zachytit mnohem širší koncepty na bázi souvislostí mezi vývojem cen i několika kryptoměnových párů.



Obrázek 4.1: Vývoj cen mezi kryptoměnami v období od 4. 4. 2020 do 28. 2. 2021

Z obrázku je zřejmé, že cena prakticky všech kryptoměn vztažených k tetheru, USDT, ve sledovaném období prudce vzrostla.

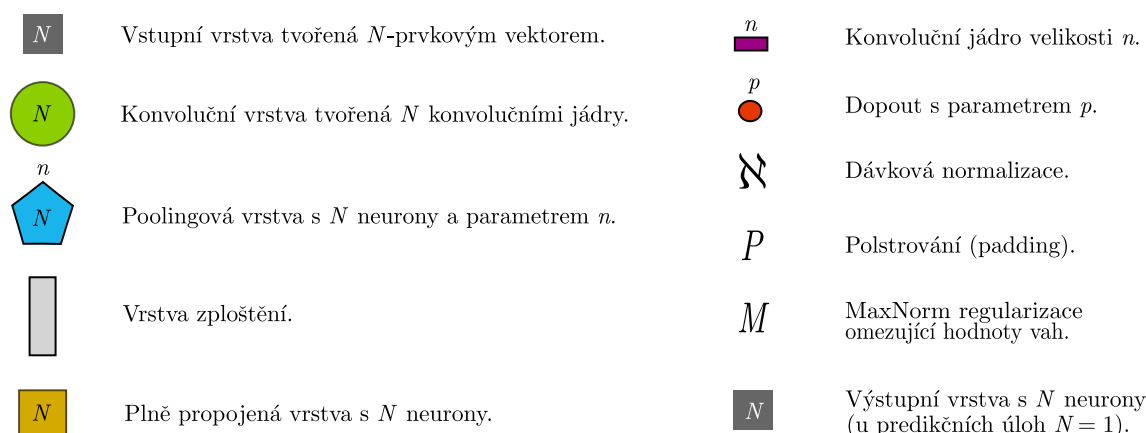
### 4.3 Programovací prostředí

Veškerá práce s daty i samotná implementace 1D konvoluční neuronové sítě byla provedena v programovacím jazyce Python verzi 3.8.5 v rámci distribuce Anaconda. Samozřejmě by se dalo využít i jakéhokoli jiného programovacího jazyka jako např. C++, Java, Julia nebo MATLAB. Python se však vyznačuje jednoduchým přístupem k objektově orientovanému programování, ale hlavně možností rychlého vývoje nového softwaru [61]. Distribuce Anaconda pak výrazně zjednodušuje správu knihoven, kterou zajišťuje open source správce balíčků a systém pro správu prostředí zvaný conda. Správce conda instaluje, spouští a aktualizuje veškeré knihovny a je součástí každé distribuce Anaconda. Součástí této distribuce je i knihovna Tensorflow, která umožňuje tvorbu a trénování modelu neuronových sítí pomocí rozhraní Keras API [62].

Samotné výpočty pak byly spuštěny na vysoce výkonném výpočetním klastru, tj. systému vzájemně propojených výpočetních serverů, kde se plánují a spouštějí výpočetní úlohy, zvaném HELIOS a vlastněném KM FJFI. Celý systém se skládá z výpočetních uzlů, vysokokapacitního úložného subsystému a serveru pro interaktivní přihlášení. Vzdálený přístup k tomuto serveru umožňuje např. open source software X2Go [63].

### 4.4 Vizualizace sítě

V dalším textu budou jednotlivé komponenty architektury 1D konvolučních neuronových sítí pro přehlednost znázorněny pomocí značek znázorněných na obr. 4.2.



Obrázek 4.2: Značení jednotlivých komponent architektur 1D konvolučních neuronových sítí

## 4.5 Seznámení s konvoluční neuronovou sítí

Samotné fungování 1D konvolučních neuronových sítí na regresních úlohách bylo nejprve otestováno pomocí jednodušší úlohy, a sice úlohy předpovědi následujícího členu aritmetické posloupnosti na základě znalosti 10 členů předchozích.

Prvním krokem bylo pro tuto úlohu opět zvolit vhodná data, která byla tentokrát vyrobena. Díky tomu byla struktura dat a v nich se vyskytující závislosti předem známy, což umožnilo jasně definovat funkce, které by se síť měla naučit, aby její předpovědi byly správné.

Za tímto účelem byly z intervalu  $(-10\,000, 10\,000)$  vygenerovány jedenáctičlenné segmenty 28 různých aritmetických posloupností s hodnotami diferencí  $\pm 2, \pm 5, \pm 7, \pm 10, \pm 14, \pm 18, \pm 20, \pm 25, \pm 29, \pm 31, \pm 37, \pm 42, \pm 44, \pm 46$ . Početní zastoupení jednotlivých segmentů příslušných aritmetických posloupností bylo ve vzniklé datové sadě zvoleno náhodně. V datové sadě tak byla zajištěna nerovnoměrná četnost výskytu jednotlivých aritmetických posloupností, resp. jejich segmentů. Náhodně bylo zvoleno i seřazení segmentů v rámci souboru. To zabránilo tomu, aby se síť naučila datový soubor jako telefonní seznam bez pochopení hlubších závislostí. Z každého jedenáctičlenného segmentu byl nakonec oddělen poslední prvek. Celý datový soubor se tak rozdělil na sobory dva. První, tvořený desetičlennými segmenty, představoval množinu dat vstupujících do sítě, druhý pak množinu příslušných dat výstupních (tzv. štítků). Část vstupní a výstupní množiny je, pro představu, zachycena na obrázku 4.3. Výsledná datová sada, resp. sada dvojic vstup-výstup, byla tvořená 9406 prvky.

Na základě takto vygenerovaných dat je možné jasně určit vlastnosti, které se má síť naučit rozeznávat - vzdálenost mezi jednotlivými členy posloupnosti a její monotonii.

Část vstupní množiny				Část výstupní množiny			
-2840	80	-3938	1090				
-2842	85	-3920	1080				
-2846	90	-3902	1070				
-2848	100	-3884	1060				
-2850	105	-3866	1050				
-2852	110	-3848	1040				
-2854	115	-3830	1030				
-2856	120	-3812	1020				
-2858	125	-3794	1010				
-2860	130	-3776	1000				

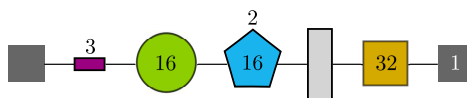
  

-2862	135	-3758	990
-------	-----	-------	-----

Obrázek 4.3: Část vstupní a výstupní množiny dat

Výsledná datová sada byla následně rozdělena na trénovací množinu, tvořenou 8906 prvky, a validační množinu, tvořenou 500 prvky. Validační množina tedy vznikla odebráním 500 prvků z konce výsledné datové sady a byla tak tvořena segmenty různých aritmetických posloupností, avšak se stejnými hodnotami diferencí vyskytujícími se u segmentů tvořících trénovací množinu. Obrázek 4.3 tedy obecně představuje část trénovací nebo validační množiny. Každý takový vstup byl pak předkládán 1D konvoluční neuronové síti, která na jeho základě učinila předpověď následujícího prvku v posloupnosti. Předpovězený prvek byl následně porovnán s příslušným štítkem.

Vzhledem k malému množství vstupních dat obsahujících pouze lineární závislosti byla zvolena velmi jednoduchá architektura 1D konvoluční neuronové sítě, jejíž schéma zachycuje obrázek 4.4.



Obrázek 4.4: Schéma použité 1D konvoluční sítě

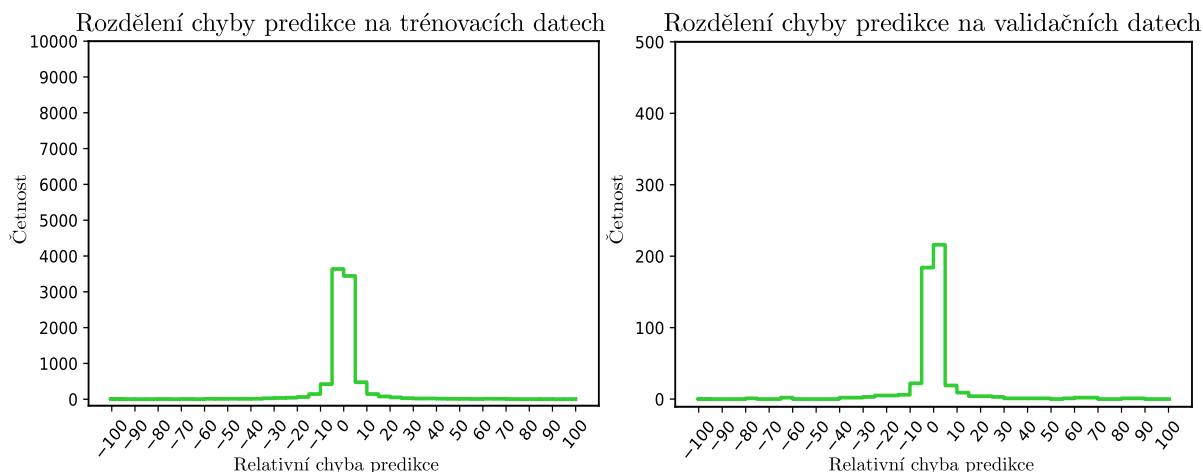
Tato architektura se skládá pouze z 1 konvoluční vrstvy tvořené 16 jádry o velikosti  $3 \times 1$ , jejíž výstup je aktivován funkcí ReLu, poolingové vrstvy s parametrem 2, výstupem této vrstvy je tedy maximum vzaté z každých dvou prvků tvořících výstup konvoluční vrstvy, dále vrstvy zploštění a vrstvy plně propojené tvořené 32 neurony. Jakožto ztrátová funkce byla zvolena MSE. Pro každou instanci dat byla vypočtena relativní chyba predikce pro všechna data pomocí vzorce

$$u(y_{p_i}, y_{s_i}) = \frac{y_{p_i} - y_{s_i}}{|y_{s_i}| + \epsilon}, \quad (4.1)$$

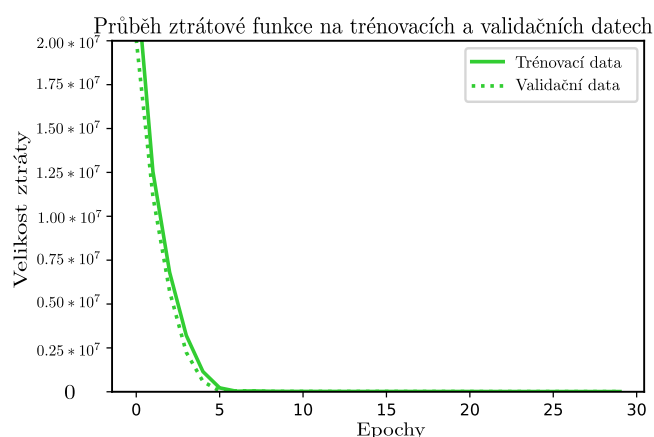
kde  $y_{p_i}$  značí hodnotu, která byla predikována sítí a  $y_{s_i}$  hodnotu štítku pro  $i$ -tý vzorek vstupních dat. Úspěšnost sítě pro tuto konkrétní úlohu byla následně vyhodnocena pomocí histogramu těchto relativních chyb. Relativní chyba predikce i výsledná úspěšnost sítě byly přitom vyhodnoceny pro testovací i validační data. Jak bylo uvedeno v sekci 3.2.1.2 k tomuto vyhodnocení dochází vždy po průchodu určité dávky trénovací množiny (batchy) sítí. Velikost této dávky byla v tomto případě nastavena na 1000 prvků, počet epoch, neboli počet jednorázových adaptací vah přes všechna trénovací data, pak na 30.

Výsledky tohoto experimentu zachycuje obrázek 4.5. Velikost chyby predikce se jak na trénovacích, tak i validačních datech pohybuje nejčastěji v rozmezí od -10% do 10%, což není vzhledem k charakteru

vstupních dat příliš. Jak je vidět z obrázku 4.6, ztrátová funkce s rostoucím počtem epoch rychle klesá k nule, a to jak na trénovacích, tak i validačních datech, což je žádoucí.



Obrázek 4.5: Rozdělení chyby predikce na trénovacích a validačních datech.

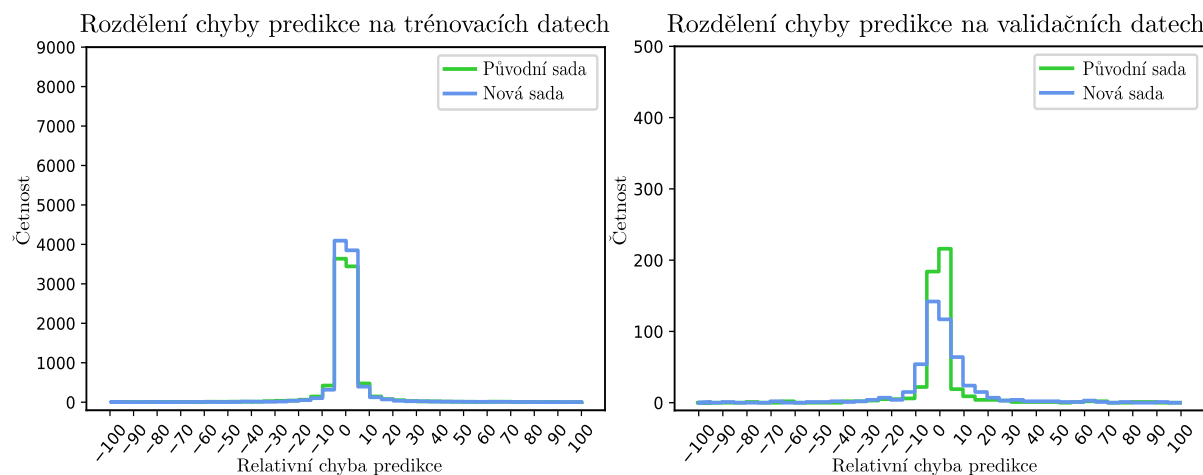


Obrázek 4.6: Průběh ztrátové funkce pro trénovací a validační data.

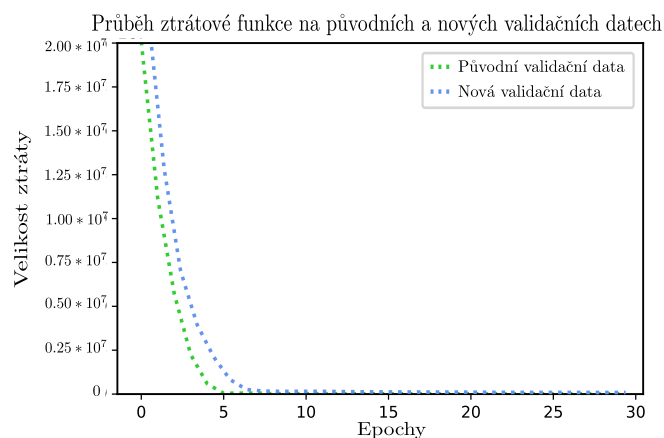
Při stejném nastavení, jaké bylo použito pro syntetická data z počátku sekce 4.5, tj. stejné architektury, velikosti dávek, epoch, ztrátové funkce i funkce vyhodnocující úspěšnost, se následně testovalo, jak bude síť úspěšná v predikci na validační množině obsahující segmenty aritmetických posloupností s jinými hodnotami diferencí, než které se vyskytovaly v trénovacích datech. Za tímto účelem byly na konec stávajícího souboru přidány dvojice segmentů (vstup-výstup) 5 aritmetických posloupností s diferencemi  $-100$ ,  $-85$ ,  $67$ ,  $72$  a  $93$ . Počet přidávaných dvojic segmentů jednotlivých aritmetických posloupností tentokrát nebyl náhodný, ale pevně nastaven na 100. Seřazení dvojic segmentů pak bylo náhodné v rámci původní a přidávané části. Výsledná datová sada, resp. sada dvojic vstup-výstup, čítala 9906 prvků. Opětným rozdělením výsledné datové sady pak bylo získáno 9406 původních dvojic tvořících trénovací množinu a 500 nových dvojic tvořících množinu validační.

Porovnání úspěšnosti predikce při využití nové a původní datové sady zachycuje obrázek 4.7. Lze nahlédnout, že v případě nové trénovací datové sady je četnost výskytu relativní chyby v intervalu  $-10\%$  do  $10\%$  vyšší než u původní trénovací datové sady. Vysvětlení lze najít ve skutečnosti, že nová trénovací množina obsahuje o 500 prvků více než množina původní, což vedlo i k nepatrné změně výsledku. Roz-

dělení chyby predikce na nových validačních datech ukazuje, že síť je v této úloze schopna generalizace. Hodnota chyby predikce sice častěji nabývá hodnot od -20% do 20%, namísto původních -10% do 10%, avšak to opět není vzhledem k charakteru dat příliš. Na obrázku 4.8 lze ještě porovnat průběh ztrátové funkce pro nová a původní validační data. Pro nová validační data klesá ztrátová funkce k nule zhruba o 3 epochy později.



Obrázek 4.7: Rozdělení chyby predikce na původních a nových trénovacích a validačních datech.



Obrázek 4.8: Porovnání průběhů ztrátových funkcí pro původní a nová validační data

Úspěšnost řešení výše uvedené úlohy za pomoci 1D konvolučních neuronových sítí byla motivací pro využití konvolučních sítí na úlohu formulovanou v sekci 4.2. Je však důležité zmínit, že formulovaná úloha je oproti úloze výše uvedené mnohem komplexnější. Časové řady ve formě cen kryptoměn mohou totiž obsahovat lineární, nelineární nebo také vůbec žádné závislosti.

## 4.6 Aplikace konvolučních neuronových sítí na formulovanou úlohu

### 4.6.1 Zpracování datového souboru

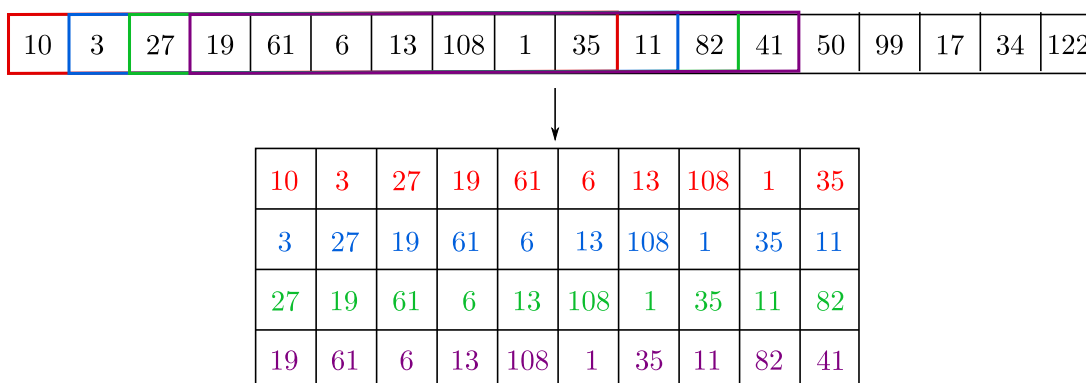
Zpracování datového souboru v podobě časové řady obsahující ceny Etherea vůči Bitcoinu se skládalo ze dvou částí - tvorby vstupních a výstupních dat a dalších úprav těchto dat. Mezi takové úpravy obecně



patří transformace, normalizace, či standardizace. Jednalo se tedy o proces výrazně složitější, než jaký byl popsán v sekci 4.5.

#### 4.6.1.1 Tvorba vstupních a výstupních dat

Časová řada byla nejprve rozdělena na vektory tvořící množinu vstupních dat. Generování těchto vektorů bylo prováděno s minutovými rozestupy, tj. sousední vektory se vzájemně lišily pouze o jednu cenu. Počet prvků v jednotlivých vektorech mohl být nastaven libovolně, viz 4.6.4. Ukázka vzniku vstupní množiny generováním desetiprvkového vektoru z časové řady je zachycena na obrázku 4.9.



Obrázek 4.9: Vznik vstupní množiny dat.

Členy výstupní množiny dat neboli štítky příslušející jednotlivým vstupním vektorům byly následně vypočteny pomocí vzorce

$$y_r = \frac{x_f - x_t}{x_t}, \quad (4.2)$$

kde  $x_t$  značí cenu příslušnou poslednímu prvku vstupního vektoru a  $x_f$  cenu o 10 minut později.

Výsledná vstupní množina byla tedy tvořena posloupnostmi cen Ethera vůči Bitcoinu, přičemž délka těchto posloupností odpovídala časovému úseku vzhledem k němuž byla činěna předpověď. Výsledná výstupní množina pak byla tvořena štítky, které vždy k dané posloupnosti představovaly relativní změnu v ceně Ethera vůči Bitcoinu v následujících 10 minutách.

#### 4.6.1.2 Předzpracování dat

Data z kryptoměnové burzy mohou být navíc, oproti datům vygenerovaným v úloze popsané v sekci 4.5 různých škálových rozpětí a také mohou obsahovat mnoho anomálií, tj. např. chybějící data v důsledku výpadku internetového připojení. To vše je před samotným trénováním sítě žádoucí odstranit, neboť by již v důsledku toho mohla být neuronová síť zcela nefunkční - poskytovala by zcela irelevantní výsledky. Za tímto účelem byla do procesu zpracování dat přidána ještě fáze tzv. předzpracování. Fáze předzpracování je stěžejní nejen pro tuto konkrétní úlohu, nýbrž obecně, proto bude rozebrána důkladněji.

Mezi široce používané metody předzpracování patří *transformace*, *normalizace* a *standardizace*. Kromě samotného účinku těchto metod na „surová“ data tkví hlavní rozdíl v jejich aplikaci. Transformace se provádí ještě před jakýmkoliv jiným zpracováním datové sady, na rozdíl od normalizace a standardizace, které probíhají na každém prvku vstupních dat samostatně.

V oblasti finančních časových řad lze pro *transformaci* s výhodou použít některý z technických ukazatelů, typicky klasické nebo exponenciální klouzavé průměry. Ve formulované úloze by to tedy znamenalo

brát jako množinu vstupních dat nikoliv přímo ceny, ale právě zmíněné průměry. Účelem transformace je poskytnout síti užitečné informace na rozumné úrovni podrobností, využitím klouzavých průměrů totiž dochází k vyhlazení dat, a tak snížení šumu vstupujícího do sítě. Zároveň je však třeba počítat s tím, že díky této transformaci mohou být ztraceny i některé užitečné informace [64]. V případě konvolučních neuronových sítí ale není tento typ předzpracování potřebný, neboť ho zajišťuje samotná konvoluční vrstva.

Normalizace a standardizace jsou metody, které lze použít ke konzistentní změně měřítka dat. Velké rozpětí hodnot v datech totiž může výrazně zvýšit obtížnost modelování úlohy. Pokud jsou např. vstupní hodnoty příliš velké (v řádech stovek nebo tisíců jednotek), může se stát, že se model naučí velké hodnoty vah. Modely s velkými váhovými hodnotami jsou však často nestabilní (nevýkonné a citlivé na vstupní hodnoty), což vede k vyšší chybě generalizace [65].

Data je možné *normalizovat* mnoha způsoby. Pojem normalizace dat obecně vyjadřuje změnu původního rozsahu měřítka do rozsahu předem určeného. Nejčastěji používaným druhem normalizace je změna původního rozsahu měřítka do rozsahu 0-1. Matematicky lze proces normalizace popsat pomocí vzorce

$$x_n = \frac{x - \min(X)}{\max(X) - \min(X)}, \quad (4.3)$$

kde  $x$  označuje hodnotu, která je normalizována,  $\min(X)$  minimální hodnotu z množiny vstupních dat označené jako  $X$ ,  $\max(X)$  maximální hodnotu z množiny vstupních dat  $X$ . V jazyce Python je však pro normalizaci dat možno využít funkci *MinMaxScaler*, která je součástí balíčku *preprocessing* z knihovny *sklearn* [66].

*Standardizace* datové sady naproti tomu zahrnuje změnu distribuce hodnot. Rozdělení standardizovaných dat má střední hodnotu 0, data jsou tzv. vycentrována, a směrodatnou odchylku 1. Důležitým předpokladem pro využití standardizace je, aby data měla normální rozdělení. Pokud tento předpoklad není splněn, mohou být výsledky měření nespolehlivé [65]. Matematicky lze standardizaci vyjádřit pomocí vzorce

$$x_s = \frac{x - \mu}{\sigma}, \quad (4.4)$$

kde  $x$  označuje hodnotu, která je standardizována a hodnoty  $\mu$  a  $\sigma$  dány vzorcem

$$\mu = \frac{1}{n} \sum_{i=0}^n x_i \quad \sigma = \sqrt{\frac{1}{n} \sum_{i=0}^n (x_i - \mu)^2}. \quad (4.5)$$

V jazyce Python je však stejně jako u normalizace k této práci s daty navržena funkce *StandardScaler*, která je součástí balíčku *preprocessing* z knihovny *sklearn* [66].

Často však nemusí být na první pohled zřejmé, jak data nejlépe, či zda vůbec, škálovat. Data se proto obvykle nejprve normalizují a pokud neuronová síť nefunguje příliš dobře, je možné se zamyslet nad jinými možnostmi předzpracování. Data určená pro úlohu predikce relativní změny v ceně tak byla normalizována. Takto předzpracovaná data byla následně uložena, aby se celý proces nemusel neustále opakovat.

Ukázka implementace celého procesu zpracování je obsahem přílohy 1.

#### 4.6.2 Sestavení modelu konvoluční neuronové sítě

Po předzpracování „surových“ dat přišlo na řadu sestavení modelu a jeho nastavení. Ještě před tím však byla data rozdělena na trénovací a validační množinu. Nepsaným pravidlem pro velké datové soubory je jejich rozdělení v poměru 98 : 2, tj. 98% všech dat tvoří trénovací množinu a zbylé 2% množinu validační. Tohoto pravidla bylo následováno, trénovací množina tak byla tvořena zhruba 440000 prvky, zbylých 10000 prvků pak tvořilo množinu validační. Samotnou tvorbu modelu umožňuje knihovna Keras,

jejíž součástí je i model *Sequential*, který byl použit u všech zkoumaných architektur. V rámci modelu *Sequential* za sebe byly postupně řazeny jednotlivé vrstvy, přičemž jejich druh a počet závisel na zvolené architektuře. Výstupy z každé vrstvy byly aktivovány funkcí ReLu.

U všech architektur bylo použito stejné nastavení ve smyslu stejné ztrátové funkce, velikosti dávek a počtu epoch. Za ztrátovou funkci, tj. funkci na základě jejíž hodnoty bude docházet k úpravě vah za účelem její minimalizace, byla zvolena Hubrova ztrátová funkce, velikost dávky byla nastavena na 50000 prvků vstupní množiny a počet epoch téměř ve všech případech na 50.

### 4.6.3 Vyhodnocení úspěšnosti sítě

Vyhodnocení úspěšnosti sítě je klíčovým a zároveň nejobtížnějším bodem celého procesu. Knihovna Keras obsahuje mnoho metrik, který tento proces výrazně usnadňují. Mezi metriky používané v regresních modelech, patří např. MSE, MAE či logaritmická chyba měření [67]. Běžně dostupné metriky však často nejsou postačující pro kompletní interpretaci fungování modelu na dané úloze. Tyto metriky v sobě sice nesou jistou informaci o jeho fungování, tato informace však nemusí být v kontextu konkrétní úlohy relevantní. V rámci knihovny Keras je však možné navrhnout i vlastní metriku pro vyhodnocování. Metrika, která byla použita pro vyhodnocení fungování všech testovaných architektur 1D konvoluční neuronové sítě na úloze predikce relativní změny v ceně byla definována vzorcem

$$z = \frac{1}{n} \sum_{i=1}^n u(y_{p_i}, y_{s_i}), \quad (4.6)$$

kde

$$u(y_{p_i}, y_{s_i}) = \frac{y_{p_i} - y_{s_i}}{|y_{s_i}| + \epsilon}, \quad (4.7)$$

přičemž  $y_{p_i}$  značí hodnotu predikovanou sítí a  $y_{s_i}$  hodnotu štitku pro  $i$ -tý prvek vstupní množiny dat. Vzorec (4.6) lze interpretovat jako průměr relativních chyb predikce pro všechna data. Úspěšnost sítě pro formulovanou úlohu pak byla vyhodnocena pomocí histogramu těchto relativních chyb.

Schéma implementace modelu včetně definice metriky, která byla použita k jeho vyhodnocení je obsahem přílohy 2.

### 4.6.4 Výsledky

Jak bylo nastíněno již v úvodu kapitoly, neexistuje univerzální způsob, pomocí kterého by pro řešenou úlohu bylo možné okamžitě zvolit správný model a navrhnout vhodnou architekturu neuronové sítě. V praxi se proto nejprve zvolí typ neuronové sítě (v této práci 1D konvoluční neuronová síť) a základní architektura. U této základní architektury se následně volí počet neuronů, přidávají jednotlivé komponenty, tj. vrstvy, či nejrůznější druhy regularizací, čím dochází ke vzniku nových architektur. Někdy se však může stát, že ani tímto způsobem nelze dospět k architektuře, která by v kontextu řešené úlohy poskytovala uspokojivý výsledek. Na tento problém se narazilo i při řešení úlohy predikce relativní změny ceny Etherea vztažené k ceně Bitcoinu v následujících 10 minutách, a to na základě cen Etherea vztažených k cenám Bitcoinu za určité historické období. Hlavním účelem této sekce je tak demonstrovat možný postup hledání vhodné architektury, představit možná vylepšení a sledovat jejich vliv na úspěšnost předpovědi. Za daných podmínek, tj. při zvoleném typu neuronové sítě a formátu datové sady, lze tímto postupem dospět nejlepší možné, avšak zdaleka ne optimální architektuře.

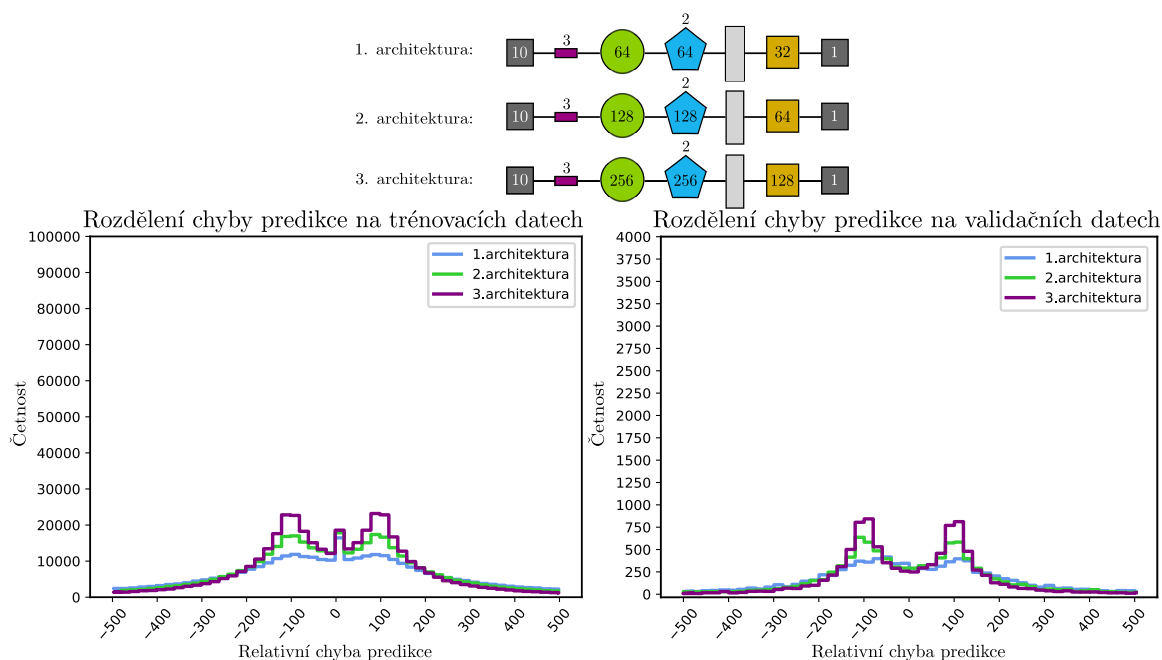
Za základní architekturu byla zvolena 1D konvoluční neuronová síť tvořená 1 konvoluční, 1 poolingovou

a 1 plně propojenou vrstvou. Vstupní vrstvu přitom tvořil vektor **10 hodnot**, tj. úkolem sítě tedy konkrétně bylo se naučit předpovídat relativní změnu ceny Etheria vztažené k ceně Bitcoinu v následujících 10 minutách, a to na základě cen z předchozích **10 minut**.

#### 4.6.4.1 Vliv počtu neuronů na úspěšnost předpovědi

Prvním předmětem zkoumání byl vliv počtu filtrů (dále souhrnně neuronů) v konvoluční, resp. neuronů v plně propojené vrstvě, na úspěšnost předpovědi. Volba vhodného počtu neuronů je velmi důležitá, neboť v případě nedostatečného, či naopak nadměrného počtu neuronů sítě ztrácí schopnost generalizace. Tuto skutečnost je velmi snadné vyčíst z grafu popisujícího průběh relativní chyby, které se síť během predikcí dopustila, resp. obecně z průběhu ztrátové funkce. Pokud by byl počet neuronů příliš malý, pak by relativní chyba predikce byla velmi vysoká, a to jak na trénovací, tak i validační datové sadě, tj. docházelo by k tzv. podučení. Naopak pokud by byl počet neuronů příliš vysoký, mohlo by dojít k tzv. přeučení, tj. relativní chyba predikce by na trénovací množině dat byla sice po celou dobu malá a na množině validačních dat by se tak zpočátku také mohla jevit, avšak s rostoucím počtem epoch by začala růst. Alternativně by mohla být relativní chyba predikce na validační datové sadě vysoká po celou dobu [68].

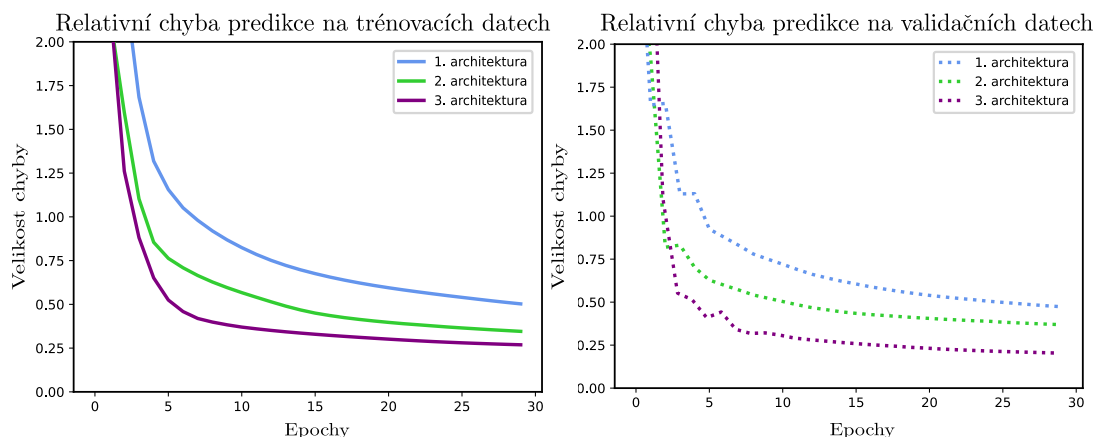
Výsledky zkoumání vlivu počtu neuronů v konvoluční a plně propojené vrstvě na rozdělení chyby predikce relativní změny ceny Etheria vztažené k ceně Bitcoinu v následujících 10 minutách, včetně schémat použitých architektur zachycuje obrázek 4.10. průběhy relativních chyb v predikci na trénovací a validační datové sadě pak obrázek 4.11.



Obrázek 4.10: Vliv počtu neuronů na rozdělení chyby predikce (4.7).

Z vyhodnocení rozdělení chyby predikce již na trénovací množině dat je zřejmé, že úspěšnost předpovědi ani u jedné z architektur není dostatečná na to, aby bylo možné výsledky využít k budování obchodní strategie. Nedostatečná úspěšnost předpovědi již pro data trénovací, pak přímo implikuje i nedostatečnou úspěšnost pro data validační, viz druhý graf. Oba grafy pak obsahují dva význačné vrcholy v oblasti kladných a záporných 100%, které naznačují, že síť má ve většině případů tendenci předpoví-

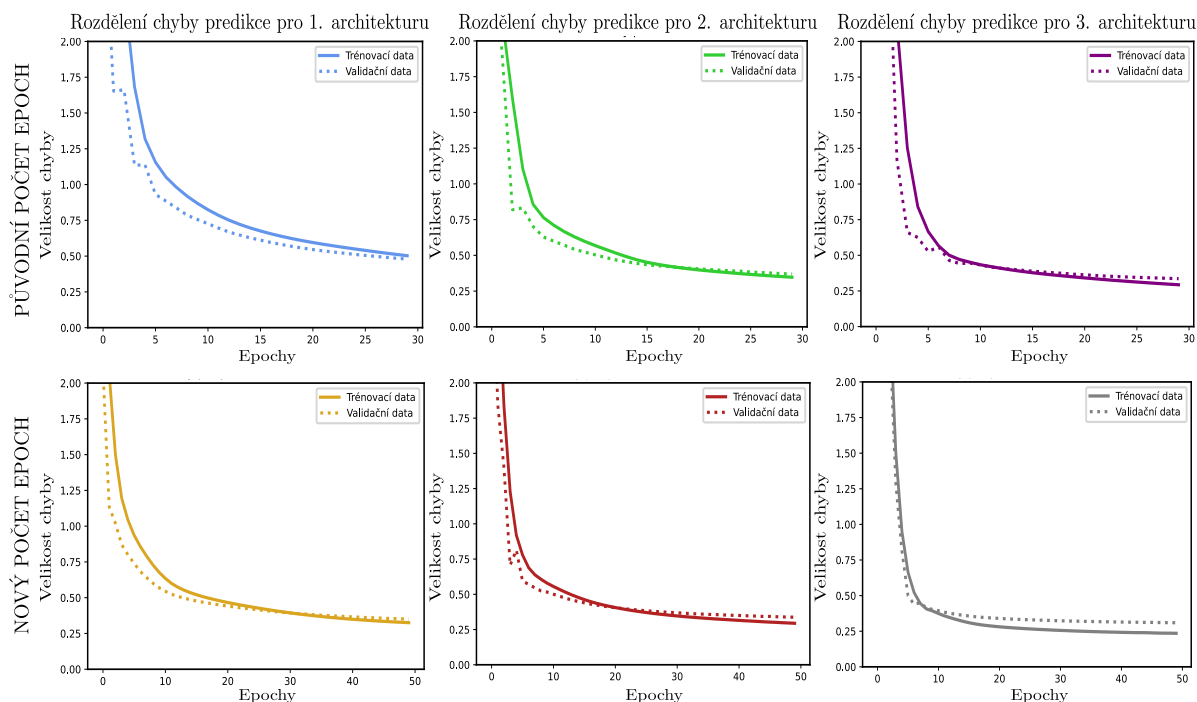
dat hodnotu blízkou nule. Vrchol na místě 0%, který lze zaznamenat v grafu rozdělení chyby predikce na trénovací datové sadě pak pravděpodobně, díky charakteru dat, která nečíslo obsahují nulové hodnoty, vyjadřuje právě správnou předpověď těchto hodnot, nikoliv však naprosto přesnou předpověď jiných hodnot. Výška obou grafů pak ještě, vzhledem k výrazně vyššímu celkovému počtu trénovacích a validačních dat, signalizuje, že chybovost sítě je velmi často mnohem vyšší než 500%, resp. -500%. Rozumným měřítkem, na jehož základě bude možné mezi sebou jednotlivé sítě porovnávat se zdá být rozdělení chyby predikce na škále od -100% do 100%.



Obrázek 4.11: Vliv počtu neuronů na průběh relativní chyby predikce (4.6).

Z obrázku 4.11 je zřejmé, že v případě všech zkoumaných architektur relativní chyba predikce klesá, a to jak na trénovací, tak i validační datové sadě. To tedy znamená, že ani v jednom případě jistě nedochází k přeučení. Architektura č.1 má však velikost chyby znatelně vyšší než zbylé dvě architektury, což značí nedostatečný počet parametrů. Na grafu zachycujícím relativní chybu predikce na validační množině dat lze dále u všech architektur zaznamenat počáteční oscilace, což může být způsobeno tím, že sítě jednoduše nějaké závislosti unikají.

Za účelem bližšího zkoumání průběhů relativních chyb predikce byl proveden další experiment, který spočíval ve zvýšení počtu epoch z 30 na 50 a následném sledování dalšího vývoje chyb. Je však třeba brát v úvahu fakt, že i nadměrný počet epoch může vést k přeučení sítě. To v tomto případě nastává, když tréninková datová sada jednoduše neobsahuje, vzhledem k počtu opakování tréninků, tolik závislostí. Sít' si v pozdějších epochách osvojuje již naučené závislosti a ztrácí schopnost generalizace [69]. Přetrénování sítě v důsledku vysokého počtu epoch lze však předejít zavedením funkce tzv. předčasného zastavení do modelu, což bylo také provedeno. Tato funkce sleduje jak ztrátovou funkci, tak i metriku, a to jak na trénovací, tak i validační množině dat a automaticky zastaví trénování, pokud jsou pozorovány výrazné přírůstky v hodnotách ztráty, resp. konstantní hodnoty příslušné metriky, tj. model se již dále nezlepšuje [70]. Výsledky tohoto experimentu při zachování jednotlivých architektur zachycuje obrázek 4.12, obsahující průběhy relativních chyb predikce u všech zkoumaných architektur pro 30 a 50 epoch.



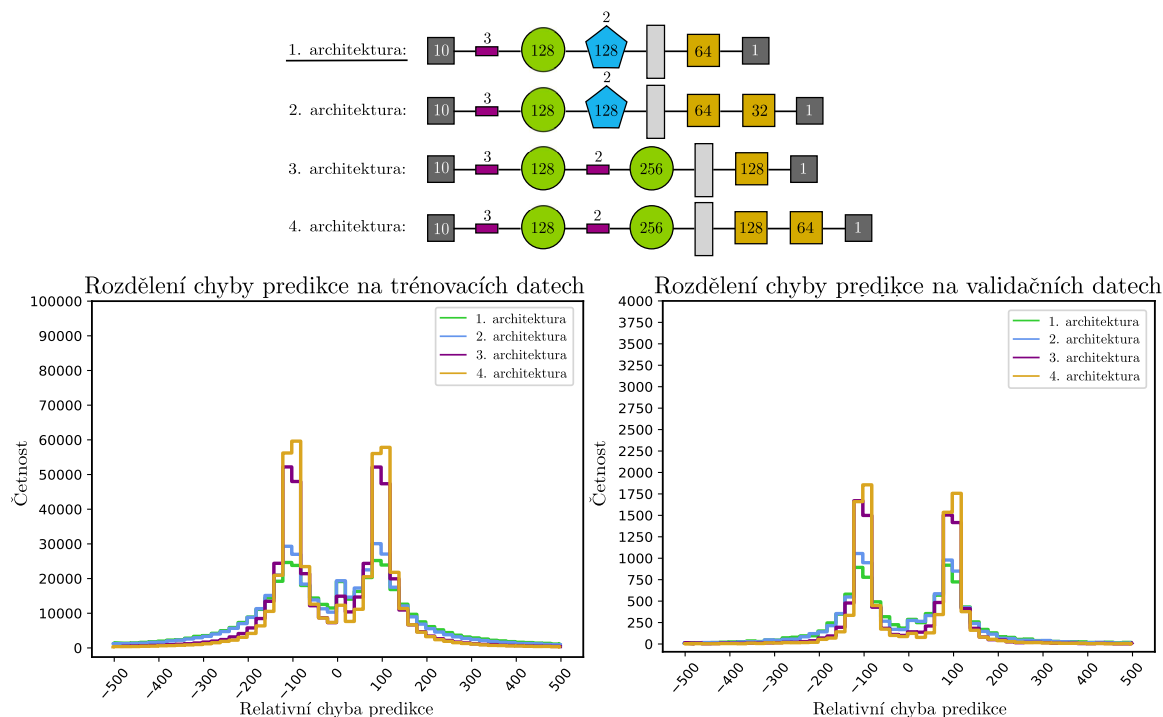
Obrázek 4.12: Vliv počtu neuronů na průběh relativní chyby predikce (4.6) při zvýšení počtu epoch z 30 na 50.

Obrázek 4.12 ukazuje, že v pokročilejších stádiích validace modelu již k žádným výkyvům relativní chyby predikce nedochází. Lze také nahlédnout, že ani u jedné z architektur nedochází k předčasnému zastavení, tj. zvýšení epoch z 30 na 50 nezpůsobilo přetrénování. Hodnota relativní chyby predikce pro validační množinu dat sice v určitém okamžiku u všech architektur překročí hodnotu relativní chyby predikce pro množinu dat trénovacích, avšak stále má klesající, nikoliv rostoucí, charakter.

Na základě porovnání všech výše uvedených výsledků byla jakožto výstupní (nejoptimálnější) architektura této sekce zvolena architektura č. 2 a nastavení počtu epoch na 50. Zdálo by se být možná vhodnější zvolit architekturu č. 3, ta však již nyní obsahuje poměrně mnoho parametrů a vzhledem k následujícímu postupu při hledání nejlepší možné architektury, tj. přidávání dalších vrstev, by mohlo lehce dojít k přetrénování.

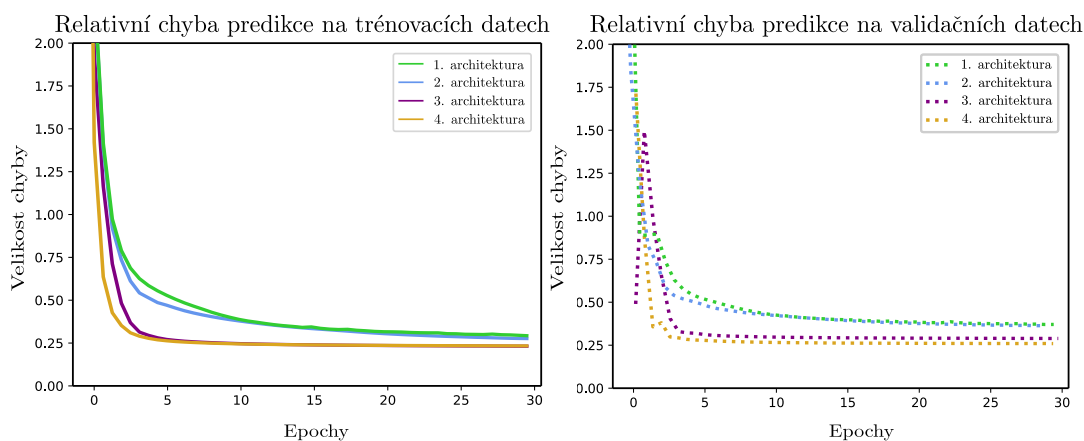
#### 4.6.4.2 Vliv počtu a typu vrstev na úspěšnost předpovědi

Předmětem dalšího zkoumání byl vliv počtu jednotlivých vrstev 1D konvoluční neuronové sítě na úspěšnost předpovědi. Jak bylo demonstrováno v sekci 4.5, jednovrstvé sítě jsou postačující, pokud je datová sada velmi jednoduchá a obsahuje lineární závislosti, reprezentované spojitými funkcemi. Komplexní datové sady, mezi něž patří i časové řady, však obvykle vyžadují vyšší počet vrstev, díky nimž lze ze vstupní datové sady extrahovat i mnohem složitější funkce [68]. V kontextu konvolučních neuronových sítí je třeba rozhodnout o vhodném počtu konvolučních vrstev, které jsou následovány vrstevmi poolingovými, a počtu plně propojených vrstev. Zde se však ještě nabízí otázka, zda by pro zpracování časové řady obsahující ceny Etherea vztažené k ceně Bitcoinu nebylo vhodnější poolingovou vrstvu nahradit další konvoluční vrstvou s velikostí jádra 2. Výsledek aplikace této úvahy na rozložení chyby predikce včetně příslušných schémat zkoumaných architektur zachycuje obrázek 4.13. Průběhy relativních chyb predikce pak obrázek 4.14.



Obrázek 4.13: Vliv počtu a typu vrstev na rozdělení chyby predikce (4.7).

Porovnáním rozdělení chyb predikce v případě použití 1. (původní) a 2. architektury lze nahlédnout, že zvýšení počtu plně propojených vrstev z jedné na dvě vede k mírnému procentuálnímu nárůstu chyb v oblasti od  $-100\%$  do  $100\%$  a to na úkor snižování počtu chyb větších než  $-500\%$ , resp.  $500\%$ . Dále je zřejmé, že náhradou poolingové vrstvy další konvoluční vrstvou s velikostí jádra 2, tj. porovnáním 2. a 4. architektury, se zvyšuje tendence síť předpovídat hodnoty blízké nule, a to nikoliv na úkor velmi vysokých chyb, ale spíše nižších chyb v rozmezí od  $-100\%$  do  $100\%$ . Toto pozorování tedy potvrzuje vyhlazovací vlastnosti konvolučních vrstev.

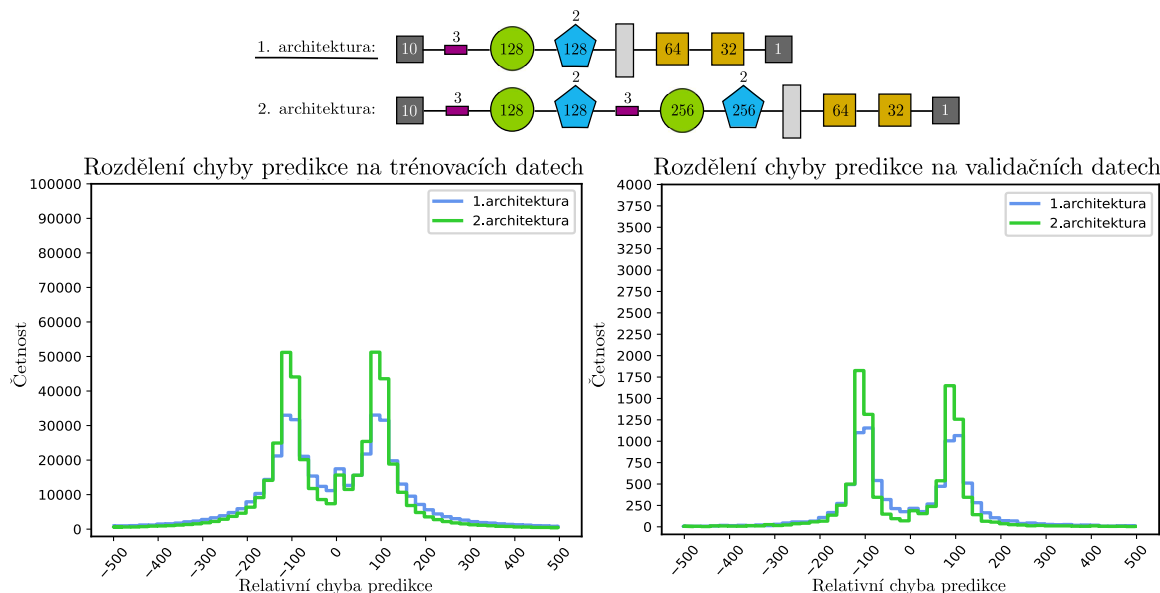


Obrázek 4.14: Vliv počtu a typu vrstev na průběh relativní chyby predikce (4.6).

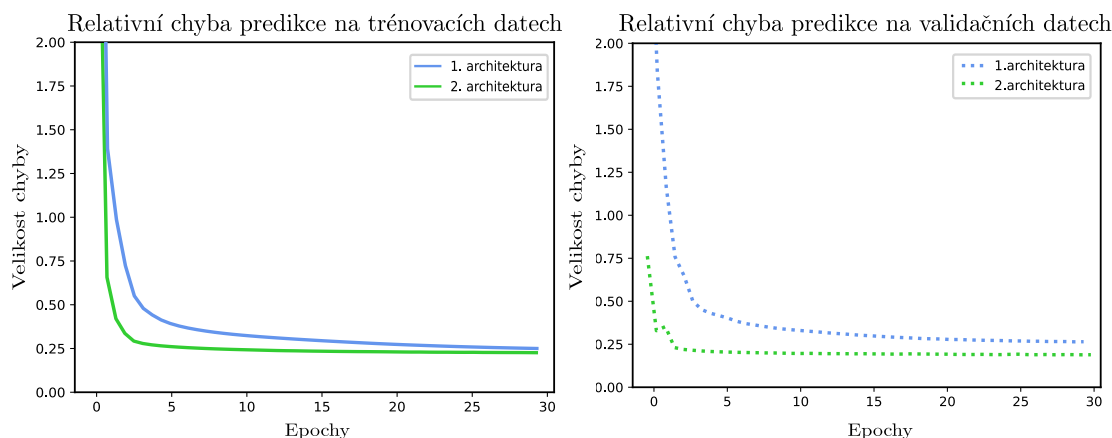
Obrázek 4.14 znázorňuje, že použití konvolučních vrstev místo poolingových, tj. relativní chyba predikce pro architekturu č. 3 a architekturu č. 4, má sice za následek strmější pokles relativní chyby predikce, v souladu s výše uvedeným to však nutně nemusí značit větší úspěšnost sítě. Relativní chyba po přidání

plně propojené vrstvy, tj. u architektury č. 2 resp. č. 4, nadále klesá, a to jak pro trénovací, tak i validační datovou sadu, což je žádoucí.

S ohledem na mírné zlepšení úspěšnosti predikce po přidání plně propojené vrstvy u architektury s jednou konvoluční vrstvou následovanou vrstvou poolingovou, spočíval další experiment v přidání druhé série těchto vrstev. Obrázky 4.15 a 4.16 zachycují výsledky tohoto experimentu.



Obrázek 4.15: Vliv počtu vrstev na relativní chybu predikce (4.7).



Obrázek 4.16: Vliv počtu vrstev na průběh relativní chyby predikce (4.6).

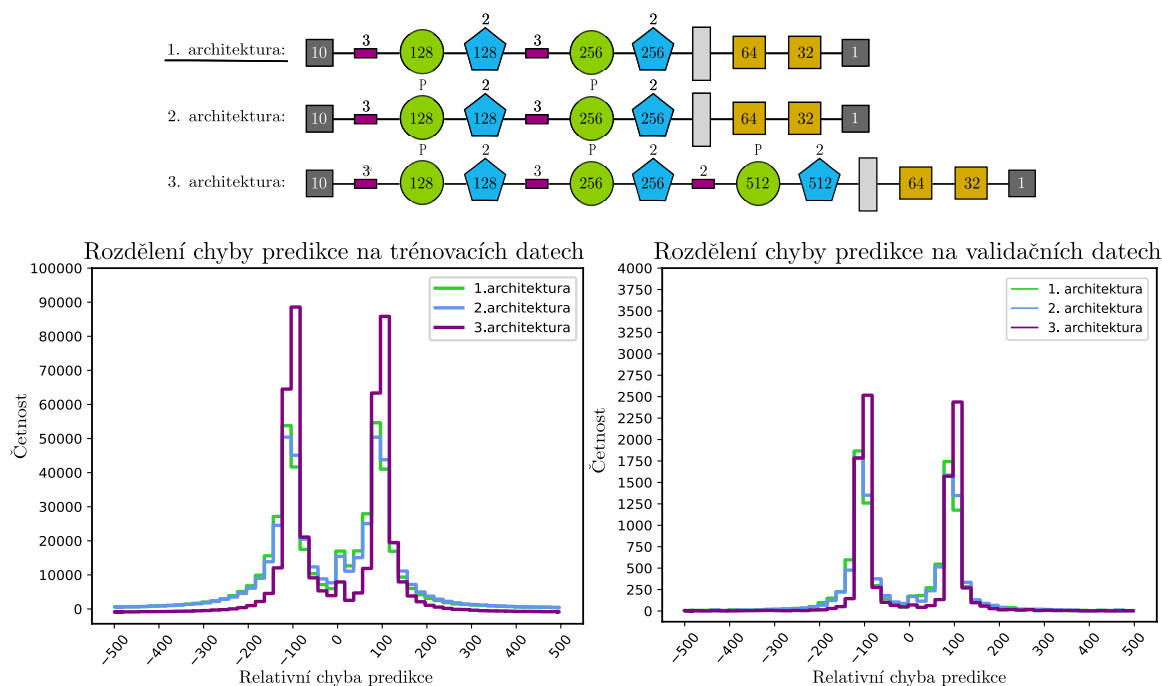
Přidáním další konvoluční vrstvy následované vrstvou poolingovou se zvyšuje četnost 100%, resp. –100% chyb, avšak tentokrát nikoliv na úkor malých chyb v rozmezí od –100% do 100%, ale chyb vyšších než –500%, resp. 500%. Zároveň opět dochází ke strmějšímu klesání relativní chyby, a to jak na testovací, tak i validační datové sadě. Pro další zvýšení úspěšnosti predikce se zdá být logické zkusit přidat další vrstvu. Jak bylo však popsáno v sekci 3.3.1.1, v důsledku aplikace konvolučních vrstev samotných dochází ke zmenšení rozměrů vstupujících vektorů, čím spíše pokud za vrstvami konvolučními následují ještě vrstvy poolingové. Vektor tvořený 10 hodnotami, který tvoří vstupní vrstvu doposud zkoumaných struktur



se průchodem 1 konvoluční vrstvy s jádrem o velikosti 3 zmenší na vektor 8 hodnot a následnou aplikací poolingové vrstvy pak dokonce na vektor tvořený pouze 4 hodnotami. Použitím další série těchto vrstev je tedy výsledná velikost vektoru vstupujícího do plně propojené vrstvy rovna 1. Přidání třetí série těchto vrstev tak obecně není možné. Pro přidání další série konvoluční vrstvy následované vrstvou poolingovou je nutné buďto v jednotlivých vrstvách využít polstrovaní nebo zvětšit velikost vstupního vektoru.

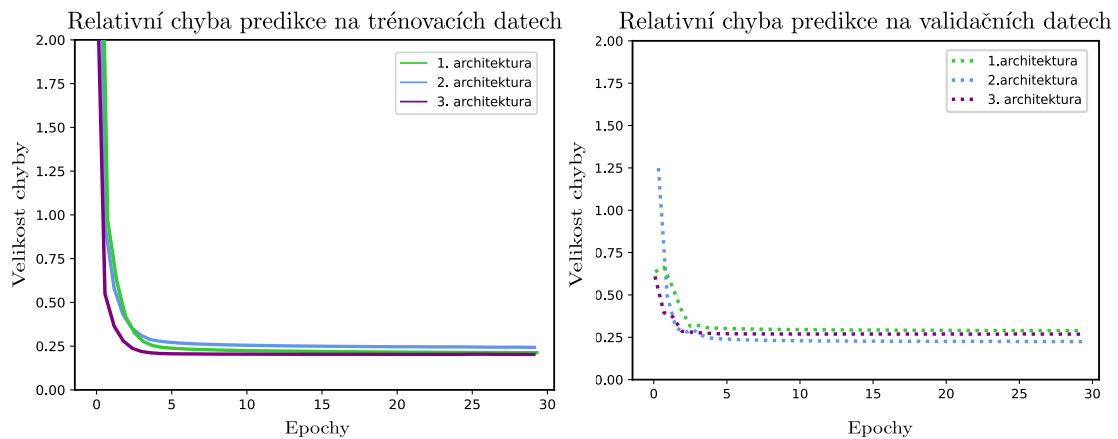
#### 4.6.4.3 Vliv polstrovaní na úspěšnost předpovědi

Jak bylo důkladně rozebráno v sekci 3.3.1.1, polstrovaní je nástroj, jehož aplikací nedochází průchodem jednotlivých vektorů konvolučními vrstvami ke zmenšení jejich velikosti, tj. ztrátě krajních hodnot. V důsledku toho je pak působení vlivu krajních hodnot těchto vektorů ve vrstvách pozdějších výraznější. Zachování velikosti vektoru po jeho průchodu konvoluční vrstvou zároveň umožňuje přidání třetí série konvoluční vrstvy následované vrstvou poolingovou. Konkrétní schéma architektur, které byly použity v tomto experimentu společně s rozdělením chyby predikce při jejich využití znázorňuje obrázek 4.17. Obrázek 4.18 pak ještě zachycuje průběhy relativních chyb predikce u všech tří architektur.



Obrázek 4.17: Vliv polstrovaní na rozdělení chyby predikce (4.7).

Porovnáním rozdělení chyby predikce při použití 1. (původní) a 2. architektury je zřejmé, že přidáním polstrovaní nedochází k výraznému zpřesnění predikcí. Zahrnutí vlivu krajních hodnot do dalších vrstev tedy na úspěšnost predikce nemá žádný efekt. Další pozorování pak souvisí s přidáním třetí série vrstev, při němž opět došlo k výraznému nárůstu předpovědí čistých nul, a to mj. na úkor předpovědí mírně odchýlených od skutečných hodnot, tj. chyb z rozmezí od -100% do 100%.

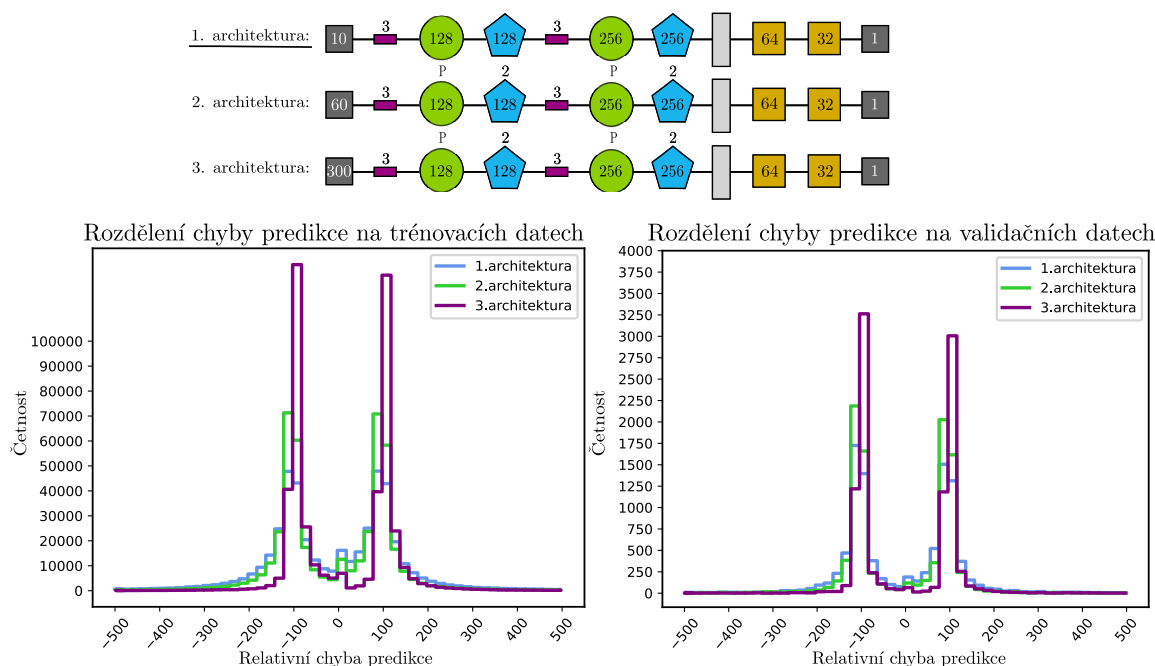


Obrázek 4.18: Vliv polstrovaní na průběh relativní chyby predikce (4.6).

Průběh relativních chyb predikce znázorněný na obrázku 4.18 vykazuje u všech architektur zhruba od 5. epochy velmi pomalý pokles, což obecně není žádoucí, neboť to znamená, že se chyba s rostoucím počtem epoch již nijak výrazně nesnižuje.

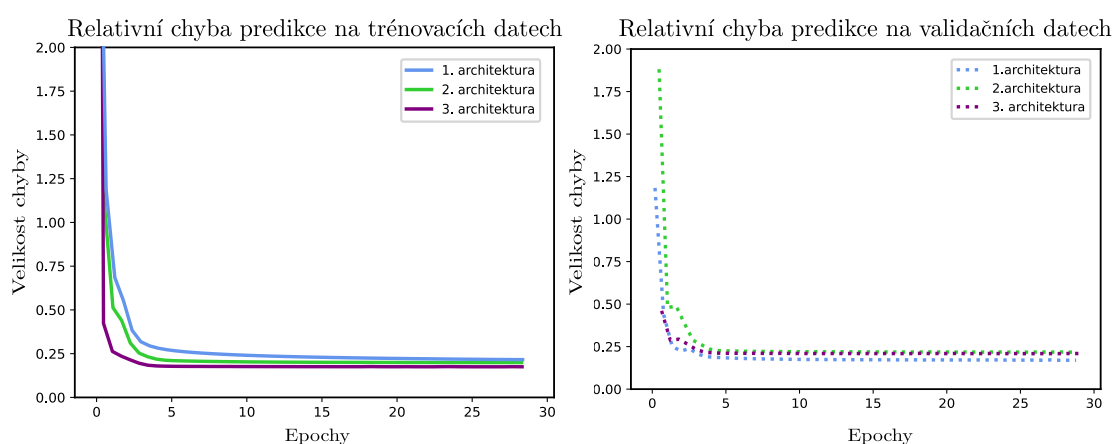
#### 4.6.4.4 Vliv velikosti vstupního vektoru na úspěšnost předpovědi

Vzhledem k tomu, že přidání třetí série konvoluční vrstvy následované vrstvou poolingovou nevedlo ke zvýšení úspěšnosti predikce, nemá smysl zvětšovat počet hodnot tvořících vektor vstupující do sítě za tímto účelem. Obecně by však velikost vstupního vektoru mohla mít na úspěšnost předpovědi nějaký vliv. Prozkoumání případného vlivu velikosti vstupního vektoru na úspěšnost předpovědi u dvouvrstvé architektury s přidaným polstrovaním do každé konvoluční vrstvy, tj. architektury č. 2 z předchozí sekce, je proto předmětem této sekce. Díky přidanému polstrovaní je totiž možné sledovat pouze vliv velikosti vstupního vektoru jako takový bez přidaných vlivů způsobených změnou velikosti jednotlivých vektorů průchodem skrz konvoluční vrstvy. Vstupní vektor byl přitom zvětšen z 10 hodnot na 60 a 300. Tzn. že síť měla za úkol předpovídat relativní změnu ceny Ethereum vztaženou k ceně Bitcoinu na základě cen za poslední 1 hodinu (tj. 60 minut) a 5 hodin (tj. 300 minut). Výsledky měření pro zmíněné velikosti vstupních vektorů zachycují obrázky 4.19 a 4.20.



Obrázek 4.19: Vliv velikosti vstupního vektoru na rozdělení chyby predikce (4.7).

Lze nahlédnout, že zvětšením vektoru z 10 na 60 hodnot, tj. porovnáním 1. a 2. architektury, došlo k mírnému poklesu četností chyby predikce v rozmezí od  $-100\%$  do  $100\%$  a opětovnému zvýšení počtu předpovědí nulových hodnot. Zdá se tedy, že s rostoucím počtem hodnot tvořících vektor vstupující do sítě se zvyšuje tendence sítě předpovídat hodnoty blízké nule. To je možné interpretovat tak, že čím delší je historický vývoj ceny Etherea vztažené k ceně Bitcoinu na základě něhož jsou činěny predikce, tím méně se síť snaží předpovídat jiné, než nulové hodnoty. Starší data tak ptavděpodobně nemají na bezprostřední vývoj ceny de facto žádný vliv. Tento fakt pak potvrzuje graf pro architekturu č. 3, kde je již tato tendence naprosto zřejmá.



Obrázek 4.20: Vliv velikosti vstupního vektoru na průběh relativní chyby predikce (4.6).

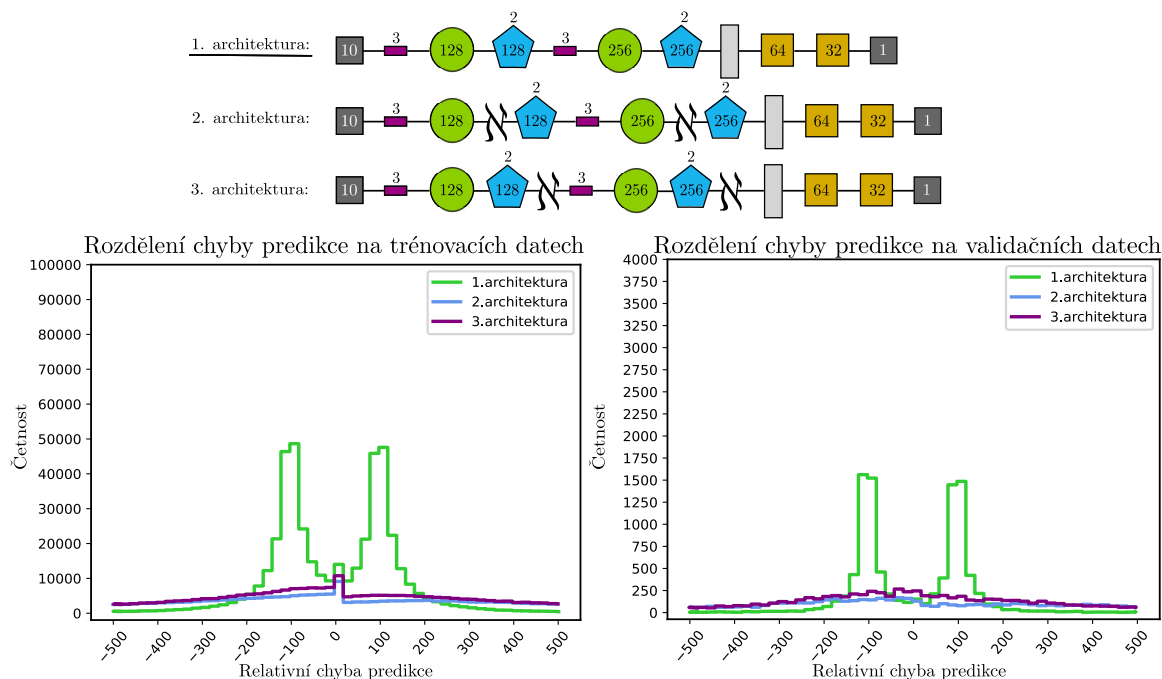
Z průběhu relativních chyb predikce u všech zkoumaných архитектур je pak opět vidno, že se zhruba od 5. epochy tento průběh téměř nemění, tj. nedochází k výraznému snižování velikosti relativní chyby predikce.

Vyhodnocením všech dosavadních výsledků byla za architekturu, která v porovnání s architekturami ostatními vykazuje nejlepší výsledky zvolena architektura se dvěma sériemi konvolučních vrstev následovaných vrstvami poolingovými bez přidaného polstrování, dvěma plně propojenými vrstvami a vstupním vektorem tvořeným 10 hodnotami. I u této architektury byl však přibližně po 5 epochách pozorován téměř neměnný průběh relativní chyby predikce, což znamená, že síť po jistém čase již nedokázala výrazně snižovat přesnost svých předpovědí. Tento problém by však mohl být alespoň částečně odstraněn zavedením některé z regularizačních technik, mezi něž patří např. dávková normalizace nebo dropout.

#### **4.6.4.5 Vliv dávkové normalizace na úspěšnost předpovědi**

Dávková normalizace je technika, jejíž aplikací dochází k automatické standardizaci vstupů do určených vrstev tvořících danou architekturu, a to vždy v rámci jedné dávky. Tato technika částečně předchází tzv. „Internal Covariate Shift“ problému, kdy se v průběhu aktualizace vah v každé dané vrstvě zároveň mění i data, která do vrstvy vstupují. Změna dat je přitom způsobena aktualizací vah v předchozích vrstvách, která probíhá současně s aktualizací vah dané vrstvy. Změna distribuce vstupních vektorů související se změnou váhových hodnot zpomaluje trénink a naprosto přesnou inicializaci parametrů, což výrazně ztěžuje řešení úloh se silně nelineárními závislostmi. Pokud jsou však vstupy předem standardizovány, pak nedochází k tak výrazným změnám v distribucích, díky čemuž je trénování sítě rychlejší a schopnost generalizace vyšší [71].

V rozhraní Keras je dávková normalizace implementována jako transformace vstupních vektorů tak, aby měly střední hodnotu rovnou 0 a směrodatnou odchylku rovnou 1 (tj. standardizace). Lze ji umístit před nebo až po aplikaci aktivační funkce. Pokud model obsahuje aktivační funkce jako např. sigmoid nebo hyperbolický tangens, pak se dávková normalizace obvykle aplikuje až po těchto funkcích. Naopak u aktivační funkce ReLu, která je součástí všech doposud zkoumaných architektur, se dávková normalizace standardně umísťuje před tuto aktivaci [72]. Tohoto doporučení bylo využito k aplikaci dávkové normalizace do níže diskutovaných modelů. Vliv dávkové normalizace na úspěšnost sítě předpovídat relativní změnu v ceně Etherea vztahenou k ceně Bitcoinu zachycuje obrázek 4.21.



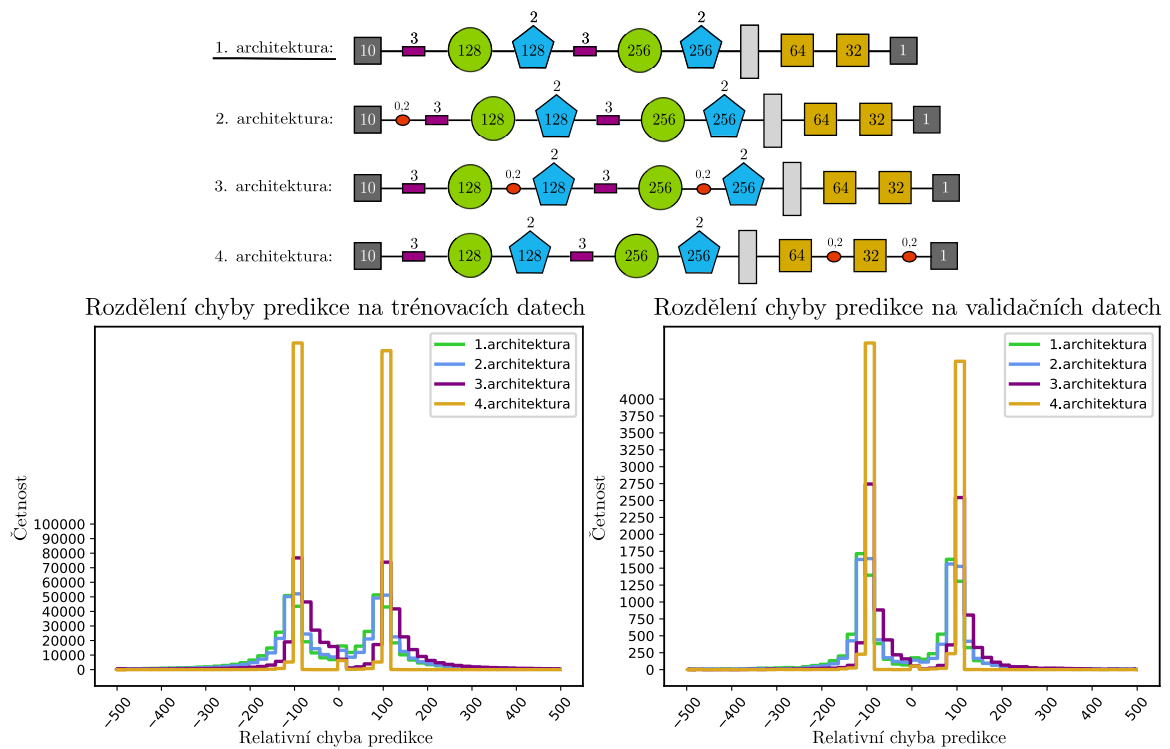
Obrázek 4.21: Vliv dávkové normalizace na rozdělení chyby predikce (4.7).

Z obrázku 4.21 je zřejmé, že využití dávkové normalizace v 1D konvoluční neuronové síti určené pro řešení této konkrétní regresní úlohy není vhodná. Vede totiž k výrazně vyšší četnosti velkých chyb, tj. chyb větších než  $-500\%$ , resp.  $500\%$ .

#### 4.6.4.6 Vliv dropoutu na úspěšnost předpovědi

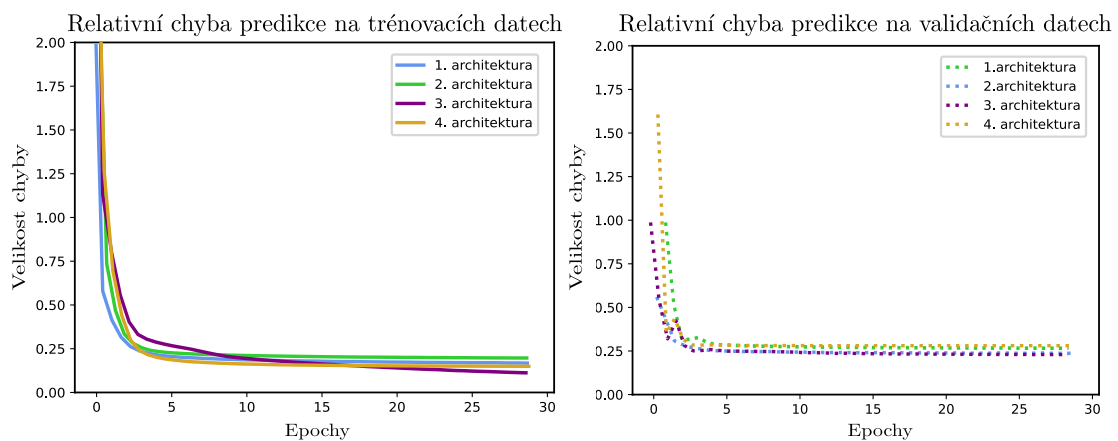
Dropout je regularizační technika, která spočívá ve vyřazení náhodně vybraných neuronů během tréninku, nikoliv však při validaci. Příspěvek těchto neuronů k aktivaci neuronů v následující vrstvě je tedy dočasně odstraněn a zároveň u těchto neuronů nedochází k aktualizaci vah během zpětného šíření. Předpokladem aplikace dropoutu je, že se síť naučí více nezávislých interních reprezentací, neboť při náhodném výpadku některých neuronů musí informace, které tyto neurony kódovaly zachytit neurony zbývající. V konečném důsledku se tak síť vlivem dropoutu stává méně citlivá na specifické váhy neuronu, což vede ke schopnosti lepší generalizace [73].

Dropout je v rozhraní Keras implementován jako náhodný výběr neuronů, jež jsou s danou pravděpodobností v každé epoše vyřazeny. Dropout s parametrem 0,2 tedy značí vyřazení 1/5 náhodně vybraných neuronů ze všech neuronů tvořících vrstvu, na níž je dropout aplikován. Hodnotu parametru, tj. velikost pravděpodobnosti, je však třeba důkladně zvážit. Příliš nízká hodnota toho parametru bude mít na síť minimální účinek, naopak příliš vysoká hodnota povede k výraznému omezení počtu neuronů v jednotlivých vrstvách a s tím spojené podučením. Tuto regularizační techniku je možné aplikovat na libovolnou vrstvu, z níž se daná architektura skládá, pouze s výjimkou vrstvy výstupní [74]. Výsledky zavedení dropoutu do vstupní, konvolučních a plně propojených vrstev zachycují obrázky 4.22 a 4.23.



Obrázek 4.22: Vliv dropoutu na rozdělení chyby predikce (4.7).

Na základě rozdělení chyby predikce při použití 1. (původní) a 2. architektury lze usoudit, že náhodné vyřazení neuronů ve vstupní vrstvě, tj. vyřazení některých hodnot vstupního vektoru, nemá na výslednou úspěšnost sítě žádný účinek. Zdá se však být výhodné zavést dropout do vrstev konvolučních, viz. 3. architektura, neboť tím dochází k výraznému nárůstu –100%, resp. 100% chyb, a to na úkor chyb větších než –500%, resp. 500%. Velmi špatné výsledky jsou pak zřejmé v případě 4. architektury, kde byl dropout aplikován na plně propojené vrstvy. Sít' v tomto případě předpovídá pouze čistě nulové hodnoty.

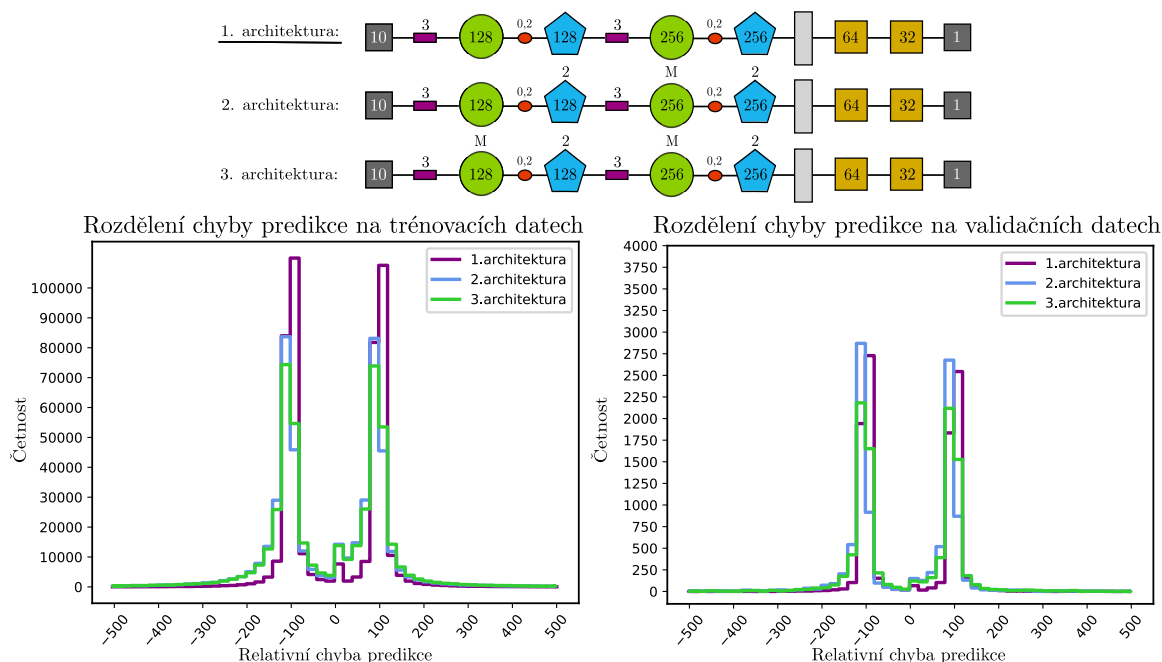


Obrázek 4.23: Vliv dropoutu na průběh relativní chyby predikce (4.6).

Z průběhů relativních chyb predikce je v případě 3. architektury taktéž zřejmé zlepšení. Alespoň na trénovacích datech má totiž průběh relativní chyby klesající charakter. U zbylých архитектур je průběh opět téměř konstantní.

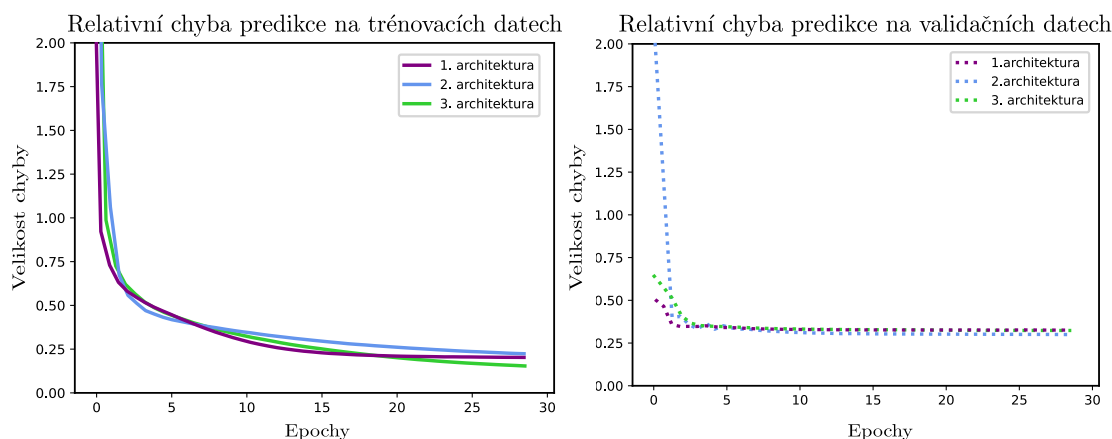
#### 4.6.4.7 Vliv dropoutu a MaxNorm regularizace omezující hodnoty vah na úspěšnost předpovědi

V mnoha modelech strojového učení bylo pozorováno výrazné zlepšení v jejich fungování při použití dropoutu v kombinaci s regularizací omezující velikost váhového vektoru příslušejícího každému neuronu v maximové normě, tzv. MaxNorm regularizací. Tato regularizace zajišťuje, že velikost vektoru vah příslušícímu danému neuronu nepřekročí stanovenou hranici. V praxi k tomuto omezení dochází až po aktualizaci všech váhových hodnot, tj. po zpracování jedné dávky [73]. Kombinace dropoutu s Max-Norm regularizací obecně zvyšuje stabilitu modelu, její účinnost u řešené úlohy je pak zřejmá z obrázku 4.24, resp. obrázku 4.25.



Obrázek 4.24: Vliv dropoutu a MaxNorm regularizace na rozdělení chyby predikce (4.7).

Lze nahlédnout, že již zavedení MaxNorm regularizace omezující váhové hodnoty do jedné konvoluční vrstvy, viz architektura 1. vs. architektura 2., vede ke zvýšení četnosti relativně malých chyb predikce, tj. chyb v rozmezí od  $-100\%$  do  $100\%$ , v důsledku čehož klesá četnost předpovídání pouze nulových hodnot. Tento trend se pak potvrdil rozdělením chyby predikce u 3. architektury, kde byla MaxNorm regularizace aplikována na obě konvoluční vrstvy. Zároveň se zvyšováním četnosti malých chyb se však zvyšuje i četnost chyb mírně vyšších než  $-100\%$ , resp.  $100\%$ . Sít' tak sice nemá tolik tendenci předpovídat hodnoty blízké nule, její četnost správných predikce je však stejně velmi malá.



Obrázek 4.25: Vliv dropoutu a MaxNorm regularizace na průběh relativní chyby predikce (4.6).

Po kombinaci dropoutu s MaxNorm regularizací omezující váhové hodnoty průběh relativní chyby predikce na trénovacích datech i nadále klesá, což je žádoucí.

#### 4.6.4.8 Vliv dalších možných úprav na úspěšnost předpovědi

Pro formulovanou úlohu byla výše uvedeným postupem prozkoumána řada architektur a možných nastavení. Bohužel se však prostřednictvím nich nepodařilo dospět k výsledkům, které by se daly použít v praxi. Tyto výsledky však poskytují jistý odrazový můstek pro budoucí práci, neboť jsou z nich zřejmé vlivy jednotlivých komponent sítě a využitých technik pro časovou řadu obsahující speciálně data z kryptoměnových finančních trhů. Zároveň tyto výsledky stojí za mnohými nápady na další možná vylepšení. Prvním z těchto nápadů souvisí s ověřením fungování tohoto typu modelu neuronové sítě na velmi podobné úloze k úloze formulované. V podstatě by se jednalo o simulaci problému pomocí průběhu nelineárních funkcí, konkrétně funkcí sinus. Vstupní datovou sadu by v tomto případě představovaly úseky funkcí sinus, přičemž jednotlivé funkce by se lišily svou frekvencí a amplitudou. Na základě této vstupní množiny dat by pak byly činěny předpovědi o směru a tvaru pokračování jednotlivých úseků.

Jistého zlepšení, by se dále možná dalo dosáhnout kombinací konvolučních vrstev s LSTM (angl. long-short term memory) vrstvami. Použití konvolučních vrstev společně s vrstvami LSTM totiž již v minulosti přineslo velmi dobré výsledky v řadě úloh založených na práci s časovými řadami obsahujícími data pocházející z finančních trhů [75].

Další možností, jak vylepšit přesnost předpovědí, je poskytnout síti více informací, na základě nichž budou tyto předpovědi činěny, nebo data nějakým nestandardním způsobem transformovat. S touto změnou však opět souvisí i modifikace původní 1D konvoluční neuronové sítě.

Poskytnutí síti více informací může v kontextu formulované úlohy znamenat např. přidat vstupní vektory obsahující nabídkové ceny (angl. Ask price), poptávkové ceny (angl. Bid price), maximální obchodované objemy, ceny Etherea vztažené k USDT nebo ceny jiných kryptoměn vztažené k Bitcoinu, a to vše v uplynulých 10 minutách. Takový formát vstupní datové sady by následně bylo možné zpracovat např. pomocí tzv. *vícehlavých 1D konvolučních neuronových sítí* (angl. multi-headed convolutional neural network) nebo *2D konvolučních neuronových sítí*. Vícehlavé 1D konvoluční neuronové sítě umožňují zpracovávat všechny vektory obsahující různé informace, avšak ze stejných 10 minut zároveň, a to pomocí totožných nebo odlišných architektur 1D konvolučních neuronových sítí. Výsledky všech subarchitektur jsou pak následně zkombinovány a na základě této kombinace je učiněna předpověď [76]. V případě využití 2D konvolučních neuronových sítí by bylo třeba vkládat jednotlivé vektory ve formátu matice, např. tak, že by každý sloupec matice obsahoval kromě samotné ceny Etherea vztažené k ceně



Bitcoinu i ceny ostatních dostupných kryptoměn vztažené k ceně Bitcoinu ve stejných 10 minutách. Datovou sadu obsahující ceny Etherea vztažené k cenám Bitcoinu by také bylo možné transformovat prostřednictvím tzv. HFD (angl. High Frequency Density) spektrogramu. Během této transformace je nejprve na základě zvolené délky a tvaru časového okna (např. Hanovo, Welchovo apod.) vypočtena diskrétní Fourierova transformace, čímž je získána informace o výskytu daných frekvencí v časové řadě v době, která odpovídá délce časového okna. Dále je zvolen celkový počet těchto oken pro danou časovou řadu. Každé takové okno pak po transformaci tvoří jeden sloupec matice. Na takto zpracovaná data lze následně aplikovat 2D konvoluční neuronové síť [77].

V případě, že by ani jedna z výše uvedených alternativ nepřinesla žádoucí výsledky, je možné modifikovat formulaci úlohy.

# Závěr

V rámci této bakalářské práce bylo zkombinováno několik oblastí, které jsou v současném světě technologií velmi aktuální. Ať už se jedná o samotný koncept kryptoměn jakožto možnou alternativu k historickým fiatovým měnám a obchodování s nimi, umělou inteligenci snažící se o vyšší schopnost generalizace, než které dosahuje inteligence lidská, či aplikaci konvolučních neuronových sítí konkrétně na regresní typy úloh.

Hlavní přínos této práce tak tkví, spíše než ve vývoji fungujícího systému či aplikace, v pečlivém a rozsáhlém výzkumu všech výše uvedených oblastí, nalezení a představení souvislostí mezi nimi a první zkoumání potenciálu umělé inteligence, resp. konvolučních neuronových sítí, v dané oblasti.

Pomocí experimentů učiněných v sekci 4.5 bylo ověřeno, že aplikace konvolučních neuronových sítí na řešení regresní typy úloh je smysluplná. Lze tedy říci, že pokud data obsahují jednoduché vzory v podobě lineárních závislostí, je možné je pomocí tohoto typu neuronových sítí odhalit a rozpoznat. Výsledky sekce 4.6.4 však ukazují, že ve snaze zobecnit konkrétně pojem technické analýzy na reálných datech z kryptoměnových trhů jsou konvoluční neuronové sítě zatím neúspěšné. Na otázku, jaká bude relativní změna ceny Etherea vztažená k ceně Bitcoinu v následujících 10 minutách na základě zadání cen z předchozích 10 minut nám síť sice dá nějakou odpověď, avšak tato odpověď bude v praxi nevyužitelná. Jak bylo konkrétně zjištěno, odpověď bude velmi často taková, že žádná změna nenastává, což ve většině případů není pravda. Existuje celá řada aspektů, které mohly nefunkčnost sítě výrazně ovlivnit, a proto tyto výsledky vedly k dalšímu zamyšlení. Data sice obsahují jisté vzory, ale závislosti se zde ani zdaleka vyskytovat nemusí. To vlastně znamená, že v rámci těchto dat nemusí existovat příčinné souvislosti mezi minulostí a budoucností na základě kterých by byla učiněna správná předpověď. Chyba tedy může být jak na straně formulované úlohy, tak i k ní nevhodně zvoleným datům. Minutové rozlišení dat dále nemusí být pro odhalení případných vzorů dostačující. Prostřednictvím těchto prvních výsledků proto zatím není možné definitivně rozhodnout, zda je neúspěch aplikace konvolučních neuronových sítí na tuto komplexní úlohu způsoben právě některým z výše uvedených, či mnohých dalších aspektů, nebo přímo nevhodným použitím toho konkrétního typu neuronových sítí. Otázky sekce 4.1 tak zůstávají i nadále otevřené, objevují se další a společně s nimi široké pole budoucí působnosti.

Pokud by se podařilo dospět k takovému modelu neuronové sítě, který by byl schopen předpovídat jakékoliv informace o budoucím vývoji kryptoměnového trhu, mohly by se jeho výsledky využít ve vhodné strategii automatického obchodování. Implementaci takové strategie by přitom bylo rozumné ověřit pomocí simulace obchodování na základě historických dat.

# Literatura

- [1] Summit Saha: *A Comprehensive Guide to Convolutional Neural Network*, 15.12.2018. URL: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53> [cit. 20.06.2021].
- [2] Alae Chouiekh, EL Hassane IbnEL Hajb: *ConvNets for Fraud Detection analysis*, Procedia Computer Science, vol. 127, 2018, pp.: 133-138.
- [3] Jake Frankenfield: *Cryptocurrency*, 25.5.2021. URL: <https://www.investopedia.com/terms/c/cryptocurrency.asp> [cit. 22.06.2021].
- [4] Luke Conway: *Blockchain Explained*, 1.6.2021. URL: <https://www.investopedia.com/terms/b/blockchain.asp> [cit. 22.06.2021].
- [5] Jake Frankenfield: *Bitcoin Mining*, 1.6.2021. URL: <https://www.investopedia.com/terms/b/bitcoin-mining.asp> [cit. 22.06.2021].
- [6] Alza.cz: *Proof of Work a Proof of Stake (VŠE, CO VÍME)*, 13.1.2020. URL: <https://www.alza.cz/proof-of-work-a-proof-of-stake> [cit. 22.06.2021].
- [7] Chia: *FAQ*, 20.5.2020. URL: <https://www.chia.net/faq/#faq-1> [cit. 28.06.2021].
- [8] Luke Conway: *What are the Safest Ways to Store Bitcoin?*, 1.5.2021. URL: <https://www.investopedia.com/news/bitcoin-safe-storage-cold-wallet/> [cit. 23.06.2021].
- [9] Jake Frankenfield: *Paper Wallet*, 1.4.2021. URL: <https://www.investopedia.com/terms/p/paper-wallet.asp> [cit. 23.06.2021].
- [10] Jake Frankenfield: *Hot Wallet*, 26.5.2021. URL: <https://www.investopedia.com/terms/h/hot-wallet.asp> [cit. 23.06.2021].
- [11] Nathan Reiff: *What Are the Advantages of Paying With Bitcoin*, 13.7.2020. URL: <https://www.investopedia.com/ask/answers/100314/what-are-advantages-paying-bitcoin.asp> [cit. 23.06.2021].
- [12] Nathan Reiff: *Could Cryptocurrencies Replace Cash?*, 25.6.2019. URL: <https://www.investopedia.com/news/could-cryptocurrencies-replace-cash-bitcoin-flipping/> [cit. 23.06.2021].
- [13] Binance Academy: *A Complete Guide to Cryptocurrency Trading for Beginners*, 2018. URL: <https://academy.binance.com/en/articles/a-complete-guide-to-cryptocurrency-trading-for-beginners#what-is-the-spot-market> [cit. 21.02.2021].

- [14] Adam Hayes: *Bull Market*, 24.1.2021. URL: <https://www.investopedia.com/terms/b/bullmarket.asp> [cit. 21.02.2021].
- [15] James Chen: *Bear Market*, 23.12.2020. URL: <https://www.investopedia.com/terms/b/bearmarket.asp> [cit. 21.02.2021].
- [16] Adam Hayes: *Liquidity*, 15.1.2021. URL: <https://www.investopedia.com/terms/l/liquidity.asp> [cit. 21.02.2021].
- [17] Adam Hayes: *Market Depth*, 13.2.2020. URL: <https://www.investopedia.com/terms/m/marketdepth.asp> [cit. 06.04.2021].
- [18] Justin Kuepper: *Liquidity*, 22.2.2021. URL: <https://www.investopedia.com/terms/v/volatility.asp> [cit. 24.02.2021].
- [19] Binance Academy: *Market Makers and Market Takers Explained*, 2018. URL: <https://academy.binance.com/en/articles/what-are-makers-and-takers> [cit. 21.02.2021].
- [20] INVESTOPEDIA: *Spot Market*, 20.2.2021. URL: <https://www.investopedia.com/terms/s/spotmarket.asp> [cit. 24.02.2021].
- [21] Pierce Moore: *Advanced Order Types and Functions*, 15.7.2020. URL: <https://honestcrypto.io/advanced-order-types-and-functions/> [cit. 24.02.2021].
- [22] Binance Academy: *Spot Market*, 24.5.2020. URL: <https://www.binance.com/en/blog/421499824684900602/Binance-Margin-Differences-Between-the-New-Isolated-Margin-Mode-and-Cross-Margin-Mode> [cit. 24.02.2021].
- [23] Jason Fernando: *Derivative*, 5.12.2020. URL: <https://www.investopedia.com/terms/d/derivative.asp#advantages-of-derivatives> [cit. 20.12.2020].
- [24] Finančník.cz: *Co jsou to komodity a jak to funguje*, 2.11.2004. URL: <https://www.financnik.cz/clanky/serialy/komoditni-manual/komodity-jak-to-funguje-r3/> [cit. 20.12.2020].
- [25] Lukáš Koťátko: *Obchodování Futures: Průvodce & odpovědi na 10 zásadních otázek*, 5.4.2016. URL: <https://www.lynxbroker.cz/vzdelavani/futures-obchodovani-pruvodce/> [cit. 20.12.2020].
- [26] Leslie Kramer: *Long Position vs. Short Position: What's Difference?*, 5.11.2020. URL: <https://www.investopedia.com/ask/answers/100314/whats-difference-between-long-and-short-position-market.asp> [cit. 21.12.2020].
- [27] INVESTOPEDIA: *The Difference Between Initial Margin vs. Maintenance Margin*, 14.12.2020. URL: <https://www.investopedia.com/ask/answers/033015/what-difference-between-initial-margin-and-maintenance-margin.asp> [cit. 21.12.2020].
- [28] Binance Academy: *What Are Perpetual Futures Contracts?*, 2019. URL: <https://academy.binance.com/en/articles/what-are-perpetual-futures-contracts> [cit. 22.12.2020].
- [29] Binance Academy: *Differences Between a Perpetual Contract and Traditional Futures Contract*, 9.9.2019. URL: <https://www.binance.com/en/support/faq/360033524991> [cit. 22.12.2020].

- [30] Binance Academy: *Price Index*, 29.10.2020. URL: <https://www.binance.com/en/support/faq/547ba48141474ab3bdc5d7898f97928> [cit. 25.12.2020].
- [31] Binance Academy: *Margin Trading Index Price Explained*, 15.12.2020. URL: <https://www.binance.com/en/support/faq/52b5970fc4e54994a7869d9b0db13754> [cit. 25.12.2020].
- [32] Binance Academy: *Mark Price in USDT-M futures*, 7.3.2020. URL: <https://www.binance.com/en/support/faq/360033525071> [cit. 25.12.2020].
- [33] Binance Academy: *A Guide to Trading Binance Quarterly Futures Contracts*, 2020. URL: <https://academy.binance.com/en/articles/a-guide-to-trading-binance-quarterly-futures-contract> [cit. 25.12.2020].
- [34] Binance Academy: *Mark Price in Coin-Margined Futures*, 6.11.2020. URL: <https://www.binance.com/en/support/faq/fff6ec1b3f5845d19f1fb924644a0877> [cit. 25.12.2020].
- [35] Binance Academy: *The Pros and Cons of COIN- & USDT-margined contracts: How do they impact your returns?*, 23.9.2020. URL: <https://www.binance.com/en/blog/421499824684901012/The-Pros-and-Cons-of-COIN--USDTmargined-contracts-How-do-they-impact-your-returns> [cit. 30.12.2020].
- [36] Binance Academy: *The Ultimate Guide to Trading on Binance Futures*, 2019. URL: <https://academy.binance.com/en/articles/the-ultimate-guide-to-trading-on-binance-futures> [cit. 30.12.2021].
- [37] Binance Academy: *Liquidation*, 7.5.2020. URL: <https://www.binance.com/en/support/faq/360033525271> [cit. 30.12.2020].
- [38] Binance Academy: *Understanding the Different Order Types*, 2020. URL: <https://academy.binance.com/en/articles/understanding-the-different-order-types> [cit. 15.02.2021].
- [39] Binance Academy: *Types of Order*, 6.10.2020. URL: <https://www.binance.com/en/support/faq/360033779452> [cit. 15.02.2021].
- [40] Aishwarya Singh: *Demystifying the Mathematics Behind Convolution Neural Networks (CNNs)*, 11.02.2020. URL: <https://www.analyticsvidhya.com/blog/2020/02/mathematics-behind-convolutional-neural-network/> [cit. 16.03.2021].
- [41] Ing. Milan Blaha, Ph.D.: *Matematická biologie: e-learningová učebnice* [online], Brno, 2015. ISBN: 9788021080959. URL: <https://portal.matematickabiologie.cz/index.php?pg=analiza-a-hodnoceni-biologickyh-dat--umela-inteligence> [cit. 13.03.2021].
- [42] Sagar Sharma: *Activation Functions in Neural Networks*, 06.09.2017. URL: <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6> [cit. 13.03.2021].
- [43] Mgr. Martin Pilát, Ph.D.: *Neuronové sítě - RBF síť a rekurentní síť* 18.4.2019. URL: <https://martinpilat.com/cs/prirodou-inspirovane-algoritmy/neuronove-site-rbf-site-rekurentni-site> [cit. 24.06.2021].
- [44] J. Feldman, Raul Rojas: *Neural Networks: A Systematic Introduction*, 1996. ISBN: 9783645610684. URL: <https://public.ebookcentral.proquest.com/choice/publicfullrecord.aspx?p=3093727> [cit. 14.03.2021].

- [45] Petr Kačenka: *Neuronové sítě*, Mariánská, vol. 98., URL: <https://prase.cz/library/NeuronoveSitePK/NeuronoveSitePK.pdf>.
- [46] Prince Grover: *5 Regression Loss Functions All Machine Learners Should Know*, 05.06.2018. URL: <https://heartbeat.fritz.ai/5-regression-loss-functions-all-machine-learner-s-should-know-4fb140e9d4b0> [cit. 16.03.2021].
- [47] Deep Learning Demystified: *Loss Functions Explained*, 2020. URL: <https://deeplearningdemystified.com/article/fdl-3> [cit. 16.03.2021].
- [48] Jason Brownlee: *How To Choose Loss Functions When Training Deep Learning Neural Networks*, 25.8.2020. URL: <https://machinelearningmastery.com/how-to-choose-loss-function-s-when-training-deep-learning-neural-networks/> [cit. 26.06.2021].
- [49] IBM Cloud Education: *Artificial Intelligence (AI)*, 3.6.2020. URL: <https://www.ibm.com/cloud/learn/what-is-artificial-intelligence> [cit. 24.06.2021].
- [50] Shai Shalev-Schwartz and Shai Ben-David: *Understanding machine learning from theory to algorithms*. Cambridge University Press, 2014. ISBN: 9781107057135.
- [51] IBM Cloud Education: *Machine Learning*, 15.7.2020. URL: <https://www.ibm.com/cloud/learn/machine-learning> [cit. 24.06.2021].
- [52] C. C. Aggarwal: *Neural Networks and Deep Learning*. Springer International Publishing AG, 2018. ISBN: 9783319944630.
- [53] Jiwon Jeong: *The Most Intuitive and Easiest Guide for Convolution Neural Network*, 24.01.2019. URL: <https://towardsdatascience.com/the-most-intuitive-and-easiest-guide-for-convolutional-neural-network-3607be47480> [cit. 17.03.2021].
- [54] Arc: *Convolutional Neural Network*, 25.12.2018. URL: <https://towardsdatascience.com/convolutional-neural-network-17fb77e76c05> [cit. 17.03.2021].
- [55] Piotr Skalski: *Gentle Dive into Math Behind Convolutional Neural Network*, 04.12.2019. URL: <https://towardsdatascience.com/gentle-dive-into-math-behind-convolutional-neural-networks-79a07dd44cf9> [cit. 17.03.2021].
- [56] Eugene Fama: *Efficient Capital Markets: A Review of Theory and Empirical Work*. The Journal of Finance, vol.2, 1970. pp: 383-417.
- [57] Jack D. Schwager: *Market Wizards: Interview with top traders*. HarperBus, 1990. ISBN: 9780887306105.
- [58] James Chen: *High Frequency Trading*, 14.5.2021. URL: <https://www.investopedia.com/terms/h/high-frequency-trading.asp> [cit. 27.06.2021].
- [59] Shobhit Seth: *Debunking 8 myths about technical analysis*, 8.5.2020. URL: <https://www.investopedia.com/articles/active-trading/062215/debunking-8-myths-about-technical-analysis.asp> [cit. 27.06.2021].
- [60] Theresa W. Carey: *Top Technical Analysis Tools for Traders*, 14.5.2021. URL: <https://www.investopedia.com/articles/active-trading/121014/best-technical-analysis-trading-software.asp> [cit. 27.06.2021].

- [61] Python Documentation: *The Python Tutorial*, 17.04.2021. URL: <https://docs.python.org/3/tutorial/index.html> [cit. 18.04.2021].
- [62] Wikipedia, the free encyclopedia: *Anaconda*, 7.04.2021. URL: [https://en.wikipedia.org/wiki/Anaconda\\_\(Python\\_distribution\)](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)) [cit. 18.04.2021].
- [63] Pavel Strachota: *Helios cluster documentation*. P. Strachota, 16.7.2020.
- [64] Lou MendelSohn: *Preprocessing data for Neural Networks*, 12.12.2020. URL: [https://www.vantagepointsoftware.com/mendelsohn/preprocessing-data-neural-networks/?fbclid=IwAR2IYm7QBd3bIFh6uunlt0aIQ0AxrjKmn6lVbBsa04erp7Byi8aR\\_-C8zDY](https://www.vantagepointsoftware.com/mendelsohn/preprocessing-data-neural-networks/?fbclid=IwAR2IYm7QBd3bIFh6uunlt0aIQ0AxrjKmn6lVbBsa04erp7Byi8aR_-C8zDY) [cit. 19.04.2021].
- [65] Jason Brownlee: *How to use data Scaling Improve Deep Learning Model Stability and Performance*, 4.2.2019. URL: <https://machinelearningmastery.com/how-to-improve-neural-network-stability-and-modeling-performance-with-data-scaling/> [cit. 20.04.2021].
- [66] Jason Brownlee: *How to Normalize and Standardize Time Series Data in Python*, 4.2.2019. URL: <https://machinelearningmastery.com/normalize-standardize-time-series-data-python/> [cit. 20.04.2021].
- [67] Keras: *Metrics*. URL: <https://keras.io/api/metrics/> [cit. 20.04.2021].
- [68] Jason Brownlee: *How to Configure The Number of Layers and Nodes in a Neural Network*, 6.8.2019. URL: <https://machinelearningmastery.com/how-to-configure-the-number-of-layers-and-nodes-in-a-neural-network/> [cit. 20.04.2021].
- [69] GeeksforGeeks: *Choose optimal number of epochs to train a neural networks in Keras*, 8.6.2020. URL: <https://www.geeksforgeeks.org/choose-optimal-number-of-epochs-to-train-a-neural-network-in-keras/> [cit. 21.04.2021].
- [70] Jason Brownlee: *A Gentle Introduction to Early Stopping to Avoid Overtraining Neural Networks Models*, 6.8.2019. URL: <https://machinelearningmastery.com/early-stopping-to-avoid-overtraining-neural-network-models/> [cit. 21.04.2021].
- [71] Jason Brownlee: *A Gentle Introduction to Batch Normalization for Deep Neural Networks*, 4.12.2019. URL: <https://machinelearningmastery.com/batch-normalization-for-training-of-deep-neural-networks/> [cit. 21.04.2021].
- [72] Jason Brownlee: *How to Accelerate Learning of Deep Neural Networks with Batch Normalization*, 25.8.2020. URL: <https://machinelearningmastery.com/how-to-accelerate-learning-of-deep-neural-networks-with-batch-normalization/> [cit. 21.04.2021].
- [73] Jason Brownlee: *A Gentle Introduction to Dropout for Regularizing Deep Neural Networks*, 6.8.2019. URL: <https://machinelearningmastery.com/how-to-accelerate-learning-of-deep-neural-networks-with-batch-normalization/> [cit. 25.04.2021].
- [74] Jason Brownlee: *How to Reduce Overfitting with Dropout Regularization in Keras*, 25.8.2020. URL: <https://machinelearningmastery.com/how-to-reduce-overfitting-with-dropout-regularization-in-keras/> [cit. 25.04.2021].

- [75] Wenjie Lu, Jiazheng Li, Yifan Li, Aijun Sun, Jingyang Wang: *A CNN-LSTM-Based Model to Forecast Stock Prices*, Complexity, vol. 2020, 2020, pp: 10.
- [76] Jason Brownlee: *How to Develop Convolutional Neural Network for Time Series Forecasting*, 28.8.2020. URL: <https://machinelearningmastery.com/how-to-develop-convolutional-neural-network-models-for-time-series-forecasting/> [cit. 30.04.2021].
- [77] Martin Kovanda *Rozpoznávání ultrazvukových signálů pomocí konvolučních neuronových sítí*, bakalářská práce, FJFI, 7.7.2020.



# Přílohy

## Příloha 1: Implementace procesu zpracování

```
1 import datareader as reader
2 from datetime import datetime as dt
3 import numpy as np
4 import pandas as pd
5 import csv
6 import copy
7 from loadCachedData import loader
8 from numpy import array
9 from sklearn import preprocessing
10
11 series = 10
12
13 # tvorba vstupnich vektoru
14 def split_sequence(sequence, x_series):
15     dic = {}
16     for key in sequence.columns:
17         x = list()
18         for i in range(len(sequence[key])):
19             end_ix = i + x_series
20             if end_ix > len(sequence)-1:
21                 break
22             seq_x = pd.DataFrame({key: sequence[key][i:end_ix]})
23             x.append(seq_x)
24         dic[key] = x
25     return pd.DataFrame(dic)
26
27 # tvorba skutecnych vystupnich hodnot
28 def y_value(arr, price, x_series):
29     a, b = list(), list()
30     for i in arr:
31         a.append(i.values[0:x_series])
32     for i in range(0, len(price)):
33         first_ix = i + x_series - 1
34         second_ix = first_ix + series
35         if second_ix > len(price)-1:
36             break
37         b.append((price[second_ix] - price[first_ix])/price[first_ix])
38     return np.squeeze(np.array(a)), np.squeeze(np.array(b))
39
40 # normalizace vstupnich dat
41 def normalize(df):
42     names = df.columns
43     x = df.values
```

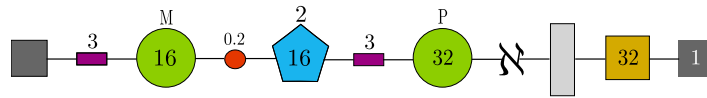
```

44 min_max_scaler = preprocessing.MinMaxScaler()
45 normalized_df = pd.DataFrame(min_max_scaler.fit_transform(x), columns=names)
46 return normalized_df
47
48 # spojeni jednotlivych casti za vzniku zpracovane vstupni a vystupni mnoziny
49 def proces(number_of_element, x_series):
50     ticker_btc = loader()
51     test_btc = {}
52
53     for key in ["ETHBTC"]:
54         test_btc[key] = ticker_btc.price[key]
55
56     df_btc = pd.DataFrame(test_btc)
57     z_btc = split_sequence(df_btc, x_series)
58     preprocessed_btc = {}
59     for key in df_btc.columns:
60         preproc_btc = list()
61         dic_btc = {}
62         for i in z_btc[key]:
63             preproc_btc.append(normalize(i))
64         x, y = y_value(preproc_btc, np.array(df_btc[key]), x_series)
65         for n in range(0, x_series):
66             dic_btc["x_" + str(n)] = x[:,n][0:len(y)]
67         dic_btc["y_0"] = y
68         preprocessed_btc[key] = pd.DataFrame(dic_btc)
69
70     return preprocessed_btc
71
72 # ulozeni zpracovanych dat
73 def save(pole1, pole2, name):
74     np.savetxt(name + ".csv", zip(pole1, pole2), delimiter=",")

```

## Příloha 2: Implementace modelu 1D konvoluční neuronové sítě

Následující definice modelu se týká 1D konvoluční neuronové sítě, které zachycuje obrázek 4.26



Obrázek 4.26: Implementované schéma 1D konvoluční sítě

```
1 import tensorflow as tf
2 import numpy as np
3 import load_price
4 from numpy import array
5 from loadCachedData import loader
6 from tensorflow import keras
7 from keras.models import Sequential
8 from keras.layers import Activation, Conv1D, MaxPooling1D, Flatten, Dense, Dropout,
   BatchNormalization
9 from keras import backend as K
10 from keras.constraints import max_norm
11
12 # pocet prvku v jednom vektoru tvoricim vstupni mnozinu
13 x_series = 10
14
15 # nacteni dat
16 data_btc = load_price.load("preprocessed_5_10-price", x_series)
17
18 # presun dat z formatu DataFrame do formatu Numpy
19 a, b = list(), list()
20 for key in data_btc["ETHBTC"].columns:
21     if key[0]=="x":
22         a.append(data_btc["ETHBTC"][key])
23     if key[0]=="y":
24         b.append(data_btc["ETHBTC"][key])
25
26 x_train, y_train = np.array(a), np. array(b)
27
28 n_features = 1
29 x_train=x_train.reshape(x_train.shape[1], x_train.shape[0], n_features)
30 y_train=y_train.reshape(y_train.shape[1], y_train.shape[0])
31
32 # rozdeleni dat trenovaci a validacni mnozinu
33 x_val = x_train[-10000:]
34 y_val = y_train[-10000:]
35 x_train = x_train[:-10000]
36 y_train = y_train[:-10000]
37
38 # inicializace ztratove funkce
39 h = tf.keras.losses.Huber()
40 # nastaveni poctu batchu
41 batch_size = 50000
42 # nastaveni poctu epoch
43 epochs = 30
44
45 # metrika pro vyhodnoceni uspesnosti
```

```

46 def relative_error(y_true, y_pred):
47     return K.abs((y_pred - y_true) / (K.abs(y_true)+K.epsilon()))
48
49
50 model = Sequential()
51 model.add(Conv1D(16, kernel_size=3, kernel_constraint=max_norm(2.), input_shape=(
    x_train.shape[1], x_train.shape[2])))
52 model.add(Dropout(0.2))
53 model.add(Activation("relu"))
54 model.add(MaxPooling1D(pool_size=2))
55 model.add(Conv1D(32, kernel_size=3, activation="relu", padding="same", input_shape=(
    x_train.shape[1], x_train.shape[2])))
56 model.add(BatchNormalization())
57 model.add(Flatten())
58 model.add(Dense(32, activation = "relu"))
59 model.add(Dense(1))
60
61 model.compile(optimizer= 'adam', loss=h, metrics=relative_error)
62 model.fit(x_train, y_train, epochs=epochs, batch_size=batch_size, validation_data=(
    x_val, y_val))

```