



Vaasan yliopisto
UNIVERSITY OF VAASA

Kristian Lahtinen

Aikasarja-aineiston ennustaminen neuroverkolla

Tarkastelussa OMXH25-indeksi

Laskentatoimen ja rahoituksen yksikkö
Taloustieteen Pro Gradu -tutkielma
Taloustieteen koulutusohjelma

Vaasa 2023

VAASAN YLIOPISTO**Laskentatoimen ja rahoituksen yksikkö**

Tekijä:	Kristian Lahtinen	
Tutkielman nimi:	Aikasarja-aineiston ennustaminen neuroverkolla : Tarkastelussa OMXH25-indeksi	
Tutkinto:	Kauppätieteiden maisteri	
Oppiaine:	Taloustiede	
Työn ohjaaja:	Petri Kuosmanen, Mikko Ranta	
Valmistumisvuosi:	2023	Sivumäärä: 73

TIIVISTELMÄ:

Pitkään vallinneet teoriat tehokkaista markkinoista ja niiden satunnaiskulusta ovat saaneet vuosien aikana myös kilpailevia teorioita eri tutkimuksiin pohjautuen. Hypoteesit esittävät, että markkinoilla esiintyy epätehokkuuksia ja momentumia. Tällaisia teorioita ovat esimerkiksi tyyli-tellyt faktat sekä aikasarjan momentum -teoria. Niiden mukaan markkinoiden omilla liikkeillä on yhteys tulevaan ja liike markkinoilla ei ehkä ole niin sattumanvaraista. Tällaiset teoriat avaavat mielenkiintoisen mahdollisuuden tutkia rahoitusmarkkinoiden lainalaisuuksia erilaisin keinoin. Eräs tällainen menetelmä ovat neuroverkot, joilla on kyky analysoida aikasarja-aineistoja sekä epälineaarista dataa.

Tässä pro gradu -tutkielmassa tarkastellaan neuroverkkojen ennustekykyä Suomen markkinoilla, tarkemmin OMXH25-indeksissä, jossa tämänkaltaisia tutkimuksia ei ole juurikaan tehty. Tutkimuksessa haetaan vastauksia kolmeen tutkimuskysymykseen. Minkälainen ennustekyky neuroverkolla on eri aikasarjoilla OMXH25-indeksissä? Onko tutkimuksessa käytettävien neuroverkkomalleilla eroja ennustekyvyyssä? Voiko sijoittaja saada hyötyä neuroverkkojen käytöstä? Tutkielmassa käytetään neuroverkkona pitkäkestoista työmuistia. Tutkimus on myös rajattu siltä osin, että käytössä on kolme eri aikasarjaa ja neljä eri neuroverkkomallia, joiden hyperparametrit eroavat hieman toisistaan. Neuroverkkomallien ennustekykyä tullaan myös vertaamaan autoregressiiviseen malliin.

Teoriaosuudessa käsitellään tutkimuksen teoreettinen viitekehys, sisältäen sekä talousteoriat että neuroverkon teorian. Neuroverkoista käsitellään sen toimintaperiaate, rakenne sekä koulutus. Lisäksi tutkielma avaa tarkemmin käytettäviä neuroverkkoja sekä tarjoaa perustelut, miksi juuri kyseinen neuroverkko ja parametrit valikoituivat käyttöön testissä. Tutkielman empiirisessä osiossa käydään läpi aineiston kerääminen ja esikäsittely. Lisäksi empiirinen osio käsittää mallien rakentamisen sekä koulutuksen. Lopuksi esitellään ja analysoidaan tutkimuksesta saatavat tulokset sekä tehdään johtopäätökset tutkimuskysymysten perusteella.

Tulosten pohjalta voitiin sanoa, että neuroverkkojen ennustekyvyyllä oli eroja eri aikasarjojen välillä. Mitä pidempi aikasarja oli käytössä, sitä useampi neuroverkkomalli kykeni voittamaan vertailumallin. Tulokset myös osoittivat, ettei tutkimuksessa käytettävien neuroverkkomallien ennustekyvyyssä oli eroja erityisesti lyhyemmillä aikasarjoilla. Lisäksi tulosten pohjalta voitiin vetää johtopäätös siitä, ettei neuroverkkojen käytöstä ollut käytännön hyötyä sijoittajalle ennustekyvyn ollessa käytännössä liian heikko kaikissa aineistoissa. Neuroverkko kykeni kuitenkin tunnistamaan aineistoista trendejä, joka avaa mahdollisuuden tutkia neuroverkon käyttöä pidemmän aikavälin kaupassa.

AVAINSANAT: Neuroverkko, Pitkäkestoinen työmuisti, Aikasarjan momentum -teoria, tyyli-tellyt faktat, OMX Helsinki 25, Aikasarja-analyysi

UNIVERSITY OF VAASA**School of Accounting and Finance**

Author: Kristian Lahtinen
Title of the thesis: Aikasarja-aineiston ennustaminen neuroverkolla :
Tarkastelussa OMXH25-indeksi
Degree: Master of Science in Economics and Business Administration
Programme: Master's Degree Programme in Economics
Supervisor: Petri Kuosmanen, Mikko Ranta
Year: 2023 **Pages:** 73

ABSTRACT:

Over the years, the long-held theories about efficient markets and random walk have also received competing theories about inefficiencies and momentum in the market. Such theories are, for instance, stylized facts and the time series momentum theory. According to these theories, the historical movements of a stock or index are believed to be connected to future movements, challenging the idea that movement in the market may not be so random. That opens up an exciting opportunity to study the laws of financial markets in different ways. One method is neural networks, which have the ability to analyze time series data as well as non-linear data.

This master's thesis examines the predictive ability of neural networks in the Finnish stock markets, specifically in the OMXH25 index. The research seeks answers to three research questions. What kind of predictive ability does the neural network have for different time series in the OMXH25 index? Do the neural network models used in the study have differences in prediction ability? Can an investor benefit from the use of neural networks? In this thesis, we are going to use a long short term memory. The research is also limited to three different time series and four different neural network models with slightly different hyperparameters. The predictive ability of neural network models will also be compared to an autoregressive model.

The theory part goes through the thesis's theoretical background, including both economic and neural network theories. The study of neural networks pertains to their operational principles, structural characteristics, and training methodologies. Furthermore, this thesis elaborates on the neural networks that can be utilized in greater depth and offers justifications for selecting a particular neural network and its corresponding parameters for employment in the experiment. The empirical section of this thesis entails a review of the data collection and pre-processing procedures. Moreover, it involves constructing models and neural network training processes. Subsequently, the research findings are presented, analyzed, and conclusions are made based on the research questions.

Based on the results, there were differences in the predictive abilities of neural networks between different time series. The longer the time series used, the more neural network models were able to outperform the comparison model. The results also showed that there were differences in the predictive abilities of the neural network models used in the study, especially with shorter time series. In addition, based on the results, there was no practical benefit for investors in using neural networks due to their weak predictive ability in all datasets. However, neural networks were able to identify trends in the data, which opens up the possibility of studying the use of neural networks in longer-term trading.

KEYWORDS: Neural Networks, Long Short Term Memory, Time Series Momentum, Stylized Facts, OMX Helsinki 25, Time series analysis

Sisällys

1	Johdanto	7
2	Osakemarkkinoiden ja neuroverkon teoria	9
2.1	Tehokkaiden markkinoiden hypoteesi	9
2.1.1	Tyylitellyt faktat	12
2.1.2	Aikasarjan momentum-teoria	13
2.2	Neuroverkkojen elementit ja rakenne	15
2.2.1	Neuronin toiminta	17
2.2.2	Aktivaatiofunktio	18
2.2.3	Neuroverkon koulutus	22
2.2.4	Kustannusfunktio	23
2.2.5	Gradienttimenetelmä	24
2.2.6	Vastavirta-algoritmi	26
2.2.7	Stokastinen gradienttimenetelmä ja optimointialgoritmit	27
2.2.8	Neuroverkon hyperparametrit sekä ali- ja ylisovittaminen	28
2.2.9	Monikerroksiset eteenpäin syöttävät neuroverkot	29
2.2.10	Takaisinkytketyt neuroverkot	30
2.2.11	Pitkäkestoinen työmuisti	32
2.3	Aikaisemmat tutkimukset	34
3	Empiirinen osuus	42
3.1	Käytettävä aineisto	42
3.2	Tutkimusmenetelmät	48
3.3	Tulokset ja analysointi	51
3.3.1	Analyysi	57
4	Johtopäätökset	60
	Lähteet	61
	Liitteet	67
	Liite 1. Neuroverkon koodi	67

Kuvat

Kuva 1. Eteenpäin syöttävän neuroverkko	16
Kuva 2. Neuronin k toiminta	17
Kuva 3. Porrasfunktio	19
Kuva 4. Sigmoid-funktio	20
Kuva 5. Hyperbolinen tangenttifunktio	20
Kuva 6. ReLU-aktivaatiefunktio	21
Kuva 7. Kustannusfunktion minimointi	24
Kuva 8. Auton hinnan määrittely	30
Kuva 9. Esimerkki RNN rakenteesta	31
Kuva 10. LSTM rakenne	33
Kuva 11. Logaritmiset tuotot Aikasarja 1	44
Kuva 12. Päivän päätöshinta Aikasarja 1	44
Kuva 13. Logaritmiset tuotot Aikasarja 2	45
Kuva 14. Päivän päätöshinta Aikasarja 2	45
Kuva 15. Logaritmiset tuotot Aikasarja 3	46
Kuva 16. Päivän päätöshinta Aikasarja 3	46
Kuva 17. Neuroverkon häviö koulutuksen aikana	51
Kuva 18. AR-malli ennusteet Aikasarjalla 1	53
Kuva 19. Neuroverkkojen ennusteet Aikasarjalla 1	53
Kuva 20. AR-mallin ennusteet Aikasarjalla 2	54
Kuva 21. Neuroverkkojen ennusteet Aikasarjalla 2	55
Kuva 22. AR-mallin ennusteet Aikasarjalla 3	56
Kuva 23. Neuroverkkojen ennusteet Aikasarjalla 3	56

Taulukot

Taulukko 1. Tulokset aikaisemmista tutkimuksista	41
Taulukko 2. Tutkimuksessa käytettävät herätteet ja lähtöarvot	43
Taulukko 3. Aineiston jako koulutus- ja testausaineistoon aikasarjoittain	47
Taulukko 4. Neuroverkkojen hyperparametrit	48

Taulukko 5. Aikasarjan 1: 1.1.2014-31.12.2019 tulokset	52
Taulukko 6. Aikasarjan 2: 1.1.2016-31.12.2017 tulokset	54
Taulukko 7. Aikasarjan 3: 1.1.2020-30.9.2022 tulokset	55
Taulukko 8. Tutkimuksen tulokset kootusti aikasarjoittain	58

1 Johdanto

Markkinoiden hinnan muodostuminen sekä sen liikkeiden ennustaminen on läpi historian ollut suosittu aihe rahoitusmarkkinoiden tutkimuksessa. Esille ovat nousseet teorian markkinoiden tehokkuudesta ja satunnaiskulusta sekä sijoittajan mahdottomuudesta voittaa markkinat pitkällä aikavälillä. Toisaalta nämä teorian ovat saaneet osakseen kritiikkiä ja vastakkaisia hypoteeseja markkinoiden epätehokkuuksista, ja osa tutkimuksista on tunnistanut markkinoilta myös erilaisia säännönmukaisuuksia. Eräs tällainen esimerkki on vuonna 2012 Moskowitzin ja muiden esiin nostama teoria aikasarjan momentumista. Tutkijoiden mukaan sijoitusinstrumenttien omilla historiallisilla liikkeillä on yhteys tulevaan. Tällaiset teorian avaavat mielenkiintoisen mahdollisuuden tutkia rahoitusmarkkinoiden lainalaisuuksia erilaisin keinoin.

Uusia poikkeuksellisia keinoja löytyy jatkuvasti enemmän esimerkiksi koneoppimisen maailmasta, jossa menetelmät ja laskentateho kehittyvät jatkuvasti. Eräs tällainen koneoppimisen osa-alue on syväoppiminen ja siihen vahvasti linkittyvät neuroverkot. Neuroverkot ovat informaation käsittelyn malleja, jotka rakenne pohjautuu luonnollisiin hermoverkkoihin (Elements of AI, 2020). Neuroverkoilla on tehokas kyky prosessoida dataa, kuten aikasarja-aineistoja. Rakenteensa ansiosta ne kykenevät myös analysoimaan epälineaarista dataa eivätkä vaadi niiden analysoinnissa samanlaisia oletuksia kuin perinteisemmät tilastolliset mallit (Puolitaival & Kananen, 2019, s. 127–128). Näin ollen ne sopivatkin hyvin markkinoiden aikasarjojen analysoinnin tutkimukseen sekä aikasarjojen liikkeiden ennustamiseen. Neuroverkot ovatkin potentiaalinsa ansiosta olleet monien tutkimusten kohteena. Tutkimukset ovat kuitenkin pääosin keskistyneet Yhdysvaltojen markkinoille tai muihin suuriin kansainvälisiin indekseihin ja esimerkiksi Suomen osakemarkkinoilla tämänkaltaista tutkimusta ei ole juurikaan tehty.

Tässä tutkielmassa tullaankin keskittymään neuroverkkojen käyttöön Suomen osakemarkkinoilla ja tarkoituksena on tarkastella neuroverkon kykyä OMXH25-indeksin liikkeiden ennustamisessa eri aikasarjoilla. Näiden ennusteiden tekemisen pohjaksi on

tarkoitus hyödyntää indeksin omaa historiallista liikettä aikasarjan momentum -teorian mukaisesti. Tutkielmassa tullaan hakemaan vastauksia seuraaviin tutkimuskysymyksiin:

- *Minkälainen ennustekyky neuroverkolla on eri aikasarjoilla OMXH25-indeksissä?*
- *Onko tutkimuksessa käytettävien neuroverkkomalleilla eroja ennustekyvyyssä?*
- *Voiko sijoittaja saada hyötyä neuroverkkojen käytöstä?*

Ensimmäisen kysymyksen, eli millainen neuroverkon ennustekyvyn tarkkuus on, vastaukseen käytetään vertailuna autoregressiivista mallia. Neuroverkon ennustekykyä verrataan siis perinteisempään menetelmään. Lisäksi tutkielma tullaan rajaamaan siten, että käytössä on kolme eri aikasarjaa OMXH25-indeksistä. Ensimmäinen aikasarjoista edustaa pidempää aikaväliä ajanjaksolta 1.1.2014-31.12.2019. Toinen aikasarja edustaa lyhyttä aikaväliä ajanjaksolta 1.1.2016-31.12.2017. Kolmas aikasarja ajanjaksolta 1.1.2020-30.9.2022 edustaa hyvin poikkeuksellista aikaa markkinoilla, joten on mielenkiintoista selvittää, kuinka neuroverkkojen ennustekyky toimii näin poikkeuksellisella ajanjaksolla. Tutkielma tullaan rajaamaan myös siten, että käytössä on neuroverkkomenetelmistä vain pitkäkestoinen työmuisti. Lisäksi tutkielmassa käytetään neljää eri neuroverkkomallia, jotka eroavat toisiltaan vain muutamalla eri hyperparametrillä, piilokerrosten määrällä sekä koulutuksen kestolla.

Tutkielma etenee seuraavasti. Luvussa 2 esitellään tutkielman teoreettinen viitekehys ja siinä avataan tärkeimmät talousteoriat tutkimuksen aiheeseen liittyen. Luvussa esitellään myös neuroverkkojen yleistä teoriaa keskittyen neuroverkkojen rakenteisiin, toimintaperiaatteisiin sekä kouluttamiseen liittyviin asioihin, jotta lukija saa käsityksen tässä tutkimuksessa käytettyjen mallien toimintaperiaatteista. Lisäksi luvussa esitellään aikaisempia tutkimuksia neuroverkkojen ennustekyvyyden liittyen ja tehdään tämän pohjalta johtopäätöksiä tutkimukseen. Luvussa 3 käsitellään tutkimuksen empiirinen osuus. Luvussa käydään läpi aineisto ja sen käsittely, tutkimuksessa käytettyjen mallien rakentaminen sekä kouluttaminen. Lopuksi analysoidaan malleista saatavat tulokset ja tehdään johtopäätökset tutkimuskysymyksiin pohjautuen.

2 Osakemarkkinoiden ja neuroverkon teoria

2.1 Tehokkaiden markkinoiden hypoteesi

Vuonna 1970 julkaisemassaan tutkimuksessa *Efficient Capital Markets: A Review of Theory and Empirical Work*, Eugene Fama kokosi yhteen siihen asti tehtyjen tutkimuksien tuloksia markkinoiden tehokkuudesta, joka johti lopulta määritelmään *tehokkaiden markkinoiden hypoteesista*. Tehokkaiden markkinoiden hypoteesilla tarkoitetaan markkinoilla vallitsevaa tilaa, jossa osakkeisiin kohdistuva informaatio vaikuttaa niiden hintoihin välittömästi. Teorian mukaan informaatio ei siis niin sanotusti ”valu” hintoihin (Knüpfer & Puttonen, 2018 s. 167). Käytännössä tehokkaiden markkinoiden hypoteesin mukaan sijoittajan ei tulisi saada markkinoilla riskikorjattua tuottoa parempaa tuottoa, sillä myös muut sijoittajat pystyvät hyödyntämään samaa informaatiota välittömästi osakkeiden hinnoittelussa (Knüpfer & Puttonen, 2018 s. 172).

Fama (1970) asetti kuitenkin tehokkaiden markkinoiden hypoteesille ehtoja, joiden tulisi markkinoilla täyttyä. Ehtojen mukaan tehokkailla markkinoilla ei tulisi olla transaktiokustannuksia. Tämän lisäksi kaikki informaatio tulisi olla jokaiselle sijoittajalle ilmaiseksi saatavilla. Ehtona tehokkaille markkinoille oli myös sijoittajien kyky käsitellä rationaalisesti sekä yksimielisesti markkinoille tulevaa informaatiota. Tehokkaiden markkinoiden määritelmä ei myöskään ole niin yksiselitteinen, vaan tehokkuuden asteita on useita. Tehokkuuden aste riippuu siitä, mitä kaikkea informaatiota osakkeiden hinnat jatkuvasti heijastelevat. Markkinat voivat teorian mukaan jakaantua heikosti tehokkaisiin markkinoihin, puolivahvasti tehokkaisiin markkinoihin sekä vahvasti tehokkaisiin markkinoihin (Knüpfer & Puttonen, 2018 s. 172).

Knüpfer & Puttosen (2018, s. 171) mukaan heikosti tehokkailla markkinoilla osakkeiden hinnat heijastelevat kaikkia historiallisia tietoja ja teorian mukaan teknisellä analyysillä ei ole merkittävää hyötyä tämän kaltaisella markkinalla. Aikaisempien tuottojen tutkiminen on turhaa, koska sama tieto on kaikkien saatavilla. Puolivahvasti tehokkailla markkinoilla osakkeiden hinnat heijastelevat edellisen informaation lisäksi myös kaikkea julkista

tietoa, kuten yritysten tulosjulkistuksia. Tämän perusteella fundamenttianalyysin käytöstä ei ole hyötyä sijoittajalle. Vahvasti tehokkailla markkinoilla osakkeiden hinnat heijastelevat kaikkea mahdollista tietoa. Edellä mainittujen tietojen lisäksi tämä pitää sisälleen sisäpiiritiedot. Hallituksen kokouksista lähtien kaikkien päätösten tulisi näkyä heti hinnoissa. Tällaisilla markkinoilla ei teknisen analyysin eikä fundamenttianalyysin lisäksi ole mitään hyötyä sisäpiiritiedoista. Tehokkailla markkinoilla ei siis millään sijoitustyyllillä tai strategialla tulisi saada markkinoita korkeampaa riskikorjattua tuottoa.

Tehokkaista markkinoista puhuttaessa nousee esille myös teoria niin sanotusta satunnaiskulusta (random walk theory). Teorian mukaan menneillä hinnoilla ei tulisi olla korrelaatiota tulevien hintojen kanssa. Tämä riippumattomuus johtuu markkinoille koko ajan tulevasta uudesta ennustamattomasta informaatiosta, joka muokkaa osakkeiden hintoja jatkuvasti (Knüpfer ja Puttonen, 2018, s. 169–170).

Tehokkaiden markkinoiden hypoteesi pohjautuu myös tiettyihin talousteoreettisiin oletuksiin, joita esimerkiksi Shleifer (2000, s. 2–5) on kirjassaan avannut. Ensimmäisenä oletuksena pidetään sitä, että suurin osa sijoittajista toimii rationaalisesti tai lähes rationaalisesti. Toisena oletuksena teoriassa pidetään sitä, että epärationaalisten sijoittajien tekemät transaktiot ovat satunnaisia ja näin ollen niiden oletetaan jakautuvan siten, että näiden epärationaalisten päätösten vaikutukset kumoavat toisensa. Samanlaisista syistä myös markkinoilla esiintyvien arbitraasien oletetaan kumoutuvan. Jokaista epärationaalista sijoittajaa kohtaan löytyy markkinoilta rationaalisesti toimiva sijoittaja.

Tälle näkemykselle voi hakea tukea myös Kollerin ja muiden (2020, s. 671–673) julkaisusta. Kirjassaan he jakoivat sijoittajat neljään eri luokkaan sijoitustyylien mukaan. Ensimmäisenä ryhmänä olivat arvosijoittajat (intrinsic investors), joiden sijoitustoiminta perustui yrityksen sisäiseen arvoon liittyvään analyysiin. Kirjan mukaan he myös käyttivät enemmän aikaa fundamenttien tutkimukseen. Tämä sijoitusryhmä kävi kauppaa myös verrattain harvoin. Toisena ryhmänä olivat treidaajat (traders). Heidän sijoitustyylinensä perustui nopeaan kaupankäyntiin ja teknisen analyysin käyttämiseen. Kolmantena

sijoitusryhmänä olivat mekaaniset sijoittajat (mechanical investors), joiden kaupankäynti perustui ennalta määritettyihin tiukkoihin kriteereihin tai sääntöihin, esimerkiksi yrityksen arvostaminen vain matemaattisin mallein, ilman syvällisempää fundamenttianalyysiä. Viimeisenä ryhmänä olivat kaappi-indeksoijat (closet indexers), joiden tarkoituksena oli pääosin kopioida jokin indeksi ja häivyttää se aktiivisen kaupankäynnin taakse.

Kollerin ja muiden (2020, s. 674–675) mukaan arvosijoittajat olivat merkittävin pitkän aikavälin hintoihin vaikuttava sijoittajaryhmä suurista ostoista ja myynneistä johtuen. Näin ollen suurin hintoihin vaikuttava ryhmä myös teki harkittuja kaupankäyntejä ja epärationaalisesti käyttäytyvien sijoittajien määrä ei siis vaikuttaisi suuresti osakkeiden hintoihin. Tämä irrationaalisesti sijoittavien joukko avaa kuitenkin mahdollisuuden esimerkiksi treidaajille, jotka voivat hyödyntää näitä väärin hinnoiteltuja paikkoja.

Vaikka tehokkaiden markkinoiden hypoteesia on pidetty pohjana ja teoreettisena lähestymistapana osakemarkkinoiden hinnoittelulle ja informaation välittymiselle osakkeiden hintoihin, on se saanut osakseen myös suurta kritiikkiä. Merkittävä kritiikki teoriaa kohtaan nousee esiin jo siihen liittyvien taustaoletusten vuoksi. Todellisuudessa esimerkiksi veroilla, kaupankäyntikustannuksilla ja informaation keräämiseen liittyvillä kustannuksilla on vaikutuksia ja ne voivat johtaa tietynasteiseen tehottomuuteen markkinoilla. Esimerkiksi, jos informaation katsotaan vaikuttavan osakkeen hintoihin kaupankäyntikuluja vähemmän, ei todellisuudessa tieto vaikuta välittömästi osakkeen hintoihin. Esimerkiksi vuonna 1978 Jensen nosti esille, että sijoittajien tulee huomioida päätöksissään riskikorjatun tuoton lisäksi myös muut kustannukset.

Samanlaisiin johtopäätöksiin ovat päätyneet myös Grossman (1976) ja Grossman & Stiglitz (1980). Heidän mukaansa tehokkaat markkinat ovat mahdottomuus, sillä tehokkailla markkinoilla informaation keräämisestä ei saisi minkäänlaista etua ja näin ollen kannustimia kaupankäyntiin ei markkinoilla olisi. Tämä johtaisi taas lopulta markkinoiden romahtamiseen. Heidän mukaansa markkinoilla tulisi esiintyä aina sen verran tehottomuutta, että kaupankäyntiin sekä informaation keräämiseen liittyvät kustannukset

voidaan kuitata ”ylituottolla”. Fama (1991) on myös itse myöhemmin tarkentanut teoriaa ja sanookin tehokkaiden markkinoiden olevan eräänlainen äärimmäinen ehto ja nollahypoteesi, jonka pohjalta markkinaa voidaan analysoida.

2.1.1 Tyylitellyt faktat

Tehokkaat markkinat ovat saaneet osakseen kritiikkiä myös empiirisiin tutkimuksen pohjalta, joissa esiin nousevat muun muassa edellä esitetyt hintakuplat ja anomaliat. Lisäksi osakemarkkinoille on huomattu syntyvän niin sanottua momentumia eli liikemäärää (Balsara ja muut, 2006). Tällä tarkoitetaan käytännössä sitä, että osakemarkkinoiden hintaliike pyrkii jatkumaan siihen suuntaan, mihin se on myös aikaisempina päivinä mennyt. Tämä empiirinen tutkimus sotii vahvasti tehokkailla markkinoilla olevaa satunnaiskulkua vastaan.

Lisäksi taloudellisista aikasarjoista on empiirisissä tutkimuksissa pystytty havaitsemaan tilastollisia ominaisuuksia eli niin sanottuja *tyylitellyjä faktoja* (stylized facts). Näillä tyylitellyillä faktoilla tarkoitetaan sitä, että näennäisesti satunnaisilla omaisuuserien hintojenvaihteluilla on huomattu samankaltaisia tilastollisia ominaisuuksia (Cont, 2001). Nämä tilastolliset aikasarjat käyttäytyvät esimerkiksi eri markkinoilla samankaltaisesti, vaikka näennäisesti markkinoiden välillä ei huomattaisikaan korrelaatiota (Mostafa, 2017, s. 19–20). Osa näistä aikasarjoista löytyvistä tyylitellyistä faktoista sotivat myös tehokkaiden markkinoiden hypoteesia vastaan.

Eräs tyylitely fakta liittyy tuottojen jakautumiseen. Tämän tyylitellyn faktan mukaan tuotot jakautuvat ns. ”häntiin” (heavy tails) (Cont, 2001). Tämä häntiin jakautuminen ei vastaa täysin tehokkaiden markkinoiden ehtoja ja satunnaiskulut teoriaa, sillä tämä tyylitely fakta sulkisi pois ”stabiilin lain” loputtomasta varianssista ja normaalijakautuneisuudesta. Satunnaiskulunteoria pohjautuu Brownin liikeyhtälöön, jonka mukaan tuottojen tulisi olla normaalisti jakautuneita (Cont, 2001; Mostafa, 2017, s. 19–20).

Toinen tyyllitelty fakta on niin kutsuttu voittojen ja tappioiden epäsymmetria (gain/loss asymmetry). Osakkeiden ja indeksien arvonlaskut voivat olla todella suuria, mutta arvonousut eivät ole koskaan niin poikkeuksellisen suuria verrattuna suuriin laskuihin (Cont, 2001; Mostafa, 2017, s. 19–20). Toisin sanoen osakkeilla ja indekseillä kestää kauemmin nousta takaisin ylös kuin laskea. Toinen tähän läheisesti liittyvä asia on vipuvaikutus (Leverage Effect). Tämän teorian mukaan omaisuuserien volatilitteetilla on negatiivinen korrelaatio kyseisen omaisuuserän voittojen kanssa (Cont, 2001).

Viimeinen tyyllitelty fakta liittyy volatilitteetin klusteroitumiseen (volatility clustering). Contin (2001) mukaan vaihtelevat volatilitteetit osoittavat positiivista autokorrelaatiota useiden päivien ajan. Tämä tukee sitä tutkimustulosta, että suuren volatilitteetin tapahtumilla on taipumus klusteroitua ajan kuluessa. Volatilitteetin klusteroitumisella tarkoitetaan käytännössä siis sitä, että tuottojen suuria muutoksia seuraa suuri vaihtelu ja päinvastoin tuottojen pieniä muutoksia seuraa pieni vaihtelu. Tämän katsotaan olevan ristiriidassa satunnaiskulunteorian kanssa.

2.1.2 Aikasarjan momentum-teoria

Vuonna 2012 Moskowitz ja muut julkaisivat tutkimuksen, jonka aiheena oli niin kutsuttu aikasarjan momentum -teoria. Julkaisussaan he tutkivat 58 eri sijoitusinstrumenttia osakeindekseistä aina valtion velkakirjoihin. He huomasivat 1–12 kuukauden aikajaksolla pysyvyyttä tuotoissa, joka aikajakson jälkeen kääntyy päinvastaiseksi. He siis loivat trendiä hyödyntävän sijoitusstrategian, joka pohjautui vain sijoitusinstrumentin omiin historiallisiin liikkeisiin. Tämä poisti siis tarpeen vertailla omaisuusluokkia keskenään voittajien ja häviäjien löytämiseksi. Moskowitzin ja muiden (2012) mukaan nämä tulokset ovat linjassa sentimenttiteorioiden kanssa, alkuperäistä alireagoinnin ja viivästyneen ylireagoinnin kanssa. Tutkimuksen tulosten mukaan aikasarjan momentumin avulla jokaisessa omaisuusluokassa pääsi poikkeuksellisiin tuottoihin. He tulivat myös siihen johtopäätökseen, että tämä strategia toimii erityisen hyvin äärimmäisillä markkinoilla (extreme

markets). Heidän johtopäätöksensä oli se, että spekulioijat voivat tehdä poikkeuksellisia voittoja tällä strategialla suojaustoimintaa käyttäviä sijoittajia vastaan.

Moskowitzin ja muiden (2012) julkaisun jälkeen aiheen tiimoilta on tehty myös paljon uutta tutkimusta, joissa aikasarjan momentum-teoriaa on testattu kansainvälisillä markkinoilla eri omaisuusluokilla ja eri ajanjaksoilla. Moskowitzin ja muiden teoriaa tukevia tutkimustuloksia ovat saaneet esimerkiksi Hurst ja muut (2017). Tässä tutkimuksessa tutkijat hyödynsivät aikasarjan momentum -teoriaa aina vuodesta 1880 vuoteen 2016 saakka. He tutkivat 67 eri markkinaa neljässä eri omaisuusluokassa: 29 hyödykkeissä, 11 osakeindekseissä, 15 bondeissa ja 12 valuutoissa. Tulosten mukaan momentum oli tuottanut positiivista keskimääräistä tuottoa, jolla on alhainen korrelaatio perinteisten omaisuusluokkien kanssa.

Tutkijoiden mukaan aikasarjan momentum oli toiminut hyvin myös erilaisissa makroympäristöissä. Ham ja muut (2019) tutkivat aikasarjan momentum -teoriaa Kiinan hyödykefutuuri-markkinoilla. Tulosten mukaan aikasarjan momentum -teoria voitti verrokkinsa sekä riskikorjatun että kumulatiivisen voiton suhteen. Tutkijat kuitenkin huomauttivat, että momentumin käyttö Kiinassa kestää vähemmän aikaa verrattuna Yhdysvaltoihin, sillä Kiinan futuurimarkkinoilla oli enemmän spekulatiivisia sijoittajia. Aikasarjan momentum -teoriasta ovat tämän lisäksi antaneet positiivisia näyttöjä Gao ja muut vuoden 2018 tutkimuksessa, Jin ja muut vuonna 2020 sekä Pitkäjärvi ja muut vuonna 2020.

Kaikki empiiriset tutkimukset eivät kuitenkaan ole olleet puhtaasti teoriaa tukevia. Esimerkiksi vuonna 2016 Kim ja muut tulivat tutkimuksessaan siihen johtopäätökseen, että aikasarjan momentum-teorian hyöty katoaa, jos kunkin omaisuusluokan tuottoa ei skaalata niiden volatiliteettien mukaan, kuten Moskowitz ja muut (2012) ovat esittäneet. He & Li (2015) nostivat esiin myös sen seikan, että teoria olisi tuottoisa vain silloin, kun momentum-kauppiat hallitsevat markkinoita ja teoriaa tarkastellaan vain hyvin lyhyellä aikavälillä. Myös Huang ja muut (2020) pitivät aikasarjan momentum-teorian tuomia tuottoja melko heikkona, kun tutkitaan eri omaisuuseriä laajalla skaalalla. Tutkimuksissa he

kuitenkin nostivat esiin sen, etteivät he sulje strategialla saatavia tuottoja pois, jos strategiaa esimerkiksi optimoidaan aikahorisontin suhteen.

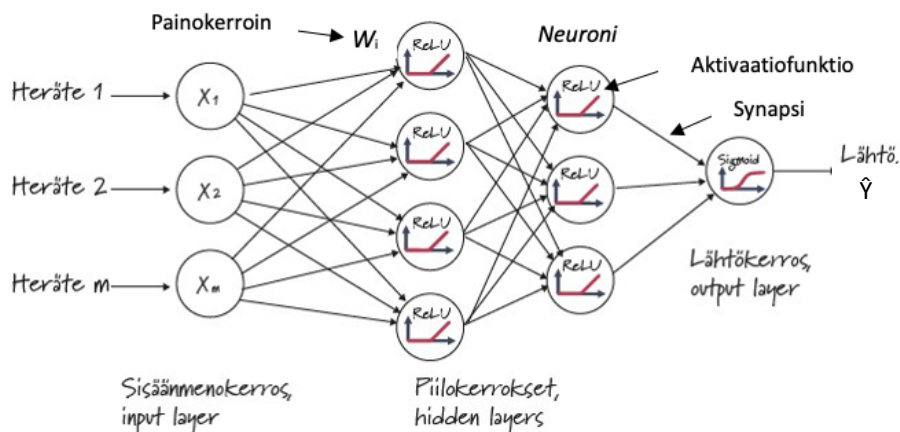
Näiden edellä mainittujen teorioiden ja huomioiden pohjalta voidaan vetää johtopäätöksiä sen suhteen, etteivät markkinat todellisuudessa olekaan täydellisesti tehokkaat ja osakkeiden historialliset aikasarjat voivat pitää sisällään jonkinasteista informaatiota myös tulevista hinnoista. Vaikkakin vaikutus markkinoilla on vähäinen, sijoittajien epärationaalisen käyttäytymisen havainnot myös tukevat sitä, että tehokkaiden markkinoiden teoria ei ole täydellinen. Tämä avaakin mielenkiintoisen mahdollisuuden siihen, että taloudellisia aikasarjoja analysoimalla voidaan tehdä ennustuksia myös tulevista markkinan liikkeistä.

2.2 Neuroverkkojen elementit ja rakenne

Koneoppimismenetelmistä puhuttaessa esille nousee usein termi *syväoppiminen* (deep learning) sekä *neuroverkot* (neural networks). Yksinkertaisesti syväoppimisella tarkoitetaan verkkoa, joka sisältää prosessointiyksiköistä koostuvia kerroksia, joiden läpi järjestelmässä prosessoitu tieto kulkee (Elements of AI, 2020). Neuroverkko on juuri tällainen syväoppimismalli. Neuroverkon rakenne perustuu aivojen biologiseen rakenteeseen, mutta se ei kuitenkaan toimi täysin aivojen kaltaisesti (Puolitaival & Kananen, 2019, s. 127–128). Rakenteensa takia neuroverkolla on mahdollisuus prosessoida valtavia määriä dataa yhtäaikaisesti. Tämän lisäksi neuroverkolla on kyky sopeutua ja oppia kokemukseräisesti dataa hyödyntäen, minkä pohjalta se pystyy yleistämään jo opittuja asioita. Tämä rakenne mahdollistaa myös tiedon hajautetumman varastoinnin eri neuroneihin (Elements of AI, 2020). Tästä johtuen neuroverkolla on hyvä vikasietoisuus ja se siis kykenee jatkamaan toimintaansa, vaikka osaan neuronien välisistä yhteyksistä ”vaurioituisikin”.

Toimintaperiaatteensa ansiosta neuroverkko kykenee tehokkaasti tarkastelemaan erilaisia epälineaarisia ilmiöitä (Puolitaival & Kananen, 2019, s. 128). Tämä mahdollistaa neuroverkon käyttämisen useissa erilaisissa tehtävissä aina kuvan tunnistuksesta aikasarja-

analyysiin. Aikasarja-analyysissä neuroverkolla on useita vahvuuksia perinteisempiin menetelmiin nähden. Se kykenee ratkaisemaan ongelmia aineistoista, jotka ovat epätäydellisiä, virheille alttiita tai jotka sisältävät paljon melua (noise). Neuroverkon kyky toimia erilaisissa tehtävissä vaatii pohjalle kuitenkin suuren määrän dataa (Elements of AI, 2020). Tämän datan avulla neuroverkko oppii ja tämän oppimisen avulla neuroverkon kyky yleistää opittuja asioita tapahtuu.



Kuva 1. Eteenpäin syöttävän neuroverkko (mukaillen Puolitaival & Kananen, 2019, s. 133)

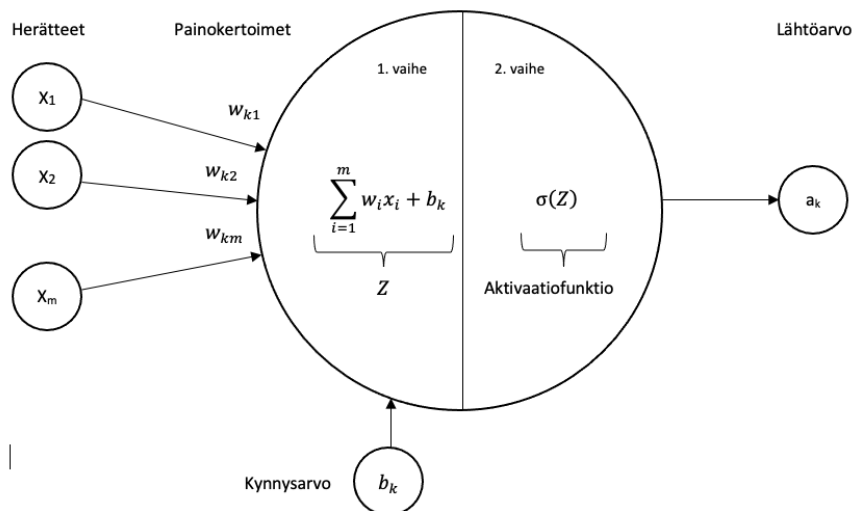
Neuroverkko rakentuu yksinkertaisesti kolmesta erityyppisestä kerroksesta (Kuva 1). Kerrokset ovat *sisäänmenokerros* (myös syöttökerros), *piilokerrokset* sekä *lähtökerros* (myös ulostulokerros) (Puolitaival & Kananen, 2019, s. 129). Sisäänmenokerroksessa neuroverkko vastaanottaa datan. Tässä kerroksessa ei kuitenkaan tapahdu varsinaista laskentaa, mutta kerroksessa on mahdollista skaalata tai normalisoida data, jos sitä ei ole tehty aikaisemmin. Piilokerroksessa tapahtuu varsinainen prosessointi ja siellä neuroverkko pyrkii löytämään käytetystä datasta säännönmukaisuuksia (Elements of AI, 2020). Piilokerroksia voi olla yksi tai useampi. Lähtökerroksessa neuroverkko laskee lopullisen tuloksen, joka pohjautuu aikaisemmin piilokerroksissa saatuihin arvoihin. Tämä rakenne toimii pohjana myös kehittyneemmille neuroverkkomalleille (Elements of AI, 2020).

Neuroverkkojen kohdalla datassa olevia muuttujia kutsutaan *herätteiksi* (Kuva 1). Herätteiksi annetaan siis lopputulokseen oletettavasti vaikuttavat syötearvojen ominaisuudet.

Herätteet menevät sisäänmenokerroksen kautta piilokerroksiin, joissa sijaitsevat *neuro- nit*. Neuroneissa dataa prosessoidaan *painokertoimien* ja *aktivaatiofunktioiden* avulla (Puolitaival & Kananen, 2019, s. 129–130). Painokertoimien avulla voidaan nähdä, kuinka vahvasti sidoksissa kaksi asiaa on keskenään. Neuroverkkojen koulutus on yksinkertaisimmillaan siis tietynlaisten säännönmukaisuuksien löytämistä datasta, toisin sanoen painokertoimien optimaalisten arvojen etsimistä. Neuronista tieto lähtee uutena herätteenä ja uudella painokertoimella *synapsia* pitkin seuraavaan neuroniin (Elements of AI, 2020).

2.2.1 Neuronin toiminta

Neuroverkot rakentuvat siis peräkkäisistä toisiinsa kytketyistä neuroneista. Näitä neuroneita voidaan pitää yksinkertaisina tietojenkäsittely-yksikköinä (Puolitaival & Kananen, 2019, s. 129–131). Neuroverkon koulutus ja älykkyys pohjautuvat näihin yksittäisiin neuroneihin. Neuronin sisällä tapahtuva toiminta on kaksivaiheinen. Kuvassa 2 on esimerkki neuronin k toiminnasta. Ensimmäiseksi neuroni saa herätteet aikaisemman kerroksen neuroneista. Lisäksi herätteisiin vaikuttavat niitä vastaavat painokertoimet.



Kuva 2. Neuronin k toiminta (mukaillen Puolitaival & Kananen, 2019, s. 132)

Neuronin toiminnan ensimmäisessä vaiheessa lasketaan herätteen X_i ja sitä vastaavan painokertoimen W_i lineaarinen summafunktio ja lisätään siihen kyseisen neuronin k kynnyisarvo:

$$\sum_{i=1}^m w_i x_i + b_k \tag{1}$$

Jossa m kuvaa neuronin k tulevien herätteiden määriä. Seuraavaksi summafunktioon lisätään neuronin oma kynnyisarvo b_k , jossa b kuvaa kynnyisarvoa (bias) ja k neuronin. Kynnyisarvon tehtävänä on mahdollistaa neuronin ulostulon poikkeaminen nolasta, vaikka summafunktio saisikin arvon nolla (Tuominen, 2023). Summafunktion ja kynnyisarvon summasta saadaan aktivaatiopotentiaali Z (Puolitaival & Kananen, 2019, s. 131–132).

Neuronin toiminnan toisessa vaiheessa aktivaatiopotentiaali Z vietään aktivaatiofunktioon σ . Aktivaatiofunktion tehtävänä on toimia niin sanotusti kytkimenä, jonka on tarpeen mukaan aktivoitava neuroni (Puolitaival & Kananen, 2019, s. 130–131). Neuronin aktivoitumisella tarkoitetaan sitä, että neuroni lähettää signaalin a_k eteenpäin kohti seuraavaa neuronin (Tuominen, 2023). Aktivaatiofunktio siis määrittelee herätetietojen ja lähtöarvon keskinäisen riippuvuuden. Seuraavassa kappaleessa tarkastellaan aktivaatiofunktion toimintaa ja vaatimuksia hieman tarkemmin.

2.2.2 Aktivaatiofunktio

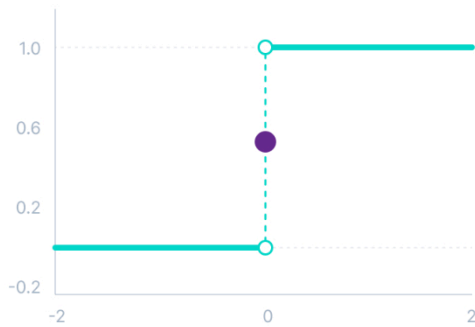
Aktivaatiofunktioita käytetään neuroverkon piilo- ja lähtökerroksen neuroneissa. Kuten kaavasta 1 huomataan, aktivaatiofunktio on lineaarinen. Neuroverkon tehokas käyttö pohjautuu kuitenkin epälineaariseen laskentaan, joten aktivaatiopotentiaali on muutettava epälineaariseksi (Elements of AI, 2020; Puolitaival & Kananen, 2019, s. 129–130). Aktivaatiofunktion tärkein tehtävä onkin nykyaikaisessa neuroverkkolaskennassa muuttaa herätteet epälineaariseksi. Lisäksi aktivaatiofunktion yksi tärkeimmistä ominaisuuksista on mahdollistaa herätteen derivoituvuus (Tuominen, 2023). Derivoituvuuden vaatimus liittyy neuroverkon kouluttamiseen niin kutsutulla gradienttimenetelmällä, johon

palataan tarkemmin alaluvussa 2.2.5. Aktivaatiofunktion avulla neuronin lähtöarvoa kyetään halutessa myös rajaamaan, minkä avulla neuroverkon koulutukseen käytettyä aikaa kyetään vähentämään ja oppimista voidaan myös vakauttaa (Tuominen, 2023).

Aktivaatiofunktioita on lukuisia erilaisia ja niiden kehitys alkoi aivan perinteisestä lineaarisesta funktiosta (Elements of AI, 2020). Lineaarissa funktiossa aktivaatiofunktion arvo on sama kuin aktivaatiopotentialin arvo. Kuten edellä mainittiin, lineaarinen funktio ei kuitenkaan toiminut epälineaaristen funktioiden kanssa. Tällaista aktivaatiofunktiota käytettiin lähinnä funktioiden approksimointiin. Tästä kehitys jatkui porraskunktion eli niin kutsuttuun Heavisiden funktioon (Elements of AI, 2020; Wang, 2010). Porraskunktion matemaattinen esitys on seuraava:

$$\sigma(Z) = \begin{cases} 1, & \text{jos } Z \geq 0 \\ 0, & \text{jos } Z < 0 \end{cases} \quad (2)$$

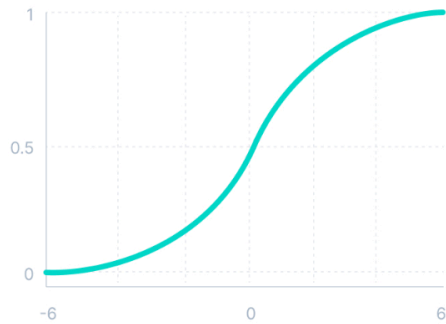
Jossa σ kuvaa aktivaatiofunktioita ja Z aktivaatiopotentialia. Porraskunktion mukaan neuronin ulostulo saa arvon 1, jos aktivaatiopotentialia on suurempi tai yhtä suuri kuin nolla ja arvon 0, jos aktivaatiopotentialia Z saa negatiivisen arvon (Tuominen, 2023).



Kuva 3. Porraskunktion (Baheti, 2023)

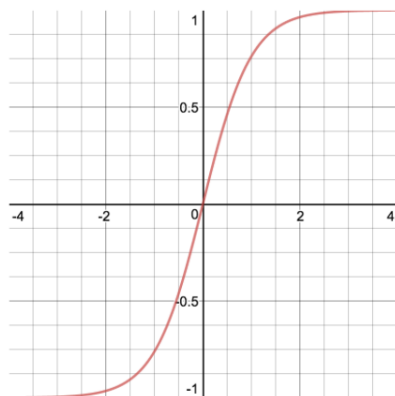
Porraskunktion ansiosta neuronit saivat rajatut ulostuloarvot, minkä ansiosta neuroverkon oppimisnopeus kasvoi (Kuva 3). Tämä funktio ei kuitenkaan täyttänyt vielä jatkuvuuden ehtoa, eikä se näin ollen ollut kaikkialla derivoituva. Neuroverkkojen kehittyessä tulivat

käyttöön kuitenkin epälinearisuutta hyödyntävät aktivaatiofunktiot (Elements of AI, 2020).



Kuva 4. Sigmoid-funktio (Baheti, 2023)

Epälinearisista aktivaatiofunktioista yksi yleisimmistä on Sigmoid-funktio (Kuva 4). Sigmoid-funktio on s-mallinen kuvaaja, joka rajaa lähtöarvot välille $[0, 1]$. Edellisestä aktivaatiofunktioista poiketen sen derivaatta on jatkuva koko välillä. Sigmoid-funktion ongelmaksi muodostuu kuitenkin sen muoto, josta johtuen sen derivoituvuus lähellä ääriarvoja on todella hidasta (Tuominen, 2023). Tästä johtuen sen numeerinen laskenta koulutusvaiheessa on todella hidasta.



Kuva 5. Hyperbolinen tangenttifunktio (Tuominen, 2023)

Toinen hyvin samankaltainen aktivaatiofunktio on hyperbolinen tangenttifunktio (tanh) (Kuva 5). Eroina Sigmoid-funktioon on symmetrisyys nolla suhteen, jolloin se myös

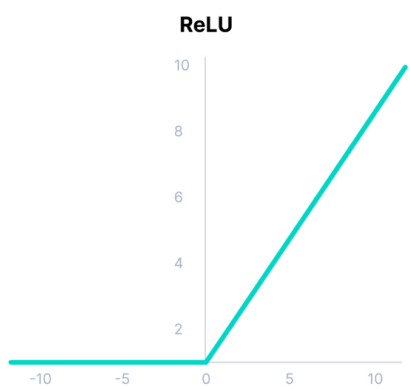
kasvaa nollan lähellä nopeammin ja täten sen derivaatta on suurempi (Tuominen, 2023). Kyseisen funktion matemaattinen ilmaisu on seuraava:

$$\begin{aligned} \sigma(Z): \mathbb{R} &\rightarrow]-1,1[, \\ \sigma(Z) &= \frac{1-e^{-2Z}}{1+e^{-2Z}} \end{aligned} \quad (3)$$

Myöhemmin kehittyi niin kutsuttu Rectified Linear Unit (ReLU) aktivaatiofunktio, jonka käyttö on yksinkertaisempaa ja tehokkaampaa kuin aiemmilla funktioilla (Tuominen, 2023). ReLUn matemaattinen tulkinta on seuraavanlainen:

$$\begin{aligned} \sigma(Z): \mathbb{R} &\rightarrow [0, \infty[, \\ \sigma(Z) &= \max\{0, Z\} \end{aligned} \quad (4)$$

Jos aktivaatiopotentiali Z on positiivinen, myös ulostulo saa positiivisen arvon Z (Kuva 6). Muussa tapauksessa lähtöarvo on nolla. Tämän ansiosta koulutusvaiheessa numeerinen laskenta on paljon tehokkaampaa, sillä lähtöarvon derivaatta pysyy joko vakiona tai se on nolla (Tuominen, 2023). ReLUn yksinkertaisuudesta ja tehokkuudesta johtuen sitä tullaan käyttämään tässä tutkielmassa kaikissa neuroverkkomalleissa.



Kuva 6. ReLU-aktivaatiofunktio (Baheti, 2023)

2.2.3 Neuroverkon koulutus

Neuroverkon oppiminen voidaan jakaa kolmeen eri tyyppiin, joita ovat ohjattu oppiminen, ohjaamaton oppiminen sekä vahvistava oppiminen. Ohjatussa oppimisessa neuroverkolle syötetään heräte-lähtöarvo-pareja. Neuroverkko muuttaa painokertoimia sen mukaan, että verkosta saatava heräte päättyy mahdollisimman lähelle oikeaa lopputulosta (Puolitaival & Kananen, 2019, s. 136). Ohjaamattomassa oppimisessa neuroverkolle syötetään vain herätteitä, joista neuroverkko pyrkii löytämään säännönmukaisuuksia, joiden pohjalta se pyrkii jakamaan herätteitä ryhmiin niiden ominaisuuksien perusteella (Russell ja muut, 2022, s. 826). Vahvistavassa oppimisessa halutun lopputuloksen saavuttamiseksi verkolle syötetään positiivista tai negatiivista palautetta. Neuroverkko mukautuu siten, että positiivisen palautteen määrä kasvaa ja negatiivisen palautteen määrä vähenee (Goodfellow ja muut, 2016). Tässä tutkielmassa tullaan käyttämään ohjattua oppimista, sillä koulutuksessa käytettävässä aineistossa on tiedossa sekä herätteet että niitä vastaavat lähtöarvot.

Neuroverkko käyttää oppimiseensa numeerista dataa, jonka pohjalta neuroverkko koulutetaan yleistämään sen oppimia asioita. Russelin ja muut (2022, s. 683) käsittelevät kirjassaan aineiston jakoa seuraavasti. Neuroverkon käytössä aineisto jaetaan yleensä koulutus- sekä testausaineistoon. Näiden lisäksi voidaan käyttää myös validointiaineistoa. Suurin osa aineistosta käytetään koulutusaineistoksi neuroverkolle. Tämän aineiston avulla neuroverkko tekee tarvittavat muutokset painokertoimiin. Validointiaineisto on koulutusaineistosta erillään olevaa dataa, jonka avulla neuroverkon oppimiskykyä voidaan arvioida. Validointiaineiston avulla pystytään tekemään päätelmiä neuroverkon analysointikyvystä ja siitä, antavatko käytössä oleva neuroverkkotyyppi ja hyperparametrit vaadittavat tulokset. Lisäksi validointiaineiston avulla pystytään tekemään havaintoja yli- tai alisovittamisesta. Näitä asioita tullaan käsittelemään tarkemmin seuraavissa kappaleissa.

Neuroverkon kouluttamisessa painokertoimia ja kynnyksarvoja pyritään muuttamaan niin, että neuroverkon lähtöarvo eli approksimaatio \hat{Y} olisi mahdollisimman lähellä todellista

arvoa Y . Tätä ennustetun arvon ja todellisen arvon välisestä erotuksesta voidaan laskea *kustannusfunktio* (Puolitaival & Kananen, 2019, s. 136). Neuroverkon kouluttamisella tähdätään yksinkertaisesti siis kustannusfunktion minimoimiseen. Koulutuksen aikana kustannusfunktion minimoiminen tapahtuu laskentakierrosten aikana, joissa painoker-toimia ja kynnsarvoja säädetään. Tällaisesta yhdestä laskentakierroksesta käytetään ni-mitystä *epookki* (Puolitaival & Kananen, 2019, s. 136). Yleinen tapa optimoida neurover-
kon parametreja on käyttää niin sanottua *gradienttimenetelmää* (Gradient Descent). Menetelmän tarkoitus on yksinkertaisuudessaan etsiä funktion minimiarvoa, eli arvoa, jossa kustannusfunktio minimoituu (Russell ja muut, 2022, s. 128). Tätä gradienttimene-
telmää tullaan hyödyntämään myös tässä tutkimuksessa.

Seuraavissa kappaleissa neuroverkon kouluttamista tullaan esittelemään yksinkertaisen monikerroksisen eteenpäin syöttävän neuroverkon avulla. Tämä tullaan tekemään siitä syystä, että sen matemaattinen tulkinta on yksinkertaisempi, ja samat periaatteet koskevat myös monimutkaisempien neuroverkkomallien kouluttamista.

2.2.4 Kustannusfunktio

Kustannusfunktiolla on merkittävä osa neuroverkon koulutuksessa ja gradienttimenetel-mässä sen minimikohta pyritään löytämään osittaisderivoimalla kustannusfunktio sekä painokertoimen w_i että kynnsarvon b_i suhteen (Tuominen, 2023). Kustannusfunktioita on erilaisia ja sen valinnalla on suuri merkitys koulutukseen. Yksi yleisimmin käytetyistä kustannusfunktioista on *keskineliövirhe* (mean square error, MSE).

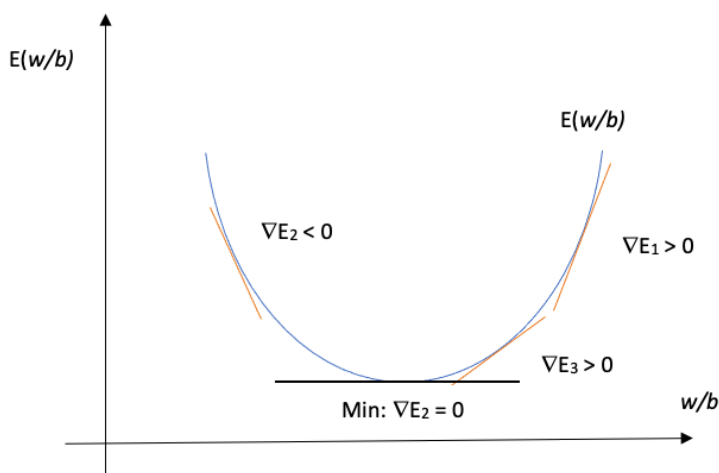
Koska kustannusfunktion halutaan mittaavaan koko neuroverkon jokaisen neuronin vir-hettä, normalisoidaan keskineliövirheen kaava ennen käyttöä (Tuominen, 2023). Sen ma-
temaattinen esitystapa on tässä tapauksessa seuraava:

$$E = \frac{1}{2N} \sum_{n=1}^N (Y(n) - \hat{Y}(n))^2 \quad (5)$$

Missä N kuvaa harjoitusaineistossa olevia esimerkin määriä ja n kuvaa epookin askelta. \hat{Y} kuvaa neuroverkon approksimaatiota ja Y vastaavaa todellista arvoa.

2.2.5 Gradienttimenetelmä

Neuroverkon kouluttaminen lähtee gradienttimenetelmässä liikkeelle siitä, että kaikki neuroverkossa olevat painokertoimet ja kynnsarvot saavat ensin satunnaiset arvot (Tuominen, 2023). Tämän jälkeen neuroverkkoon syötetään herätteet ja lasketaan neuroverkon lähtöarvo \hat{Y} . Tästä käytetään myös nimitystä *feedforward*-laskenta (Puolitaival & Kananen, 2019, s. 136–136). Tämän jälkeen lasketaan herätteitä vastaavan kustannusfunktion osittaisderivaatat kaikkien neuroverkon painokertoimien w_i ja vakiotermien b_i suhteen (Tuominen, 2023). Tästä johtuu myös syy sille, miksi aktivaatiofunktioiden tulee olla derivoituvia kaikkialla. Nämä osittaisderivaatat lasketaan niin kutsutun vastavirta-algoritmin (Backpropagation algorithm) avulla (Tuominen, 2023). Vastavirta-algoritmi käsitellään tarkemmin alaluvussa 2.2.6. Tulosten pohjalta painokertoimien ja vakioker-toimien parametreja muutetaan kustannusfunktion nopeimmin pienenevään suuntaan eli derivaatan negatiiviseen suuntaan.



Kuva 7. Kustannusfunktion minimointi

Kuva 7 havainnollistaa tätä toimintaa esimerkin avulla. Gradienttimenetelmässä laskeaan ensin kustannusfunktiolle E osittaisderivaatta painokertoimen ja vakiotermin suhteen. Vastaukseksi saadaan derivaatan tulos E_1 . Tästä neuroverkon parametreja muutetaan siten, että siirrytään nykyisestä pisteestä derivaatan negatiiviseen suuntaan (vasemmalle). Tämän jälkeen uusilla derivaatan arvoilla päädytään funktiolla pisteeseen E_2 . Tätä siirtymän suuruutta kontrolloidaan oppimisnopeuden avulla. Jälleen parametrit muutetaan ja siirrytään negatiivisen derivaatan suuntaan (nyt oikealle). Seuraavalla derivoinnilla päädytään pisteeseen E_3 . Tätä samaa hakeutumista kohti kustannusfunktion minimiarvoa tehdään koko neuroverkon koulutuksen ajan ja tämä on yksinkertaisimmillaan gradienttimenetelmän ydin.

Matemaattisesti ilmaistuna parametrin arvon muuttaminen yksittäisille neuronin painokertoimille w_i ja kynnsarvoille b_i saadaan seuraavien kaavojen avulla:

$$w_i \rightarrow w_i - \alpha \frac{\Delta E}{\Delta w_i} \quad (6)$$

Sekä

$$b_i \rightarrow b_i - \alpha \frac{\Delta E}{\Delta b_i} \quad (7)$$

Jossa α kuvaa oppimisnopeutta ja $\frac{\Delta E}{\Delta w_i}$ sekä $\frac{\Delta E}{\Delta b_i}$ vastavirta-algoritmissa laskettuja osittaisderivaattoja. Oppimisnopeudella siis kuvataan funktiolla tehtyä siirtymää kohti minimikohtaa. Oppimisnopeuden suuruudella on merkitystä verkon koulutuksessa (Tuominen, 2023). Suurella oppimisnopeudella optimaalisen kohdan yli saatetaan hyppiä useita kertoja, kun taas pienellä oppimisnopeudella verkon koulutus voi olla erittäin hidasta. Pienellä oppimisnopeudella kustannusfunktion lokaalit minimiarvot saattavat myös aiheuttaa ongelmia (Tuominen, 2023).

2.2.6 Vastavirta-algoritmi

Tuominen (2023) käsittelee vastavirta-algoritmin toimintaan seuraavasti. Vastavirta-algoritmin avulla osittaisderivaatat $\frac{\Delta E}{\Delta w_i}$ ja $\frac{\Delta E}{\Delta b_i}$ lasketaan piilokerroksen ja lähtökerroksen kaikille parametreille. Ensimmäiseksi vastavirta-algoritmi laskee lähtökerroksen aktivaatioiden virheen delta δ jokaiselle neuronille k :

$$\delta_k^L = \frac{\Delta E}{\Delta a_k^L} \sigma(Z_k^L) \quad (8)$$

Jossa L kuvaa neuroverkon lähtökerrosta, a neuronin lähtöarvoa ja σ aktivaatiofunktioita. Lähtökerroksen virheen avulla saadaan laskettua neuroverkon edellisen kerroksen virheet:

$$\delta_k^{L-1} = ((w_k^{L+1})^T \delta_k^{L+1} \sigma(Z_k^L) \quad (9)$$

Jossa neuroverkon kerros $l = L-1$. Tästä kaava yleistetään koskemaan neuroverkon kaikkia kerroksia l :

$$\delta_k^l = ((w_k^{l+1})^T \delta_k^{l+1} \sigma(Z_k^l) \quad (10)$$

Kaavoista huomataan, että kustannusfunktion E valinnalla on suuri merkitys neuroverkon koulutuksessa, sillä kaikki parametrien virheiden laskeminen tehdään sen derivaatan pohjalta. Nyt virheen δ avulla kyetään laskemaan osittaisderivaatat kaikkien parametrien suhteen seuraavasti:

$$\frac{\Delta E}{\Delta w_i} = \delta_k^l \quad (11)$$

$$\frac{\Delta E}{\Delta b_i} = a_k^{l-1} \delta_k^l \quad (12)$$

Nyt kyseiset arvot voidaan sijoittaa gradienttimenetelmän kaavoihin 6 ja 7.

2.2.7 Stokastinen gradienttimenetelmä ja optimointialgoritmit

Edellä esitettyssä gradienttimenetelmässä ja vastavirta-algoritmin toiminnassa virhe siis lasketaan jokaisen neuronin kohdalta ja nämä tiedot tulevat lopulta kustannusfunktiolle. Neuronit saavat päivitettyt arvot vasta, kun koko opetusjoukko on käyty läpi eli feedforward-laskennan jälkeen. Tämä johtaa siihen, että neuroverkon tulee säilyttää kaikki opetusjoukon tiedot, joka saattaa suurilla datamäärillä hidastaa kouluttamista. Lisäksi päivityksiä tulee harvoin, joten minimointi saattaa tapahtua liian aikaisin. Tästä johtuen neuroverkon parametrit eivät välttämättä ole täysin optimaalisia.

Perinteisen gradienttimenetelmän lisäksi on olemassa muita menetelmiä ja eräs näistä on niin kutsuttu *stokastinen gradienttimenetelmä*. Tuomisen (2023) mukaan stokastisessa gradienttimenetelmässä neuroverkko oppii käytännössä koko ajan, sillä tässä menetelmässä kustannusfunktio ja neuronien parametrit päivitetään jokaisen opetusjoukon herätteen jälkeen. Parametrien päivityksen ei tarvitse siis odottaa koko opetusjoukon läpikäyntiä. Kuitenkin tällainen tiheä päivittäminen hidastaa osaltaan oppimista ja samalla tiheä päivitys voi aiheuttaa myös parametrien arvojen suurta heiluntaa. Heilunta saattaa kuitenkin edesauttaa optimaalisempien parametrien löytämisen ja näin johtaa parempaan kustannusfunktion minimoimiseen.

Nykyaikaisessa neuroverkon kouluttamisessa nämä gradienttimenetelmien ongelmat parametrien heilunnan ja koulutuksen hitauteen liittyen on kuitenkin pyritty minimoimaan. Näiden ongelmien ratkaisuun käytetään *optimointialgoritmeja*. Nykyaikaiset optimointialgoritmit huomioivat gradienttimenetelmässä myös liikemäärän suunnan eli suunnan, jonne liikutaan kustannusfunktion derivaatan perusteella (Kuva 7). Optimointialgoritmeja on monia erilaisia, mutta tässä tutkimuksessa tullaan käyttämään algoritmina Adamia. Tutkielmassa ei tarkemmin syvennyttä optimointialgoritmin toimintaan tai sen matemaattiseen tulkintaan. Brownleen (2017) mukaan optimointialgoritmi Adamin

voidaan kuitenkin sanoa olevan tehokkaampi versio stokastisesta gradienttimenetelmästä, sillä edellisen menetelmän lisäksi se huomioi epookkien parametrimuutosten suunnat myös nykyisten epookkien parametrimuutoksissa (vrt. momentum). Lisäksi se hyödyntää mukautuvaa oppimisnopeutta. Näin ollen koulutuksessa parametrien arvojen heilunta vähenee ja neuroverkkojen koulutus nopeutuu. Käytännössä Kuvan 7 esimerkin mukaan kustannusfunktion E eri puolille ei tarvitse hyppiä niin montaa kertaa, koska parametrit löytävät optimiarvonsa paremmin.

2.2.8 Neuroverkon hyperparametrit sekä ali- ja ylisovittaminen

Neuroverkon koulutukseen kuuluu olennaisena osana neuroverkon hyperparametrien valinta. Neuroverkon hyperparametreja ovat muuan muassa sen rakenne eli piilokerroksien määrä sekä yksittäisten neuronien määrä näissä kerroksissa. Lisäksi yksi hyperparametreista on oppimisnopeus. Hyperparametrien arvot määritellään aina ennen neuroverkon kouluttamista ja niillä on vaikutusta neuroverkon oppimiskykyyn. Hyperparametrien valintoihin ei ole olemassa suoria sääntöjä vaan sopivimpien hyperparametrien arvojen valinta selviää usein testaamalla.

Neuroverkon kouluttamisen aikana voi tapahtua niin sanottua alisovittamista tai ylisovittamista. Tuominen (2023) sekä Goodfellow ja muut (2016, s. 108–114) kuvaavat ylisovittamista seuraavalla tavalla. Ylisovittaminen on tila, jossa neuroverkko oppii aineiston ”liian hyvin”. Tässä tapauksessa neuroverkon parametrit ovat säädetty optimaalisesti harjoitusaineistolle, mutta sen kyky analysoida testausaineistoa on heikko. Ylisovittamisen voi havaita myös koulutuksen aikana, jos kustannusfunktion pieneneminen hidastuu tai ennustamisen tarkkuus heikkenee. Tämä ongelma ilmenee yleensä silloin, jos harjoitusdata on liian pieni suhteessa neuroverkon kokoon. Neuroverkko kykenee silloin oppimaan pienen koulutusaineiston liian tehokkaasti, eikä pysty yleistämään informaatiota muihin aineistoihin.

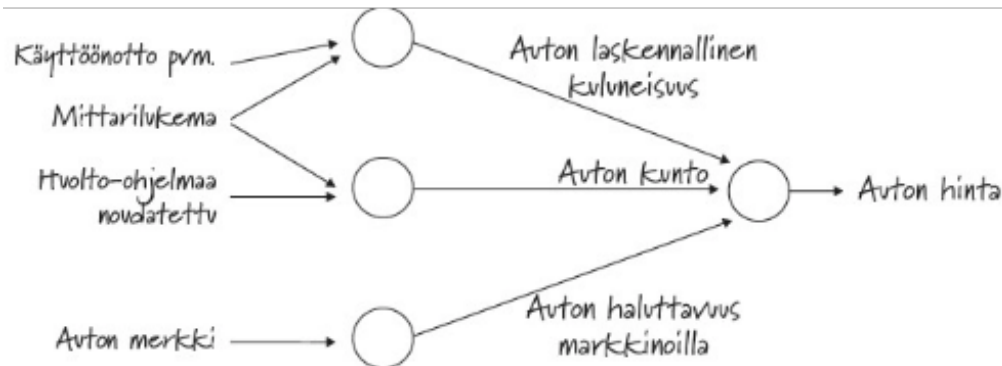
Tuomisen (2023) sekä Goodfellown ja muiden (2016, s. 108–114) mukaan ylisovittamisen ehkäisyyn on kuitenkin muutamia eri keinoja. Yksi vaihtoehto on mahdollisuuksien mukaan kasvattaa koulutuksessa käytettävän datan määrää. Näin neuroverkko ei kykene niin tarkasti omaksumaan vain koulutusaineiston eritysominaisuuksia. Toinen vaihtoehto on neuroverkon pienentäminen (hyperparametrien muuttaminen). Kolmas vaihtoehto on lopettaa neuroverkon kouluttaminen aikaisemmin, jotta ylisovittamista ei ehdi tapahtumaan. Ylisovittamista pyritään koulutuksen aikana pienentämään myös niin sanottu pudotusmenetelmän avulla (drop-out technique) (Aaryan ja muut, 2022; Srivastava ja muut, 2014). Tässä tekniikassa koulutuksen aikana neuroverkon kerroksesta valitaan satunnaisesti neuroneita, joiden lähtöarvoksi annetaan arvo 0.

Päinvastainen ongelma neuroverkkojen kouluttamisessa on alisovittaminen. Alisovittamisessa neuroverkko ei kykene oppimaan käytetystä harjoitusaineistosta tarvittavia ominaisuuksia, joiden pohjalta se kykenisi tekemään yleistykset (Tuomisen, 2023). Kun neuroverkkoa käytetään testausaineistoon, on neuroverkon tekemä virhe suuri. Yleisin syy alisovittamiseen on neuroverkon liian pieni koko käytettyyn harjoitusaineistoon nähden. Alisovittamista voidaan pienentää esimerkiksi neuroverkon kokoa kasvattamalla eli vaikuttamalla neuroverkon hyperparametreihin (Goodfellow ja muut, 2016, s. 108–114). Neuroverkon koulutuksessa neuroverkon tulosten analysointi ja sen pohjalta muutosten tekeminen on siis ensisijaisen tärkeää, jotta neuroverkon oppimiskyky saadaan vaadittavalle tasolle.

2.2.9 Monikerroksiset eteenpäin syöttävät neuroverkot

Monikerroksinen eteenpäin syöttävä neuroverkko (multilayer perceptron, MLP) on yksinkertainen myötäsyötteinen neuroverkkomalli (Elements of AI, 2020). Tämän neuroverkkomallin toiminta ja koulutus on teorioiden osalta käsitelty aikaisemmissa kappaleissa. Neuroverkko siis sisältää sisäänmenokerroksen, vähintään yhden piilokerroksen sekä lähtökerroksen (Puolitaival & Kananen, 2019, s. 129). Tieto syötetään sisäänmenokerroksesta kohti lähtökerrosta ja neuronit hoitavat datan analysoinnin. Neuroverkon

koulutus voidaan tehdä esimerkiksi aikaisemmin käsitellyn gradienttimenetelmän avulla.

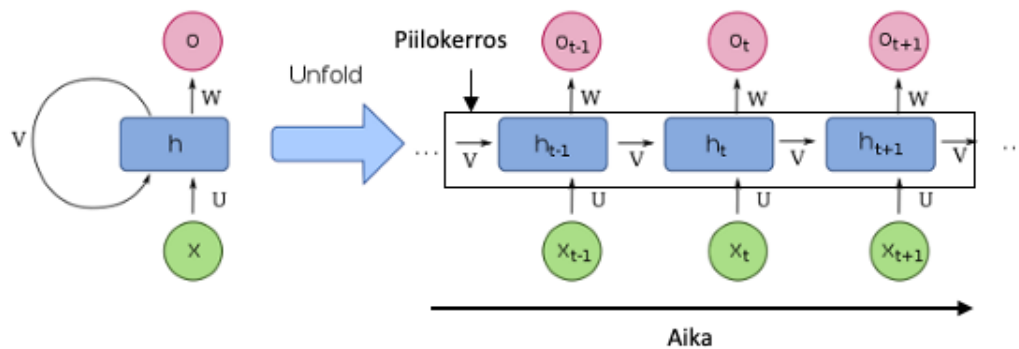


Kuva 8. Auton hinnan määrittely (Puolitaival & Kananen, 2019, s. 135)

Puolitaival ja Kananen (2019) käyttivät MLP toiminnan havainnollistamiseen käytännön esimerkkiä auton hinnan määrittelystä (Kuva 8). Auton hintaan uskotaan vaikuttavan Kuvan 8 mukaiset syötearvojen ominaisuudet eli herätteet. Tämän jälkeen neuroverkko etsii aineistosta säännönmukaisuudet, joiden pohjalta määritellään lopullinen auton hinta. Neuroverkkojen tapauksessa harjoitteludataksi kerättäisiin tiedot näistä ominaisuuksista ja autojen hinnoista. Tämän jälkeen kaikki neuronit yhdistettäisiin keskenään ja harjoitteluvaiheessa niiden välille muodostettaisiin parhaiten kuvaavat painoarvot. Tämän pohjalta neuroverkko kykenisi määrittelemään myös muille autoille hinnat, jos niistä saataisiin kerättyä edellä mainitut parametrit.

2.2.10 Takaisinkytketyt neuroverkot

Kuten edellisissä luvuissa on esitetty, MLP laskee neuronin lähtöarvon vain siihen tulevien herätteiden avulla. Tämä laskenta tapahtuu aina kerros kerrokselta ja täten MLP ei kykene hyödyntämään menneitä tietoja enää uusissa lähtöarvoissa. Näin ollen eri kerrosten tietoja ei pystytä hyödyntämään niin tehokkaasti keskenään.



Kuva 9. Esimerkki RNN rakenteesta (mukaillen Wikipedia, 2023)

Tämä ongelma on kuitenkin pystytty ratkaisemaan kehittyneemmällä neuroverkkorakenteella, takaisinkytketyllä neuroverkolla (recurrent neural network, RNN). Goodfellown ja muiden (2016, s. 372–399) mukaan RNN kykenee rakenteensa ansiosta hyödyntämään koko herätearvojen historiaa lähtöarvon laskennassa. Tämä tarkoittaa käytännössä sitä, että takaisinkytketyllä neuroverkolla on muistiominaisuus, jota se pystyy hyödyntämään. Tämän tyyppinen neuroverkko onkin erittäin hyödyllinen ajallisesti jäsennetyn datan käsittelyyn.

Kuvassa 9 vasemmalla näkyy takaisinkytketyn neuroverkon yksinkertaistettu rakenne, jossa x kuvaa herätettä, h neuronissa olevaa muistisolua ja o lähtökerrosta. u kuvaa herätettä vastaavaa painokerrointa, w lähtökerroksen neuroniiin vaikuttavaa painokerrointa ja v kuvaa syötettä, joka vaikuttaa seuraavaan muistisoluun. Kuvassa 9 oikealla on avattu RNN toiminta eri aika-askelilla. Aikasarja-aineistolla tämä tarkoittaa eri ajanhetkiä. Ensimmäisessä kohdassa neuroni saa herätteen x_{t-1} , jossa t kuvaa ajanhetkeä. Herätteen vaikuttaa myös painokerroin u . Lisäksi neuroniiin vaikuttaa edellisen neuronin lähtöarvo v . Muistisolussa h_{t-1} tapahtuu prosessointi ja neuronista lähtee lähtöarvo o_{t-1} , johon vaikuttaa myös painokerroin w . Lisäksi neuronista lähtee uusi lähtöarvo v taas seuraavalle neuronille seuraavaan aika-askeleelle.

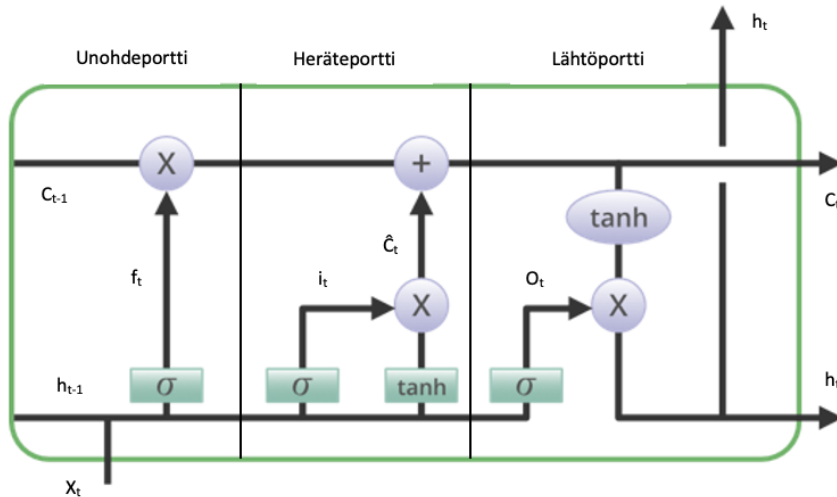
Goodfellown ja muiden (2016, s. 396–399) mukaan takaisinkytketyillä neuroverkoilla on kuitenkin niin sanottu *katoavan gradientin ongelma* (vanishing gradient problem). Tämä

ongelma syntyy neuroverkon koulutuksen aikana vastavirta-algoritmia käytettäessä. Neuronissa oleva aktivaatiofunktio (esim. sigmoid-funktio) rajaa herätteistä tulleet arvot tietylle välille, usein $[0, 1]$. Tästä johtuen suuretkin muutokset herätteissä aiheuttavat vain pieniä muutoksia lähtöarvoissa eli derivaatta (muutosnopeus) pienenee. Kuten alaluvuissa 2.2.5 ja 2.2.6 esitettiin, gradienttimenetelmässä neuronien virheiden laskeminen alkaa aina lähtökerroksesta kohti ensimmäisiä kerroksia ja parametrien muutokset pohjautuvat osittaisderivaattaan. Tämä aiheuttaa käytännössä sen ongelman, että laskeittaessa yhä pieneneviä derivaattoja kerros kerrokselta taaksepäin, parametrien arvojen muutos pienenee eksponentiaalisesti. Tämä taas aiheuttaa sen, etteivät neuroverkon alkupäässä olevien parametrien arvot muutu lainkaan tai muutos on erittäin vähäistä. Ongelma siis vain pahenee, mitä enemmän neuroverkossa on kerroksia. Tämä taas heikentää kustannusfunktion minimointia ja täten neuroverkon yleistämiskyky heikkenee. Katoavan gradientin ongelmaa voidaan minimoida käyttämällä esimerkiksi sellaista aktivaatiofunktioita, joka ei rajaa lähtöarvoa tietylle välille. Tällainen aktivaatiofunktio on esimerkiksi ReLU, jota myös tässä tutkimuksessa käytetään. Toinen ratkaisu on käyttää takaisinkytketyn neuroverkon kehittyneempiä versioita, kuten *pitkäkestoista työmuistia*, joka esitellään seuraavassa alaluvussa.

2.2.11 Pitkäkestoinen työmuisti

Pitkäkestoinen työmuisti (long short term memory, LSTM) on takaisinkytketyn neuroverkon kehittyneempi versio. Se on ratkaisu edellä mainitun takaisinkytketyn neuroverkon katoavan gradientin ongelmaan. Takaisinkytketyn neuroverkon tavoin LSTM kykenee huomaamaan aineistoista lyhyen aikavälin riippuvuuksia ja tämän lisäksi se sopii myös pidemmän aikavälin tarkasteluun. Tämä johtuu siitä, että pitkäkestoisessa työmuistissa muistisolut ovat pidemmälle kehittyneet kuin takaisinkytketyssä neuroverkoissa (Goodfellow ja muut, 2016, s. 404–407). Yleisesti voidaan sanoa, että näiden LSTM-solut päättävät, mitä dataa neuroverkko unohtaa ja muistaa aikaisemmista soluista. Lisäksi näihin neuroverkon muistisoluihin on lisätty

takaisinkytkettyjä syklejä, jolloin katoavan gradientin ongelmaa ei pääse syntymään (Goodfellow ja muut, 2016, s. 404–407).



Kuva 10. LSTM rakenne (mukaillen GeeksforGeeks, 2023)

Kuvassa 10 on avattu pitkäkestoisen työmuistin rakennetta ja toimintaa tarkemmin. LSTM koostuu kolmesta portista, joita ovat *unohdeportti* (forget gate), *heräteportti* (input gate) sekä *lähtöportti* (output gate) (GeeksforGeeks, 2023). Unohdeportissa verkko saa herätteen x_t , jossa t kuvaa ajanhetkeä. Tämän lisäksi siihen vaikuttaa edellisen muistisolun lähtöarvo h_{t-1} . Tästä herätteet siirtyvät unohdeportin aktivaatiofunktioon σ . Tästä saadaan tulos f_t . Jos tulos on lähempänä 1, herätteistä tulleet tiedot säilytetään ja sisällytetään solun tilaan c_{t-1} . Jos tulos on lähempänä 0, informaatio unohdetaan osittain tai kokonaan. Solun tila c on yksinkertaisesti vektori muistettavista asioista, jota päivitetään (Dolphin, 2020). Tämä solun tila, joka kiertää silmukassa ajanhetkeestä toiseen, on erottavallisen RNN toimintaan.

Dolphin (2020) mukaan heräteportissa verkon tarkoituksena on analysoida, mitä uutta informaatiota lisätään solun tilaan c_t aikaisemman tiedon ja uusien herätteiden pohjalta. Tähän tarkoitukseen käytetään kahta aktivaatiofunktiota. Ensimmäinen aktivaatiofunktio toimii kuten unohdeportissa eli sen avulla verkko analysoi, mikä informaatio on tarpeellista. Toisena aktivaatiofunktiona käytetään hyperbolista tangenttifunktiota (tanh) (GeeksforGeeks, 2023). Tällä aktivaatiofunktiolla lähtöarvot saadaan välille $[-1, 1]$

(Dolphin, 2020). Tämän ansiosta herätteiden vaikutusta solun tilaan voidaan vähentää (negatiiviset arvot) tai lisätä (positiiviset arvot). Tämän jälkeen näiden kahden aktivaatiofunktion arvot voidaan yhdistää ja tulokseksi solun tilaan C_t voidaan päivittää informaatio, mikä on tarpeellista ja tiedetään heikentääkö vai vahvistako se tulosta (Dolphin, 2020).

GeeksforGeeks (2023) sekä Dolphin (2020) käsittelevät lähtöportin toimintaa seuraavasti. Lähtöportissa päivitetystä solun tilasta C_t erotellaan tärkeät tiedot neuronin lähtöarvoksi h_t . Tämän jälkeen solun tila C_t siirtyy seuraavaan neuroniin. Jotta lähtöarvosta saadaan tarvittava tieto, muutetaan data ensin vektoriksi hyperbolisen tangenttifunktion avulla (GeeksforGeeks, 2023). Lähtöportissa päivitetystä solun tilasta C_t erotellaan tärkeät tiedot lähtöarvoiksi h_t neuronin arvoksi ja seuraavalle muistisolulle. Ensin solun tilasta data muutetaan vektoriksi hyperbolisen tangenttifunktion avulla. Tämän jälkeen herätteistä X_t ja H_{t-1} muodostetaan tulos O_t aktivaatiofunktio σ avulla. Lopuksi vektorin arvot ja arvo O_t yhdistetään. Tästä saadaan neuronin lähtöarvo H_t . Toimintaperiaatteensa ansiosta LSTM toimii hyvin aikasarja-aineistojen analysoinnissa ja siitä syystä LSTM:ää tullaan käyttämään neuroverkkomallina tässä tutkimuksessa.

2.3 Aikaisemmat tutkimukset

Macielin ja Ballinin (2010) tutkivat monikerroksisen eteenpäin syöttävän neuroverkon kykyä ennustaa tulevia liikkeitä Pohjois-Amerikan, Euroopan sekä Brasilian osakemarkkinoilla. Tutkimuksessa neuroverkon ennustekykyä verrattiin perinteiseen GARCH-malliin kahdeksassa eri indeksissä. Yhdysvaltojen osakemarkkinoilta mukana olivat Dow Jones ja S&P 500. Euroopasta mukaan oli valittu DAX, CAC 40, FTSE, IBEX 35 sekä PSI 20. Lisäksi mukana oli Brasilian Ibovespa.

Kenton (2020) mukaan vertailussa käytetty yleistetty autoregressiivinen ehdollinen heteroskedastinen malli eli GARCH-malli on epälineaarinen malli aikasarja-analyysin

tekemiseen. Mallia käytetään erilaisten rahoitusvälineiden volatilitteen estimointiin. Malli toimii erityisen hyvin tuottojen ennustamisessa.

Maciel & Ballini (2010) käyttivät tekniseen analyysiin perustuvia aikasarja-aineistoja kustakin indeksistä, jotka valittiin korrelaatio- ja autokorrelaatioanalyysin perusteella. Tutkimuksessa otoksena toimivat kunkin indeksin päätösarvot. Tutkijat keräsivät indeksien päivittäiset arvot aikaväliltä 12.1.2000 - 27.7.2008. Varsinainen ennustaminen suoritettiin aikavälillä 2.5.–27.7.2008 ja loput aineistosta käytettiin neuroverkon kouluttamiseen ja validointiin. Kunkin indeksin ennustamista varten luotiin omanlaisensa neuroverkko, joiden hyperparametrit määriteltiin koulutuksessa. (Maciel & Ballini, 2010.)

Tulosten perusteella Maciel & Ballini (2010) tulivat siihen tulokseen, että molemmat menetelmät kykenivät ennustamaan markkinoiden liikkeitä. Neuroverkko suoriutui kuitenkin verrokkiinsa nähden paremmin. Neuroverkon selitysaste oli indekseille keskimäärin 0,951 GARCH-mallin selitysasteen ollessa 0,819. Lisäksi tutkimuksessa huomattiin, että MPL pystyi ennustamaan n. 75 % markkinan liikkeistä, GARCH-mallien kyetessä ennustamaan n. 60 % liikkeistä. Empiirisen tutkimuksen perusteella neuroverkko oli suorituskykyisempi kaikissa indekseissä tarkasteluperiodilla. Kokonaisuutena testi todisti, että neuroverkolla oli kyky ennustaa kaikkia indeksejä klassista GARCH-mallia tarkemmin sekä pienemmällä virhemarginaalilla. Tutkijoiden mukaan yksittäinen sijoittaja voi hyötyä neuroverkkojen käytöstä perinteisiä menetelmiä enemmän.

Desai ja muut (2011) tutkivat neuroverkon ennustamiskykyä Intian S&P CNX Nifty 50-indeksissä. Myös tässä tutkimuksessa tutkijat käyttivät monikerroksista eteenpäin syöttävää neuroverkkoa ja vertasivat sen ennustekykyä samassa aikaperiodilla suoritettuun osta ja pidä-strategiaan.

Tutkimuksessa käytettiin indeksin päätöshintoja ja kaupankäyntiä suoritettiin aikavälillä 1.9.2009 – 30.4.2011 (Desai ja muut, 2011). Tutkijoiden mukaan kyseinen aikaväli valikoitui indeksissä vallinneen nousumarkkinan takia ja tällä markkinalla neuroverkko

kykeni käymään kauppaa sekä pitkäksi että lyhyeksi myymällä. Kokeessa verrattiin keskenään myös kahdeksaa erilaista neuroverkkoa, jotka poikkesivat hyperparametreiltaan ja aktivaatiofunktioiltaan hieman toisistaan. (Stuhlsatz ja muut, 2011; Desai ja muut, 2011). Malli ennusti seuraavan seitsemän päivän liukuvan keskiarvon edellisen seitsemän päivän perusteella.

Desai ja muut (2011) huomasivat, että neuroverkko suoriutui osta ja pidä-strategiaan verrattuna selkeästi paremmin. Yksikään neuroverkkomalli ei saanut huonompaa tuottoa kuin verrokkiryhmä ja parhaimmillaan neuroverkko tuotti aikaperiodilla yli 30 %, osta ja pidä-strategian jäädessä vain -4,25 % tuottoihin. Tuloksen pohjalta tutkijat päätyivät siihen johtopäätökseen, että neuroverkon kyvystä ennustaa trendejä on hyötyä sijoittajalle erityisesti Intian kaltaisessa volatiliteetissä markkinassa. (Desai ja muut, 2011.)

Myös Moghaddamin ja muut (2016) hyödynsivät kokeessaan MLP-neuroverkkoa ennustamaan Nasdaq-indeksin päivittäisiä pörssikursseja. Empiirisessä osuudessa tutkijat jakoivat neuroverkot kahteen ryhmään siten, että toisessa ryhmässä neuroverkot tekivät ennusteen neljän edellisen päivän pohjalta ja toisessa ryhmässä yhdeksän edellisen päivän perusteella. Tutkijoiden tarkoituksena oli selvittää, oliko aikaperiodeilla merkitystä ennusteisiin.

Neuroverkon käytössä Moghaddamin ja muut (2016) hyödynsivät syöteinä indeksin lyhyen aikavälin historiallisia osakekursseja sekä viikonpäiviä. Kahden neuroverkkoryhmän sisällä testattiin myös erilaisia neuroverkkorakenteita, joista parhaat valittiin varsinaiseen kokeeseen. Aineistona käytettiin Nasdaq indeksin osakekursseja aikaväliltä 28.1.-18.6.2015, joista ensimmäistä 70 päivää käytettiin harjoitusjoukkona neuroverkolle ja viimeistä 29 käytettiin ennustekyvyn testaamiseen (Moghaddamin ja muut, 2016.; Yahoo finance, 2023).

Tulosten perusteella Moghaddamin ja muut (2016) havaitsivat, että sekä neljän päivän että yhdeksän edellisen päivän tilastojen avulla neuroverkko pystyi ennustamaan

markkinoiden liikkeitä hyvinkin tarkasti, lukuun ottamatta muutamia vahvoja liikkeitä markkinoilla. Tutkijoiden mukaan myöskään päivien määrällä ei ollut juurikaan eroa tuloksia seurattaessa.

Vuonna 2017 Wu ja Tingtin vertasivat kahden erilaisen neuroverkon ennustekykyä Kiinan CSI 300-indeksin tuottoihin. Tutkimuksessa he käyttivät monikerroksista eteenpäin syöttävää neuroverkkoa (MLP) sekä eräänlaista Elmanin takaisinkytkettyä neuroverkkoa (RNN). Verkko eroaa tavallisesta RNN siten, että se sisältää lisäksi kontekstikerroksen (context layer), jonka ansiosta se pystyy hyödyntämään lähtökerroksen tietoja uudesta kerroksen herätteenä (Guanghua ja muut, 2018).

Wu & Tingtin (2017) valitsivat CSI 300-indeksistä 1000 havaintoa aikaväliltä 13.5.2010 – 30.6.2014. 800 valituista havainnoista otettiin harjoitteluaineistoon ja loput 200 testiaineistoon. Tulosten pohjalta voitiin sanoa, että MLP seurasi indeksiä tarkasti ja suhteellinen virhe oli todellisen ja ennustetun arvon välillä keskimäärin 1,46 %. Suhteellisen virheen varianssi taas oli 1,37 %. RNN suhteellisen virheen keskiarvo oli 0,86 % ja varianssin 0,51 %.

Tulosten perusteella Wu & Tingtin (2017) totesivat, että kummatkin neuroverkkotyypit kykenivät ennustamaan markkinan liikkeitä hyvin. Tutkijat kuitenkin huomauttivat, että RNN toimi verrokkiinsa nähden paremmin ja rakenteensa ansiosta se kykeni hyödyntämään menneitä lähtöarvoja myös nykyisten arvojen analysoinnissa ja ei näin ollen kärsinyt MLP:n rajoitteista.

Koska nykyään on käytössä monenlaisia tekoälymenetelmiä, ei puhdas neuroverkko ole ainoa työkalu markkinoiden ennustamiseen. Käyttöön on tullut esimerkiksi erilaisia hybridineuroverkkoja, joissa neuroverkon lisäksi on yhdistetty joitain muita tekoälymenetelmiä. Guresen ja muut (2011) vertailivat erilaisten neuroverkkojen ennustekykyä keskenään. Neuroverkkoina olivat käytössä tavallinen monikerroksinen eteenpäin syöttävä neuroverkko, dynaaminen keinotekoinen hermoverkko (dynamic artificial neural

network) sekä hybridineuroverkko, jossa käytettiin neuroverkon lisäksi GARCH-mallia tulomuuttujien poimimiseksi. Malleja vertailtiin keskenään keskimääräisen neliövirheen ja keskimääräisen absoluuttisen poikkeaman avulla. Tutkimuksessa käytettiin Nasdaq indeksin markkinatuottoja 7.8.2008 – 26.6.2009. Ensimmäisiä 146 päivää käytettiin koulutukseen ja validointiin ja viimeistä 36 päivää käytettiin testijaksona.

Tutkimuksen pohjalta Guresen ja muut (2011) tulivat siihen tulokseen, että perinteinen eteenpäin katsova neuroverkko toimi hieman paremmin kuin dynaamisen keinotekoisien hermoverkon ja hybridineuroverkot. Lisäksi dynaamisen neuroverkon käyttö osoittautui ominaisuuksiensa takia hieman huonosti ennustamiseen sen toimiessa enemmänkin tilastollisena menetelmänä kuin neuroverkkona. Tämän johdosta tutkijat päätyivät tulokseen, että normaali eteenpäin katsova neuroverkko oli paras ja käytännöllisin neuroverkko ennustamiseen.

Toisenlaiseen tulokseen päätyivät Bao ja muut (2017), jotka vertasivat sekä takaisinkytkettyä neuroverkkoa (RNN) että pitkäkestoista työmuistia (LSTM) hybridineuroverkkoon. Hybridineuroverkko koostui aallokemuunnoksesta (wavelet transforms, WT), jota hyödynnettiin kohinan poistamisessa aikasarjoissa, pinotuista autoenkoodereista (stacked autoencoders, SAEs), joilla löydettiin aikasarjoista pörssikursseihin vaikuttavat ominaisuudet, ja LSTM:stä, jolla ennustettiin yhden päivän tuotot.

Baoll ja muilla (2017) oli ennustettavia indeksejä käytössä maailmanlaajuisesti. Mukana olivat CSI 300, Nifty 50, Hang Seng, Nikkei 225, S&P 500 sekä DJIA. Jokaisesta indeksistä kerättiin dataa aikaväliltä 10.2010–9.2016. Aikaväli jaettiin 24 osaan, joiden aikana toteutettiin sekä harjoittelu että testivaihe. Syötömuuttujina käytettiin 15 eri taloudellista mittaria ja makrotaloudellista indikaattoria indeksien mukaan. Tulosten perusteella tutkijat totesivat hybridimallin olevan tarkempi ennustamaan päivittäisiä päätöskursseja takaisinkytkettyyn neuroverkkoon ja pitkäkestoiseen työmuistiin nähden.

Vaikka monissa tutkimuksissa on saatu tuloksia neuroverkkojen tarkemmista ennusteista perinteisiin tilastollisiin menetelmiin verrattuna, ei ero ole kuitenkaan täysin selkeä (Dzikevičius ja Stabužytė, 2012). Vuonna 2018 Makridakis ja muut tekivät laajan tutkimuksen, jossa he vertasivat kahdeksaa tilastollista menetelmää ja 10 koneoppimismenetelmää keskenään. Koneoppimismenetelmissä olivat mukana sekä MLP, RNN että LSTM. Siinä käytetään 1045 kuukausittaista aikasarjaa Makridakisin aikaisemmasta tutkimuksesta (Makridakis ja muut, 2018; Makridakis ja muut, 2000.) Tulosten pohjalta Makridakis ja muut (2018) tulivat siihen tulokseen, että perinteiset tilastolliset menetelmät ovat koneoppimismenetelmiä tarkempia. He kuitenkin huomauttavat julkaisussa, että käytetyt aikasarjat saattavat vaikuttaa tuloksiin ja koneoppimismenetelmiä ei voida sulkea pois.

Vuonna 2012 Dzikevičius ja Stabužytė vertasivat MLP neuroverkon ennustekykä usean muuttujan regressioanalyysiin sekä liukuvaan keskiarvoon. Aikasarjaa ennustettiin OMX Vilnus -indeksissä. Kokeen aikana tehtiin sekä päivittäisiä että kuukausittaisia ennustuksia sekä indeksin todellisesta tulevasta arvosta että indeksin suunnasta. Koetta varten tutkijat loivat useamman erilaisen neuroverkkorakenteen. Kaikissa käytetyissä neuroverkkomalleissa oli yksi piilokerros ja yhdestä kuuteen neuronua. Päivittäistä ennustetta tehtäessä neuroverkon ennustekyky oli parempi verrokkeihinsa nähden ja parhaiten pärjasi verkko, joka sisälsi yhdessä piilokerroksessa kaksi neuronua. Kuukauden ennusteita tehtäessä neuroverkon ennustekyky hävisi kuitenkin usean muuttujan regressioanalyysille. Kuitenkin kuukauden ennusteita tehdessä kaikki tulokset olivat yleisesti ottaen hyvinkin epätarkkoja. Tässä tapauksessa neuroverkoista parhaiten pärjasi verkko, joka sisälsi yhden piilokerroksen neljällä neuronilla.

Vuonna 2022 tehdyssä tutkimuksessa Bhandari ja muut ennustivat S&P 500 -indeksin seuraavan päivän päätöshintoja hyödyntämällä pitkäkestoista työmuistia. Tutkimuksessa he vertailivat yksikerroksisen LSTM ennustekykä kuuteen erilaiseen monikerroksiseen LSTM-malliin. Aineistoa he keräsivät vuodesta 2006 vuoteen 2020 asti. Aineisto valikoitui tälle välille siitä syystä, että se piti sisällään pitkiä nousumarkkinoita, mutta myös voimakkaita laskumarkkinoita, kuten finanssikriisin vuonna 2008 ja koronaviruspandemian

vuonna 2020. Tutkijat uskoivat laajan markkinadatan johtavat parempiin ennusteisiin. Varsinainen aineisto piti sisällään yhdeksän erilaista herätettä, jotka olivat fundamenteista, makrotaloudesta ja teknisestä analyysistä kerättyjä indikaattoreita. Fundamenteista mukana olivat päivän avaus- ja päätöskurssit. Makrotaloudellisia indikaattoreita olivat volatilitiiteetti-indeksi VIX, korkotasoa, työttömyysaste, kuluttajasentimentti-indeksi sekä US dollari-indeksi. Teknisistä indikaattoreista olivat käytössä liukuva keskiarvo konvergenssille/divergenssille (MACD), keskimääräinen todellinen alue (ATR) ja suhteellisen vahvuuden indeksi (RSI).

Bhandari ja muut (2022) muuttivat sekä yksikerroksisen LSTM että monikerroksisten LSTM hyperparametreja (neuronien määrää, optimointialgoritmi) koulutuksen aikana ja valitsivat niistä optimaalisimmat varsinaiseen ennustamiseen. Tulokset osoittivat, että selkeästi paras ennustekyky saavutettiin käyttämällä yksikerroksista pitkäkestoista työmuistia, jonka piilokerros sisälsi 150 neuronaa. Tutkijat tulivat myös siihen johtopäätöksen, että kyseisen mallin käyttöä voi räätälöidä myös käytettäväksi muihin laajoihin markkinaindekseihin, joissa data käyttäytyy samalla lailla. Tutkijat vielä lisäsivät, että sijoittajat voivat käyttää kyseistä mallia hahmottaakseen markkinatilanteen ennen sijoituspäätöstensä tekoa.

Taulukkoon 1 on kerätty aikaisempien tutkimusten tuloksia ja perustietoja. Näiden tutkimuksen perusteella neuroverkkojen käyttö osakemarkkinoiden ennustamisessa on tuottanut hyviä tuloksia. Etuja on tullut niin perinteisiin malleihin nähden kuin myös osta- ja pidä strategiaan. Kaikki tulokset eivät kuitenkaan ole tukeneet tuloksia ja selvien johtopäätösten tekeminen aikaisemman tutkimustuloksen perusteella on vaikeaa.

Taulukko 1. Tulokset aikaisemmista tutkimuksista

Kirjoittaja ja vuosiluku	Artikkeli	Neuroverkkotyyppi	Verrokki	Johtopäätös
MACIEL ja BALLINI (2010)	NEURAL NETWORKS APPLIED TO STOCK MARKET FORECASTING: AN EMPIRICAL ANALYSIS	MLP	GARCH-MALLI	Neuroverkko kykeni ennustamaan indeksejä verrokiaan tarkemmin ja pienemmällä virhemarginaalilla
Desai et al. (2011)	FORECASTING OF INDIAN STOCK MARKET INDEX S&P CNX Nifty 50 USING ARTIFICIAL INTELLIGENCE	MLP	Osta ja pidä-strategia	Kaikilla neuroverkkomalleilla saatiin parempaa tuottoa osta ja pidä-strategiaan verrattuna
Guresen, Kaykutlu ja Daim (2011)	Using artificial neural network models in stock market index prediction	MLP	Dynaaminen keinotekoinen neuroverkko, hybridineuroverkko ja GARCH-malli	Perinteinen MLP neuroverkko kykeni ennustamaan markkinan liikkeitä verrokkejaan paremmin
Dzikevičius ja Stabužytė (2012)	FORECASTING OMX VILNIUS STOCK INDEX – A NEURAL NETWORK APPROACH	MLP	Usean muuttujan regressioanalyysi ja liukuva keskiarvo	Päivittäisiä ennusteita tehtäessä neuroverkon ennustekyky oli verrokkejaan parempi. Kuukauden ennusteissa usean muuttujan regressioanalyysi suorituti paremmin. Kuukauden ennusteissa kaikki menetelmät olivat kuitenkin epätarkkoja.
Moghaddamin, Moghaddamin ja Esfandiyarin (2016)	Stock market index prediction using artificial neural network	MLP	Verrattiin neuroverkkoja 4 ja 9 edellisen päivän datalla	Kummassakin tapauksessa neuroverkot pystyivät ennustamaan markkinoita hyinkin tarkasti, eikä mallien välillä ollut juurikaan eroja
Wu ja Duan (2016)	A Performance Comparison of Neural Networks in Forecasting Stock Price Trend	MLP ja RNN (Elman neuroverkko)	Vertasivat neuroverkkoja keskenään	Kummatkin neuroverkkotyytit pystyivät ennustamaan markkinoiden liikkeitä hyvin, mutta RNN pohjainen ratkaisu kykeni sovittamaan datan paremmin malliin
Bao, Yue ja Rao (2017)	A deep learning framework for financial time series using stacked autoencoders and long-short term memory	RNN ja LSTM	Hybridineuroverkko	Hybridineuroverkko kykeni ennustamaan päivittäisiä päätöskursseja RNN ja LSTM paremmin
Makridakis, Spiliotis ja Assimakopoulos (2018)	Statistical and Machine Learning forecasting methods: Concerns and ways forward	MLP, RNN ja LSTM	8 tilastollista menetelmää ja 7 muuta koneoppimismenetelmää	Perinteiset tilastolliset menetelmät ovat koneoppimismenetelmiä tarkempia
Bhandari ja muut (2022)	Predicting stock market index using LSTM	LSTM	Vertasivat LSTM keskenään eri hyperparametreilla	Yksikerroksinen LSTM pystyi ennustamaan indeksin hintoja monikerroksisia LSTM menetelmiä tarkemmin

Aikaisemmissa tutkimuksissa nousi esiin myös se seikka, että takaisinkytketyt neuroverkot, erityisesti pitkäkestoinen työmuisti, ovat tuottaneet tulosta verrokkeihinsa sekä monikerroksiseen eteenpäin syöttävään neuroverkkoon nähden. Lisäksi neuroverkkojen hyödyistä volatiileilla reunamarkkinoilla oli tutkimuksessa näyttöä. Tutkimustulokset antavat kokonaisuudessaan tukea sille ajatukselle, että pitkäkestoisen työmuistin testaa- mista OMXH25-indeksissä eri aikasarjoilla kannattaa tutkia tarkemmin.

3 Empiirinen osuus

Tässä luvussa käydään läpi tutkielman empiirinen osuus, ja se jakaantuu kolmeen alalukuun. Ensimmäisessä alaluvussa käydään läpi tutkimuksen aineiston kerääminen, esitellään käytettävät aikasarjat sekä aineistolle tehtävä esikäsittely ennen varsinaista analysointia. Toisessa alaluvussa käydään läpi tutkimuksessa käytettävät menetelmät sekä esitellään niihin liittyvien mallien rakentaminen sekä kouluttaminen. Viimeisessä alaluvussa esitellään saadut tulokset ja analysoidaan niitä. Kaikki tutkimuksessa tehty aineiston kerääminen, muokkaus sekä mallien rakentaminen toteutetaan käyttäen Python-ohjelmointikieltä sekä useita avoimeen lähdekoodiin perustuvia kirjastoja. Ohjelmointiympäristönä käytössä on Jupyter Notebook 6.4.5., joka on verkkopohjainen interaktiivinen lasentaympäristö (Jupyter Team, 2015).

3.1 Käytettävä aineisto

Tutkimuksessa käytettävä aineisto on OMX Helsinki 25 -indeksistä kolmelta eri aikaväliltä kerättävä aikasarja-aineisto. OMXH25 on Helsingin Pörssin johtava osakeindeksi, joka koostuu 25 pörssin vaihdetuimmasta osakkeesta. Nasdaqin (2023) mukaan rajoitettu osakemäärä takaa sen, että kaikilla kohde-etuutena olevilla osakkeilla on tarpeeksi suuri likviditeetti. Indeksillä on myös markkina-arvo painotettu ja yksittäisen yrityksen paino on OMXH25-indeksissä rajoitettu maksimissaan 10 prosenttiin (Nasdaq, 2023).

Tutkimuksessa käytettävä aineisto on kerätty Yahoo! Finance -tietokannasta. Varsinainen data kerättiin käyttämällä Yahoo Finance -ohjelmointirajapintaa, jonka avulla tiedot saatiin malleille suoraan käyttöön edellä mainitusta tietokannasta. Tutkimuksessa käytettävät aikasarjat kerättiin seuraavilta aikaväleiltä:

- Aikasarja 1: 1.1.2014 – 31.12.2019
- Aikasarja 2: 1.1.2016 – 31.12.2017
- Aikasarja 3: 1.1.2020 – 30.9.2022

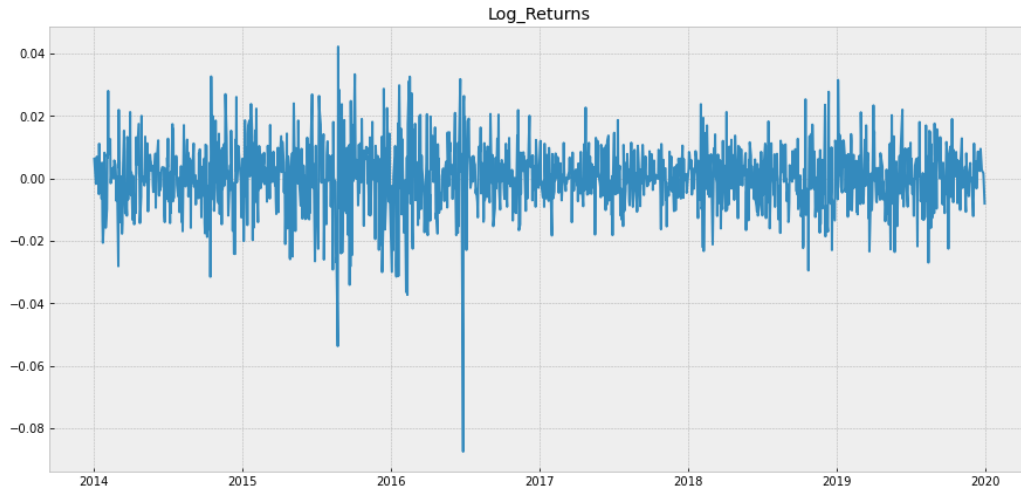
Aineistona käytettiin OMXH25-indeksin päivittäisiä tietoja. Jokaisesta aikasarjasta kerättiin päivittäiset arvot indeksin avaushinnasta, päätöshinnasta, päivän ylimmästä ja alimmasta hinnasta sekä volyymistä. Tarkemmat tiedot tutkimuksessa käytettävistä parametreista löytyvät Taulukosta 2. Tutkimuksen tarkoitus on ennustaa päivän päätöshintaa muiden parametrien avulla. Muut arvot toimivat siis sekä koulutuksessa että testauksessa herätteinä ja päätöshinta on malleista saatava lähtöarvo.

Taulukko 2. Tutkimuksessa käytettävät herätteet ja lähtöarvot

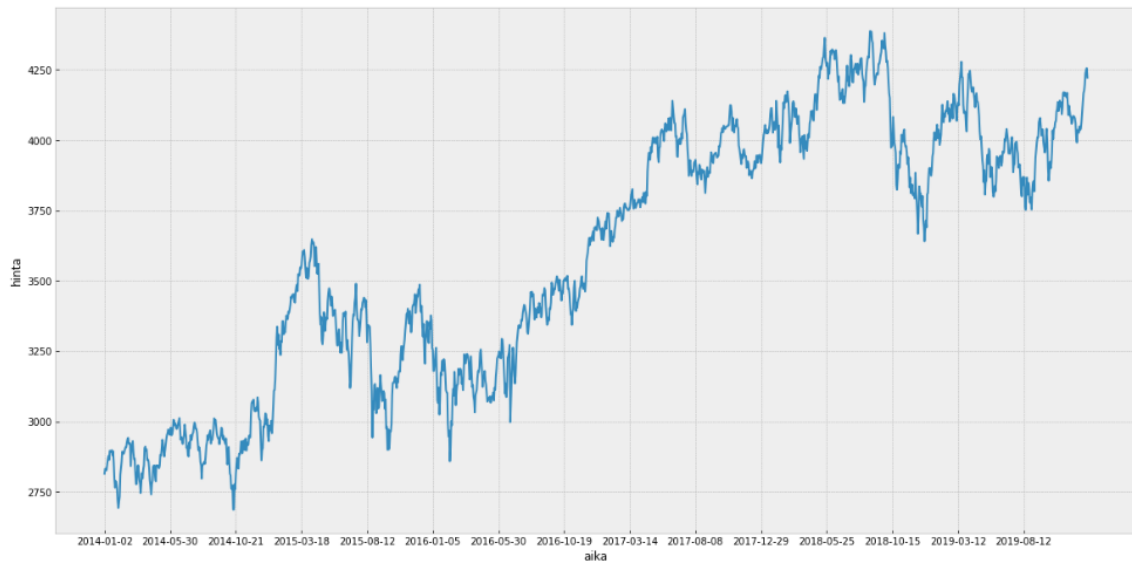
Lähtöarvo	Selitys
Päätöshinta	Hinta, joka määräytyy päätöshuutokaupan tarjousten täsmäytyksen perusteella
Herätteet	Selitys
Avaushinta	Avaushuutokaupassa määräytyvä indeksin tai osakkeen hinta pörssin avauksessa
Ylin hinta	Päivän korkein vaihdettu hinta
Alin hinta	Päivän matalin vaihdettu hinta
Volyymi	Tietyinä päivinä tehdyn päivän kokonaisarvo

Seuraavaksi jokaisesta aikasarjasta on luotu kuvaajat päivittäisten päätöshintojen mukaan. Lisäksi jokaiselle aikasarjan päätöshinnoilla on laskettu logaritmiset tuotot ja luotu myös niistä kuvaajat. Nämä logaritmiset tuotot osoittavat aikasarjojen tuottojen vaihtelun ajan kuluessa.

Kuvassa 11 on esitetty Aikasarjan 1 logaritmistien tuottojen kuvaaja ja Kuvassa 12 kuvaaja indeksin päätöshinnoista koko aikasarjalta. Aikasarja 1 on tutkimuksen pisin aikasarja ja ajalle mahtuu sekä suuria nousuja että suuria laskuja. Lisäksi logaritmisissa tuotoissa voidaan nähdä jonkin verran vaihtelua ja muutamia isoja piikkejä aikavälillä. Kuitenkin kovimmat vaihtelut logaritmisissa tuotoissa näkyvät aineiston alkupäässä.

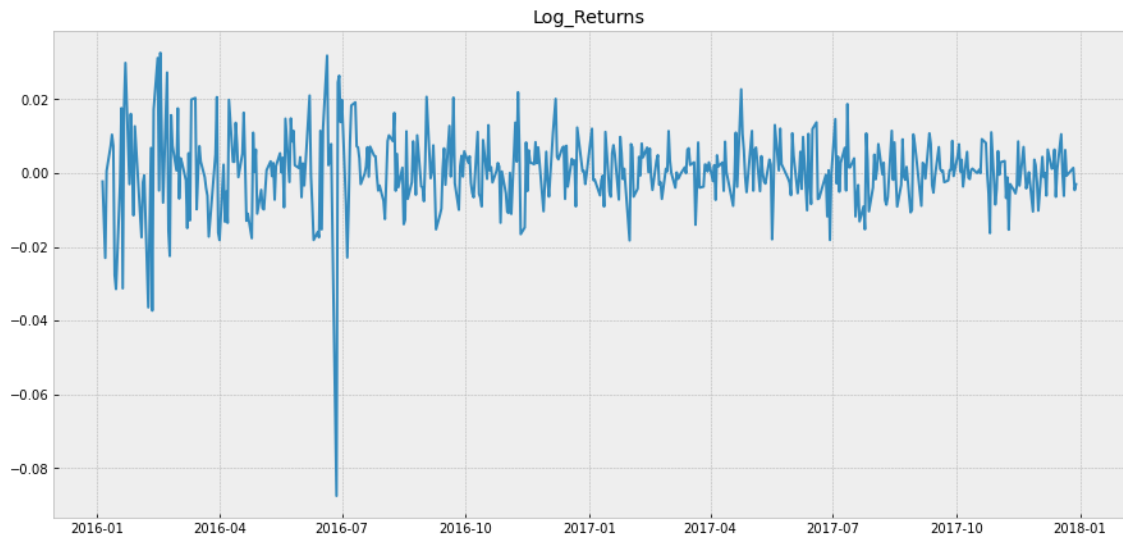


Kuva 11. Logaritmiset tuotot Aikasarja 1

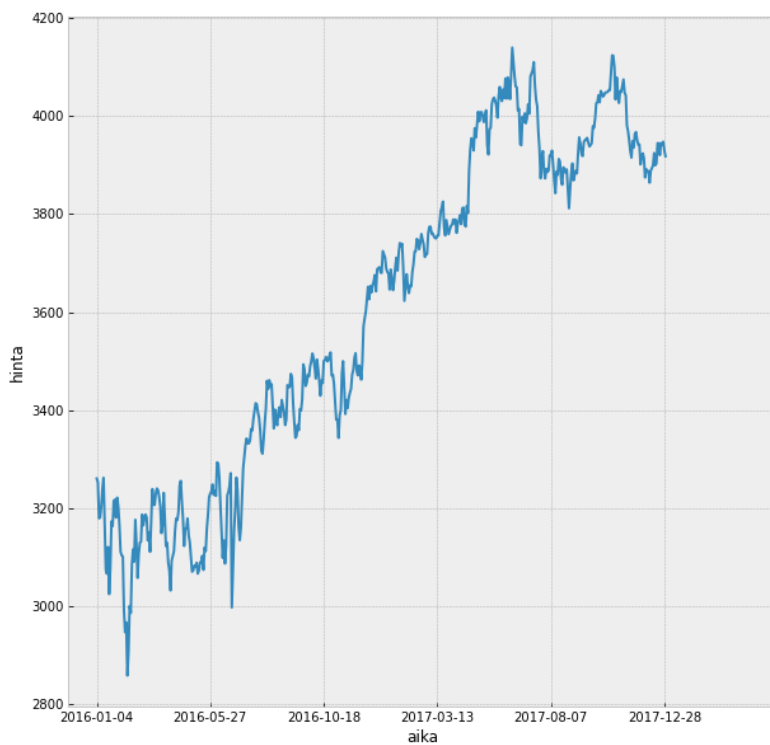


Kuva 12. Päivän päätöshinta Aikasarja 1

Kuvassa 13 on esitetty Aikasarjan 2 logaritmistien tuottojen kuvaaja ja Kuvassa 13 kuvaaja indeksin kehityksestä koko aikaväliltä. Aikasarja 2 on tässä tutkimuksessa olevista aikasarjoista kaikista lyhin. Aikasarjalle osuu markkinoilla oleva vahva nousutrendi. Indeksiin osuu kuitenkin vahvojen ylöspäin menevien liikkeiden lisäksi myös vahvoja korjauksia alaspäin. Logaritmisista tuotoista nähdään, että aikasarjan alkupäähän on osunut suurempaa vaihtelua, mutta loppua kohden logaritmistien tuottojen vaihtelu tasaantuu.



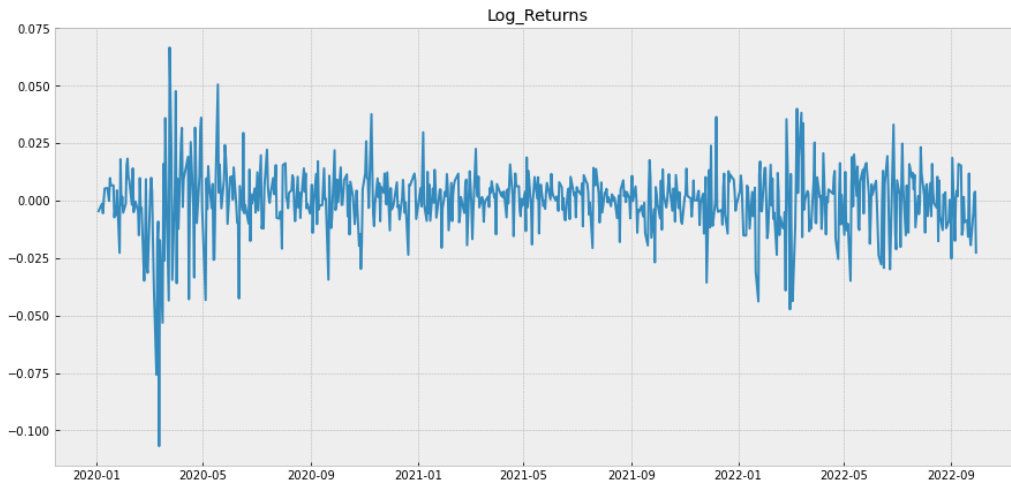
Kuva 13. Logaritmiset tuotot Aikasarja 2



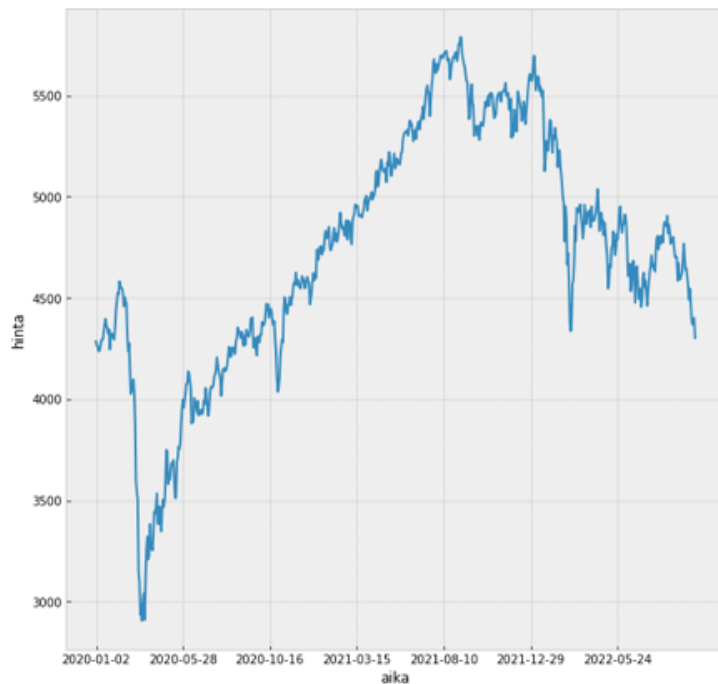
Kuva 14. Päivän päätöshinta Aikasarja 2

Kuvassa 15 on esitetty kuvaaja Aikasarjan 3 logaritmisista tuotoista ja kuvassa 16 on puolestaan esitetty kuvaaja indeksin kehityksestä kyseisellä aikavälillä. Aikasarjalle osuu todella vahvoja liikkeitä sekä ylöspäin että alaspäin. Ensimmäiseksi kuvasta voidaan nähdä koronaviruksesta johtuva historiallisen kova lasku alaspäin. Tätä seurasi vielä vahvempi

nousumarkkina. Lopulta nousu kuitenkin hidastui ja se joutui uuteen kovaan laskuun Venäjän hyökkäyssodan alkamisen jälkeen. Indeksien vahvat liikkeet nähdään myös logaritmisista tuotoista. Suurimmat vaihtelut osuvat kuitenkin erityisesti aikasarjan alkuun ja loppuun.



Kuva 15. Logaritmiset tuotot Aikasarja 3



Kuva 16. Päivän päätöshinta Aikasarja 3

Kun tutkimuksen aineisto on kerätty, se esikäsitellään. Pitkäkestoinen työmuisti on herkkä mittakaavan vaihteluille. Jotta alkuperäisen aineiston erilaiset vaihteluvälit eivät

aiheuttaisi ongelmia neuroverkon kouluttamiselle, normalisoidaan käytettävä aineisto välille [0, 1]. Tässä tutkielmassa aineiston normalisointiin käytetään niin kutsuttua Min-MaxScaler-funktiota, joka löytyy Pythonille tarkoitetusta Scikit-learn-kirjastosta (Aaryan ja muut, 2022; Scikit-learn, n.d.). Matemaattisesti funktio on ilmaistu seuraavalla tavalla:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (13)$$

Jossa x' kuvaa normalisoitua arvoa, x alkuperäistä solun arvoa, $\max(x)$ ja $\min(x)$ sarakkeen suurinta ja pienintä arvoa (Scikit-learn, n.d.).

Kun tutkimuksessa käytettävä aineisto on normalisoitu, jaetaan se koulutus-, validointi sekä testausaineistoon. Tässä tutkimuksessa aineistot tullaan jakamaan siten, että ensimmäiset 80 % aineistosta käytetään neuroverkon kouluttamiseen sekä validointiin ja loput 20 % aineistosta käytetään varsinaiseen testaamiseen. Kuten alaluvussa 2.2.8. käsiteltiin, datan jakaminen oikeankokoisiin aineistoihin on tärkeää, jotta neuroverkko kykenee oppimaan ja yleistämään dataa kuitenkin ylisovittamatta sitä varsinaiseen koulutusaineistoon. Taulukossa 3 on esitetty kaikkien aikasarja-aineiston havaintojen määrä sekä jakautuminen koulutus- ja testausaineistoihin.

Taulukko 3. Aineiston jako koulutus- ja testausaineistoon aikasarjoittain

Aikasarja	Koulutusaineisto	Testausaineisto	Havainnot yht.
Aikasarja 1: 1.1.2014-31.12.2019	1207	291	1498
Aikasarja 2: 1.1.2016-31.12.2017	411	91	502
Aikasarja 3: 1.1.2020-30.9.2022	562	128	670

Kun aineisto on normalisoitu ja jaettu koulutus- ja testausaineistoon, voidaan sitä hyödyntää mallien kouluttamisessa ja varsinaisen ennusteen tekemisessä.

3.2 Tutkimusmenetelmät

Tutkimusta varten rakennettiin neljä erilaista pitkäkestoista työmuistia, jotka poikkesivat toisistaan neuroverkon kerrosten tai koulutuksessa käytettävien epookkien määrällä. Tutkimuksessa käytettävän neuroverkon ohjelmakoodi löytyy kokonaisuudessaan Liitteestä 1. Lähteenä koodille on käytetty the PythonCoden verkkotutoriaalia (Rockikz, 2022) sekä sivustoa *Introduction to data analytics in accounting* (Ranta, 2020).

Kaikki neuroverkkomallit ennustivat aina seuraavan päivän päätöshintaa liukuvasti 50 edellisen päivän perusteella. Kahdessa ensimmäisessä neuroverkkomallissa, LSTM 1:ssä ja LSTM 2:ssa, käytettiin 200 epookkia koulutukseen. Ensimmäisessä mallissa käytettiin kahta piilokerrosta ja toisessa mallissa viittä piilokerrosta. Kahdessa seuraavassa neuroverkkomallissa, LSTM 3:ssa ja LSTM 4:ssä, käytetään 500 epookkia koulutukseen. Neuroverkossa kolme käytetään kahta piilokerrosta ja neuroverkossa neljä käytetään viittä piilokerrosta. Muilta osin neuroverkkojen parametrit ovat yhteneväiset.

Taulukko 4. Neuroverkkojen hyperparametrit

Parametrit	LSTM 1	LSTM 2	LSTM 3	LSTM 4
Piilokerrokset	2	5	2	5
Neuronit	200	200	200	200
Epookit	200	200	500	500
Kustannusfunk.	MSE	MSE	MSE	MSE
Optimointifunk.	Adam	Adam	Adam	Adam
Aktivaatiofunk.	ReLU	ReLU	ReLU	ReLU
Pudotus-%	20	20	20	20

Taulukossa 4 on esitetty neuroverkkomallien käyttämät parametrit. Kussakin neuroverkkomallissa tullaan käyttämään 200 neuronina. Koulutuksessa kustannusfunktiona tullaan käyttämään keskineliövirhettä. Aktivaatiofunktiona neuroneissa tullaan käyttämään ReLUa ja optimointifunktiona koulutuksessa Adamia.

Lisäksi koulutuksessa tullaan käyttämään niin sanottua *pudotusmenetelmää* (drop-out technique). Sen tarkoituksena on pyrkiä ehkäisemään ylisovittamista koulutuksen aikana. Tässä tekniikassa koulutuksen aikana neuroverkon kerroksesta valitaan satunnaisesti neuroneita, joiden lähtöarvoksi annetaan arvo 0 (Aaryan ja muut, 2022; Srivastava ja muut, 2014). Tällöin neuroni ei aktivoidu ja siitä ei lähde signaalia eteenpäin seuraavalle neuronille. Tässä tutkimuksessa käytetään kaikilla neuroverkoilla pudotusprosenttia 20. Tämä tarkoittaa käytännössä sitä, että todennäköisyys yksittäiselle neuronille tulla valituksi ja saada lähtöarvoksi nolla on 0,2.

Kaikissa neuroverkoissa käytetään koulutuksessa myös niin kutsuttua bidirektionaalista neuroverkkoa (bidirectional neural network, BRNN). Tämä muuttaa käytetyn neuroverkon rakennetta hieman. Normaalisti neuroverkoissa, kuten LSTM:ssä, informaatio neuronista siirtyy aina seuraavalle muistisolulle. Eli tieto siirtyy menneisyydestä kohti tulevaa, kuten esimerkiksi Kuvassa 9 on esitetty. BRNN ansiosta tietoa muistisolusta voi siirtyä kahteen suuntaan (Salehinejad ja muut, 2017). Neuronin saa käyttöönsä siis informaatiota sekä menneisyydestä että tulevaisuudesta samanaikaisesti. Tämän ansiosta neuroverkko saa tietoa tulevaisuudesta jatkuvasti, eikä sen tarvitse ainoastaan odottaa, että takaisinkytkentävaiheessa tietoa tulee takaisin. Tämä tehostaa neuroverkon oppimista. Tämä ei kuitenkaan vaikuta neuroverkon kouluttamiseen, vaan sen toimii samalla lailla kuin teoriaosuudessa on esitetty.

Neuroverkkomallien rakennus tapahtuu käyttämällä TensorFlow ja Keras kirjastoja. TensorFlow on avoimen lähdekoodin ohjelmistokirjasto, joka on kehitetty koneoppimiseen ja tekoälysovelluksiin (TensorFlow, 2023). Sen avulla luodaan puitteet neuroverkon rakentamiselle. Keras on puolestaan taas korkean tason ohjelmointirajapinta, jota käytetään tässä tutkielmassa rajapintana TensorFlow-kirjastolle (Keras, 2023). Keras tarjoaa yksinkertaisen käyttöliittymän neuroverkkojen rakentamiseen ja kouluttamiseen. Kerasin avulla pystytään luomaan syväoppimismalleja, jotka tukevat erilaisia aktivaatiofunktioita ja se sisältää myös apuohjelmia mallin arviointia varten (Keras, 2023). Aineiston analysointiin ja esikäsittelyyn käytetään Pandas ja Scikit-learn kirjastoja, sillä molemmat

tarjoavat tehokkaat ja yksinkertaiset välineet datan käsittelyyn. Lopuksi datan visualisointiin hyödynnetään Matplotlib-kirjastoa.

Tässä tutkimuksessa neuroverkkomallien vertailumenetelmänä tullaan käyttämään autoregressiivistä (AR)-mallia. AR-malli on yleisesti käytetty aikasarja-analyysissä ja neuroverkon tavoin sen ennustukset pohjautuvat aikaisemmin tehtyihin havaintoihin. Matemaattisesti sitä voidaan kuvata lineaarisena regressiomallina, jossa aikaisemmin saadut havainnot toimivat sen selittäjinä (Dickey & Fuller, 1979).

AR-mallin matemaattinen määritelmä on seuraava:

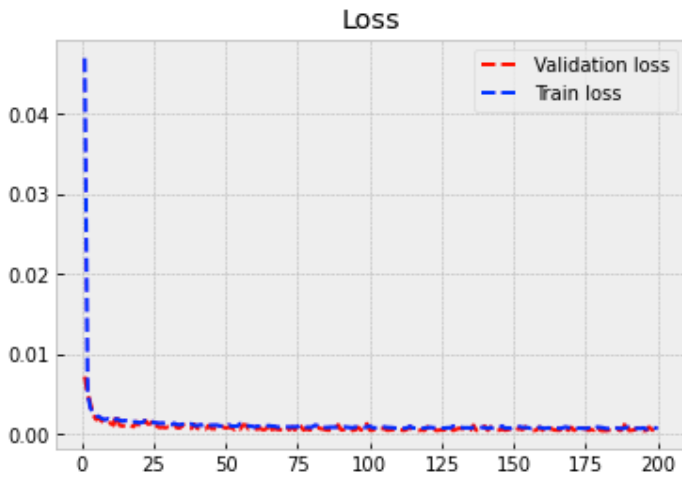
$$y_t = c + \Phi_1 y_{t-1} + \Phi_2 y_{t-2} + \dots + \Phi_p y_{t-p} + \epsilon_t \quad (14)$$

Jossa c on vakiotermi, Φ kuvaava regressiokertoimia, y ovat viivefunktioita, p on kertaluku, t kuvaa tarkasteltavaa ajanjaksoa väliltä $[1, T]$ ja ϵ on virhetermi (Dickey & Fuller, 1979). Tämä virhetermi ϵ kuvaa niin kutsuttua valkoista kohinaa, jonka avulla määritellään satunnaisten arvojen riippumattomuutta toisistaan. Kertaluvulla kuvataan taas sitä, kuinka pitkälle taaksepäin ajassa malli pohjautuu. Tässä tutkielmassa tullaan käyttämään autoregressiivistä mallia, joka pohjautuu vain yhden ajanjakson päähän eli niin sanottua AR(1)-mallia.

Autoregressiivisen mallin luontiin käytettiin myös Pythonin eri ohjelmakirjastoja. Datan esikäsittelyyn käytettiin Pandas ja Scikit-learn kirjastoja. Jälkimmäistä kirjastoa käytettiin myös mallin suorituskykymittauksiin. Varsinainen AR-malli luotiin käyttämällä Statsmodels-kirjastoa. Datan visualisointiin hyödynnettiin Matplotlib-kirjastoa.

Kun mallit on saatu rakennettua, aloitetaan kaikkien neuroverkkojen kouluttaminen kaikille kolmelle aikasarjalle. Tämä tarkoittaa käytännössä 12 eri neuroverkon kouluttamista. Neuroverkoille annetaan siis 80 % esikäsitellystä datasta, jonka avulla ne oppivat kierros

kierrokselta. Neuroverkkojen koulutusta seurataan laskentakierrosten edetessä (Kuva 15).



Kuva 17. Neuroverkon häviö koulutuksen aikana

Kuvassa 17 nähdään, kuinka neuroverkon häviö pienenee eli kustannusfunktion arvo laskee koulutuskierrosten edetessä. Kuvaajan avulla nähdään, kuinka neuroverkko pystyy käyttämään oppimaansa validointiaineiston kanssa. Tämä mahdollistaa neuroverkon koulutuksen siten, että neuroverkolle mahdollisesti tapahtuvaa yli- tai alisovittamista voidaan tarvittaessa kontrolloida.

Neuroverkkomallit ovat rakennettu siten, että koulutuksen aikana neuronien arvot, joiden pohjalta virhe minimoituu, tallennetaan. Lopulta arvot, joilla virhe minimoituu parhaiten, valitaan automaattisesti käytettäväksi varsinaiseen ennustamiseen.

3.3 Tulokset ja analysointi

Tässä kappaleessa esitetään ennusteiden tulokset kullekin aikasarjalle. Kussakin kappaleessa esitetään tulokset, sekä graafiset esitykset autoregressiivisestä mallista sekä neuroverkkomalleista. Matemaattiset tulosten tarkasteluun käytetään keskineliövirhettä. Tuloksissa lasketaan keskineliövirhe todellisen ja ennustetun indeksin arvon väliltä.

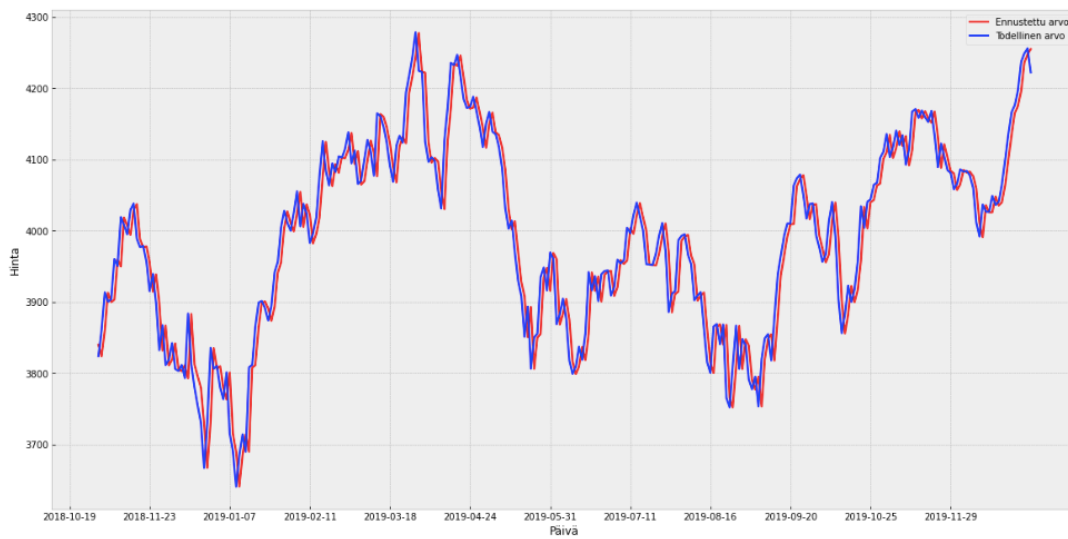
Keskineliövirheen matemaattinen tulkinta on esitetty alaluvussa 2.2.4. Virheet on laskettu normalisoidusta aineistosta.

Aikasarjan 1 tulokset on esitetty kokonaisuudessaan Taulukossa 5. Tuloksista nähdään, että paras ennustekyky oli malleilla LSTM 1 sekä LSTM 3, joiden keskineliövirhe oli sama. Näiden jälkeen paras ennustekyky oli autoregressiivisellä mallilla. Toiseksi huonoin ennustekyky oli mallilla LSTM 4 ja heikoin ennustekyky oli mallilla LSTM 2. Kuitenkin käytännössä mallien ennustekyvyyssä ei ollut suuria eroja toistensa välillä.

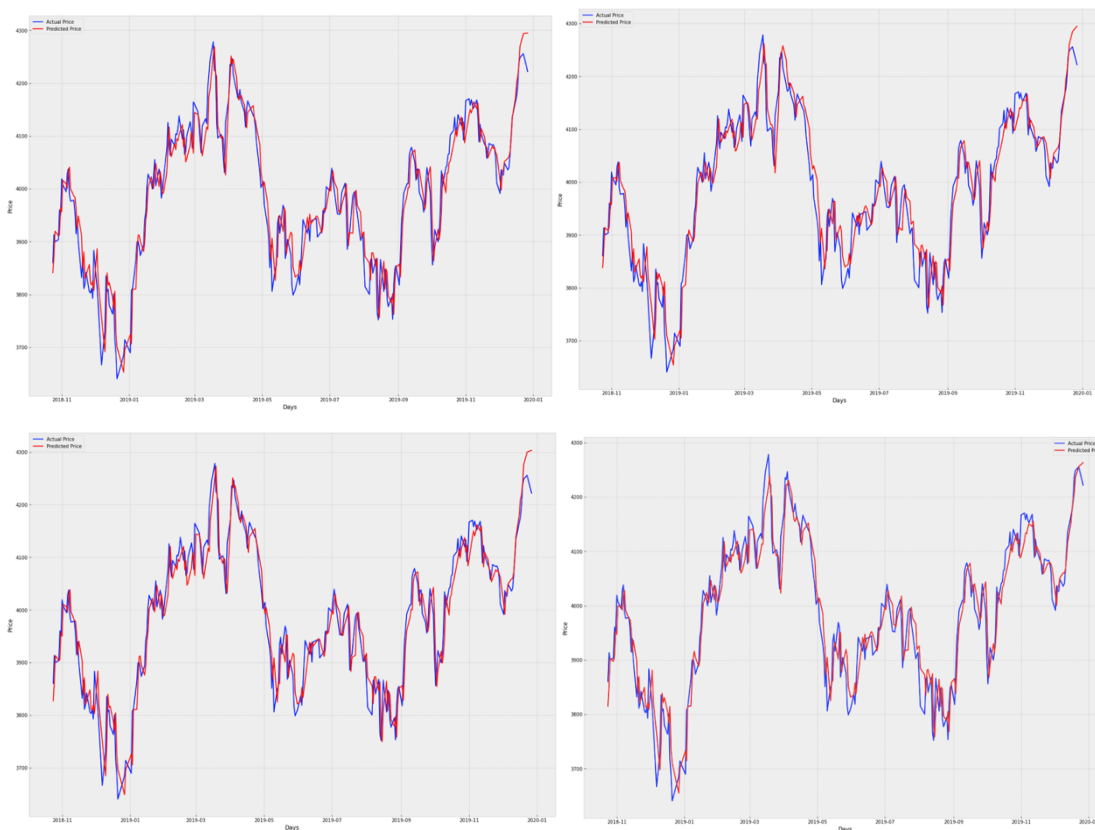
Taulukko 5. Aikasarjan 1: 1.1.2014-31.12.2019 tulokset

Malli	MSE
AR	0,0005060
LSTM 1	0,0004877
LSTM 2	0,0005313
LSTM 3	0,0004877
LSTM 4	0,0005075

Kuvassa 18 on esitetty kuvaaja AR(1)-mallin ennusteista ja kuvassa 19 puolestaan kuvaajat neuroverkkomallien ennusteista. Mallit on esitetty järjestyessä LSTM 1, LSTM 2, LSTM 3 sekä LSTM 4. Todelliset arvot kuvaajissa on esitetty sinisellä ja mallin ennuste punaisella. Koska käytössä on AR(1)-malli, tekee se päivän päätöskurssin ennustamisen edellisen päivän todellisen arvon mukaan, joka voidaan huomata AR(1)-mallin kuvaajasta. Tästä syystä se toimii myös hyvänä vertailukohtana neuroverkkomalleille. Neuroverkkojen kuvaajista nähdään, että kaikki mallit kykenivät melko hyvin ennustamaan yleistä trendiä indeksillä, mutta haasteita kaikilla malleilla oli erityisesti pohjien sekä huippujen tunnistamisessa.



Kuva 18. AR-malli ennusteet Aikasarjalla 1



Kuva 19. Neuroverkkojen ennusteet Aikasarjalla 1

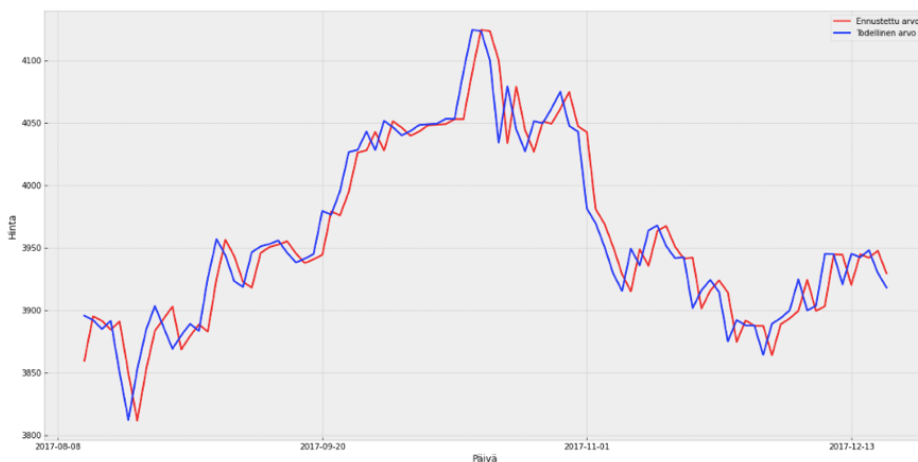
Aikasarjan 2 tulokset on esitetty kokonaisuudessaan Taulukossa 6. Kyseinen aikasarja oli kaikista lyhin aikasarja tutkimuksissa olevista aikasarjoista, mikä tarkoitti neuroverkoilla

olevan käytössä kaikista vähiten koulutusdataa. Tämä voidaan havaita selkeästi myös tuloksista, joissa autoregressiivinen mallin ennustekyky oli selkeästi neuroverkkomalleja parempi. Parhaiten neuroverkoista pärjäs LSTM 3. Selvästi huonoiten pärjäs jälleen malleistä LSTM 2, jonka virhe oli selkeästi suurempi parhaiten pärjänneeseen malliin nähden.

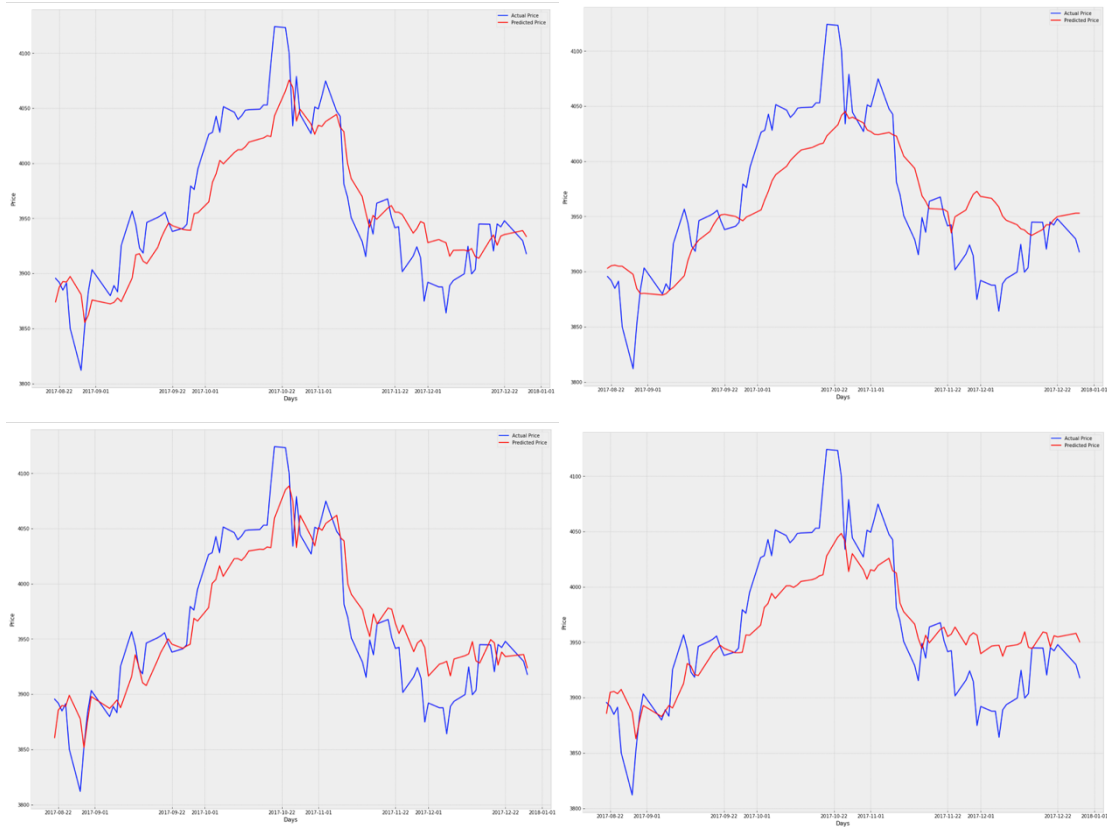
Taulukko 6. Aikasarjan 2: 1.1.2016-31.12.2017 tulokset

Malli	MSE
AR	0,0003143
LSTM1	0,0006839
LSTM2	0,0011027
LSTM3	0,0005546
LSTM4	0,0008314

Kuvassa 20 on esitetty kuvaaja AR(1)-mallin ennusteista ja kuvassa 21 puolestaan kuvaajat neuroverkkomallien ennusteista. Kuvaajista voidaan havaita samat tulokset kuin keskineliövirheistä. Neuroverkkomallien ennuste oli selvästi heikompi autoregressiiviseen malliin nähden. Kuvaajista huomataan kaikkien neuroverkkomallien ennusteiden olevan heikkoja ja selviä eroja voidaan nähdä myös mallien välillä. LSTM 3 kykeni hieman tunnistamaan indeksin liikkeitä, mutta LSTM 2 mallin ennustekyky oli todella heikko.



Kuva 20. AR-mallin ennusteet Aikasarjalla 2



Kuva 21 Neuroverkkojen ennusteet Aikasarjalla 2

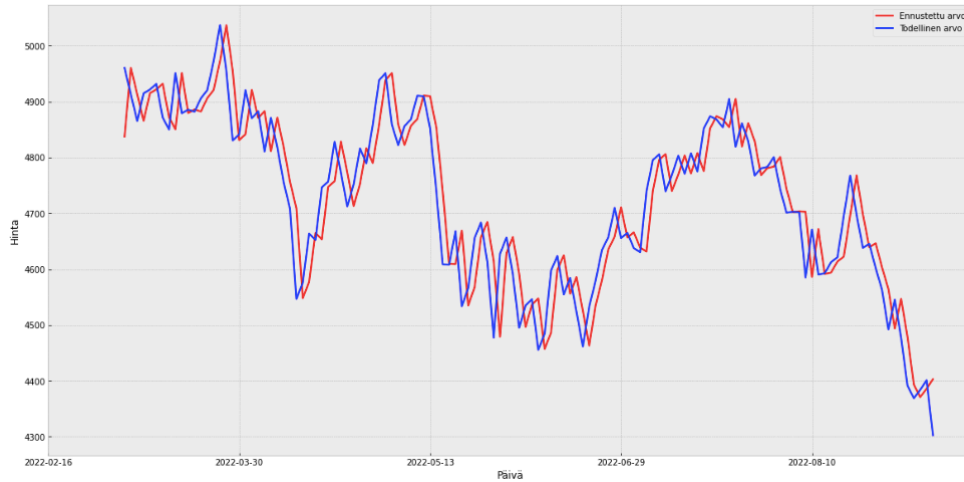
Aikasarjan 3 tulokset on esitetty Taulukossa 7. Kyseisellä aikasarjalla LSTM 3 oli ennustekyvyltään malleista paras. Lähes samoihin tuloksiin pääsi myös autoregressiivinen malli. Toiseksi paras ennustekyky neuroverkoista oli mallilla LSTM 4, kolmanneksi huonoin mallilla LSTM 1 ja heikoin ennustekyky mallilla LSTM 2.

Taulukko 7. Aikasarjan 3: 1.1.2020-30.9.2022 tulokset

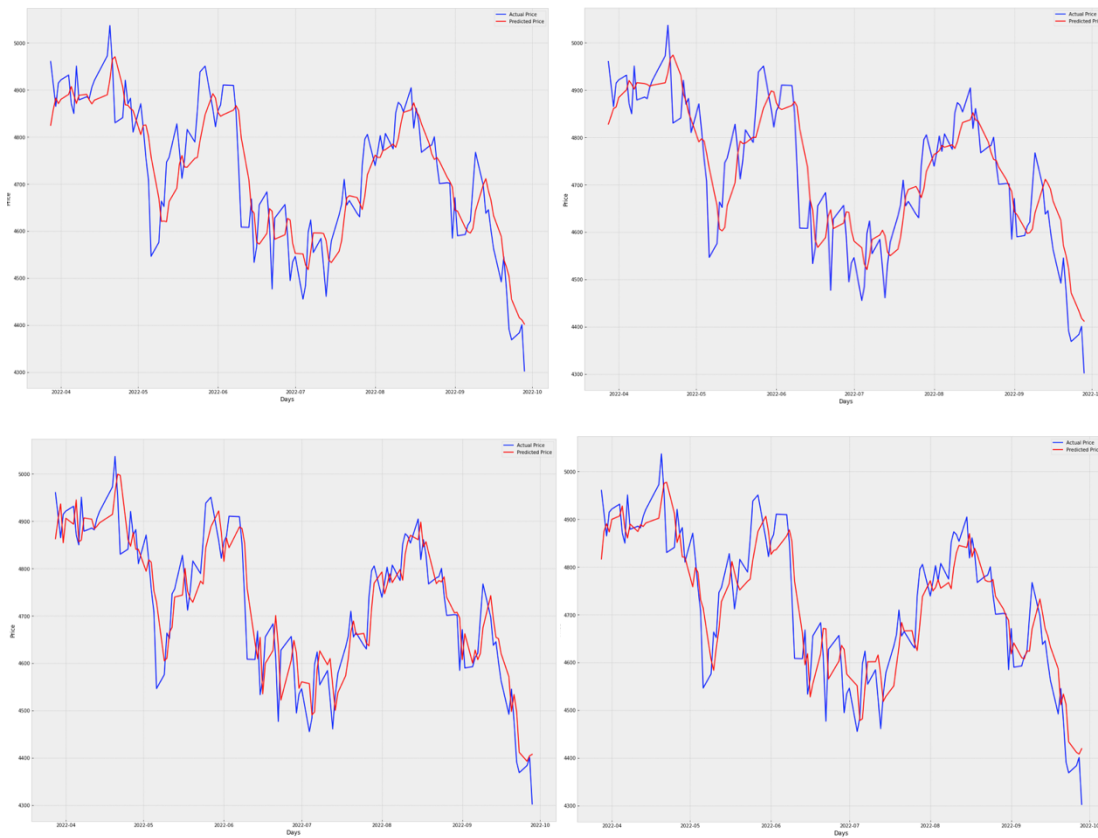
Malli	MSE
AR	0,0005495
LSTM 1	0,0006276
LSTM 2	0,0007321
LSTM 3	0,0005337
LSTM 4	0,0006178

Kuvassa 22 on esitetty kuvaaja AR(1)-mallin ennusteista ja kuvassa 23 puolestaan kuvaajat neuroverkkomallien ennusteista. LSTM 3 ja LSTM 4 mallit kykenivät melko hyvin

seuraamaan indeksin trendiä, mutta malleilla oli vaikeaa tunnistaa aikasarjalta huippuja ja pohjia. Mallien LSTM 1 ja LSTM 2 ennustekyky oli edellisiä malleja heikompi, ja niiden kyky tunnistaa markkinoiden liikkeitä oli verrattain huono.



Kuva 22. AR-mallin ennusteet Aikasarjalla 3



Kuva 23 Neuroverkkojen ennusteet Aikasarjalla 3

3.3.1 Analyysi

Neuroverkkomallien sekä autoregressiivisen mallin ennustekykyä testattiin kolmella eri aikasarjalla, ja tulosten pohjalta on tarkoitus vastata tutkimuskysymyksiin. Tutkimustulokset on koottu kokonaisuutena Taulukkoon 8. Ensimmäisenä tutkimuskysymyksenä oli selvittää neuroverkon ennustekyky eri aikasarjoilla sekä verrata tuloksia autoregressiiviseen malliin. Tuloksista voitiin havaita, että aikasarjojen välillä oli eroja neuroverkon ennustekyvyyssä. Ensimmäisessä aikasarjassa, aikaväliltä 1.1.2014-1.1.2019, neuroverkko-mallit LSTM 1 sekä LSTM 3 kykenivät verrokkimalliaan parempaan ennustekyvyyteen. Kaikki neuroverkot kykenivät havaitsemaan trendejä ja neuroverkkojen ennusteet olivat oikean suuntaisia.

Toisen aikasarjan kohdalla, aikaväliltä 1.1.2016-1.1.2017, taas autoregressiivinen malli oli selkeästi parempi kaikkiin neuroverkkomalleihin nähden. Ennustekyky jäi kaikkien neuroverkkojen osalta huonoksi. LSTM 3 -malli kykeni jollain tasolla tunnistamaan trendiä, mutta erityisesti mallit LSTM 2 ja LSTM 4 olivat hyvin heikkoja ennustekyvyltään kyseisellä aikasarjalla. Kolmannen aikasarjan kohdalla, aikaväliltä 1.1.2016-1.1.2017, malli LSTM 3 oli verrokkimalliaan parempi, mutta ero oli todella marginaalinen. Muut neuroverkkomallit hävisivät vertailumallille. Malleilla LSTM 3 sekä LSTM 4 oli kykyä seurata indeksin liikkeitä, mutta niilläkin ennustekyky huippujen sekä pohjien suhteen oli heikko.

Tulosten pohjalta nähdään, että neuroverkolla oli kykyä voittaa vertailumalli ennustekyvyyssä, kun aikaväli oli pidempi. Esimerkiksi ensimmäisen aikasarjan kohdalla, joka oli pisin aikasarja, kaksi neuroverkkomallia kykeni voittamaan verrokkimallin. Viimeisen aikasarjan kohdalla, joka oli toiseksi pisin aikasarja, kaksi neuroverkkomallia kykeni parempaan ennustekyvyyteen verrokkimalliin nähden. Toisen aikasarjan kohdalla, mikä oli kaikista lyhyin aikasarja, kaikki neuroverkkomallit jäivät selvästi vertailumallin ennustekyvyltä. Koulutusaineiston koolla on siis selvä vaikutus neuroverkon ennustekyvyyteen ja johtopäätöksenä voidaan sanoa tutkimuksessa olevan kahden vuoden tarkastelujakson olevan liian lyhyt neuroverkkojen tehokkaaseen käyttöön. On kuitenkin hyvä nostaa esille se, että vaihtelu ennustekyvyydestä saattaa johtua myös autoregressiivisen mallin heikosta

toiminnasta kuin neuroverkkomallien ylivertaisuudesta. Esimerkiksi Aikasarja 3 oli hyvin volatiili, joka on erityisen huono autoregressiiviselle mallille. Ensimmäisen tutkimuskysymyksen tulokset avaavat mielenkiintoisen mahdollisuuden jatkaa tutkimuksia neuroverkojen käytöstä vieläkin pidemmällä aikasarjoilla.

Taulukko 8. Tutkimuksen tulokset kootusti aikasarjoittain

Aikasarja	1.1.2014- 31.12.2019	1.1.2016- 31.12.2017	1.1.2020- 30.9.2022
Aineisto (koulutu/testaus)	(1207/291)	(411/91)	(562/128)
Malli / MSE			
AR(1)	0,0005060	0,0003143	0,0005495
LSTM 1	0,0004877	0,0006839	0,0006276
LSTM 2	0,0005313	0,0011027	0,0007321
LSTM 3	0,0004877	0,0005546	0,0005337
LSTM 4	0,0005075	0,0008314	0,0006178

Toisena tutkimuskysymyksenä oli selvittää, onko tutkimuksessa käytettävien neuroverkkomallien välillä eroa ennustekyvyyssä. Tulosten perusteella voidaan sanoa neuroverkkomallien ennustekyvyyden välillä olevan eroja. Kaikissa aikasarjoissa paras ennustekyky oli mallilla LSTM 3, joka sisälsi kaksi piilokerrosta ja sen koulutukseen käytettiin 500 epookkia. Heikoin malli kaikilla aikasarjoilla oli sen sijaan LSTM 2, joka sisälsi viisi piilokerrosta ja sen koulutukseen käytettiin 200 epookkia. Huomioitavaa on myös se, että koulutusaineiston pienentyessä neuroverkkomallien väliset erot tulivat paremmin esiin. Mielenkiintoista on myös nostaa esille se, että kahden ensimmäisen aikasarjan kohdalla parhaimmat tulokset olivat kaksi piilokerrosta sisältävää neuroverkkomallia. Huonommat tulokset olivat sen sijaan viisi piilokerrosta olevilla neuroverkoilla riippumatta koulutukseen käytettävien epookkien määrästä. Sen sijaan viimeisen aikasarjan kohdalla, mikä oli myös kaikista volatiilein, mallit, joissa koulutukseen käytettiin 500 epookkia, pärjäsivät 200 epookkia käyttäviä malleja paremmin, riippumatta piilokerrosten määrästä. Johtopäätöksenä voidaan sanoa, että neuroverkkomallien väliset erot pienenevät, mitä

enemmän aineistoa oli käytössä. On kuitenkin nostettava esille, että tutkimuksessa käytettävät mallit sisälsivät eroja vain epookkien sekä piilokerrosten määrässä. Uusissa tutkimuksissa neuroverkkojen ennustekykyä voidaan pyrkiä parantamaan testaamalla useampia neuroverkon hyperparametrejä, kuten aktivaatiefunktiota tai optimointifunktiota.

Tutkimuksen kolmantena kysymyksenä oli selvittää, voiko sijoittaja hyödyntää neuroverkkoja omassa päätöksenteossaan. Tutkimustulosten pohjalta voidaan vastata, ettei neuroverkkojen antamista ennusteista ole kovinkaan paljon hyötyä lyhyen aikavälin kauppoissa. Kuvista 19, 21 sekä 23 huomataan, että neuroverkkojen tekemien ennusteiden ja indeksin todellisten arvojen välillä on käytännössä suuri ero, joten sijoituspäätösten teko ei ole kovinkaan järkevää. Lisäksi tulokset näyttävät, että neuroverkoilla oli vaikeuksia havaita huippuja sekä erityisesti pohjia markkinoilla. Lisäksi tutkimustulokset näyttävät, että neuroverkon ennustekyvillä on vaihtelua eri aikasarjoilla. Tämä tuo epävarmuutta niiden käytännön hyödyntämiseen, koska markkinatilanteet vaihtelevat, jolloin neuroverkon ennustekyky saattaa olla poikkeuksellisen huono. Kuitenkin osassa aikasarja-aineistoissa neuroverkot kykenivät havaitsemaan trendejä. Tutkimuksessa neuroverkoilla tehtiin ennustuksia yhden päivän päähän 50 edellisen päivän perusteella. Tulevissa tutkimuksissa neuroverkon ennustekykyä voisi testata pidemmällä aikavälillä ja seurata, pystyykö neuroverkko järkevästi ennustamaan trendejä useamman päivän päähän.

4 Johtopäätökset

Tämän tutkielman tarkoituksena oli selvittää erilaisten neuroverkkomallien ennustekykä eri mittaisilla aikasarjoilla ja vastata myös kysymykseen, pystyykö sijoittaja hyödyntämään neuroverkon ennustekykä päätöksenteon tukena. Tutkimuksen pohjalta voitiin vetää johtopäätös siitä, että neuroverkkojen ennustekykä oli parempi aikasarja-aineiston pidentyessä. Lisäksi tulokset osoittivat, että tutkimuksessa käytettävien neuroverkkomallien ennustekyvillä oli eroja keskenään. Erityisesti erot kasvoivat aikasarja-aineistojen lyhentyessä. Tulosten pohjalta tehtiin myös johtopäätös, ettei neuroverkon käytöstä ole sijoittajalle hyötyä ainakaan lyhyellä aikavälillä. Neuroverkot kykenivät kuitenkin seuraamaan joissakin tapauksissa indeksin trendejä.

Tutkielman tulokset ja rajaukset antavat mahdollisuuden tutkia neuroverkkojen hyödyntämistä osakemarkkinoiden toiminnan ymmärtämisessä jatkossa myös syvemmin. Tulevissa tutkimuksissa neuroverkkojen hyödyntämää dataa voisi laajentaa myös indeksien omien historiallisten tuottojen lisäksi muihin taloudellisiin aineistoihin. Neuroverkon ennustekykä voi myös lähteä parantamaan testaamalla laajemmin neuroverkkojen erilaisia parametreja, sillä tämän tutkielma rajoitti muutoksia vain koulutuksen määrään ja neuroverkon rakenteen suhteen. Vaihtoehtona on myös pyrkiä ennustamaan pidemmän aikavälin trendejä yhden päivän hintakehityksen sijaan. Lisäksi neuroverkon analysointikykyä voidaan hyödyntää myös tutkimalla esimerkiksi salkunhallinnan tai riskienhallinnan näkökulmia, joista molemmista on saatu positiivisia tuloksia aikaisemmassa tutkimuksessa (Khodayari ja muut, 2020; Ko & Lin, 2008).

Lisäksi varmaa on se, että koneoppimismenetelmät kehittyvät jatkuvasti, minkä johdosta niiden analysointikyky tulee paranemaan entisestään. Uutta tekoälyteknologiaa otetaan jatkuvasti enemmän käyttöön yrityksissä ja nämä tekniset ratkaisut valtaavat entisestään alaa myös rahoitusmarkkinoiden piirissä (AlphaSense, 2023). Tästä johtuen uusien menetelmien tutkiminen on jatkossa entistä tärkeämpää, sillä niiden mahdollisuuksien ymmärtäminen avaa yhä paremmat edellytykset tutkijoille sekä asiantuntijoilla havainnoimaan rahoitusmarkkinoiden luonnetta.

Lähteet

- Aaryan, Kanisha, B., Jayaraman, V. & Kanisha, B. 2022. Forecasting stock market price using LSTM-RNN. *IEEE Xplore*, konferenssijulkaisu. doi: 10.1109/ICA-CITE53722.2022.9823818
- AlhpaSense. (2023). *Hi, we're AlphaSense*. Built In. Noudettu 26.3.2023 osoitteesta <https://builtin.com/company/alphasense>
- Baheti, P. (2023, 2. helmikuu). *Activation Functions in Neural Networks [12 Types & Use Cases]*. V7. Noudettu 13.2.2023 osoitteesta <https://www.v7labs.com/blog/neural-networks-activation-functions>
- Balsara, N. J., Zheng, L., Vidozzi, A. & Vidozzi, L. (2006). *Explaining momentum profits with an epidemic diffusion model*. *Journal of economics and finance*, 30(3), 407-422. doi:10.1007/BF02752744
- Bao, W., Yue, J. & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLoS ONE* 12(7). <https://doi.org/10.1371/journal.pone.0180944>
- Bhandari, H, Rimal, B, Pokhrel, N, Rimal, R, Dahal, K & Khatri, R. (2022). Predicting stock market index using LSTM. *Machine Learning with Applications* (9). <https://doi-org.proxy.uwasa.fi/10.1016/j.mlwa.2022.100320>
- Brownlee, J. (2017, 3. heinäkuu). *Gentle Introduction to the Adam Optimization Algorithm for Deep Learning*. Machine Learning Mastery. Noudettu 13.2.2023 osoitteesta <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>
- Cont, R. 2001. Empirical properties of asset returns: Stylized facts and statistical issues. *Quantitative finance* 1(2), 223-236. <https://doi-org.proxy.uwasa.fi/10.1080/713665670>
- Desai, J, Desai, K, Joshi, N, Juneja, A & Dave, A. (2011). Forecasting of Indian Stock Market Index S&P CNX Nifty 50 Using Artificial Intelligence. *Behavioral & Experimental Finance eJournal*, 3(79). Noudettu 18.2.2023 osoitteesta <https://ssrn.com/abstract=1842283>
- Dickey, D. A. & Fuller, W. A. (1979). Distribution of the Estimators for Autoregressive Time Series With a Unit Root. *Journal of the American Statistical Association*, 74(366), 427. <https://doi.org/10.2307/2286348>

- Dolphin, R. (2020, 21. elokuu). *LSTM Networks | A Detailed Explanation*. Towards Data Science. Noudettu 13.2.2023 osoitteesta <https://towardsdatascience.com/lstm-networks-a-detailed-explanation-8fae6aefc7f9>
- Dzikevičius, A. & Stabužytė, N. (2012). Forecasting OMX Vilnius stock index--a neural network approach. *Business: Theory and Practice* 13(4), 324–332. <https://doi.org/10.3846/btp.2012.34>
- Elements of AI. (2020). *Tekoälyn perusteet* (luku 5). Reaktor. Noudettu 13.2.2023 osoitteesta <https://course.elementsofai.com/fi/5>
- Fama, E. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. *The Journal of finance (New York)* 25(2), 383-417. <https://doi.org/10.2307/2325486>
- Fama, E. F. (1991). Efficient Capital Markets: II. *The Journal of finance (New York)*, 46(5), pp. 1575-1617. <https://doi.org/10.2307/2328565>
- Gao, L., Han, Y., Zhengzi Li, S. & Zhou, G. (2018). Market intraday momentum. *Journal of financial economics*, 129(2), pp. 394-414. <https://doi-org.proxy.uwasa.fi/10.1016/j.jfineco.2018.05.009>
- Geeksforgeeks. (2023, 19. tammikuu). *Deep Learning | Introduction to Long Short Term Memory*. GeeksforGeeks. Noudettu 13.2.2023 osoitteesta <https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/>
- Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Deep learning*. MIT Press. Noudettu 13.2.2023 osoitteesta <https://www.deeplearningbook.org>
- Grossman, S. & Stiglitz, J. (1980). On the Impossibility of Informationally Efficient Markets. *The American Economic Review* 70(3), 393-408. <https://www.jstor.org/stable/1805228>
- Grossman, S. (1976). ON THE EFFICIENCY OF COMPETITIVE STOCK MARKETS WHERE TRADES HAVE DIVERSE INFORMATION. *The Journal of Finance (New York)*, 31(2), 573-585. <https://doi.org/10.2307/2326627>
- Guresen, E., Kayakutlu, G. & Daim, T. U. (2011). Using artificial neural network models in stock market index prediction. *Expert systems with applications* 38(8), 10389-10397. <https://doi-org.proxy.uwasa.fi/10.1016/j.eswa.2011.02.068>
- Ham, H., Cho, H., Kim, H. & Ryu, D. (2019). Time-series momentum in China's commodity futures market. *The journal of futures markets* 39(12), 1515-1528. <https://doi-org.proxy.uwasa.fi/10.1002/fut.22053>

- He, X. & Li, K. (2015). Profitability of time series momentum. *Journal of banking & finance*, 53, 140-157. <https://doi-org.proxy.uwasa.fi/10.1016/j.jbankfin.2014.12.017>
- Huang, D., Li, J., Wang, L. & Zhou, G. (2020). Time series momentum: Is it there? *Journal of financial economics*, 135(3), 774-794. <https://doi-org.proxy.uwasa.fi/10.1016/j.jfineco.2019.08.004>
- Hurst, B., Ooi, Y. H. & Pedersen, L. H. (2017). A Century of Evidence on Trend-Following Investing. *Journal of portfolio management*, 44(1), 15-29. <https://doi.org/10.3905/jpm.2017.44.1.015>
- Jensen, M. (1978). Some Anomalous Evidence Regarding Market Efficiency. *Journal of financial economics* 6(2), 95–101. [https://doi.org/10.1016/0304-405X\(78\)90025-9](https://doi.org/10.1016/0304-405X(78)90025-9)
- Jin, M., Kearney, F., Li, Y. & Yang, Y. C. (2020). Intraday time-series momentum: Evidence from China. *The journal of futures markets*, 40(4), 632–650. <https://doi-org.proxy.uwasa.fi/10.1002/fut.22084>
- Jupyter Team. (2015). The Jupyter Notebook. *Jupyter Notebook*. Noudettu 22.2.2023 osoitteesta <https://jupyter-notebook.readthedocs.io/en/stable/notebook.html>
- Kananen, H. & Puolitaival, H. (2019). *TEKOÄLY - Bisneksen uudet työkalut*. Alma Talent Oy.
- Kenton, W. (2020, 25. elokuu). GARCH Process. *Investopedia*. Noudettu 12.2.2023 osoitteesta <https://www.investopedia.com/terms/g/generalizedautogressiveconditionaltheteroskedasticity.asp>
- Keras. (2023). *About Keras*. Keras. Noudettu 16.2.2023 osoitteesta <https://keras.io/about/>
- Khodayari, M., Yaghobnezhad, A. & Eraghi, K. (2020). A Neural-Network Approach to the Modeling of the Impact of Market Volatility on Investment. *Advances in mathematical finance & applications*, 5(4), 1-16. <https://doi.org/10.22034/amfa.2020.674953>
- Kim, A. Y., Tse, Y. & Wald, J. K. (2016). Time series momentum and volatility scaling. *Journal of financial markets (Amsterdam, Netherlands)*, 30, 103-124. <https://doi-org.proxy.uwasa.fi/10.1016/j.finmar.2016.05.003>
- Knüpfer, S. & Puttonen, V. (2018). *Moderni rahoitus* (10. painos). Alma Talent Oy.

- Ko, P. & Lin, P. (2008). Resource allocation neural network in portfolio selection. *Expert systems with applications*, 35(1), 330-337. <https://doi.org/10.1016/j.eswa.2007.07.031>
- Inc., M. & C., Koller, T., Goedhart, M. & Wessels, D. (2020). *Valuation: Measuring and Managing the Value of Companies*.
- Maciel, L & Ballini, R. (2010). Neural Networks Applied to Stock Market Forecasting: An Empirical Analysis. *Learning and Nonlinear Models (L&NLM) – Journal of the Brazilian Neural Network Society*, 8(1), 3-22. Noudettu 13.2.2023 osoitteesta https://www.researchgate.net/publication/307748108_Neural_Networks_Applied_to_Stock_Market_Forecasting_An_Empirical_Analysis
- Makridakis, S & Hibon, M. (2000). The M3-Competition: results, conclusions and implications. *International Journal of Forecasting* 16(4), 451-47. [https://doi.org/10.1016/S0169-2070\(00\)00057-1](https://doi.org/10.1016/S0169-2070(00)00057-1)
- Makridakis, S., Spiliotis, E. & Assimakopoulos, V. (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PLoS ONE* 13(3). <https://doi.org/10.1371/journal.pone.0194889>
- Moghaddam, A., Moghaddam, M. & Esfandyari, M. (2016). Stock market index prediction using artificial neural network. *Journal of Economics, Finance and Administrative Science* 21(41), 89-93. <https://doi.org/10.1016/j.jefas.2016.07.002>
- Moskowitz, T. J., Ooi, Y. H. & Pedersen, L. H. (2012). Time series momentum. *Journal of financial economics*, 104(2), 228-250. <https://doi-org.proxy.uwasa.fi/10.1016/j.jfineco.2011.11.003>
- Mostafa, F., Dillon, T. & Chang, E. (2017). Time series modelling.
- Nasdaq. (2023, 13. helmikuu). *OMXH25*. Nasdaq, Inc. Noudettu 13.2.2023 osoitteesta <https://indexes.nasdaqomx.com/index/overview/omxh25>
- Novy-Marx, R. (2012). Is momentum really momentum? *Journal of financial economics*, 103(3), 429-453. <https://doi-org.proxy.uwasa.fi/10.1016/j.jfineco.2011.05.003>
- Pitkäjärvi, A., Suominen, M. & Vaittinen, L. (2020). Cross-asset signals and time series momentum. *Journal of financial economics*, 136(1), 63-85. <https://doi-org.proxy.uwasa.fi/10.1016/j.jfineco.2019.02.011>
- Ranta, M. (2020) *Introduction to data analytics in accounting*. Github. Noudettu 4.3.2023 osoitteesta https://mranta-ai.github.io/Data_analytics_in_accounting/0_installing_Python.html

- Ren, G, Cao, Y, Wen, S, Huang, T & Zeng, Z (2018). A modified Elman neural network with a new learning rate scheme. *Neurocomputing (Amsterdam)*, Volume 286, 11-18. <https://doi-org.proxy.uwasa.fi/10.1016/j.neucom.2018.01.046>
- Rockikz, A. (2022, toukokuu). *How to Predict Stock Prices in Python using TensorFlow 2 and Keras*. PythonCode. Noudettu 4.3.2023 osoitteesta https://www.thepythoncode.com/article/stock-price-prediction-in-python-using-tensorflow-2-and-keras#google_vignette
- Russell, S. J., Norvig, P., Chang, M., Devlin, J., Dragan, A., Forsyth, D., . . . Wooldridge, M. (2022). *Artificial intelligence: A modern approach* (4). Global edition. Pearson.
- Salehinejad, H., Sankar, S., Barfett, J., Colak, E. & Valaee, S. (2017). *Recent Advances in Recurrent Neural Networks*. Cornell University. <https://doi.org/10.48550/arXiv.1801.01078>
- Shleifer, A. (2000). *Inefficient markets: An introduction to behavioral finance*. Oxford University Press.
- Scikit-learn. (n.d.). *sklearn.preprocessing.MinMaxScaler*. Scikit-learn.org. Noudettu 15.2.2023 osoitteesta <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html>
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15, 1929-1958. Noudettu 16.2.2023 osoitteesta <https://www.jmlr.org/papers/v15/>
- Stuhlsatz, A., Meyer, C., Eyben, F., Zielke, T., Meier, G. & Schuller, B. (2011). Deep neural networks for acoustic emotion recognition: Raising the benchmarks. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 4577-5691. doi: 10.1109/ICASSP.2011.5947651.
- TensorFlow. (2023). *Why TensorFlow*. TensorFlow. Noudettu 16.2.2023 osoitteesta <https://www.tensorflow.org/about>
- Tuominen, H. (2023, 7. helmikuu). *Johdatus tekoälyn taustalla olevaan matematiikkaan (Heli Tuominen)*. Jyväskylän yliopisto. Noudettu 13.2.2023 osoitteesta <https://tim.jyu.fi/view/143092#stokastinen-gradienttimenetelmä>
- Wikipedia (2023, 3. tammikuu). *Recurrent neural network*. Wikipedia. Noudettu 13.2.2023 osoitteesta https://en.wikipedia.org/wiki/Recurrent_neural_network
- Wu, B. & Duan, T. 2017. A performance comparison of neural networks in forecasting stock price trend. *International journal of computational intelligence systems*, 10(1), 336-346. doi:10.2991/ijcis.2017.10.1.23

Yahoo finance. (2023). NASDAQ Composite (^IXIC). Yahoo! Finance. Noudettu
13.2.2023 osoitteesta
[https://finance.yahoo.com/quote/%5EIXIC/?guccounter=1&guce_refer-
rer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_sig=AQAAAD559rLZ
Udut-VYw-
KUjjsJr_gORDD74Px2AEdd96FIEGmdQBleHrQHWixlMKHOpEOSR4Cwfman-
rqIXTPZp4HLsy9lxsUGDavX4ZK2wbpjlf9Fo7rXE5CHF0I2NcDt6sljZeDYmlhMnkvQr
fi-JVNSZGC3X_81gCF7-hwSNhXAINqS wi](https://finance.yahoo.com/quote/%5EIXIC/?guccounter=1&guce_refer-
rer=aHR0cHM6Ly93d3cuZ29vZ2xlLmNvbS8&guce_referrer_sig=AQAAAD559rLZ
Udut-VYw-
KUjjsJr_gORDD74Px2AEdd96FIEGmdQBleHrQHWixlMKHOpEOSR4Cwfman-
rqIXTPZp4HLsy9lxsUGDavX4ZK2wbpjlf9Fo7rXE5CHF0I2NcDt6sljZeDYmlhMnkvQr
fi-JVNSZGC3X_81gCF7-hwSNhXAINqS wi)

Liitteet

Liite 1. Neuroverkon koodi

```
# Koodi mukaillen Rockikz (2022)
# Muutoksia:
#     lisätty aineiston lataamiseen mahdollisuus valita päivämäärät
#     muutettu koulutuksessa käytettävät kustannusfunktiot ja aktivaatiofunktiot
#     lisätty accuracy- ja loss-kuvaajat (Ranta, 2020)
#     lisätty paremmat kuvaajat ennusteista
#     lisätty mahdollisuus muuttaa validointiaineiston määrää neuroverkkoa kouluttaessa
#     lisätty MSE:n kaavat (tulokset normalisoituna ja palautettuna)
#     lisätty selitteet koodin eri vaiheisiin
#     poistettu koodista tutkimukseen liittymättömiä mittareita

"TARVITTAVIEN KIRJASTOJEN ASENNUS"
import tensorflow as tf
import matplotlib.pyplot as plt
import os
import numpy as np
import pandas as pd
import random
import os
import time

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense, Dropout, Bidirectional
from tensorflow.keras.callbacks import ModelCheckpoint, TensorBoard
from sklearn import preprocessing
from sklearn.model_selection import train_test_split
from yahoo_fin import stock_info as si
from collections import deque

from sklearn.metrics import mean_squared_error

get_ipython().run_line_magic('load_ext', 'tensorboard')

"AINEISTON LATAAMINEN YAHOO FINANCESTA JA ESIKÄSITTELY"

# Asetetaan seed, jolloin eri kierrosten väliset tulokset ovat verrattavia (käytössä vain, kun
Shuffle = True)
np.random.seed(80)
tf.random.set_seed(80)
random.seed(80)

def shuffle_in_unison(a, b):
    # sekoitetaan kaksi taulukkoa samalla tavalla
    state = np.random.get_state()
    np.random.shuffle(a)
    np.random.set_state(state)
    np.random.shuffle(b)
```

```

# datan lataaminen
def load_data(ticker, n_steps=50, scale=True, shuffle=True, lookup_step=1, split_by_date=True,
              test_size=0.2, feature_columns=["volume", "close", "open", "high", "low"]):

    # testataan, onko aineisto jo ladattu
    if isinstance(ticker, str):
        # ladtaan data yahoo_fin kirjastosta / aineiston rajaus (kk/pp/vvvv)
        df = si.get_data(ticker, start_date = '1/1/2020', end_date = '9/30/2022',
                        index_as_date = True, interval = '1d')
    elif isinstance(ticker, pd.DataFrame):
        # jos aineisto on jo ladattu, käytetään suoraan
        df = ticker
    else:
        # jos virhe
        raise TypeError("ticker can be either a str or a `pd.DataFrame` instances")
    # kerätään kaikki elementit, jotka palautettu funktiosta
    result = {}
    # palautetaan myös alkuperäinen dataframe
    result['df'] = df.copy()
    # testataan, että aineisto sisältää kaikki halutut herätteet (=feature_columns)
    for col in feature_columns:
        assert col in df.columns, f"'{col}' does not exist in the dataframe."
    # lisätään päivämäärä sarakkeeksi
    if "date" not in df.columns:
        df["date"] = df.index
    if scale:
        column_scaler = {}
        # datan normalisointi välille [0, 1]
        for column in feature_columns:
            scaler = preprocessing.MinMaxScaler()
            df[column] = scaler.fit_transform(np.expand_dims(df[column].values, axis=1))
            column_scaler[column] = scaler
        # lisätään MinMaxScalerin tapaukset palautettuun tulokseen
        result["column_scaler"] = column_scaler
    # lisätään kohdesarake siirtämällä "lookup_stepiä"
    df['future'] = df['close'].shift(-lookup_step)
    # viimeiset `lookup_step` sarakkeet sisältävät epäluvun tulevassa sarakkeessa,
    # joten kerätään ne ennen muiden epälukujen poistamista
    last_sequence = np.array(df[feature_columns].tail(lookup_step))
    # poistetaan muut epäluvut
    df.dropna(inplace=True)
    sequence_data = []
    sequences = deque(maxlen=n_steps)
    for entry, target in zip(df[feature_columns + ["date"]].values, df['future'].values):
        sequences.append(entry)
        if len(sequences) == n_steps:
            sequence_data.append([np.array(sequences), target])
    # liitetään "lookup_step" "n_steppiin", jotta saadaan viimeinen sekvenssi,
    # "last_sequence" saadaan ennustettua tulevat osakekurssit, mitkä eivät ole saatavilla da-
    tasetissä
    last_sequence = list([s[:len(feature_columns)] for s in sequences]) + list(last_sequence)
    last_sequence = np.array(last_sequence).astype(np.float32)
    # lisätään tulokset
    result['last_sequence'] = last_sequence

```

```

# Luodaan X ja y
X, y = [], []
for seq, target in sequence_data:
    X.append(seq)
    y.append(target)
# muutetaan taulukoiksi
X = np.array(X)
y = np.array(y)
if split_by_date:
    # jaetaan data koulutus- ja testausaineistoon päivän mukaan (SPLIT_BY_DATE = True)
    train_samples = int((1 - test_size) * len(X))
    result["X_train"] = X[:train_samples]
    result["y_train"] = y[:train_samples]
    result["X_test"] = X[train_samples:]
    result["y_test"] = y[train_samples:]
    if shuffle:
        # aineiston sekoitus, jos Shuffle = True
        shuffle_in_unison(result["X_train"], result["y_train"])
        shuffle_in_unison(result["X_test"], result["y_test"])
    else:
        # aineiston jako, jos SPLIT_BY_DATE = False
        result["X_train"], result["X_test"], result["y_train"], result["y_test"] =
train_test_split(X, y,
                 test_size=test_size, shuffle=shuffle)
# luettelo testipäivistä
dates = result["X_test"][:-1, -1]
# noudetaan testiominaisuudet alkuperäisestä dataframesta
result["test_df"] = result["df"].loc[dates]
# poistetaan päällekkäiset päivämäärät
result["test_df"] = result["test_df"][~result["test_df"].index.duplicated(keep='first')]
# poistetaan päivät harjoitus- ja testausaineistoista ja muutetaan float32
result["X_train"] = result["X_train"][:, :, :len(feature_columns)].astype(np.float32)
result["X_test"] = result["X_test"][:-1, :, :len(feature_columns)].astype(np.float32)
return result

"NEUROVERKKOMALLIN RAKENTAMINEN"

def create_model(sequence_length, n_features, units=256, cell=LSTM, n_layers=2, dropout=0.3,
                 loss="mean_squared_error", optimizer="adam", bidirectional=True):
    model = Sequential()
    for i in range(n_layers):
        if i == 0:
            # sisäänmenokerros
            if bidirectional:
                model.add(Bidirectional(cell(units, return_sequences=True),
                                         batch_input_shape=(None, sequence_length, n_features)))
            else:
                model.add(cell(units, return_sequences=True,
                               batch_input_shape=(None, sequence_length, n_features)))
        elif i == n_layers - 1:
            # lähtökerros
            if bidirectional:
                model.add(Bidirectional(cell(units, return_sequences=False)))
            else:

```

```

        model.add(cell(units, return_sequences=False))
    else:
        # piilokerrokset
        if bidirectional:
            model.add(Bidirectional(cell(units, return_sequences=True)))
        else:
            model.add(cell(units, return_sequences=True))
    # pudotusmenetelmä (dropout jokaisen kerroksen kohdalla)
    model.add(Dropout(dropout))
    model.add(Dense(1, activation="relu"))
    model.compile(loss=loss, metrics=["mean_squared_error"], optimizer=optimizer)
    return model

"NEUROVERKON PARAMETRIT"

# sekvenssin pituus (montako aikaisempaa päivää käytetään ennustamiseen)
N_STEPS = 50
# Ennusteen pituus, 1 on seuraava päivä
LOOKUP_STEP = 1
# aineiston skaalaus
SCALE = True
scale_str = f"sc-{{int(SCALE)}}"
# aineiston sekoitus
SHUFFLE = False
shuffle_str = f"sh-{{int(SHUFFLE)}}"
# aineiston jako koulutus- ja testausaineistoon päivän mukaan
SPLIT_BY_DATE = True #!
split_by_date_str = f"sbd-{{int(SPLIT_BY_DATE)}}"
# testausaineiston koko, 0.2 on 20%
TEST_SIZE = 0.2
# herätteet
FEATURE_COLUMNS = ["volume", "close", "open", "high", "low"]
# tämän hetkinen pvm
date_now = time.strftime("%Y-%m-%d")
### mallin hyperparametrit
#piilokerrokset
N_LAYERS = 2
# neuronin solu
CELL = LSTM
# neuronit
UNITS = 200
# pudotusprosentit, 0.2 on 20%
DROPOUT = 0.2
# käytetäänkö BRNN
BIDIRECTIONAL = True
### koulutusparametrit
#kustannusfunktio
LOSS = "mse"
#optimointifunktio
OPTIMIZER = "adam"
#erän koko
BATCH_SIZE = 64
#epookit
EPOCHS = 5

```

```

# pörsikoodi
ticker = "^OMXH25"
ticker_data_filename = os.path.join("data", f"{ticker}_{date_now}.csv")
# mallin nimeäminen
model_name = f"{date_now}_{ticker}-{shuffle_str}-{scale_str}-{split_by_date_str}-{LOSS}-{OPTI-
MIZER}-{CELL.__name__}-seq-{N_STEPS}-step-{LOOKUP_STEP}-layers-{N_LAYERS}-units-{UNITS}"
if BIDIRECTIONAL:
    model_name += "-b"

"LUODAAN KANSIOT, JOIHIN TULOKSET LADATAAN"

if not os.path.isdir("results"):
    os.mkdir("results")
if not os.path.isdir("logs"):
    os.mkdir("logs")
if not os.path.isdir("data"):
    os.mkdir("data")

"NEUROVERKON KOULUTUS --> TALLENNETAAN AUTOMAATTISESTI PARAS MALLI ENNUSTAMISEEN"

# aineiston lataus
data = load_data(ticker, N_STEPS, scale=SCALE, split_by_date=SPLIT_BY_DATE,
                 shuffle=SHUFFLE, lookup_step=LOOKUP_STEP, test_size=TEST_SIZE,
                 feature_columns=FEATURE_COLUMNS)
# aineiston tallennus
data["df"].to_csv(ticker_data_filename)
# mallin rakennus
model = create_model(N_STEPS, len(FEATURE_COLUMNS), loss=LOSS, units=UNITS, cell=CELL, n_lay-
ers=N_LAYERS,
                    dropout=DROPOUT, optimizer=OPTIMIZER, bidirectional=BIDIRECTIONAL)

# callback-funktiot ja osoitin tallentamaan optimaalisimmat painot
checkpointer = ModelCheckpoint(os.path.join("results", model_name + ".h5"),
                              save_weights_only=True, save_best_only=True, verbose=1)
tensorboard = TensorBoard(log_dir=os.path.join("logs", model_name))
# neuroverkon koulutus
# käytetään ModelCheckpointtia tallentamaan aina optimaalisimmat painot
#voidaan määrittää, kuinka suuri osa koulutusaineistosta käytetään validointiin
history = model.fit(data["X_train"], data["y_train"],
                   batch_size=BATCH_SIZE,
                   epochs=EPOCHS,
                   validation_split=0.1,
                   callbacks=[checkpointer, tensorboard],
                   verbose=1)

"ACCURACY JA LOSS KUVAAJAT"
plt.style.use('bmh')
burnout = 0
epochs = range(1, len(history.history['val_mean_squared_error']) + 1)
plt.plot(epochs[burnout:], history.history['val_mean_squared_error'][burnout:], 'r--', la-
bel='Validation accuracy')
plt.plot(epochs[burnout:], history.history['mean_squared_error'][burnout:], 'b--', label='Train
accuracy')

```

```

plt.legend()
plt.title('Accuracy')
plt.figure()
plt.plot(epochs[burnout:], history.history['val_loss'][burnout:], 'r--', label='Validation loss')
plt.plot(epochs[burnout:], history.history['loss'][burnout:], 'b--', label='Train loss')
plt.legend()
plt.title('Loss')
plt.show()

"ENNUSTEEN KUVAAJA"
def plot_graph(test_df):
    plt.rcParams["figure.figsize"] = (30,20)
    plt.plot(test_df[f'true_close_{LOOKUP_STEP}'], c='b')
    plt.plot(test_df[f'close_{LOOKUP_STEP}'], c='r')
    plt.xlabel("Days")
    plt.ylabel("Price")
    plt.legend(["Actual Price", "Predicted Price"])
    plt.show()

"DATAKEHYS ENNUSTUKSILLE"
def get_final_df(model, data):
    X_test = data["X_test"]
    y_test = data["y_test"]
    y_pred = model.predict(X_test)
    if SCALE:
        y_test = np.squeeze(data["column_scaler"]["close"].inverse_transform(np.expand_dims(y_test, axis=0)))
        y_pred = np.squeeze(data["column_scaler"]["close"].inverse_transform(y_pred))
    test_df = data["test_df"]
    # lisää ennustetut tulevat hinnat datakehukseen
    test_df[f"close_{LOOKUP_STEP}"] = y_pred
    # lisätään todelliset tulevat hinnat datakehukseen
    test_df[f"true_close_{LOOKUP_STEP}"] = y_test
    # lajitellaan datakehys päivän mukaan
    test_df.sort_index(inplace=True)
    final_df = test_df

    return final_df

"ENNUSTE HINNALLE"
def predict(model, data):
    # haetaan viimeinen sekvenssi datasta
    last_sequence = data["last_sequence"][-N_STEPS:]
    # laajennetaan dimensiota
    last_sequence = np.expand_dims(last_sequence, axis=0)
    # ennuste (skaalattu nyt [0, 1])
    prediction = model.predict(last_sequence)
    # saadaan hinta (käännetään skaalaus)
    if SCALE:
        predicted_price = data["column_scaler"]["close"].inverse_transform(prediction)[0][0]
    else:
        predicted_price = prediction[0][0]
    return predicted_price

```

```

# ladataan optimaaliset painokertoimet kansioista, johon ne tallennettiin koulutuksen aikana
model_path = os.path.join("results", model_name) + ".h5"
model.load_weights(model_path)

"KESKINELIÖVIRHEEN LASKEMINEN"
loss, mse = model.evaluate(data["X_test"], data["y_test"], verbose=0)

# lasketaan MSE (käännetty skaalaus)
if SCALE:
    mean_squared_error = data["column_scaler"]["close"].inverse_transform([[mse]])[0][0]
else:
    mean_squared_error = mse

print(mean_squared_error)

# tulostetaan MSE ilman, että "puretaan" normalisointia

loss, mse = model.evaluate(data["X_test"], data["y_test"], verbose=0)

mean_squared_error = mse

print("Mean Squared Error:", mean_squared_error)

# lopullinen datakehys testausjoukolle
final_df = get_final_df(model, data)

# ennustetaan tulevia hintoja
future_price = predict(model, data)

# tehdään kuvaaja tuloksista
plot_graph(final_df)

"TULOSTEN TALLENTAMINEN"
print(final_df.tail(10))
# tallennetaan tulokset vielä csv-muodossa
csv_results_folder = "csv-results"
if not os.path.isdir(csv_results_folder):
    os.mkdir(csv_results_folder)
csv_filename = os.path.join(csv_results_folder, model_name + ".csv")
final_df.to_csv(csv_filename)

```