

Tuukka Pitkänen

**Syöttödatan vaikutukset kitaravahvistinta mallintavan
neuroverkon koulutustulokseen**

Tietotekniikan pro gradu -tutkielma

2. kesäkuuta 2024

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Tuukka Pitkänen

Yhteystiedot: tuukkapitkanen2@gmail.com

Ohjaajat: Paavo Nieminen ja Jenni Raitoharju

Työn nimi: Syöttödatan vaikutukset kitaravahvistinta mallintavan neuroverkon koulutustulokseen

Title in English: Effects of input data on the training result of a neural network modeling a guitar amplifier

Työ: Pro gradu -tutkielma

Opintosuunta: ohjelmisto- ja tietoliikennetekniikan opintosuunta

Sivumäärä: 35+0

Tiivistelmä: Kitaravahvistin on oleellinen osa sähkökitaran soittoa. Fyysisiä kitaravahvistimia on haluttu jo pitkään mallintaa digitaalisesti fyysisen laitteiston haittapuolien, kuten hinnan ja vikaherkkyuden vuoksi. Yksi vaihtoehto kitaravahvistimien mallintamiseen on neuroverkot. Kitaravahvistimen neuroverkkopohjaista emulointia on tutkittu viime vuosina. Tutkimuksissa on muun muassa vertailtu mallinnustoteutuksien eri arkkitehtuureja ja mallien realistisuutta. Kirjallisuuden perusteella vaikuttaa siltä, että syötteeltä vaadittavasta sisällöstä ja sen määrästä ei ole aiemmin tehty järjestelmällistä tutkimusta, mihin tämä tutkielma nyt tähtää. Tutkielmassa arvioidaan, miten neuroverkkopohjaisen kitaravahvistimien mallintajan, Neural Amp Modelerin, koulutustulos muuttuu, kun syöteenä käytettyä ääniteparia muokataan erilaisin tavoin. Huomattiin, että 30 sekunnin syötedatalla ei päästy yhtä hyvään koulutustulokseen kuin alkuperäisellä 3 minuutin syötteellä, kun verrattiin testidatalla. Löydettiin kuitenkin alkuperäistä syötettä lyhyempi äänite, jolla koulutustulos testidatalla verrattuna pysyi samana. Lyhyemmällä syötteellä voidaan nopeuttaa koulutusparien luomista.

Avainsanat: neuroverkot, kitaravahvistin, koulutusdatan optimointi

Abstract: A guitar amplifier is an essential part of playing an electric guitar. Physical guitar amplifiers have long been wanted to be digitally modeled due to the disadvantages of physical

hardware such as cost and susceptibility to failure. One option for modeling guitar amps is neural networks. Neural network-based emulation of a guitar amplifier has been studied in recent years. The studies include comparison of different architectures and the realism of the models. Based on the literature, it appears that there has not been a systematic study on the required content and amount of input, which this thesis now aims to address. This thesis evaluates how the training result of the Neural Amp Modeler, a neural network-based modeler of guitar amplifiers, changes when the pair of sounds used as input is modified. 30-second input data did not achieve as good a training result as the original 3-minute input, when compared with the test data. A recording shorter than the original input was found, with which the training result compared to the test data remained the same. A shorter input can speed up the creation of training pairs.

Keywords: Guitar amplifiers, neural networks. optimization of training data

Kuviot

Kuvio 1. Yhden arvon laskeminen 2-ulotteisessa konvoluutiiossa.	4
Kuvio 2. Yksinkertainen konvoluutioverkko yhdellä konvoluutiokerroksella, tiivistyskerroksella ja kahdella täysin kytketyllä kerroksella. (Zaniolo ja Marques 2020)	5
Kuvio 3. Havainnekuva dilatoituista kausaalisista konvoluutiokerroksista. (Van Den Oord ym. 2016)	7
Kuvio 4. Neural Amp Modelerin karkea rakenne WaveNetin tyyliin. Alkuperäisessä WaveNetissä on vain yksi pino residual bloqueja. (Van Den Oord ym. 2016)	10
Kuvio 5. Hughes&Kettner -vahvistin ja kitarasignaalien äänittäminen.	15
Kuvio 6. Alustavat testitulokset, kun malli on koulutettu eri koulutusdatoilla ja 100 epookilla. CT (Computation time) on koulutukseen kulunut aika.	18
Kuvio 7. Tulokset eri syötteillä input1–7 ja noin 1600 sekunnin koulutusajalla.	20
Kuvio 8. Tulokset eri syötteillä inputA–G ja noin 1600 sekunnin koulutusajalla.	21
Kuvio 9. input0:lla koulutetun mallin tuottaman äänitteen spektrogrammin ja vahvistimen tuottaman äänitteen spektrogrammin erotukset eri epookkimäärillä. Voimakkuudet on normalisoitu ensimmäisen kuvan suurimmalla arvolla.	22
Kuvio 10. input0 ja inputC -syötteillä koulutettujen mallien suoriutuminen testidatalla verrattuna mallinnettavaan vahvistimeen, koulutusajan suhteen.	23
Kuvio 11. input0 ja inputC -syötteillä koulutettujen mallien suoriutuminen testidatalla verrattuna mallinnettavaan vahvistimeen, kun testidata on ajettu eri suotimien läpi. .	24

Sisällys

1	JOHDANTO	1
2	KIRJALLISUUSKATSAUS.....	3
2.1	Konvoluutioneuroverkkojen perusteet.....	3
2.2	WaveNet	6
2.3	Kitaravahvistin	7
2.4	Kitaravahvistimien mallintaminen.....	8
2.5	Neural Amp Modeler.....	9
3	MENETELMÄ.....	13
3.1	Koulutusaineiston syötteet	13
3.2	Testiaineiston syötteet.....	13
3.3	Vahvistinsignaalit.....	14
3.4	Neural Amp Modelerin koulutus	14
3.5	Tulosten arviointimittarit.....	15
3.6	Menetelmän validointi alustavilla mittauksilla	17
4	TULOKSET.....	19
4.1	Muutokset alustavaan menetelmään.....	19
4.2	Eri syötteiden yleinen vertailu	19
4.3	Spektrogrammit	21
4.4	Syötteiden input0 ja inputC tarkempi vertailu	22
4.5	Pohdinta.....	24
5	YHTEENVETO.....	26
	LÄHTEET	27

1 Johdanto

Syväoppiminen on viime vuosina saavuttanut valtavaa suosiota monilla tieteen ja yhteiskunnan aloilla (Raitoharju 2022). Konvoluutioverkot ovat olleet suosittuja etenkin kuvien luokittelussa, mutta ne ovat osoittautuneet tehokkaiksi myös äänien luokittelussa (Hershey ym. 2017). Luokittelun lisäksi konvoluutioverkot ovat tehokkaita 1-ulotteisessa signaalien käsittelyssä (Kiranyaz ym. 2019).

Kitaravahvistin on laite, joka voimistaa sähkökitaran signaalin kuuluvaksi, mutta myös muuttaa signaalin sävyä merkittävästi. Vahvistimet sisältävät säädettäviä parametreja, joilla sävyä voi muuttaa. Lisäksi eri vahvistimet erilaisista rakenteista johtuen eivät kuulosta samalta. Erityisesti elektroniputkitekнологiaan perustuvat vahvistimet (jotka ovat suosittuja kitaristien keskuudessa) ovat kalliita, suuria, herkästi särkyviä, ne kuluttavat paljon virtaa ja tuottavat paljon lämpöä (Barbour 1998). Digitaalinen kitaravahvistimien mallintaminen korjaisi mainittuja ongelmia, sillä yhdellä tietokoneella voitaisiin emuloida useita eri vahvistimia ilman fyysisten vahvistimien haasteita.

Vahvistimien emuloinnissa on kaksi päähaastetta: emuloinnin tarkkuus ja suoritustehovaatimukset. Virtuaalisen vahvistimen tulee kuulostaa mahdollisimman paljon oikealta vahvistimelta, ja lisäksi sen tulee olla tarpeeksi kevyt tehovaatimuksiltaan, jotta sillä voidaan soittaa mahdollisimman pienellä latenssilla. Siedettävän latenssin raja reaaliaikaisissa audio-sovelluksissa on noin 20-30ms (Lago ja Kon 2004). Digitaalisissa audiosovelluksissa latenssia syntyy algoritmien suorituskestosta. Lisäksi analogi-digitaali- ja digitaali-analogi-muuntimet lisäävät latenssia (Preonus 2017). Perinteisillä digitaalisen signaalinkäsittelyn keinoilla on ollut haasteita toteuttaa nämä kaksi vaatimusta (Vanhatalo ym. 2022).

Neuroverkkoja on yritetty hyödyntää kitaravahvistimien emuloimisessa ja tulokset ovat olleet lupaavia (Vanhatalo ym. 2022). Kitaravahvistinta mallintava neuroverkko ottaa syöte-tavoite-pariksi kaksi äänitiedostoa, joista tavoite on luotu äänittämällä syöte kitaravahvistimen läpi. Idealin kouluttamisen jälkeen neuroverkolle syötettävät äänet kuulostavat samalta, kuin jos ne syötettäisiin kitaravahvistimen läpi.

Wright ym. (2020) tekemän tutkimuksen mukaan kolme minuuttia koulutusdataa riittää ki-

taravahvistinta mallintavan neuroverkon kouluttamiseen. Vanhatalo ym. (2022) katsauksessaan eivät tämän lisäksi nosta muita havaintoja koskien syötedatan vaikutuksia. Aiempaa järjestelmällisesti syötedatan sisältöön keskittyvää tutkimusta ei siis ole. Pidemmän koulutusdatan läpikäyntiin verkolla kuluu kouluttaessa enemmän aikaa, mutta toisaalta lyhyempi koulutusdata ei välttämättä riitä tarkan mallin muodostamiseen.

Tässä pro gradu-tutkielmassa on tavoitteena vastata seuraaviin kysymyksiin:

- Mitä vaikutuksia oppimistulokseen on kitaravahvistinta mallintavan neuroverkon kouluttamiseen käytettävän syöte-tavoite-parin rajoittamisella?
- Löydetäänkö syöttödatasta ominaisuuksia, joiden sisällyttäminen/poistaminen parantaa/nopeuttaa oppimistulosta?

Luvussa 2 avataan tämän tutkimuksen aihealuetta pohjaten aiempaan tutkimukseen. Luvussa 3 esitellään tässä tutkimuksessa käytetty menetelmä. Ennen yhteenvetoa käydään läpi tulokset luvussa 4.

2 Kirjallisuuskatsaus

Tässä luvussa esitellään tiiviisti konvoluutioneuroverkkojen ja kitaravahvistimien perusteet. Lisäksi käydään läpi aiempaa tutkimusta liittyen kitaravahvistimien mallintamiseen neuroverkoilla ja esitellään Neural Amp Modeler.

2.1 Konvoluutioneuroverkkojen perusteet

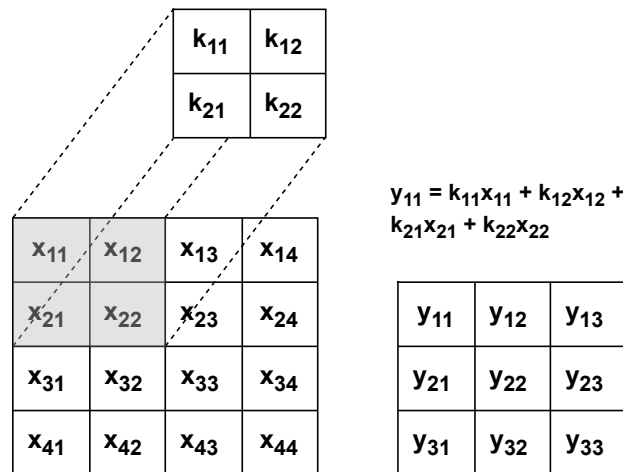
Konvoluutioneuroverkot tai konvoluutioverkot ovat keränneet viime vuosina suurta suosiota useilla eri aloilla, useissa eri käyttökohteissa. Suosio perustuu konvoluutioverkkojen kykyyn käsitellä raakaa dataa, kuten kuvia ja ääniä, sellaisenaan. Konvoluutioverkot pystyvät irrottamaan piirteitä raaka-asta datasta (feature extraction). Konvoluutioverkot ovat eteenpäin syöttäviä (feedforward) verkkoja. Informaatio siis kulkee sisääntulokerrokselta kohti ulostulokerrosta ilman takaisin kytkettyjä silmukoita, toisin kuin takaisinkytketyissä (recurrent) verkoissa. (Raitoharju 2022)

Konvoluutioneuroverkon kouluttaminen perustuu vastavirta-algoritmiin (backpropagation). Vastavirta-algoritmissa vuorotellaan eteenpäin menevän läpimenon (forward pass) ja takaisinpäin menevän läpimenon (backward pass) kanssa. Verkon kouluttamiseen tarvitaan syöte-tavoite-pari, eli koulutusdata. Eteenpäin menevässä läpimenossa lasketaan verkon ulostulo syötteelle sen hetkisillä painoilla. Ulostuloa verrataan tavoitteeseen laskemalla tappiofunktion (loss function) avulla virhe. Takaisin päin menevässä läpimenossa muutetaan gradienttimenetelmällä (gradient descent) verkon painoja, jotta laskettu virhe pienentyisi. Koulutusaineisto käydään useita kertoja läpi ja yhtä kokonaista läpikäyntiä kutsutaan epookiksi (epoch). (Raitoharju 2022)

Ylisovittaminen on yksi neuroverkkojen ongelma. Ylisovittamisessa neuroverkko oppii koulutusdatan todella hyvin, mutta suoriutuu huonosti uuden datan kanssa. Yksi tapa estää ylisovittamista on aikainen pysäyttäminen (early stopping). Aikaisessa pysäyttämisessä käytössä oleva data jaetaan koulutusdataan ja validointidataan. Verkkoa koulutetaan koulutusdatalla, mutta samalla lasketaan tappio validointidatalla. Aikainen koulutuksen pysäyttäminen tehdään, kun tappio validointidatalla laskettuna ei enää parane tai huononee. Validointidatan

käyttämistä koulutuksen arviointiin kutsutaan validoinniksi. (Zaras, Passalis ja Tefas 2022)

Konvoluutioverkko sisältää konvoluutiokerroksia. Konvoluutiokerroksen ulostulo Y on sisääntulon X ja ytimen K välinen konvoluutio. Ydin K on pieni, painoja sisältävä matriisi. Konvoluutio lasketaan liu'uttamalla K :ta X :n yli ja laskemalla kussakin kohdassa vastinparien tulojen summa. Kuviossa 1 lasketaan ulostulon Y arvo y_{11} , joka saadaan 4×4 sisääntulon X ja 2×2 ytimen K konvoluutiona. Kuviosta huomataan, että Y on pienempi kuin X . Tämä voitaisiin välttää tolppaamalla (padding) X nolilla ennen konvoluution laskemista. (Raitoharju 2022)



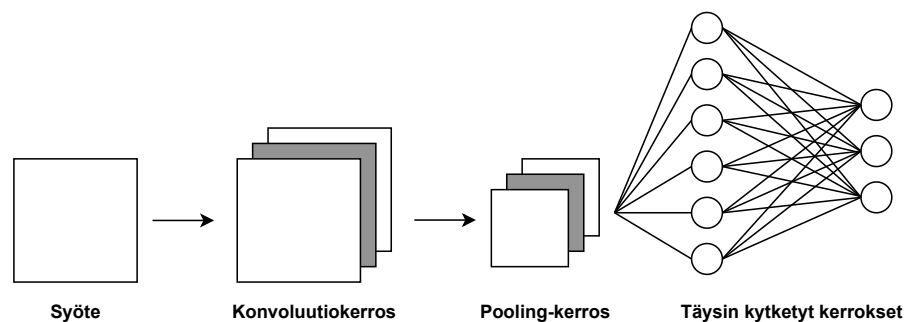
Kuvio 1. Yhden arvon laskeminen 2-ulotteisessa konvoluutiiossa.

2-ulotteinen konvoluutio voidaan kuvata yleisemmässä muodossa:

$$Y[m,n] = \sum_{i=1}^k \sum_{j=1}^k K[i,j]X[m-1+i,n-1+j], \quad (2.1)$$

missä Y on kerroksen ulostulo, X on sisääntulo, K on ydin ja k on ytimen korkeus tai leveys. Tiivistyskerroksilla (pooling layer) voidaan tiivistää sisään tuleva informaatio pienempään

matriisiin ja säästää laskennassa. Konvoluutioverkon lopussa on yleensä yksi tai useampia täysin kytkettyjä kerroksia, joita käytetään aiemmilla kerroksilla irrotettujen piirteiden luokitteluun. Täysin kytketyssä kerroksessa jokainen sisääntulon arvo kytketään jokaiseen seuraavan kerroksen elementtiin omalla painollaan ja arvot summataan (Raitoharju 2022). Esimerkki verkosta on kuviossa 2.



Kuvio 2. Yksinkertainen konvoluutioverkko yhdellä konvoluutiokerroksella, tiivistyskerroksella ja kahdella täysin kytketyllä kerroksella. (Zaniolo ja Marques 2020)

Yleensä sisääntulo koostuu useista *kanavista*. Syötteen kanavia voivat olla esimerkiksi kuvan ollessa syötteenä kuvan eri värit. Jokaiselle kanavalle on suotimessa oma ytimensä. Suotimia on samalla kerroksella yleensä useampia, sillä eri suotimet irrottavat samalta kerrokselta eri piirteitä. (Raitoharju 2022)

Ohittavat yhteydet (skip connections) ovat yhteyksiä verkon aiemmista kerroksista myöhempiin osiin verkkoa. Ohittavista yhteyksistä tulevat piirrekartat yhdistetään jonkin neuroverkon ylemmän kerroksen piirrekarttaan summaamalla tai liittämällä ne erillisiksi piirrekartan kerroksiksi. Ohittavat yhteydet ovat tehokas tapa vähentää katoavan gradientin ongelmaa, eli sitä kun vastavirta-algoritmista gradientit pienenevät kerros kerrokselta. Lisäksi näin saadaan aiemmilta kerroksilta korostettua piirteitä, jotka muuten saattaisivat hävitä myöhemmillä kerroksilta (Raitoharju 2022). Ohittavia yhteyksiä käyttivät muun muassa He ym. (2016),

ja mainitsivat lisäksi kolme kirjaa, joissa ohittavia yhteyksiä on ehdotettu: (Bishop 1995), (Ripley 2007) ja (Venables ja Ripley 2013).

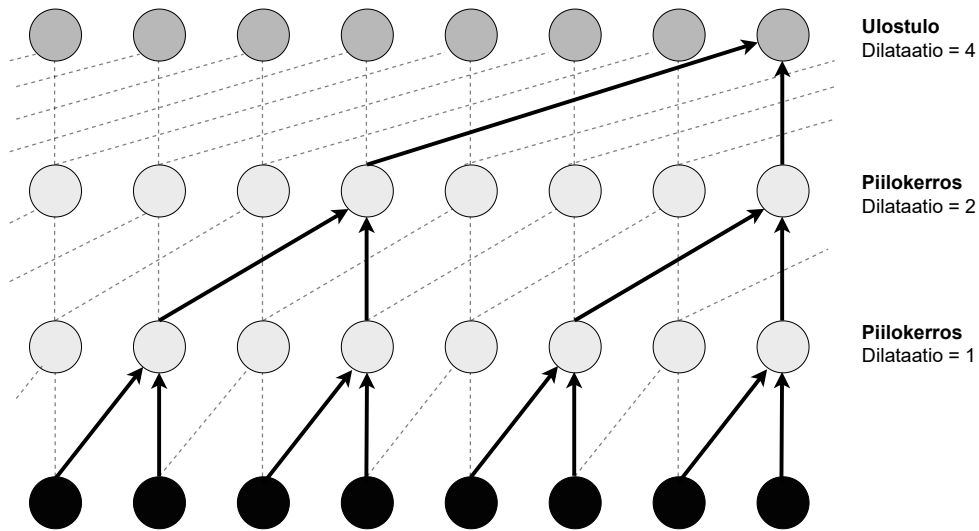
Luokittelun lisäksi neuroverkoilla voidaan myös generoida signaaleja tai muuntaa niiden ominaisuuksia. Esimerkkinä tällaisista ovat esimerkiksi tässä tutkimuksessa tarkastelussa oleva Neural Amp Modeler, ja Neural Amp Modelerin pohjana toimiva WaveNet-arkkitehtuuri. Näistä kerrotaan lisää seuraavissa luvuissa.

2.2 WaveNet

WaveNet on Google DeepMind-yhtiön kehittämä verkkoarkkitehtuuri, joka luo raakaa ääntä (Van Den Oord ym. 2016). Van Den Oord ym. (2016) näyttivät, että WaveNet kykeni luomaan ennen näkemättömän luonnollisia puheäänitteitä tekstistä. WaveNet perustuu aiemmin julkaistuun PixelCNN kuvien luomismalliin (Van Den Oord, Kalchbrenner ja Kavukcuoglu 2016).

Tärkeä elementti WaveNetissä on kausaaliset konvoluutiot (causal convolutions). Kausaalisilla konvoluutioilla varmistetaan, että mallin ennuste ei riipu tulevista näytteistä. Tällöin esimerkiksi puhesynteesissä voidaan ääntä tuottaa tekstistä reaaliajassa. Lisäksi WaveNet käyttää dilatoituja konvoluutioita. Dilatoidussa konvoluutioissa sisääntuloarvoja jätetään välistä, ja näin ydin kattaa suuremman alueen syötteestä. Kattamalla suurempi alue voidaan mallintaa pidemmän aikavälin riippuvuuksia. Dilataatio tuplataan joka kerroksella. Kuvios- ta 3 nähdään, kuinka kerroksen arvot ovat riippuvaisia vain suoraan alhaalta tai vasemmalta tulevista arvoista. Lisäksi kuvios- ta nähdään jokaisella kerroksella kasvava dilataatio. (Van Den Oord ym. 2016)

Lisäksi WaveNet koostuu residuaalilohkoista (residual blocks) ja ohittavista yhteyksistä. Residuaalilohkoista voi lukea julkaisusta, jonka tekivät He ym. (2016).



Kuvio 3. Havainnekuva dilatoiduista kausaalisista konvoluutiokerroksista. (Van Den Oord ym. 2016)

2.3 Kitaravahvistin

Sähkökitaraa soitetaan lähes poikkeuksetta kitaravahvistimen kanssa. Kitaravahvistimena voi toimia perinteisen fyysisen vahvistimen sijaan virtuaalinen kitaravahvistin, mutta tässä tutkielmassa kitaravahvistimella tarkoitetaan selvyden vuoksi analogista kitaravahvistinta. Eri valmistajien eri kitaravahvistinmalleilla on omat sävynsä, eli kitaravahvistimen tehtävä ei siis ole ainoastaan vahvistaa sähkökitara kuuluvaksi ääneksi, vaan kitaristi valitsee soitettavaan musiikkiin sopivimman vahvistimen. Herbst (2019) kertoo tutkimuksessaan kitaristien omistavan keskimäärin kolme kitaravahvistinta. Lisäksi Herbst (2019) mainitsee kolme ammattilaiskitaristia (Viljami Lindström, Stevie Ray Vaughan ja Joe Bonamassa), joiden tiedetään hyödyntäneen useita eri sävyisiä kitaravahvistimia samanaikaisesti.

Kitaravahvistin koostuu yksinkertaistetusti kahdesta vahvistimesta; esivahvistimesta ja päätevahvistimesta. Esivahvistimen tehtävä on muokata kitarasta tulevaa signaalia ja sen sävyä. Myös päätevahvistin muuntaa kitarasignaalin sävyä, mutta sen pääasiallinen tehtävä on vahvistaa esivahvistimesta tulevaa matalajännitteistä signaalia, jotta sen jännite riittää ajamaan kaiutinta (Keeports 2017). Kombovahvistimella tarkoitetaan laitetta, jossa on sekä kitaravahvistin, että yksi tai useampi kaiutin. Vahvistinnuppi sisältää esi- ja päätevahvistimen, mutta

ei kaiuttimia. Vahvistinnappi ei siis itsessään tuota ääntä, vaan se kytketään kaiuttimia sisältävään kaiutinkaappiin. Tässä tutkimuksessa kitaravahvistimen mallinnuksella tarkoitetaan vahvistinnupin mallintamista, eli kaiuttimien emulointia ei huomioida.

2.4 Kitaravahvistimien mallintaminen

Vanhatalo ym. (2022) jakavat kitaravahvistimien mallintamisen lähestymistavat kolmeen kategoriaan: white-box, gray-box ja black-box -metodeihin. White-box-metodilla mallinnettaessa tarvitaan tietoa fyysisen vahvistimen komponenteista. Komponentit ja niiden toiminta muunnetaan kaavoiksi, jotka antavat vastaavan vasteen kuin vahvistin. White-box-metodissa vaaditaan tarkkaa tietoa vahvistimen eri osista, kun taas gray-box-metodilla vahvistimen eri osat jaetaan "blokkeihin", jolloin mallinnettavia osia muodostuu vähemmän, ja tarkempaa vahvistimen tarkastelua tarvitsee tehdä paljon vähemmän. Tällöin säästetään vaikeaa algoritmin kehittämisessä sekä suoritustehoa algoritmin tehokkuudessa, toisaalta gray-boxilla ei saada emuloitua vahvistinta yhtä tarkasti kuin white-boxilla. Black-box metodin suurin etu on siinä, että vahvistimen sisäisestä toiminnasta ei tarvitse tietää mitään. Mallinnus toteutetaan pelkästään vahvistimen sisään- ja ulostulosignaalien pohjalta. (Vanhatalo ym. 2022)

Kitaravahvistimien ja muidenkin ääntä muokkaavien analogisten laitteiden neuroverkkomallinnukseen liittyvää tutkimusta on viime vuosilta (Vanhatalo ym. 2022). Cassidy ja De Sena (2023) vertasivat eri syöttödatoihin perustuvia mallinnuksia. Tutkimuksessa luotiin kolme eri genrekohtaista koulutussettiä: rock, metal ja yleinen. Kaikkien koulutettujen mallien havaittiin olevan realistisia, mutta genrekohtainen koulutusdata paransi tuloksia. Kuuntelutesteissä havaittiin, että mallit ovat tarpeeksi realistisia verrattuna oikeaan vahvistimeen, ainakin musiikkituotantokontekstissa. Tutkimuksessa verrattiin eri syöttödatojen lisäksi kahteen eri arkkitehtuuriin perustuvaa verkkoa; WaveNet ja LSTM-RNN. Näitä kahta on myös verrattu keskenään aiemmin (Wright, Damskögg ja Välimäki 2019). WaveNet on pärjännyt hyvin kuuntelutesteissä (Damskögg ym. 2019; Cassidy ja De Sena 2023).

Vanhatalo ym. (2022) kertovat katsauksessaan kolmesta julkaisusta, joissa WaveNet on muokattu kitaravahvistinta tai säröpedaalia mallintavaksi malliksi (Damskögg, Juvela ja Välimä-

ki 2019; Wright ym. 2020; Damskäg ym. 2019). Muokattu WaveNet ei enää luo uutta ääntä tekstistä, vaan ulostulo on muokattu versio sisään tulevasta äänestä. Wright ym. (2020) tekemien kuuntelutestien mukaan malli pystyi uskottavasti mallintamaan kitaravahvistinta reaaliajassa.

Verkkojen suoriutumista arvioidaan usein samojen metriikoiden avulla, joita hyödynnetään myös tappiofunktioina. Objektiiiset metriikat eivät välttämättä vastaa hyvin kuultua kokemusta, mutta ihmisten kanssa toteutetut kuuntelutestit ovat aikaa vieviä. Siksi ihmisistä riippumattomille objektiiivisille metriikoille on tarvetta (Vanhatalo ym. 2022).

Wright ym. (2020) havaitsivat, että lisäämällä koulutusdatan määrää kymmenestä sekunnista kolmeen minuuttiin, validoinnin tulos parani. Kolmea minuuttia pidemmällä datalla validointitulokset ei ollut merkittävästi parempi. Tutkimuksessa käytettiin yhteensä 5 minuuttia kitara- ja bassoäänitteitä Kehling ym. (2014) luomasta kokoelmasta. Vertailu koski vain datan määrää, mutta ei tarkemmin eri osien vaikutuksia. Väitöskirjassaan Thomas (2019) kokosi kouluttamista varten kokoelman, joku koostuu viidestä eri tyylistä, sisältäen yksittäisiä nuotteja ja sointuja oikeista kappaleista.

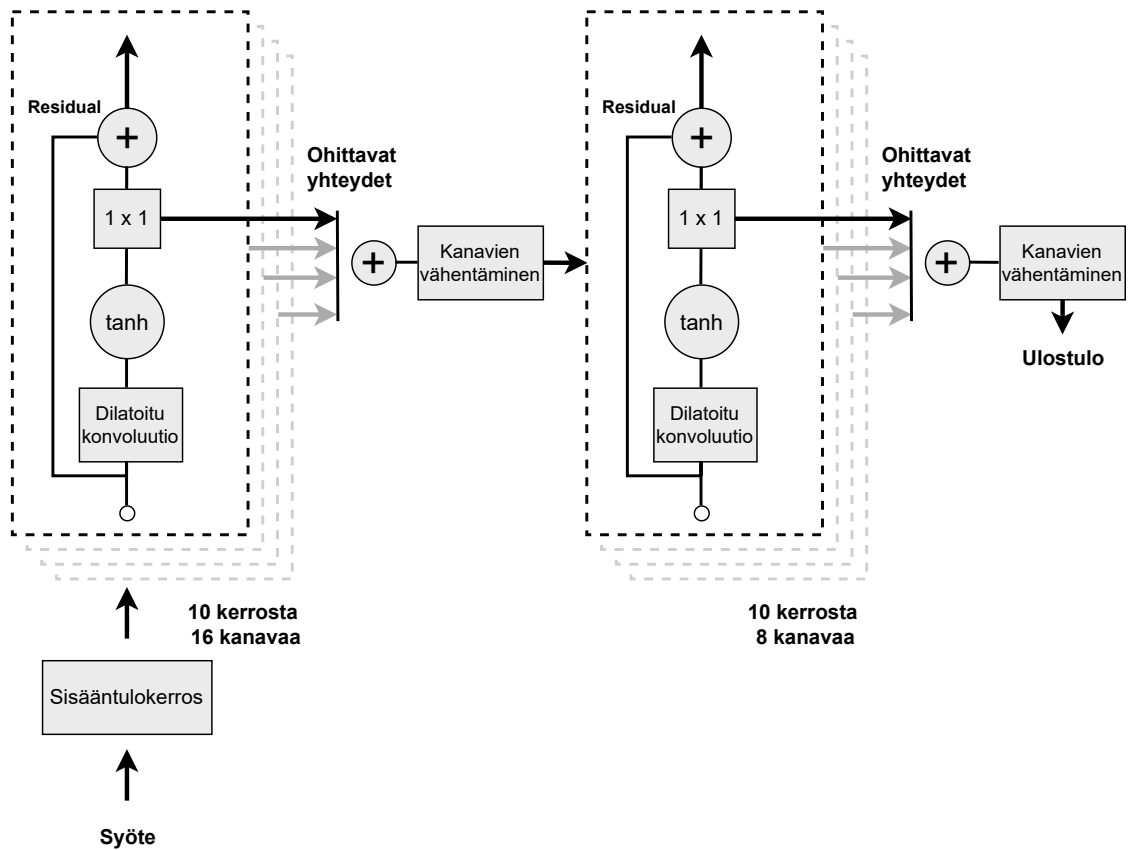
Aiemmin mainituissa julkaisuissa on syöte-tavoite-parina käytetty syöteenä digitaalisia sähkökitaraäänitteitä. Näin on tehty muissakin tutkimuksissa (Zhang ym. 2018) (Schmitz ja Embrechts 2018) (Ramírez ja Reiss 2019). Niin kutsuttuja siniaaltopyyhkäisyjä, eli taajuutta vaihtavia siniaaltosignaaleja (sine sweep), ja kohinan yhdistelmää on myös hyödynnetty (Kuznetsov, Parker ja Esqueda 2020) (Jin ym. 2016). Neural Amp Modelerin tarjoama syöte on erilaisia ääniä sisältävä kolmiminuuttinen äänite (“neural-amp-modeler GitHub” 2019). Äänit eivät pyri edustamaan erilaisia sävellajeja ja kitaransoittotekniikoita ja siten tavantomaista musiikkia, toisin kuin useimmissa tutkimuksissa.

2.5 Neural Amp Modeler

Neural Amp Modeler (NAM) on avoimen lähdekoodin projekti, joka syväoppimista hyödyntäen luo malleja kitaravahvistimista ja -pedaaleista huipputarkkuudella (“Neural Amp Modeler” 2024). Neural Amp Modelerin pääkehittäjänä toimii Steven Atkinson. NAM perustuu python-ohjelmointikielen ja se käyttää Anacondaa pakettien hallintaan ja PyTorch

Lightning -kirjastoa mallintamiskehityksenä.

Neural Amp Modeler käyttää oletuksena WaveNet-pohjaista toteutusta verkon pohjana. Rakennne on nähtävissä kuviossa 4. Kuvioista nähdään, että toisin kuin alkuperäisessä (Van Den Oord ym. 2016) WaveNet-rakenteessa, NAM-toteutus sisältää yhden sijaan kaksi pinoa residuaalilohkoja. Residuaalilohkon sisältö näkyy myös kuviossa. Pinojen välissä kanavien määrää vähennetään kuudestatoista kahdeksaan. Pinoissa on kussakin 10 blockia. Alkuperäisessä WaveNetissä aktivointiin käytettiin hyperbolisen tangentin lisäksi myös logistista funktiota. Verkko sisältää yhteensä 13801 painoa.



Kuvio 4. Neural Amp Modelerin karkea rakenne WaveNetin tyyliin. Alkuperäisessä WaveNetissä on vain yksi pino residual blockeja. (Van Den Oord ym. 2016)

Neural Amp Modeler on kerännyt suosiota viimeisen kahden vuoden aikana. Neural Amp Modeler -Facebook ryhmä on kerännyt 17000 käyttäjää alle kahdessa vuodessa ("Neural Amp Modeler (NAM) - Facebook ryhmä" 2024). Tonehunt.org on sivusto, jonka tarkoituk-

senä on helpottaa Neural Amp Modelerilla tuotettujen mallinnuksien jakamista muiden käyttäjien kanssa. Sivustolle on ladattu jo yli 14000 mallinnusta. Suosiosta huolimatta Google Scholar -hakukoneella löydettiin vain kaksi aiempaa julkaisua Neural Amp Modelerista, joista vain toiseen päästiin käsiksi. Avautuva julkaisu oli Pelita Harapanin yliopistossa tehty kandidaatin tutkielma, jossa arvioitiin subjektiivisesti Neural Amp Modelerin tarkkuutta (Tejo 2023). NAM:iin liittyvälle akateemiselle tutkimukselle vaikuttaa täten olevan tarvetta.

Neural Amp Modeler voidaan kouluttaa kolmella tavalla: pilvessä Google Colabin ja esitetydyn Jupyter Notebookin avulla, graafisen käyttöliittymän avulla tai komentoriviltä. Colab- ja graafinen-vaihtoehto mahdollistavat koulutuksen vain NAM:in omalla syötetiedostolla, joten tutkimusta varten valittiin komentorivivaihtoehto.

Anaconda-ympäristön pystyttämisen ja tarvittavien pakettien asentamisen jälkeen Neural Amp Modeler on valmis käytettäväksi. Koulutusta varten tarvitaan syöte-tavoite-pari, jossa tavoiteäänite on saatu äänittämällä syöteäänite mallinnettavan kitaravahvistimen läpi.

Varsinainen koulutus aloitetaan seuraavanlaisella komennolla:

```
python bin/train/main.py \  
bin/train/inputs/data/single_pair.json \  
bin/train/inputs/models/wavenet.json \  
bin/train/inputs/learning/default.json \  
bin/train/outputs/MyAmp
```

main.py-tiedostolle tarjottavista argumenteista single_pair.json-tiedostossa määrätään koulutuksessa käytettävät tiedostot. wavenet.json -tiedostossa määritellään käytettävä arkkitehtuuri ja muita verkkoon liittyviä parametrejä kuten aktivointifunktio. Neural Amp Modeler käyttää oletuksena WaveNet-pohjaista toteutusta, mutta valittavissa on myös yksinkertaisempi konvoluutioverkkoihin perustuva ja LSTM -pohjainen arkkitehtuuri. default.json-tiedostossa määrätään mm. eräkokko (batch size) ja epookkien lukumäärä. Viimeinen argumentti kertoo mihin hakemistoon ja millä nimellä malli tallennetaan.

Onnistuneen koulutuksen jälkeen valitussa hakemistossa on model.nam-tiedosto, joka sisältää painot. Tiedosto ladataan digitaalisen audiotyöaseman (DAW) Neural Amp Modeler

-liitännäiseen, jossa mallia voidaan käyttää.

3 Menetelmä

Tässä luvussa kuvataan käytetty menetelmä vaiheittain. Neural Amp Modelerin koulutustulosta arvioitiin kouluttamalla eri malleja eri koulutuspareilla ja syöttämällä testiäänite sekä saatujen mallien että koulutettavana olevan vahvistimen läpi. Saatuja ulostuloja verrattiin useammalla metriikalla.

3.1 Koulutusaineiston syötteen

Neural Amp Modelerin koulutustulosta eri koulutusaineistolla päätettiin arvioida kouluttamalla useita malleja eri koulutusaineistoilla. Syötetiedostona käytettiin NAM:in tarjoamaa syötettä, joka jaettiin kahdeksaan osaan seuraavasti:

- input0 = aikaväli 0:00–3:00 (koko alkuperäinen syöte)
- input1 = aikaväli 0:00–0:30
- input2 = aikaväli 0:30–1:00
- input3 = aikaväli 1:00–1:26
- input4 = aikaväli 1:26–1:48
- input5 = aikaväli 1:48–2:06
- input6 = aikaväli 2:06–2:31
- input7 = aikaväli 2:31–3:00.

Jakaminen pyrittiin tekemään noin 30 sekunnin välein, kuitenkin siten että datassa oli kuulemisen pohjalta luonnollinen kohta katkaista (esimerkiksi lyhyt tauko tai selkeä muutos datan äänissä). Tavoiteäänitteet saatiin ajamalla syötetiedostot vahvistimen läpi. Vahvistinsignaaleista kerrotaan luvussa 3.3.

3.2 Testiaineiston syötteen

Neural Amp Modelerin koulutustuloksen arviointia varten tarvitaan syöte-tavoitepareja, joita ei ole käytetty koulutusvaiheessa. Koska NAM:ia oletuksena käytetään erilaisten kitarasignaalien kanssa, valittiin tähän tutkimukseen testisyötteiksi kitarasignaaleja sisältävä kokoel-

ma, jonka Kehling ym. (2014) kehittivät tutkimustaan varten. Samaa kokoelmaa ovat sittemmin hyödyntäneet esimerkiksi Cassidy ja De Sena (2023), Wright ym. (2020), Damskäg, Juvela ja Välimäki (2019) ja Wright, Damskäg ja Välimäki (2019). Tammikuussa 2016 julkaistu kokoelman versio 2.0 sisältää neljä eri alakokoelmaa, joista *dataset 1–3* sisältävät äänitteitä yksittäisistä nuoteista ja lyhyistä soittopätkistä. *dataset 4* sisältää hieman pidempiä ottoja kahdeksan eri genren musiikkikappaleista kahdella eri sähkökitaralla ja kahden eri kitaristin soittamana. Tätä tutkimusta varten koottiin yksi neliminuuttisen testiäänite *dataset 4*:stä rock_blues ja metal -genreistä. Rock, blues ja metal -genreissä on tyypillisempää soittaa säröisemmällä vahvistimen asetuksella, kuin esimerkiksi kantrissa tai jazzissa. Tässä tutkimuksessa mallinnettiin vahvistin säröisellä asetuksella. Kokoamisvaiheessa uudelleen näytteistettiin testiäänite alkuperäisen kokoelman näytteenottotaajuudesta 44100 Hz uuteen näytteenottotaajuuteen 48000 Hz. Molemmat näytteenottotaajuudet ovat tyypillisiä musiikkiteknologiassa, mutta NAM tukee toistaiseksi vain 48 kHz näytteenottotaajuutta.

Tavoitesignaaleina käytettiin vahvistinsignaaleja, jotka on luotu samoin kuin koulutusaineistoin tavoitesignaalit.

3.3 Vahvistinsignaalit

Sekä koulutus- että testisyötteet ajettiin Hughes&Kettner-kitaravahvistimen läpi, ja ulostulo äänitettiin. Hughes&Kettner TubeMeister 20 Deluxe -vahvistin sisältää keinokuorman, jolloin siitä voidaan äänittää päätevahvistimelta tuleva signaali linjatasoisena, eikä sitä tarvitse kytkä kaiutinkaappiin (Hughes&Kettner 2015). Kitarasignaalit saatiin lähetettyä tietokoneesta instrumenttitasoisena käyttämällä USB-äänikorttina Line6 Pod Go:ta (Line6 2020).

3.4 Neural Amp Modelerin koulutus

NAM koulutettiin aiemmin kuvatuilla kahdeksalla syöte-tavoite-parilla, kukin pari usealla epookkimäärällä. Tutkimusta varten NAM:in lähdekoodia muokattiin. Muokattu NAM ajaa koulutuksen päätyttyä testiäänityksen tuoreen mallin läpi ja tallentaa ulostulon tiedostoon. Lisäksi näin koulutuksesta saa talteen siihen kulununeen ajan CT (Computation time) sekä epookkilukumäärän. Muokattu ohjelmakoodi on luettavissa GitHub:issa: <https://github.com/>



Kuvio 5. Hughes&Kettner -vahvistin ja kitarasignaalien äänittäminen.

tupepi/NAM-Neural-Amp-Modeler---modattu.

Suoritin koulutuksen Asus Rog Strix GL703GE -kannettavalla ja NVIDIA GeForce GTX 1050 Ti -grafiikkasuorittimella. Neuroverkolle käytettiin epookkien lukumäärää lukuun ottamatta NAM:in oletusparametreja.

3.5 Tulosten arviointimittarit

NAM:in eri mallien vasteita testisyötteelle sekä alkuperäistä vahvistimesta saatua testitaivoitesignaalia verrattiin Matlab-ohjelmistossa. Matlabissa kirjoitettiin sopivia skriptejä ja funktioita, joilla saatiin vertailtua äänitteiden samankaltaisuutta. Samankaltaisuuden arviointiin käytettiin useampia mittareita. Matlab-skriptit ovat luettavissa GitHub:issa: https://github.com/tupepi/progradu_matlab.

Ensimmäisenä RMSE (Root Mean Squared Error):

$$\begin{aligned}
 RMSE &= \sqrt{\frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|^2} \\
 &= \frac{\sqrt{\sum_{i=1}^n |\hat{y}_i - y_i|^2}}{\sqrt{n}} \\
 &= \frac{\|\hat{\mathbf{y}} - \mathbf{y}\|}{\sqrt{n}}
 \end{aligned} \tag{3.1}$$

missä \hat{y}_i on vahvistinsignaalin näyte ajanhetkellä i ja y_i :t on vastaavasti NAM:in emuloima

vahvistinsignaali ajanhetkellä i . Signaalien indeksointi alkaa yhdestä ja n on signaalien pituus. RMSE on yleinen signaalien vertailussa ja sitä on käytetty myös kitaravahvistinmallinusten ja kitaravahvistimien vertailussa (Covert ja Livingston 2013) (Schmitz ja Embrechts 2018).

Kaavassa merkittiin $\hat{\mathbf{y}}$ ja \mathbf{y} välisen erotuksen Euklidinen normi merkinnällä $\|\cdot\|$, jotta nähdään selvemmin yhteys ESR:n (Error-to-Signal-Ratio) kaavaan:

$$ESR = \frac{\|\hat{\mathbf{y}} - \mathbf{y}\|}{\|\hat{\mathbf{y}}\|}, \quad (3.2)$$

missä normalisointi tehdään pituuden neliöjuuren sijaan tavoitesignaalin Euklidisella normilla. ESR:ää on lähtökohtaisesti käytetty tappiofunktiona aiemmissa tutkimuksissa. Tällöin \mathbf{y} ja $\hat{\mathbf{y}}$ on suodatettu ylipäästösuotimella ennen ESR:n laskemista, sillä on huomattu, että ylätaajuuksien korostaminen verkon kouluttamisessa auttaa verkon suoriutumista ylätaajuuksilla (Wright ym. 2020) (Damskägg ym. 2019) (Damskägg, Juvela ja Välimäki 2019) (Wright, Damskägg ja Välimäki 2019). Kuitenkin verkon testaamiseen on käytetty suodattamatonta versiota ESR:stä (Damskägg, Juvela ja Välimäki 2019).

Seuraava mittari perustuu MFCC-kertoimiin (Mel-frequency cepstral coefficients):

$$MFCC_s = 1 - \frac{1}{N} \sum_{n=1}^N \frac{\hat{y}_{MFCC_n} \cdot y_{MFCC_n}}{\|\hat{y}_{MFCC_n}\| \|y_{MFCC_n}\|}, \quad (3.3)$$

missä \hat{y}_{MFCC_n} :t ovat vahvistinsignaalista lasketut MFCC-vektorit, ja y_{MFCC_n} :t ovat mallinussignaalista lasketut MFCC-vektorit. N on vektorien lukumäärä. MFCC laskettiin Matlabin `mfcc`-funktiolla. Ikkunan koko 4096 näytettä, ja ikkunoiden päällekkäisyys puolet tästä, 2048 näytettä. Näin saatiin $N=13$ vektoria kustakin signaalista, joiden väliltä laskettiin normalisoitu pistetulo, ja otettiin tulosten keskiarvo (Martínez Ramírez, Benetos ja Reiss 2020). Keskiarvo vähennettiin yhdestä, jotta saatiin 0 vastaamaan identtisyyttä, kuten aiemmissa mittareissa.

Vastaavanlaisesti muodostettiin myös kaksi muuta mittaria:

$$PSDs = 1 - \frac{\hat{y}_{PSD} \cdot y_{PSD}}{\|\hat{y}_{PSD}\| \|y_{PSD}\|}, \quad (3.4)$$

missä \hat{y}_{PSD} on vahvistinsignaalista laskettu PSD (power spectral density) ja y_{PSD} on mallinnetusta vahvistinsignaalista laskettu PSD. Ja:

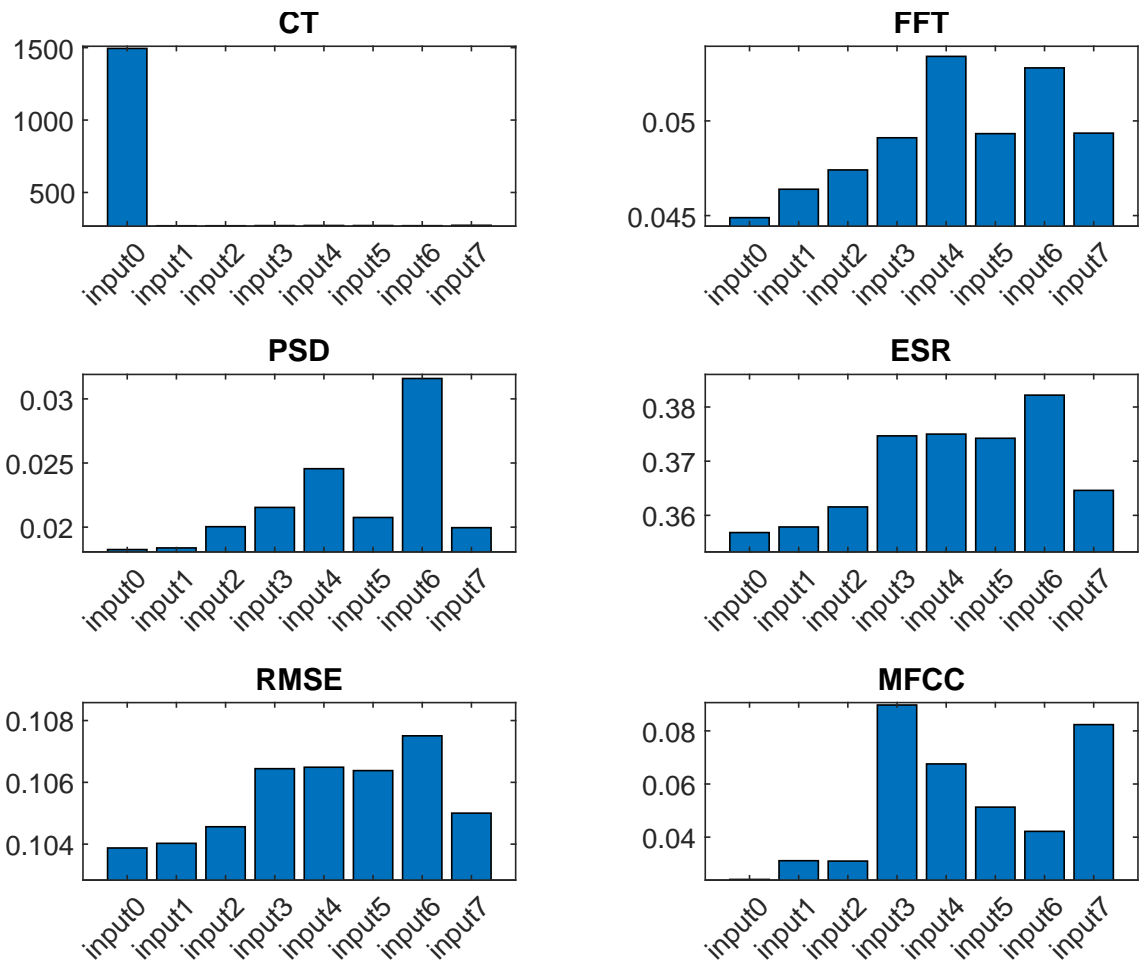
$$FFTs = 1 - \frac{\hat{y}_{FFT} \cdot y_{FFT}}{\|\hat{y}_{FFT}\| \|y_{FFT}\|}, \quad (3.5)$$

missä \hat{y}_{FFT} on vahvistinsignaalista laskettu FFT ja y_{FFT} on mallinnuksensignaalista laskettu FFT.

3.6 Menetelmän validointi alustavilla mittauksilla

Kuvatulla menetelmällä toteutettiin koulutus eri syöte-tavoite-pareilla ja seuraavilla epookkimäärillä: 100, 90, 80, 70, 60, 50, 40, 30, 20, 10, 5, 4, 3, 2, 1. Alkuperäisen vahvistimen ja mallin, joka oli koulutettu 100 epookilla ja koulutusdatalla input0, välillä ei havaittu kuuntelemalla merkittävää eroa. 100 epookilla ja koulutusdatoilla input1–7 koulutettujen mallien tuottamat äänet olivat erotettavissa vahvistimen tuottamasta äänitteestä. Erityisesti input4 oli eroteltavissa kuuntelemalla. Lyhyemmällä syöte-tavoite-äänitteillä kouluttaminen oli nopeampaa kuin pidemmällä, kuten olikin odotettavissa. Kuviossa 6 nähdään syötteen input6 heikompi suoriutuminen kaikilla mittareilla paitsi MFCC:llä, jolla input3 oli heikoin. Kuulohavaintoa syötteen input4 huonommasta suoriutumisesta tukee ainakin FFT.

Alustavien tuloksien pohjalta analyysiä muokattiin ja saatiin lopulliset tulokset, jotka esitellään luvussa (4).



Kuvio 6. Alustavat testitulokset, kun malli on koulutettu eri koulutusdatoilla ja 100 epookilla. CT (Computation time) on koulutukseen kulunut aika.

4 Tulokset

Tässä luvussa käydään läpi tutkimuksen tulokset.

4.1 Muutokset alustavaan menetelmään

Alustavien tulosten jälkeen päätettiin kouluttaa malleja syötepareilla input1–7 ja suuremmalla epookkimäärällä, jotta koulutusaika vastaisi input0:n ja 100 epookin koulutusaikaa, eli noin 1600 sekuntia. Lisäksi uusia syöte-tavoite-pareja lisättiin. Uudet syöte-tavoite-parit muodostettiin NAM:in alkuperäisestä syöte-tavoite-parista seuraavasti:

- inputA = aikaväli 0:30–3:00
- inputB = aikavälit 0:00–0:30 & 1:00–3:00
- inputC = aikavälit 0:00–1:00 & 1:26–3:00
- inputD = aikavälit 0:00–1:26 & 1:48–3:00
- inputE = aikavälit 0:00–1:48 & 2:06–3:00
- inputF = aikavälit 0:00–2:06 & 2:31–3:00
- inputG = aikaväli 0:00–2:31.

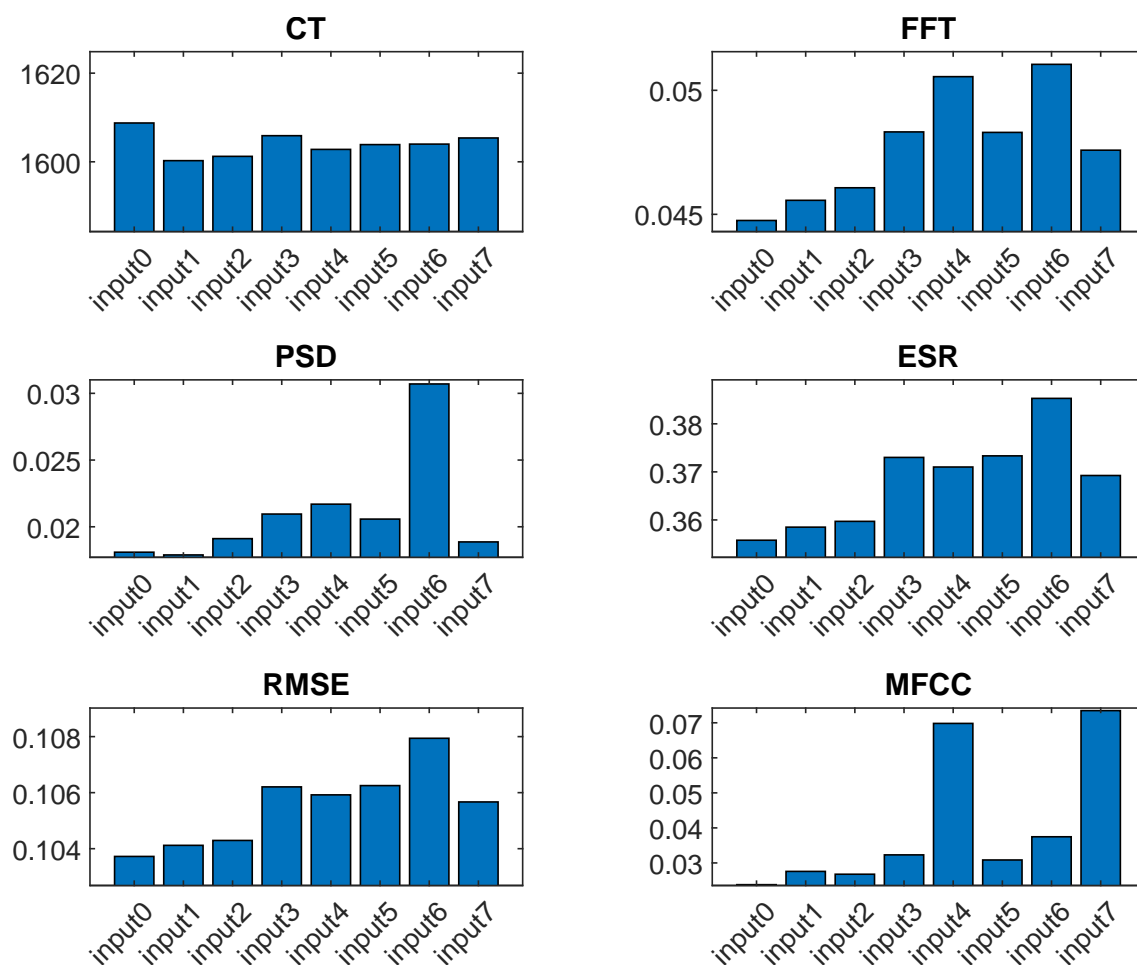
Näin saatiin arvioitua mitä vaikutusta koulutustulokseen on, jos alkuperäisestä koulutusdatasta jätetään pois jokin osajoukoista input1–7.

Uudessa vertailussa haluttiin vertailla signaaleiden taajuussisältöä tarkemmin, ajan suhteen. Taajuussisällön tarkempi vertailu tehtiin STFT:hen (Short-time Fourier Transform) perustavalla spektrogrammilla. Matlabin `spectrogram`-funktiolla laskettiin kahdelle vertailtavalle äänitteelle spektrogrammi, vähennettiin nämä toisistaan ja piirrettiin kuvaaja. Kuvaajissa tummuus vastaa samankaltaisuutta, ja vaaleus eroavaisuutta.

4.2 Eri syötteiden yleinen vertailu

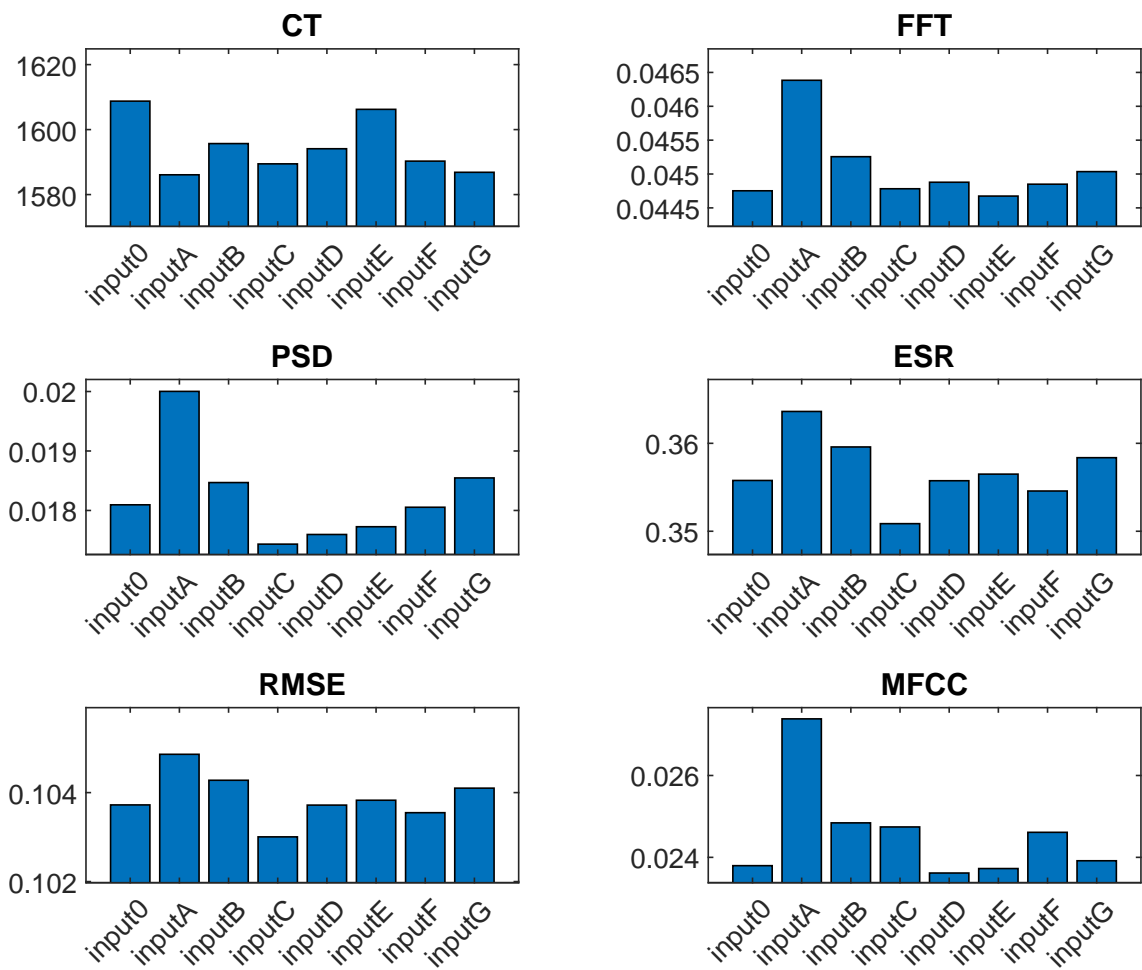
Lisäämällä lyhyemmällä pätkillä tehtyjen mallien koulutusaikaa vastaamaan input0:n ja sadan epookin koulutusaikaa (Kuvio 7), ei tapahtunut merkittävää parantumista malleissa ver-

rattuna alustaviin tuloksiin (Kuvio 6).



Kuvio 7. Tulokset eri syötteillä input1–7 ja noin 1600 sekunnin koulutusajalla.

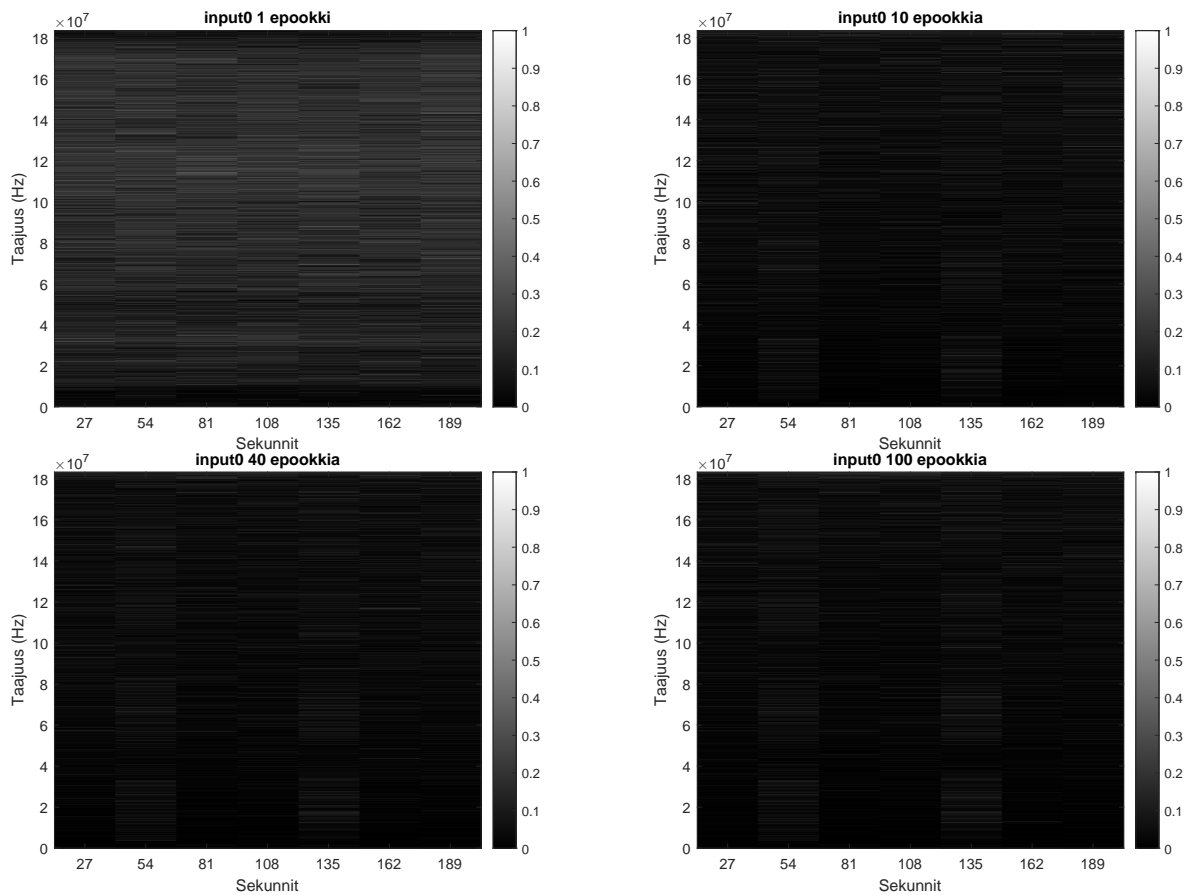
Uusilla syöte-tavoite-pareilla inputA-G tehtiin vastaavat vertailut (Kuvio 8). Kuten oli odotettavissa, inputA–G suoriutuivat vastaavassa koulutusajassa paremmin kuin input1–7. Kaikilla mittareilla inputA oli heikoin. Ottaen huomioon input1:n hyvän suoriutumisen, voitaisiin päätellä, että alkuperäisen syöttödatan osa 0:00–0:30 sisältää dataa, joka auttaa mallia oppimaan tehokkaasti, ainakin valitulla testidatalla ja vahvistimella. Vastaava pätee myös inputB (toisiksi heikoin kaikilla) ja input2 kanssa, eli myös alkuperäisen koulutusdatan osa 0:30–1:00 auttaa koulutusta. inputC pärjasi MFCC:tä lukuun ottamatta yhtä hyvin tai jopa paremmin kuin input0. input6 pärjasi useammalla mittarilla muita heikommin, mutta ilman vastaavaa osaa 2:06–2:31 (inputF) kouluttaminen ei ollut erityisesti muita parempi.



Kuvio 8. Tulokset eri syötteillä inputA–G ja noin 1600 sekunnin koulutusajalla.

4.3 Spektrogrammit

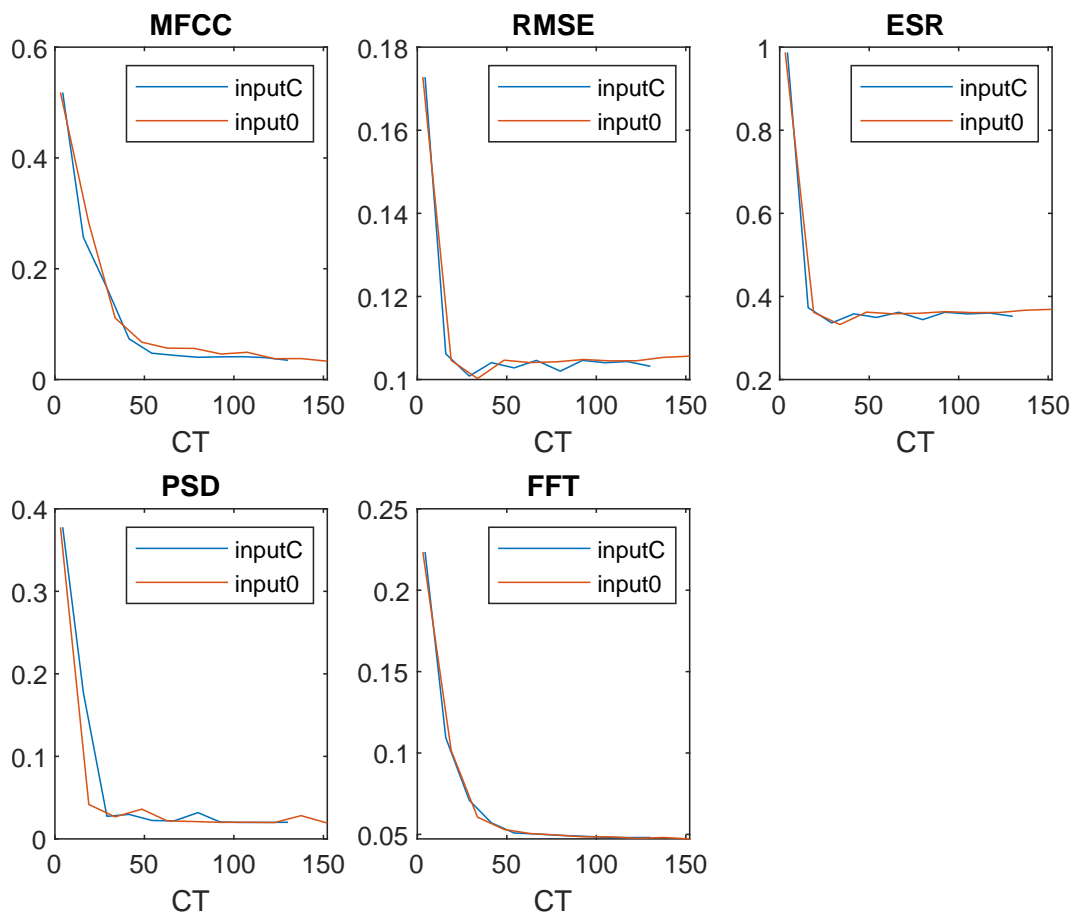
Kuviosta 9 nähdään vahvistimen ja mallien tuottamien äänitteiden spektrogrammien erotukset testidatalla. Pystyakselilla on taajudet 0–24000 Hz. Vaaka-akselilla on ikkunoinnin ajanhetket testiäänitteeltä. Huomataan, että kuvaaja tummenee epookkeja lisättäessä, ja vähintään 10 epookkimäärän jälkeen on jo vaikea nähdä eroa eri kuvaajissa. Merkittävää eroa ei myöskään ole eri taajuuksilla, poikkeuksena yhden epookin kuvaaja, jossa matalimmat taajuudet ovat muita tummempia.



Kuvio 9. input0:lla koulutetun mallin tuottaman äänitteen spektrogrammin ja vahvistimen tuottaman äänitteen spektrogrammin erotukset eri epookkimäärillä. Voimakkuudet on normalisoitu ensimmäisen kuvan suurimmalla arvolla.

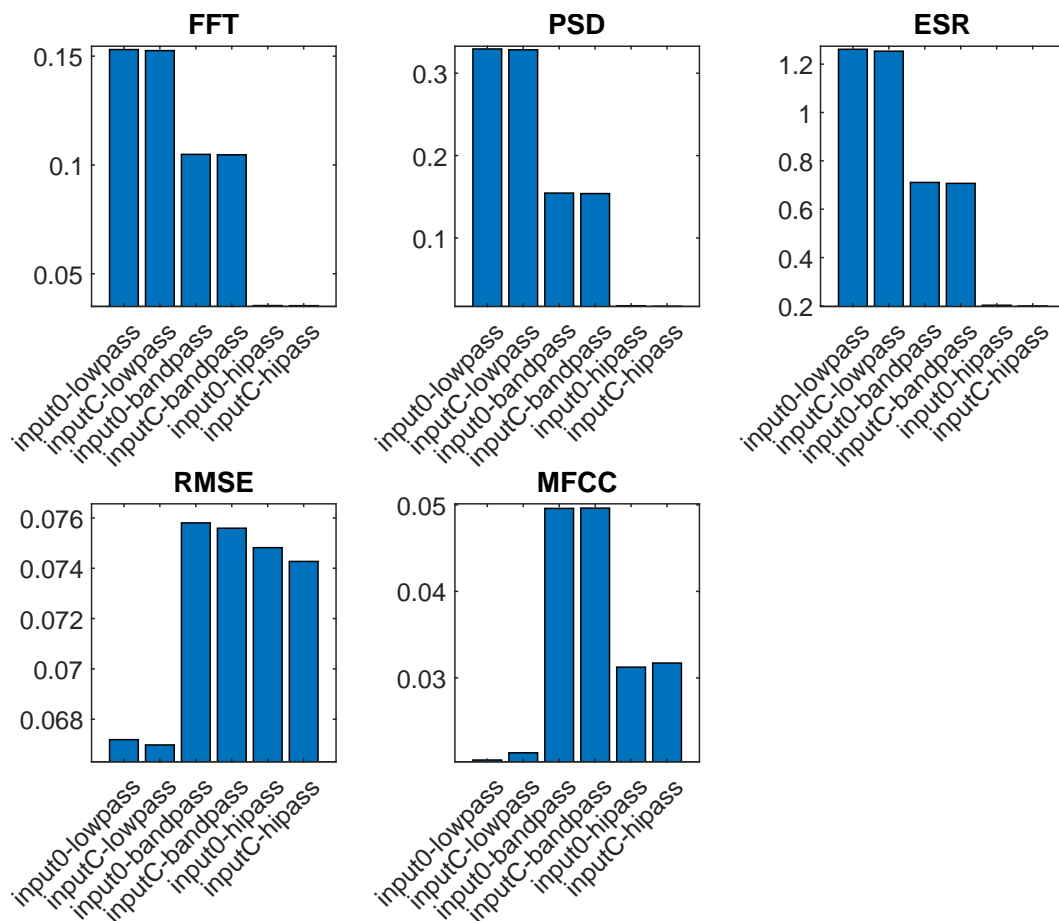
4.4 Syötteiden input0 ja inputC tarkempi vertailu

Syötteillä input0 ja inputC koulutettujen mallien ja mallinnettavan vahvistimen kanssa tehdyn vertailun tulokset koulutusajan suhteen on nähtävissä kuviossa 10. Kuvioista nähdään, että molemmilla syötteillä koulutus paranee nopeimmin ensimmäisen n. 50 sekunnin aikana (mikä vastaa noin viittä epookkia), jonka jälkeen oppiminen tasaantuu. Käyrät ovat kaikilla mittareilla melko lähellä toisiaan.



Kuvio 10. input0 ja inputC -syötteillä koulutettujen mallien suoriutuminen testidatalla verrattuna mallinnettavaan vahvistimeen, koulutusajan suhteen.

Syötteillä input0 ja inputC koulutettuja malleja vertailtiin myös ajamalla äänitteet suotimien läpi ennen vertailua. Kuviossa 11 nähdään syötteillä inputC ja input0 koulutettujen mallien samankaltaisuus vahvistimen kanssa, kun testiäänite ajettiin seuraavien suotimien läpi: alipäästösuodin 1000 Hz, kaistanpäästösuodin 1000 Hz--2000 Hz, ylipäästösuodin 2000 Hz. Kahden eri syötteen välillä ei juuri eroa näy, mutta eri taajuusalueilla ja eri mittareilla on havaittavissa eroja.



Kuvio 11. input0 ja inputC -syötteillä koulutettujen mallien suoriutuminen testidatalla verrattuna mallinnettavaan vahvistimeen, kun testidata on ajettu eri suotimien läpi.

4.5 Pohdinta

Yksittäisen lyhyen osan koulutustuloksesta ei voida välttämättä päätellä sen hyötyjä osana isompaa koulutusdataa. Tutkimusta voitaisiin jatkaa analysoimalla systemaattisesti koulutusdatan suoriutumista, kun siitä otetaan pieniä osia pois. Näin saataisiin parempi käsitys siitä, mitkä osat koulutusdatasta auttavat merkittävästi koulutustulosta. inputC:n suhteellisen hyvä suoriutuminen antaa ymmärtää, että koulutusdatasta saattaa olla löydettävissä lisää osia, jotka eivät auta merkittävästi koulutustulosta. Koulutusdata olisi siis mahdollisesti lyhennettävissä. Lyhentämällä tuskin voidaan nopeuttaa koulutusta, säilyttäen sama koulutustulos. Lyhentämällä syötettä voidaan kuitenkin nopeuttaa tavoiteäänitteen luomista, sillä vahvistimen vaste äänitetään reaaliajassa. Lyhyemmästä syötteestä olisi etua etenkin, jos vahvistimia

mallinnetaan useita, tai useilla eri asetuksilla.

Aiempaa järjestelmällistä syötteiden sisällön vertailua kitaravahvistinten mallintamisen saralla ei oltu tehty. Myöskään Neural Amp Modeleria ei juurikaan ole arvioitu akateemisissa tutkimuksissa. Tämän tutkimuksen perusteella Neural Amp Modeler tarjoamansa syötteensä kera pystyy saavuttamaan koulutustuloksen, joka on hyvin lähellä oikean vahvistimen ääntä. Rajoittamalla syöte-tavoite-paria eri tavoin, koulutustulos vaihteli. Eli koulutusdatan määrän lisäksi sisällöllä on vaikutuksia koulutustulokseen.

Jatkotutkimuksena voitaisiin yrittää korvata syötteen osia muulla sisällöllä, ja arvioida vaikuttaako korvaaminen koulutustulokseen. Tulevaisuudessa voitaisiin myös tutkia, yleistyvätkö tulokset muille neuroverkkopohjaisille kitaravahvistimien mallintajille kuin Neural Amp Modelerille. Lisäksi tutkimus voitaisiin toistaa eri vahvistimilla ja eri testidatalla.

5 Yhteenveto

Kitaravahvistimien neuroverkko mallintamista on tutkittu viime vuosina. Aiemmat tutkimukset keskittyvät mm. verkon rakenteen, mallin realistisuuden sekä reaaliaikaismahdollisuuksien arviointiin. Tässä tutkimuksessa arvioitiin, mitä vaikutuksia neuroverkon koulutustulokseen on koulutusdatan muutoksilla.

Avoimen lähdekoodin neuroverkkopohjainen kitaravahvistimien mallintaja Neural Amp Modeler on kerännyt paljon suosiota kitaristien keskuudessa. Neural Amp Modeleria käytettiin tässä tutkimuksessa neuroverkkototeutuksena.

Neural Amp Modelerin koulutustulosta arvioitiin kouluttamalla samasta vahvistimesta useita eri malleja eri koulutuspareilla. Koulutusparit luotiin syöttämällä Neural Amp Modelerin tarjoama syöteäänite vahvistimen läpi, ja samalla äänittäen ulostulo. Tätä paria pilkkomalla muodostettiin eri syöte-tavoite-pareja. Koulutuspareja luotaessa vahvistimen läpi äänitettiin myös testiäänite. Sama testiäänite ajettiin myös luotujen mallien läpi. Eri mallien ja vahvistimen vastetta testiäänitteelle vertailemalla saatiin arvioita eri mallien suoriutumisesta.

Huomattiin, että jakamalla alkuperäinen Neural Amp Modelerin koulutusdata osiin, eri osat suoriutuivat eri tavalla. Lyhyillä, alle 30 sekunnin osilla, koulutustulos on huonompi kuin alkuperäisellä datalla ja vastaavalla koulutusajalla. Vähän lyhennetyllä datalla, noin 2 min 30 s, koulutustulos ei huonontunut yhtä paljon, ja jotkin osat suoriutuivat yhtä hyvin alkuperäisen datan kanssa. Lyhyemmällä syötteellä voidaan nopeuttaa koulutusparien luomista.

Lähteet

- Barbour, Eric. 1998. “The cool sound of tubes [vacuum tube musical applications]”. *IEEE Spectrum* 35 (8): 24–35.
- Bishop, Christopher M. 1995. *Neural networks for pattern recognition*. Oxford university press.
- Cassidy, Will J ja Enzo De Sena. 2023. “Perceptual Evaluation and Genre-Specific Training of Deep Neural Network Models of a High-Gain Guitar Amplifier”. Teoksessa *International Conference on Digital Audio Effects (DAFx23)*.
- Covert, John ja David L Livingston. 2013. “A vacuum-tube guitar amplifier model using a recurrent neural network”. Teoksessa *2013 Proceedings of IEEE Southeastcon*, 1–5. IEEE.
- Damskäg, Eero-Pekka, Lauri Juvela, Etienne Thuillier ja Vesa Välimäki. 2019. “Deep learning for tube amplifier emulation”. Teoksessa *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 471–475. IEEE.
- Damskäg, Eero-Pekka, Lauri Juvela ja Vesa Välimäki. 2019. “Real-time modeling of audio distortion circuits with deep learning”. Teoksessa *Sound and music computing conference*, 332–339. Sound ja Music Computing Association.
- He, Kaiming, Xiangyu Zhang, Shaoqing Ren ja Jian Sun. 2016. “Deep residual learning for image recognition”. Teoksessa *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.
- Herbst, Jan-Peter. 2019. “Empirical explorations of guitar players’ attitudes towards their equipment and the role of distortion in rock music”.
- Hershey, Shawn, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold ym. 2017. “CNN architectures for large-scale audio classification”. Teoksessa *2017 IEEE international conference on acoustics, speech and signal processing (icassp)*, 131–135. IEEE.

- Hughes&Kettner. 2015. *Hughes&Kettner TubeMeister 20 Deluxe Manual*. Hughes&Kettner. Viitattu 20. toukokuuta 2024. https://hughes-and-kettner.com/?jet_download=07a427bec634e0727e89b3114f3b6c0f925a95ad.
- Jin, Xiangjun, Jie Shao, Xin Zhang, Wenwei An ja Reza Malekian. 2016. “Modeling of nonlinear system based on deep learning framework”. *Nonlinear Dynamics* 84:1327–1340.
- Keepports, David. 2017. “The warm, rich sound of valve guitar amplifiers”. *Physics Education* 52 (2): 025010.
- Kehling, Christian, Jakob Abeßer, Christian Dittmar ja Gerald Schuller. 2014. “Automatic Tablature Transcription of Electric Guitar Recordings by Estimation of Score-and Instrument-Related Parameters.” Teoksessa *DAFx*, 219–226.
- Kiranyaz, Serkan, Turker Ince, Osama Abdeljaber, Onur Avci ja Moncef Gabbouj. 2019. “1-D convolutional neural networks for signal processing applications”. Teoksessa *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 8360–8364. IEEE.
- Kuznetsov, Boris, Julian D Parker ja Fabián Esqueda. 2020. “Differentiable IIR filters for machine learning applications”. Teoksessa *Proc. Int. Conf. Digital Audio Effects (eDAFx-20)*, 297–303.
- Lago, Nelson Posse ja Fabio Kon. 2004. “The quest for low latency”. Teoksessa *International Computer Music Conference*.
- Line6. 2020. *Line6 Pod Go Owner’s Manual*. Yamaha Guitar Group, Inc. Viitattu 20. toukokuuta 2024. <https://line6.com/data/6/0a020a41cde35e68fab84d2fe/application/pdf/POD%20Go%201.10%20Owner’s%20Manual%20-%20English%20.pdf>.
- Martínez Ramírez, Marco A, Emmanouil Benetos ja Joshua D Reiss. 2020. “Deep learning for black-box modeling of audio effects”. *Applied Sciences* 10 (2): 638.
- “Neural Amp Modeler”. 2024. Viitattu 20. toukokuuta 2024. <https://www.neuralampmodeler.com/>.
- “Neural Amp Modeler (NAM) - Facebook ryhmä”. 2024. Viitattu 20. toukokuuta 2024. <https://www.facebook.com/groups/neuralampmodeler>.

- “neural-amp-modeler GitHub”. 2019. Viitattu 20. toukokuuta 2024. <https://github.com/sdatkinson/neural-amp-modeler>.
- Presonus. 2017. “Digital Audio Latency Explained. Technical Report.” Viitattu 20. toukokuuta 2024. <https://www.presonus.com/learn/technical-articles/Digital-Audio-Latency-Explained>.
- Raitoharju, Jenni. 2022. “Convolutional neural networks”. Teoksessa *Deep learning for robot perception and cognition*, 35–69. Elsevier.
- Ramírez, Marco A Martínez ja Joshua D Reiss. 2019. “Modeling nonlinear audio effects with end-to-end deep neural networks”. Teoksessa *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 171–175. IEEE.
- Ripley, Brian D. 2007. *Pattern recognition and neural networks*. Cambridge university press.
- Schmitz, Thomas ja Jean-Jacques Embrechts. 2018. “Real time emulation of parametric guitar tube amplifier with long short term memory neural network”. *arXiv preprint arXiv:1804.07145*.
- Tejo, Nicholas Matthew. 2023. “Evaluasi subjektif terhadap kualitas Neural Amp Modeler dalam teknologi pemodelan ulang pengeras bunyi dan efek gitar elektrik”. Tutkielma, Universitas Pelita Harapan.
- Thomas, Schmitz. 2019. “Nonlinear modeling of the guitar signal chain enabling its real-time emulation”. Tohtorinväitöskirja, University of Liège.
- Van Den Oord, Aaron, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu ym. 2016. “WaveNet: A generative model for raw audio”. *arXiv preprint arXiv:1609.03499* 12.
- Van Den Oord, Aäron, Nal Kalchbrenner ja Koray Kavukcuoglu. 2016. “Pixel recurrent neural networks”. Teoksessa *International conference on machine learning*, 1747–1756. PMLR.
- Vanhatalo, Tara, Pierrick Legrand, Myriam Desainte-Catherine, Pierre Hanna, Antoine Brusco, Guillaume Pille ja Yann Bayle. 2022. “A review of neural network-based emulation of guitar amplifiers”. *Applied Sciences* 12 (12): 5894.

- Venables, William N ja Brian D Ripley. 2013. *Modern applied statistics with S-PLUS*. Springer Science & Business Media.
- Wright, Alec, Eero-Pekka Damskägg, Lauri Juvela ja Vesa Välimäki. 2020. “Real-time guitar amplifier emulation with deep learning”. *Applied Sciences* 10 (3): 766.
- Wright, Alec, Eero-Pekka Damskägg ja Vesa Välimäki. 2019. “Real-time black-box modeling with recurrent neural networks”. Teoksessa *International Conference on Digital Audio Effects*. University of Birmingham.
- Zaniolo, Luiz ja Oge Marques. 2020. “On the use of variable stride in convolutional neural networks”. *Multimedia Tools and Applications* 79 (19): 13581–13598.
- Zaras, Adamantios, Nikolaos Passalis ja Anastasios Tefas. 2022. “Neural networks and back-propagation”. Teoksessa *Deep Learning for Robot Perception and Cognition*, 17–34. Elsevier.
- Zhang, Zhichen, Edward Olbrych, Joseph Bruchalski, Thomas J McCormick ja David L Livingston. 2018. “A vacuum-tube guitar amplifier model using long/short-term memory networks”. Teoksessa *SoutheastCon 2018*, 1–5. IEEE.