JYVÄSKYLÄN YLIOPISTO
MATEMATIIKAN JA TILASTO-
TIETEEN LAITOS

Pro gradu -tutkielma

Transformer -neuroverkon robustisuuden parantaminen Gaussisen prosessin ja neuroverkon yhdistävillä menetelmillä

Antti Yläjärvi

29. toukokuuta 2023

Tekijä
Antti Yläjärvi

Otsikko
Transformer -neuroverkon robustisuuden parantaminen Gaussisen prosessin ja neuroverkon yhdistävillä menetelmillä

Tutkinto-ohjelma
Tilastotieteen ja datatieteen maisteriohjelma

Päivämäärä	**Sivumäärä**
29. toukokuuta 2023	60

Tiivistelmä

Neuroverkkojen sovellukset ovat yleistyneet viimeisen kymmenen vuoden aikana. Nykyisin neuroverkkoja sovelletaan useilla aloilla, kuten lääketieteellisessä kuvantamisessa ja diagnostiikassa tai itseohjautuvissa autoissa, joilla ennusteen lisäksi tieto tämän uskottavuudesta voi olla kriittistä. Robustisuus ja epävarmuuden luotettava kvantifiointi on tärkeää myös muilla sovellusaloilla erityisesti, kun datan volyymi voi ajaa virheiden kumulatiiviset kustannukset merkittäviksi. On kuitenkin todettu, ettei esimerkiksi neuroverkkojen ennustamia todennäköisyyksiä voida yleisesti pitää luotettavina mittoina ennusteeseen liittyvälle epävarmuudelle.

Neuroverkkojen ennusteen epävarmuuden kvantifioimiseksi ja robustisuuden parantamiseksi on kirjallisuudessa esitetty kymmeniä erilaisia menetelmiä. Yksi tutkimussuunta on neuroverkon ja Gaussisen prosessin yhdistävät menetelmät. Gaussiset prosessit ovat joustavia parametrittomia Bayesiläisiä malleja, jotka voidaan yhdistää neuroverkkoon käytännössä kajoamatta alkuperäiseen malliin tai tämän arkkitehtuuriin. Menetelmien eduksi voidaan lukea ennusteen saaminen piste-estimaatin sijaan jakaumana, mikä mahdollistaa myös varianssin hyödyntämisen epävarmuutta koskevassa päättelyssä.

Tässä tutkielmassa esitellään kaksi verrattain uutta kirjallisuudessa esitettyä Gaussisen prosessin ja neuroverkon yhdistävää menetelmää sekä tarkastellaan, voidaanko näillä parantaa Transformer -neuroverkon robustisuutta ja epävarmuuden kvantifioinnin luotettavuutta. Menetelmiä tarkastellaan luonnollisen kielen käsittelyyn liittyvässä tehtävässä, missä neuroverkkona käytetään Googlen tutkijaryhmän kehittämää, Transformer -arkkitehtuuriin perustuvaa, BERT-mallia. Tutkielmassa osoitettiin BERT-mallin olevan verrattain robusti. Mallin suorituskykyä todettiin kuitenkin voitavan parantaa Gaussista prosessia lisäksi soveltamalla. Merkittävimmät hyödyt todettiin poikkeavien havaintojen tunnistamisessa, missä Gaussisella prosessilla ennusteelle saatava varianssi osoittautui selvästi ennustevoimaisimmaksi mitaksi.

Sisällys

Johdanto

Neuroverkkojen sovellukset ovat yleistyneet viimeisen kymmenen vuoden aikana. Suosiota voidaan ainakin osin selittää neuroverkoilla useilla sovellusaloilla saavutetuilla aikaisempia merkittävästikin paremmilla tuloksilla. Neuroverkot ovat rikkoneet aikaisempia ennätyksiä konenäköön ja puheentunnistukseen liittyvissä tehtävissä (LeCun et al., 2015) sekä vallanneet sijoitukset esimerkiksi luonnollisen kielen käsittelyyn kehitetyille malleille tarkoitetussa GLUE -suorituskykytestissä (Wang et al., 2018). Nykyisin neuroverkkoja sovelletaan mm. lääketieteellisessä kuvantamisessa ja diagnostiikassa, itseohjautuvissa autoissa sekä kyberturvallisuuden sovelluksissa.

Suosion ja sovellusten kasvun myötä myös kiinnostus neuroverkkoihin liittyvään epävarmuuteen ja tämän kvantifiointiin sekä neuroverkkojen robustisuuteen on lisääntynyt. Neuroverkot eivät ole läpinäkyviä siinä mielessä, että näiden ennusteista voitaisiin päätellä, millä tavalla ennusteeseen on päädytty ja esimerkiksi arvioida ennusteen uskottavuutta (Marcus, 2018). Useilla sovellusaloilla, kuten juuri lääketieteellisessä diagnostiikassa tai itseohjautuvissa autoissa, myös tieto ennusteen uskottavuudesta voi olla kriittistä. Läpinäkymättömyys ei välttämättä ole ongelma, mikäli malli on robusti (Marcus, 2018) tai kykenee ilmaisemaan ennusteeseen liittyvää epävarmuutta siten, että tästä voidaan luotettavasti päätellä ennusteen uskottavuus. On kuitenkin todettu, ettei esimerkiksi neuroverkkojen ennustamia todennäköisyyksiä voida yleisesti pitää luotettavina mittoina ennusteeseen liittyvälle epävarmuudelle (Guo et al., 2017; Hendrycks & Gimpel, 2016).

Robustisuus ja ennusteen epävarmuuden luotettava ilmentäminen on tärkeää myös ei-kriittisillä sovellusaloilla mallia käytäntöön sovellettaessa. Vaikka mallin sovittamiseksi kerätyn aineiston on tarkoitus kuvata reaalimaailmaa (Torralba & Efros, 2011), se harvoin kykenee täysin karakterisoimaan todellista datan generoinutta populaatiojakaumaa (Hendrycks et al., 2020). Populaatiojakauman oletetaan tyypillisesti olevan staattinen eli muuttumaton mallin sovittamisen ja käytöntään soveltamisen välillä (Quinonero-Candela et al., 2008). Käytännössä näin ei kuitenkaan usein ole, jolloin hyvän mallin tulisi sekä kyetä ekstrapoloimaan eli laajentumaan tunnetun jakauman ulkopuolelle (Hendrycks et al., 2020; Marcus, 2018) että tunnistamaan sovelluksessa kohdattavassa jakaumassa tapahtuvia muutoksia. Epäonnistumisella voi olla suuria vaikutuksia mallin suorituskykyyn (Paleyes et al., 2022). Vaikka virheellisen luokittelun seuraukset yksittäistapauksessa eivät olisi potentiaalisilta seurauksiltaan yhtä kriittisiä kuin esimerkiksi lääketieteellisessä diagnostiikassa, voi datan volyymi ajaa virheiden kumulatiiviset kustannukset merkittäviksi.

Neuroverkkojen ennusteen epävarmuuden kvantifioimiseksi ja robustisuuden parantamiseksi on kirjallisuudessa esitetty kymmeniä erilaisia menetelmiä. Gawlikowski et al. (2021) jakaa menetelmät yhden eteenpäinsyötön deterministisiin menetelmiin, Bayesiläisiin neuroverkkoihin, kokoelmiin perustuviin menetelmiin ja testiaineiston augmentointiin perustuviin menetelmiin. Yhden eteenpäinsyötön deterministisillä menetelmillä tarkoitetaan kaikkia niitä menetelmiä, joilla estimaatti ennusteen epävarmuudelle saadaan yhdellä ennustavien muuttujien syötöllä läpi deterministisen neuroverkon. Bayesiläisissä neuroverkoissa deterministisen neuroverkon sijaan tämän parametreille oletetaan priorijakauma, ja epävarmuudesta tehdään päätelmiä

posterijakauman avulla. Kokoelmiin perustuvissa menetelmissä ennuste johdetaan useista mallien kokoelmaan kuuluvien mallien ennusteista, kun taas testiaineiston augmentointiin perustuvissa menetelmissä luodaan kokoelma ennusteita alkuperäistä syötettä muuntaen monistamalla.

Yksi neuroverkkoihin liittyvä tutkimussuunta on neuroverkon ja Gaussisen prosessin yhdistävät menetelmät, joilla on saatu hyviä tuloksia sekä kuvantunnistukseen liittyvissä tehtävissä (Ovadia et al., 2019; Liu et al., 2020; van Amersfoort et al., 2021b) että luonnollisen kielen käsittelyyn liittyvissä tehtävissä (Ovadia et al., 2019; Liu et al., 2020). Gaussiset prosessit (Rasmussen & Williams, 2006) ovat joustavia parametrittomia Bayesiläisiä malleja, jotka voidaan yhdistää neuroverkkoon käytännössä kajoamatta alkuperäiseen malliin tai tämän arkkitehtuuriin. Gaussisen prosessin neuroverkkoon yhdistämällä saaduilla malleilla ennuste ja estimaatti ennusteen epävarmuudelle saadaan yhdellä eteenpäinsyötöllä deterministisestä neuroverkosta, minkä lisäksi menetelmien eduksi voidaan lukea ennusteen saaminen piste-estimaatin sijaan jakaumana, mikä mahdollistaa myös varianssin hyödyntämisen epävarmuutta koskevassa päättelyssä.

Tässä tutkielmassa esitellään kaksi verrattain uutta kirjallisuudessa esitettyä Gaussisen prosessin ja neuroverkon yhdistävää menetelmää sekä tarkastellaan, voidaanko näillä parantaa neuroverkon robustisuutta ja epävarmuuden kvantifioinnin luotettavuutta. Neuroverkkojen robustisuutta ja epävarmuuden kvantifiointia tutkittaessa kiinnostuksen kohteena tyypillisesti on mallin yleistettävyys myös opetusaineistosta poikkeaville havainnoille sekä kyky ilmaista epävarmuutta, kun malli tekee luokitteluvirheen tai kohtaa poikkeavan havainnon.

Yleistettävyys poikkeaville havainnoille tarkoittaa mallin kykyä ennustaa havaintoja, jotka poikkevat sovitettaessa kohdatuista havainnoista ja liittyy keskeisesti mallin suoriutumiseen muuttuvassa ympäristössä. Epävarmuuden ilmentäminen luokitteluvirheiden tai poikkeavien havaintojen kohdalla taas liittyy mahdollisuuteen tunnistaa näitä, mitä voidaan hyödyntää esimerkiksi ohjaamalla todennäköisemmin väärin luokiteltuja tapauksia ihmisen arvioitavaksi. Poikkeavien havaintojen tunnistaminen puolestaan voi olla tavoite itsessään joillain sovellusaloilla, kuten esimerkiksi taloudellisten tapahtumien seurannassa (Pang et al., 2021). Lisäksi poikkeavista havainnoista voidaan tunnistaa uusia trendejä sekä ympäristössä tapahtuvia muutoksia, joiden tunnistamatta jäämisellä voi olla suuria vaikutuksia mallin suorituskykyyn käytännön sovelluksessa (Paleyes et al., 2022)

Epävarmuuden kvantifiointia ja neuroverkkojen robustisuutta koskeva tutkimus vaikuttaa pääasiassa keskittyneen konenäön ja kuvantunnistuksen sovelluksiin sekä näihin tarkoitettuihin malleihin. Tässä tutkielmassa malleja tarkastellaan luonnollisen kielen käsittelyyn liittyvässä tehtävässä, missä neuroverkkona käytetään Googlen tutkijaryhmän kehittämää BERT-mallia (*Bidirectional Encoder Representations from Transformers*; Devlin et al., 2018). BERT-malli on tarkoitettu erityisesti luonnollisen kielen käsittelyyn ja käyttää samoja elementtejä, kuin esimerkiksi OpenAI:n GPT-4 -malli (*Generative Pre-trained Transformer*; OpenAI, 2023). Edelleen siinä missä kirjallisuudessa epävarmuutta luokittelumallien kohdalla tyypillisesti kuvataan ennustetuilla luokkatodennäköisyyksillä tai näistä johdetulla entropialla, tässä tutkielmassa epävarmuutta koskevassa päättelyssä hyödynnetään myös varianssia.

Tutkielma on rakennettu siten, että kaikki myöhemmissä luvuissa tarvittava teoria on koottu lukuihin 1 ja 2. Luvussa 1 käydään tarpeellisilta osin läpi neuroverkkojen teoriaa sekä esitellään BERT-mallin käyttämä Transformer -arkkitehtuuri (Vaswani et al., 2017). Luvun lopussa on lisäksi kerrottu epävarmuuden lähteistä sekä siitä, miksi epävarmuus ei aina ilmene oikein ennusteessa. Gaussisia prosesseja koskeva teoria on esitelty luvussa 2. Tältä osin esityksessä on pyritty perusteellisempaan käsittelyyn toisaalta tietoisesti ja toisaalta siksi, ettei luvussa 3 esiteltäviä Gaussisen prosessin ja neuroverkon yhdistäviä menetelmiä käytännössä kyetä kuvaamaan ilman näiden soveltamien approksimaatioiden esittämistä. Tarve approksimaatioille tulee sekä ei-konjugaattisesta uskottavuudesta että mallien skaalautuvuudesta, eivätkä liity vain Gaussisen prosessin ja neuroverkon yhdistäviin malleihin, vaan Gaussisen prosessin malleihin yleisesti. Luvussa 3 on esitelty menetelmät sekä näiden implementaatiot ja käytetyt aineistot. Lisäksi luvussa 3 on kerrottu menetelmien arvioinnista sekä tähän liittyvistä käsitteistä. Tutkimusasetelmien yksityiskohtaisempi kuvaaminen on sijoitettu tulosten raportoinnin yhteyteen lukuun 4. Tulosten perusteella tehdyt johtopäätökset on esitetty luvussa 5.

Läpi tämän tutkielman merkinnöissä noudatetaan yleisiä käytäntöjä siten, että reaaliarvoisia muuttujia merkitään pienillä kirjaimilla x, vektoriarvoisia muuttujia lihavoiduilla pienillä kirjaimilla $\mathbf{x}$ ja matriiseja lihavoiduilla isoilla kirjaimilla $\mathbf{X}$. Vastemuuttujasta y oletetaan, että $y \in \{0, 1\}$ eli vasteen oletetaan olevan dikotominen. Kerättyä aineistoa merkitään $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ tai vain vastemuuttujien y_i tai ennustavien muuttujien $\mathbf{x}_i$ osalta vastaavasti vektorina $\mathbf{y}$ tai matriisina $\mathbf{X}$, missä havaintovektorit $\mathbf{x}_i$ oletetaan matriisin riveiksi. Uusiin havaintoihin viitataan yläindeksillä * eli $y^\star$ tarkoittaa ennustetta uudelle havainnolle $\mathbf{x}^\star$. Jakaumissa riippuvuus muuttujista $\mathbf{x}_i$ tai $\mathbf{x}^\star$ pääosin käy ilmi asiayhteydestä ja jätetään tällöin usein merkitsemättä.

1 Neuroverkot

Neuroverkot ovat epälineaarisia tilastollisia malleja, joiden yhdeksi vahvuudeksi voidaan lukea niiden kyky löytää ja kuvata ennustavista muuttujista $\mathbf{x}$ sellaisia selittäviä ja erottelevia piirteitä, mitkä tukevat tavoitteena olevan tehtävän, kuten luokittelun, tehokasta suorittamista (Bengio, 2013). Edellinen on osin luettu neuroverkkojen useilla epälineaarisilla muunnoksilla hierarkkisesti suorittaman laskennan ansioksi (Goodfellow et al., 2016). Hierarkkinen rakenne (Kuva 1.1) syntyy tavasta, millä neuroverkon parametreilla $\boldsymbol{\theta}$ parametrisoima funktio $f(\mathbf{x}; \boldsymbol{\theta}) : \mathbb{R}^M \to \mathbb{R}^{M'}$ muodostetaan. Funktio muodostetaan ketjuttamalla peräkkäin useita neuroverkon *kerroksissa* $l = 0, 1, \ldots, L$ määrättyjä epälineaarisia funktioita $f^{(l)}$, jolloin koko neuroverkon parametrisoima funktio f voidaan esittää yhdistettynä funktiona $f = f^{(L)}(f^{(L-1)}(\ldots(f^{(0)})\ldots))$. Neuroverkon rakenteesta on kerrottu tarkemmin luvussa 1.1.

Tyypillisesti neuroverkon parametrisoimaa funktiota käytetään kuvaamaan jonkin todennäköisyysjakauman $p(y \,|\, \mathbf{x}; \boldsymbol{\theta})$ jotakin tuntematonta parametria π (Goodfellow et al., 2016). Läpi tämän tutkielman oletetaan, että $y \,|\, \mathbf{x} \sim \text{Bin}(1, \pi)$, jolloin neuroverkolla voidaan kuvata parametria π eli todennäköisyyttä $P(Y = 1 \,|\, \mathbf{x})$. Näin saatu

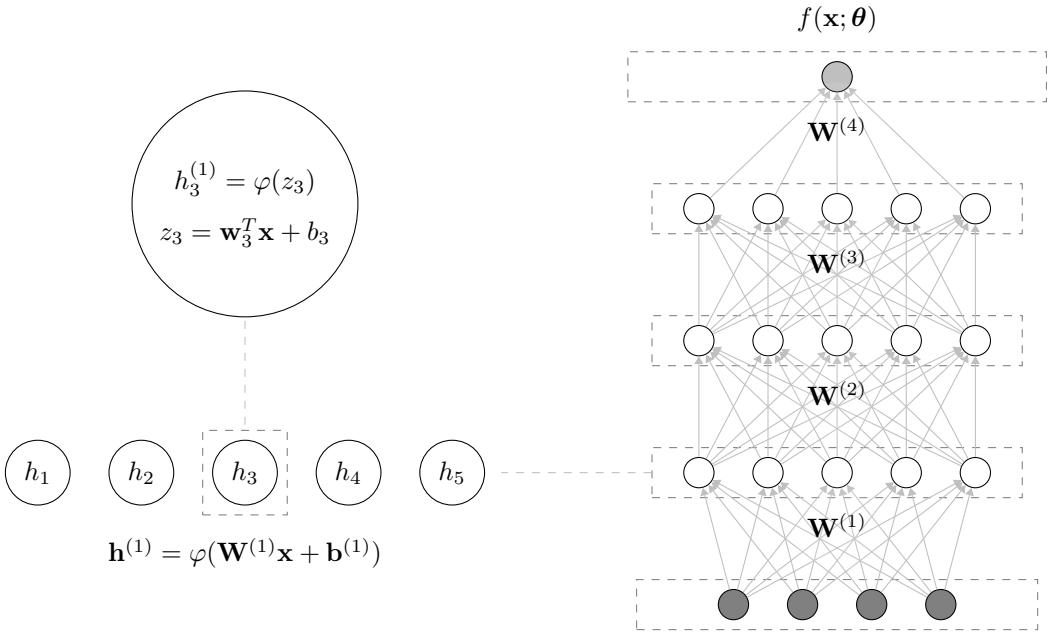
malli voidaan usein rinnastaa yleistetyksi lineaariseksi malliksi, missä ennustavina piirteinä käytetään neuroverkon määräämiä muunnoksia alkuperäisistä ennustavista muuttujista.

Kerroksissa voidaan käyttää myös muunlaisia operaatioita kuin, mitä luvun 1.1 ensimmäisessä esimerkissä laskentayksiköillä on esitetty. Monimutkaisempi esimerkki, on esitelty luvussa 1.2, missä on kuvattu Transformer -arkkitehtuuri (Vaswani et al., 2017). Transformer -arkkitehtuuri on kehitetty erityisesti luonnollisen kielen käsittelyyn liittyviin tehtäviin. Transformer -arkkitehtuuri on perustana esimerkiksi tässä työssä käytetyssä BERT-mallissa sekä OpenAI:n GPT-4 -mallissa.

Tässä tutkielmassa erityisen kiinnostuksen kohteena on neuroverkkojen robustisuus sekä neuroverkkoihin liittyvä epävarmuus ja tämän kvantifiointi. Epävarmuuden lähteistä ja välittymisestä ennusteeseen on kerrottu luvussa 1.3. Samassa luvussa on kuvattu myös, mistä syystä ennuste ei usein luotettavasti kuvaa kaikkea tähän välittyvää epävarmuutta.

1.1 Rakenne ja sovittaminen

Neuroverkolle tyypillistä hierarkkista rakennetta on havainnollistettu kuvassa 1.1. Kuvassa on syötekerroksesta, kolmesta piilokerroksesta ja ulostulokerroksesta muodostuva eteenpäinsyöttävä neuroverkko. Piilokerroksiksi kutsutaan neuroverkon muita kuin syöte- ja ulostulokerroksia. Kuvan neuroverkossa kukin piilokerros muodostuu viidestä rinnakkain järjestystä (laskenta)yksiköstä eli *neuronista* ja ulostulokerros muodostuu yhdestä yksiköstä vastaten dikotomista tai jatkuvaa vastetta.



Kuva 1.1: Neuroverkon elementit. Kerrokset (vas. alh.) muodostuvat rinnakkain järjestetyistä yksiköistä h_m. Jokaisessa yksikössä (vas. ylh.) sovelletaan aktivointifunktiota edelliseltä kerrokselta saadun vektorin lineaariselle muunnokselle. Sulkuihin merkitty yläindeksi $l = 0, 1, \ldots, L$ viittaa neuroverkon kerrokseen.

Kuhunkin kerrokseen $l = 0, 1, \ldots, L$ liittyy $M^{(l)} \times M^{(l-1)}$ matriisi $\mathbf{W}^{(l)}$ ja vektori $\mathbf{b}^{(l)} = (b_1^{(l)}, \ldots, b_{M^{(l)}}^{(l)})^T$, joilla edelliseltä kerrokselta syötteenä saadulle piirrevektorille $\mathbf{h}^{(l-1)} = (h_1^{(l-1)}, \ldots, h_{M^{(l-1)}}^{(l-1)})^T$ (kuvassa syötevektorille $\mathbf{x}$) tehdään ensin lineaarinen muunnos $\mathbf{z}^{(l)} = \mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}$

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_{M^{(l)}} \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1M^{(l-1)}} \\ w_{21} & w_{22} & \cdots & w_{2M^{(l-1)}} \\ \vdots & \vdots & \ddots & \vdots \\ w_{M^{(l)}1} & w_{M^{(l)}2} & \cdots & w_{M^{(l)}M^{(l-1)}} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_{M^{(l-1)}} \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{M^{(l-1)}} \end{bmatrix}.$$

Kerroksen (l) ulostulona saatava piirrevektori $\mathbf{h}^{(l)}$ saadaan muunnoksesta $\mathbf{z}^{(l)}$ aktivointifunktiolla $\varphi(\mathbf{z}^{(l)}) : \mathbb{R}^{M^{(l)}} \to \mathbb{R}^{M^{(l)}}$, mikä tässä on merkitty sovellettavaksi komponenteittain eli $\varphi(\mathbf{z}) = (\varphi(z_1), \ldots, \varphi(z_{M^{(l)}}))^T$. Aktivointifunktiona piilokerroksissa nykyään yleisimmin käytetään ns. *ReLu* -funktiota (*Rectified Linear Unit*) $g(z) = \max(0, z)$. Ulostulokerroksen aktivointifunktio sen sijaan määrätään sen mukaan, mitä funktiota neuroverkolla halutaan approksimoida.

Kuten luvun johdannossa mainittiin, tyypillisesti neuroverkolla parametrisoidaan jokin ehdollinen todennäköisyysjakauma $p(\mathbf{y} \,|\, \mathbf{x}; \boldsymbol{\theta})$ (Goodfellow et al., 2016), minkä tässä tutkielmassa oletettiin olevan binomijakauma. Tällöin neuroverkolla parametrisoidaan binomijakauman parametri π, toisin sanoen

$$\begin{aligned} \boldsymbol{y}_i \,|\, \mathbf{x}_i &\sim \mathrm{Bin}(1, \pi_i) \\ \pi_i &= f(\mathbf{x}_i; \boldsymbol{\theta}), \end{aligned}$$

missä f on neuroverkon parametrisoima funktio ja $\boldsymbol{\theta}$ neuroverkon parametrit eli painokertoimet $w_{11}^{(1)}, \ldots, w_{M^{(L)}M^{(L-1}}^{L}$ ja vakiotermit $b_1^{(1)}, \ldots, b_{M^{(L-1)}}^{(L)}$. Tällöin ulostulokerroksen aktivointifunktio voidaan rinnastaa linkkifunktion käänteisfunktioon g^{-1} ja neuroverkolla ajatella määritettävän yleistetty lineaarinen malli, missä ennustavina muuttujina käytetään alkuperäisten piirteiden $\mathbf{x}_i$ sijaan neuroverkolla johdettuja piirteitä $\mathbf{h}_i^{(L-1)}$. Jatkossa ilman yläindeksiä merkityllä vektorilla $\mathbf{h}$ tarkoitetaan nimenomaan viimeiseltä piilokerrokselta saatavaa vektoria $\mathbf{h}_i^{(L-1)}$. Logistisen regressiomallin tapauksessa linkkifunktiona g käytetään logit-linkkiä

$$\mathrm{logit}(\pi_i) = \log \frac{\pi_i}{1 - \pi_i}. \tag{1}$$

Koska $f(\mathbf{x}_i; \boldsymbol{\theta}) = \varphi^{(L)}(z_i)$, asettamalla ulostulokerroksen aktivointifunktioksi $\varphi^{(L)}(z_i)$ *sigmoid*-funktio

$$\varphi^{(L)}(z_i) = \frac{1}{1 + \exp(-z_i)}$$

voidaan todeta, että $\varphi^{(L)}(z_i) = g^{-1}(z_i)$. Edelleen $z_i = \mathbf{w}^T\mathbf{h} + \beta$, missä $\mathbf{w} = \mathbf{W}^{(L)}$, toisin sanoen kyseessä on logistinen regressio neuroverkolla johdetuilla piirteillä $\mathbf{h}$.

Neuroverkon piilokerrosten muodostamasta osasta käytetään usein nimitystä enkooderi. Enkooderin ulostulona saatavien johdettujen piirteiden $\mathbf{h}_i$ voidaan ajatella olevan sellainen alkuperäisille piirteille opittu muunnos, millä ulostulokerroksessa määrätty tavoite on parhaiten toteutettavissa (Goodfellow et al., 2016).

Neuroverkon parametrit estimoidaan suurimman uskottavuuden menetelmällä minimoimalla tappiofunktiona käytettävää negatiivista logaritmista uskottavuusfunktiota (Goodfellow et al., 2016). Luokittelumallien tapauksessa tappiofunktio on *ristientropia*

$$L(\boldsymbol{\theta}) = -\sum_{k=1}^{K} y_{ik} \log \pi_{ik}, \tag{2}$$

missä $k = 1, \ldots, K$ viittaa luokkaan ja $\pi_{ik} = f_k(\mathbf{x}_i; \boldsymbol{\theta})$ on neuroverkon ennuste havainnon i todennäköisyydelle kuulua luokkaan k.

Minimointi toteutetaan yleisimmin tappiofunktion gradientteihin parametrien $\boldsymbol{\theta}$ suhteen perustuvalla *stokastisella gradienttimenetelmällä* tai jollakin tämän muunnelmalla (Goodfellow et al., 2016). Useimmiten käytetään osajoukkoihin perustuvia menetelmiä (Ruder, 2016), joissa jokainen päivitysaskel perustuu opetusaineistosta palauttamatta poimittuun kokoa M olevaan otokseen. Tällöin kullakin päivitysaskeleella t käsitellään funktiota

$$L_t(\boldsymbol{\theta}_{t-1}) = -\frac{1}{M} \sum_{i \in \mathcal{S}_M} \sum_{k} y_{ik} \log \pi_{ik},$$

missä $\mathcal{S}_M \subset \mathcal{D}$ on otos opetusaineistosta $\mathcal{D}$. Otoksia poimitaan, kunnes koko opetusaineisto on käsitelty. Tämän jälkeen aloitetaan alusta, kunnes haluttu määrä kierroksia yli koko opetusaineiston on käyty läpi. Yhtä kierrosta yli koko opetusaineiston kutsutaan epookiksi (*epoch*). Parametrien päivitysaskeleella t (Goodfellow et al., 2016)

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} - \eta_t \nabla L_t(\boldsymbol{\theta}_{t-1}),$$

missä $\nabla L_t(\boldsymbol{\theta}_{t-1})$ on tappiofunktion gradientti parametrien $\boldsymbol{\theta}$ suhteen ja opetustahti η_t hyperparametri, millä säädetään päivitysaskeleen suuruutta tyypillisesti tätä vaiheittain pienentäen.

Voidaan osoittaa, että jo kahden kerroksen syvyisellä neuroverkolla voidaan approksimoida mielivaltaisella tarkkuudella mitä tahansa avaruuden $\mathbb{R}^n$ suljetussa ja rajoitetussa osajoukossa määritettyä jatkuvaa funktiota (Goodfellow et al., 2016). Neuroverkon syvyydellä tarkoitetaan neuroverkon kerrosten lukumäärää ulostulokerros mukaanlukien, mutta ilman syötekerrosta (Lu et al., 2017). Tulosta kutsutaan universaaliksi approksimointilauseeksi (Cybenko, 1989; Leshno et al., 1993 ja Castro et al., 2000).

Neuroverkkojen syvyystehokkuudella viitataan siihen, että approksimointia voidaan tehokkaammin kasvattaa neuroverkon syvyyttä kasvattamalla (Lu et al., 2017). On

esimerkiksi voitu osoittaa, että on olemassa kolmen kerroksen neuroverkolla ja leveydellä $\mathcal{O}(d^{19/4})$ esitettävä funktio, jota ei voida esittää kahden kerroksen neuroverkolla vakiota c pienemmällä tarkkuudella, jos tämän leveys on pienempi tai yhtä suuri kuin kuin $\mathcal{O}(e^{cd})$, missä d tarkoittaa syötteen dimensiota (Eldan & Shamir, 2016). Neuroverkon leveydellä tarkoitetaan rinnakkaisten yksiköiden lukumäärää siinä kerroksessa, missä yksiköiden lukumäärä on suurin (Lu et al., 2017). Edelleen jos neuroverkon syvyydelle ei aseteta rajoitteita, voidaan osoittaa, että minimileveys universaalille approksimoinnille on $d + 1$ (Lu et al., 2017; Park et al., 2020). Nykyaikaisissa neuroverkoissa onkin varsin yleistä käyttää laskentatavasta riippuen kymmeniä tai jopa satoja kerroksia (Lin & Jegelka, 2018).

1.2 Transformer -arkkitehtuuri

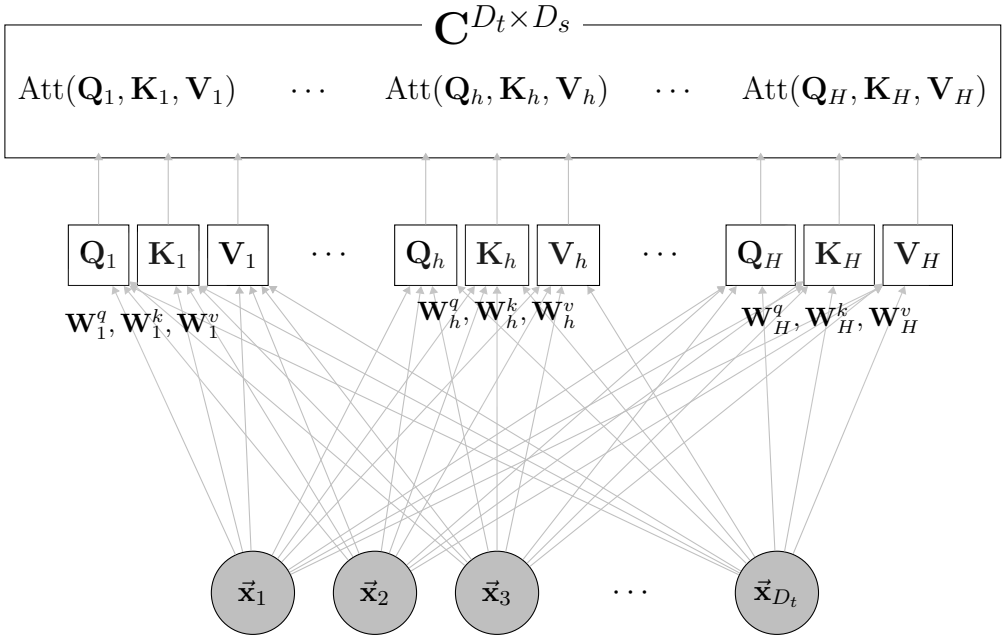
Edellisessä luvussa esitettyä, rinnakkain järjestetyistä yksiköistä muodostettua neuroverkon kerrosta kutsutaan *tiheäksi* kerrokseksi. Tässä luvussa esiteltävän Transformer -arkkitehtuurin (Vaswani et al., 2017) kerroksissa määrätään myös operaatioita, joita ei voida esittää tiheänä kerroksena. Koska arkkitehtuuri käytännössä muodostuu kuudesta peräkkäisestä Transformer-lohkosta, rajoitutaan tässä luvussa kuvaamaan yksityiskohtaisesti vain mainitun lohkon rakenne ja tässä suoritetut operaatiot. Lohkolla tarkoitetaan tässä tutkielmassa useamman kerroksen käsittävää, mutta itsenäiseksi miellettävää neuroverkon elementtiä.

Kuten luvun johdannossa todettiin, Transformer -arkkitehtuuri on kehitetty erityisesti tekstin muodossa olevien syötteiden käsittelyyn. Tekstille oletetaan tässä esitys matriisina $\mathbf{X}$ siten, että tekstin pituus on D_t ja kukin sana esitetään D_s-ulotteisena vektorina, mitkä kootaan riveiksi matriisiin $\mathbf{X}$. Tekstin muuntamisesta numeeriisiksi vektoreiksi on kerrottu luvussa 3.1.1.

Transformer -lohko muodostuu kahdesta osasta, joista jälkimmäinen käsittää kaksi peräkkäin kytkettyä tiheää kerrosta ja ensimmäisen rakennetta puolestaan on pyritty kuvaamaan kuvassa 1.2, missä nyt merkinnällä $\vec{\mathbf{x}}_i$ on haluttu korostaa sitä, että edellisen luvun esimerkistä poiketen, syötekerros muodostuu matriisin $\mathbf{X}$ rivivektoreista $\mathbf{x}_i$.

Transformer -lohkon ensimmäisessä osassa jokaisesta sanasta muodostetaan aluksi H-kappaletta kysely-, avain- ja arvovektoreita (*query, key, value*), joiden dimensioita merkitään D_k ja D_v siten, että kysely- ja avainvektoreiden dimensio on D_k ja arvovektorin dimensio on D_v. Dimensioille pätee $D_k = D_v = D_s/H$. Vektorit muodostetaan koko tekstille matriisikertolaskuina $\mathbf{Q}_h = \mathbf{X}\mathbf{W}_h^q$, $\mathbf{K}_h = \mathbf{X}\mathbf{W}_h^k$ ja $\mathbf{V}_h = \mathbf{X}\mathbf{W}_h^v$, missä $D_t \times D_k$ matriisin $\mathbf{Q}_h$ riveinä nyt on syötevektoreita vastaavat D_k-ulotteiset kyselyvektorit ja $D_s \times D_k$ matriisin $\mathbf{W}_h^q$ alkiot ovat opittavia parametereja ja vastaavasti matriiseissa $\mathbf{K}_h$ ja $\mathbf{W}_h^k$ sekä $\mathbf{V}_h$ ja $\mathbf{W}_h^v$, joiden dimensiot nyt ovat arvovektorin dimensiota vastaten $D_t \times D_v$ ja $D_s \times D_v$. Koska jatkossa kaikilla h käsittely on sama, merkitään tähän viittaava alaindeksi näkyviin vain, kun sillä on erityistä merkitystä.

Seuraavassa vaiheessa matriisit $\mathbf{Q}, \mathbf{K}$ ja $\mathbf{V}$ annetaan ns. attention -funktiolle



Kuva 1.2: Transformer-lohkon ensimmäisen osan rakenne. Syötematriisista muodostetaan matriiseiksi $\mathbf{Q}_h, \mathbf{K}_h$ ja $\mathbf{V}_h$ koottavat kysely-, avain- ja arvovektorit. Kontekstivektorit $\mathbf{c}_i$ sanoille i paikassa h saadaan matriisin $\mathbf{C}_h$ palauttavalla Attention-funktiolla. Ulostulona saatavan lohkomatriisin $\mathbf{C}$ lohkoina ovat matriisit $\mathbf{C}_1, \ldots, \mathbf{C}_H$.

$$\mathrm{Att}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{D_k}\right)\mathbf{V},$$

missä softmax -funktiota oletetaan sovellettavan $D_t \times D_t$ matriisin $\mathbf{S}/D_k = \mathbf{Q}\mathbf{K}^T/D_k$ rivivektoreille $\mathbf{s}_i/D_k$. Vektorin $\mathbf{s}_i = (s_{i1}, \ldots, s_{iD_t})^T$ komponentit määritellään

$$s_{ij} = \sum_{k=1}^{D_k} q_{ik} k_{kj},$$

mistä nähdään, kuinka vektorin $\mathbf{s}_i$ komponentteina ovat sanan $\mathbf{x}_i$ kyselyvektorin $\mathbf{q}_i$ pistetulot kaikkien sanojen avainvektoreiden suhteen. Koska vektoreihin $\mathbf{q}_i$ ja $\mathbf{k}_j$ liittyy opittavat parametrit $\mathbf{w}_i^q$ ja $\mathbf{w}_j^k$ eli sarakevektorit $\mathbf{W}_{\cdot i}^q$ ja $\mathbf{W}_{\cdot j}^k$, voidaan vektorin komponentteihin ajatella opittavan sanalle $\mathbf{x}_j$ annettava annettava paino, kun tulkitaan sanaa $\mathbf{x}_i$.

Merkitään nyt $\mathbf{S}'$ matriisia, minkä riveinä $\mathbf{s}_i' = (s_{i1}, \ldots, s_{iD_t}')^T$ ovat skaalatut ja softmax -funktiolla normalisoidut vektorit $\mathbf{s}_i$ eli

$$s_{ij}' = \mathrm{softmax}(\mathbf{s}_i/D_k)_j = \frac{\exp(s_{ij}/D_k)}{\sum_k \exp(s_{ik}/D_k)}.$$

Attention -funktion matriiseille $\mathbf{Q}_h, \mathbf{K}_h$ ja $\mathbf{V}_h$ palauttaman $D_t \times D_v$ matriisin $\mathbf{C}_h$ rivivektorit $\mathbf{c}_i$ ovat nyt

$$
\begin{aligned}
\mathbf{c}_i &= \begin{bmatrix} s'_{i1} & \cdots & s'_{iD_t} \end{bmatrix} \begin{bmatrix} v_{11} & \cdots & v_{1D_v} \\ \vdots & \ddots & \vdots \\ v_{D_t1} & \cdots & s_{D_tD_v} \end{bmatrix} \\
&= \begin{bmatrix} \sum_{j=1}^{D_t} s'_{ij} v_{j1} & \cdots & \sum_{j=1}^{D_t} s'_{ij} v_{jD_v} \end{bmatrix} \\
&= \sum_{j=1}^{D_t} s'_{ij} \mathbf{v}_j^T
\end{aligned}
$$

eli *kontekstivektoriin* $\mathbf{c}_i$ (Britz et al., 2017) kuvautuu jokaisen tekstin sanan arvovektorit $\mathbf{v}_j$ sanalle $\mathbf{x}_i$ laskettujen painojen s'_{ij} suhteessa. Kontekstivektoreiden voidaan tulkita oppivan sanalle $\mathbf{x}_i$ esityksen $\mathbf{c}_i$, mikä huomioi myös sen kontekstin tekstissä $\mathbf{X}$. Attention -kerroksen ulostulo $\mathbf{C}$ on lohkomatriisi

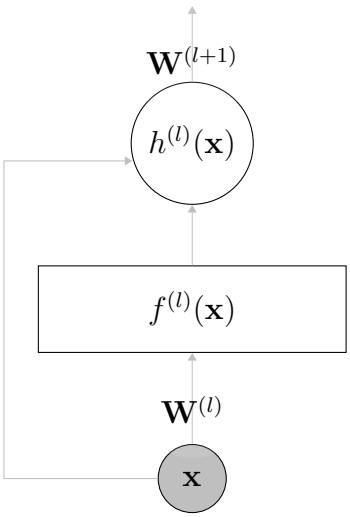
$$
\mathbf{C} = \left[\begin{array}{c|c|c|c} \mathbf{C}_1 & \mathbf{C}_2 & \cdots & \mathbf{C}_H \end{array} \right],
$$

missä kunkin matriisin $\mathbf{C}_h$ dimensio on $D_t \times D_v$ ja $D_v = D_s/H$ eli matriisin $\mathbf{C}$ dimensioksi saadaan syötettä vastaten $D_t \times D_s$.

Transformer -lohkon toisessa osassa kontekstivektorit $\mathbf{c}_i$ syötetään edelleen kahdesta tiheästä kerroksesta muodostetun neuroverkon läpi, kuten kuvassa 1.1, missä syötekerroksessa (alin rivi) nyt olisi kontekstivektori $\mathbf{c}_i = (c_{i1}, \ldots, c_{iD_s})^T$. Neuroverkon sisäkerroksen (*inner-layer*) dimensio, eli ensimmäisen tiheän kerroksen laskentayksiköiden lukumäärä, on D_{ff} ja jälkimmäisen D_s eli sama kuin syötteellä. Sisäkerroksessa aktivointifunktiona on ReLu-funktio ja toinen kerros on ns. lineaarinen kerros eli aktivointifunktion voidaan ajatella olevan identiteettifunktio $\varphi(\mathbf{z}) = \mathbf{z}$. Toisessa osassa jokaista kontekstivektoria käsitellään itsenäisesti, mutta neuroverkon parametrit, eli kerroksien painomatriisit, ovat kaikille yhteiset.

Molemmissa osissa käytetään myös jäännösyhteyksiä (*residual connection*) sekä jäännösyhteyksien otantaa (*dropout*). Jäännösyhteys (He et al., 2016) voidaan ymmärtää siten, että mielivaltaisen kerroksen l oppiman kuvauksen $f^{(l)}$ sallitaan selittää vain osa syötteelle kerroksessa tehtävästä muunnoksesta. Merkitään tässä $h^{(l)}(\mathbf{x})$ kokonaismuunnosta, mikä syötteelle kerroksessa l tehdään. Jäännösyhteyskerroksessa $h^{(l)}(\mathbf{x}) = f^{(l)}(\mathbf{x}) + \mathbf{x}$ (Kuva 1.3) eli kerroksen määrämä funktio $f^{(l)}$ selittää muunnoksessä jäännöksen $h^{(l)}(\mathbf{x}) - \mathbf{x}$. Tarkoituksena tässä on edesauttaa identiteettikuvauksen esittämistä neuroverkolla, minkä on arveltu olevan tälle hankalaa, mutta syvissä neuroverkoissa kuitenkin toisinaan tarpeellista. Esimerkiksi transformer -lohkon ensimmäisestä osasta toiseen osaan ohjataan tosiasiassa vektori $(\mathbf{c}_1^T + \mathbf{x}^T, \ldots, \mathbf{c}_H^T + \mathbf{x}^T)$.

Jäännösyhteyksien otannalla (Srivastava et al., 2014) tarkoitetaan sitä, että kullakin opetusaskeleella neuroverkossa käytetään vain osaa jäännöksistä eli poimitaan otos neuroverkon määräämistä pienemmistä neuroverkoista. Muodollisesti merkiten edelleen jäännöksiä edellä $f^{(l)}(\mathbf{x})$, jäännöksien otannassa eteenpäin välitetään vektori $\mathbf{r}^T f^{(l)}(\mathbf{x})$, missä vektorille $\mathbf{r} = (r_1, \ldots, r_{D_v})^T$ pätee $r_j \sim \text{Bern}(p)$. Toisin sanoen todennäköisyydellä p komponentti j saa arvon 0. Lähdejulkaisussa, kuten myös luvussa 3.1.1 esitellyssä BERT-mallissa, $p = 0.1$.



Kuva 1.3: Havainnekuva jäännösyhteyskerroksesta. Jäännösyhteyskerroksessa syötteelle tehtävä kokonaismuunnos $h^{(l)}(\mathbf{x})$ muodostetaan summana $f^{(l)}(\mathbf{x}) + \mathbf{x}$, mikä voidaan kuvata siten, että samanaikaisesti, kun syöte $\mathbf{x}$ annetaan kerroksen l määräämälle funktiolle $f^{(l)}$, se myös ohjataan eteenpäin kerros ohittaen.

Lopuksi edellisten vaiheiden jälkeen saatu vektori vielä normalisoidaan, kuten Ba et al. (2016) molemmissa Transformer -lohkon osissa. Normalisoinnin yksityiskohtien osalta tässä tutkielmassa viitataan em. alkuperäiseen julkaisuun.

1.3 Ennusteiden epävarmuudesta

Vaikka neuroverkkojen tarkkuus on parantunut huomattavasti mm. uusien arkkitehtuurien myötä, on samalla todettu, etteivät nykyaikaisten neuroverkkojen ennustamat luokkatodennäköisyydet luotettavasti kuvaa ennusteeseen liittyvää epävarmuutta (Guo et al., 2017; Hendrycks & Gimpel, 2016). Tässä luvussa on nostettu esiin joitakin asioita, jotka toisaalta aiheuttavat epävarmuutta ennusteeseen ja toisaalta vaikuttavat siihen, ettei ennusteeseen välittyvä epävarmuus välttämättä luotettavasti ilmene ennusteessa.

Tilastollinen päättely perustuu sekä kerättyihin havaintoihin että taustaa koskeviin oletuksiin esimerkiksi havaintojen riippumattomuudesta ja jakaumista (Huber, 2011). Draper (1995) viittaa mallin rakenteeseen, jolla tarkoitetaan esimerkiksi ennustavien muuttujien ja vastemuuttujan välisistä suhteista tehtyjä oletuksia tai linkkifunktion valintaa koskevia oletuksia. Käytännössä kyse on välttämättömistä matemaattisista yksinkertaistuksista, joihin liittyvien virheiden ajatellaan aiheuttavan vain pieniä virheitä johtopäätöksissä, mikä ei kuitenkaan aina pidä paikkaansa (Huber, 2011).

Merkitään nyt yleisesti M mallin rakennetta ja $\boldsymbol{\theta}$ tämän indusoimia mallin parametreja. Mallin rakennetta koskevien oletusten todettiin edellä olevan matemaattisia yksinkertaistuksia, joiden ei käytännössä edes oleteta olevan täsmälleen oikeita (Huber, 2011). Voidaankin pitää selvänä, että näihin oletuksiin liittyy aina epävarmuutta. Kerättyjen havaintojen puolestaan voidaan tulkita aiheuttavan epävarmuutta mallin

parametreihin $\boldsymbol{\theta}$, vaikka mallin rakenne oletettaisiinkin oikein määritetyksi. Edellä viitataan ennenkaikkea otosvaihtelun aiheuttamaan epävarmuuteen.

Epävarmuus, joka liittyy mallin rakenteeseen ja parametrien estimointiin välittyy ennusteeseen ja mikäli ei tule riittävästi määritetyksi, vaikuttaa ennusteessa ilmenevään epävarmuuteen (Draper, 1995). Ennusteeseen välittyvän epävarmuuden määrittämisestä voitaisiin teoreettisesti päästä käyttämällä ennustettaessa marginaalijakaumaa

$$p(y^\star \,|\, \mathcal{D}) = \int_{\mathcal{M}} \int_{\Theta} p(y^\star \,|\, \boldsymbol{\theta}, M, \mathcal{D})\; p(\boldsymbol{\theta}, M \,|\, \mathcal{D})\; d\boldsymbol{\theta}\, dM,$$

mistä sekä mallin rakenteeseen että parametreihin liittyvä epävarmuus tulee integroiduksi pois. Käytännössä rakenteesta tehdyistä oletuksista kuitenkin seuraa mallille tietty rakenne $M_\star$, jolloin ennustejakaumaksi efektiivisesti tulee

$$p(y \,|\, \mathbf{x}, M_\star) = \int_{\Theta_\star} p(y \,|\, \mathbf{x}, \boldsymbol{\theta}_\star, M_\star)\; p(\boldsymbol{\theta}_\star, M_\star \,|\, \mathbf{x})\; d\boldsymbol{\theta}_\star,$$

missä $\boldsymbol{\theta}_\star$ viittaa rakenteen $M_\star$ indusoimiin parametreihin. Edellinen voi olla liiaksi keskittynyt tiettyihin, rakennetta $M_\star$ koskeviin oletuksiin johtaakseen luotettavasti epävarmuutta kuvaaviin ennusteisiin (Draper, 1995). Edellistä esitystä voitaisiin vielä jatkaa kiinnittämällä mallin parametreiksi $\boldsymbol{\theta}_\star$ esimerkiksi suurimman uskottavuuden estimaatit $\hat{\boldsymbol{\theta}}_\star$, kuten luvussa 1.1. Tällöin ennusteeksi efektiivisesti tulee $y \,|\, \mathbf{x}, M_\star, \hat{\boldsymbol{\theta}}_\star$, jolloin edellisessä konstruktiossa myös parametreihin liittyvä epävarmuus jää määrittämättä.

Epävarmuuden kvantifioinnin arvioimisesta, mikä tässä voidaan ymmärtää epävarmuuden kuvautumisen luotettavuuden arvioinniksi on kerrottu luvussa 3.2 yhdessä teemaan neuroverkkokirjallisuudessa liitettyjen käsitteiden kanssa. Tämän luvun lopuksi nostetaan kuitenkin esiin vielä yksi erityisesti, vaikkakaan ei yksinomaan, neuroverkkoja koskeva seikka. Syväoppimismallien tappiofunktiot eivät yleisesti ole konvekseja, vaan niillä on useita lokaaleja minimikohtia ja satulapisteitä (Goodfellow et al., 2016). Koska optimointialgoritmit tyypillisesti alustetaan satunnaisesti ja myös näiden polkuihin liittyy satunnaisuutta, liittyy myös algoritmien konvergointiin epävarmuutta, mikä ilmenee mallin parametrien estimaateissa. Vaikka nykykäsityksen mukaan riittävän suurilla neuroverkoilla useimpiin lokaaleihin minimikohtiin liittyy alhainen tappiofunktion arvo (Goodfellow et al., 2016), konvergointi satulapisteeseen johtaa huonompaan yleistettävyyteen (Rangwani et al., 2022; Dauphin et al., 2014).

2 Gaussiset prosessit

Edellisessä luvussa esiteltiin teoreettinen Bayesiläinen viitekehys sille, kuinka epävarmuus välittyy ennusteeseen keskittyen erityisesti malliin M ja parametreihin $\boldsymbol{\theta}$. Malliin liittyvä epävarmuus tuotiin esiin, jotta voitiin motivoida tarve epävarmuuden kvantifioinnin tutkimiselle. Vaikka menetelmiä myös tämän huomioimiseksi on

olemassa, jatketaan tässä tutkielmassa olettaen, kuten yleisimmin, että mallin rakenne on tunnettu, jolloin jäljelle yhä jää mallin parametreihin liittyvä epävarmuus.

Mallin parametreihin liittyvää epävarmuutta voidaan pyrkiä huomioimaan asettamalla näille priorijakauma $p(\boldsymbol{\theta})$ ja nojaamalla päättelyssä posteriorijakaumaan

$$p(\boldsymbol{\theta} \mid y) = \frac{p(y \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta})}{p(y)},$$

ja posterioriennustejakaumaan

$$p(y^\star \mid y) = \int p(y^\star \mid \boldsymbol{\theta})\, p(\boldsymbol{\theta} \mid y)\, d\boldsymbol{\theta},$$

missä $\boldsymbol{\theta}$ tässä viittaa yleisesti päättelyn kohteena oleviin parametreihin, eikä neuroverkon parametreihin, kuten muutoin tässä tutkielmassa. Regressiomalleissa tyypillisesti $\boldsymbol{\theta} = (\beta_1, \ldots, \beta_P)^T$ ja Bayesiläisissä neuroverkoissa puolestaan $\boldsymbol{\theta}$ käsittää neuroverkon parametrit. Päättelyä Bayesiläisten neuroverkkojen tapauksessa kuitenkin hankaloittaa parametrien suuri määrä (Gawlikowski et al., 2021).

Koska y edellä riippuu parametreista $\boldsymbol{\theta}$ vain näiden parametrisoiman funktion f kautta, voidaan parametreja koskevasta päättelystä päästä, jos priorijakauma asetetaan parametrien sijaan suoraan funktiolle f. Esimerkiksi oletettaessa havaintomalliksi $y \sim \text{Bin}(1, \pi)$, missä $\pi = \beta_1 x_1 + \cdots + \beta_P x_P$, y on riippumaton parametreista β_p ehdolla f, kun $f(\mathbf{x}) = \beta_1 x_1 + \cdots + \beta_p x_P$. Priorijakauma funktiolle f voidaan asettaa Gaussisella prosessilla.

Gaussiset prosessit ovat joustavia parametrittomia Bayesiläisiä malleja (Gelman et al., 2014), joilla on mahdollista saada luotettavia estimaatteja ennusteen epävarmuudelle (Seeger, 2004). Gaussisen prosessin voidaan osoittaa vastaavan yhden kerroksen bayesiläistä neuroverkkoa äärettömässä leveydessä, kun priorina neuroverkon parametreille käytetään normaalijakaumaa (Neal, 2012). Gaussinen prosessi on mahdollista myös yhdistää deterministiseen neuroverkkoon, kuten luvussa 3 tullaan kuvaamaan ja näin pyrkiä hyödyntämään tämän luotettavana pidettyä kykyä kuvata epävarmuutta (Ober et al., 2021) myös deterministisissä neuroverkoissa.

Epävarmuutta koskevassa päättelyssä yhdeksi Gaussisten prosessien tuomaksi eduksi voidaan lukea mahdollisuus hyödyntää koko posterijakaumaa. Koska tavanomaisista deterministisistä neuroverkoista saadaan ainoastaan piste-estimaatti, mikä luokittelumallien tapauksessa kuvaa luokkatodennäköisyyttä $\pi_i = P(Y_i = 1)$, on epävarmuuden tässä tulkittava liittyvän havainnolle ennustettuun luokkaan. Käytännössä epävarmuus kuitenkin liittyy havainnoille oletetun mallin $y_i \sim \text{Bin}(1, \pi_\text{i})$ tuntemattoman parametrin π_i estimaattiin. Estimaatteihin liittyvää epävarmuutta kuvataan tilastotieteessä yleensä luottamus- tai posterioriväleillä, mitkä molemmat riippuvat estimaatin arvojen odotetusta vaihtelusta. Toisin kuin piste-estimaatti, posteriorijakauma mahdollistaa myös estimaatin varianssia koskevan päättelyn, mitä on tässä tutkielmassa hyödynnetty piste-estimaatin rinnalla luvussa 4 selostetuissa tarkasteluissa.

Gaussisen prosessin funktiolle f määräämä jakauma on aina normaalijakauma (ks. Luku 2.1). Edellisestä seuraa, ettei uskottavuutta näin esimerkiksi luokittelumallien

tapauksessa voida määrittää tälle konjugaatiksi. Ei-konjugaattisen uskottavuuden kohdalla sovelletaan latentin Gaussisen prosessin mallia, mistä on kerrottu luvussa 2.2.

Ei-konjugaattisen uskottavuuden tapauksessa posteriorijakaumaa ei saada ratkaistua suljetussa muodossa, vaan sitä on joko approksimoitava analyyttisesti tai simuloitava Monte Carlo Markov Chain (MCMC) -menetelmään perustuvilla menetelmillä (Rasmussen & Williams, 2006). Analyyttisistä approksimaatioista tämän tutkielman kontekstissa on kerrottu luvussa 2.3. MCMC-menetelmää ei käsitellä tarkemmin tässä tutkielmassa.

Gaussisen prosessin malleissa sekä posteriorijakauman että posterioriennustejakauman määrittäminen vaatii kokoa $N \times N$ olevan kovarianssimatriisin kääntämisen, mikä on vaativuusluokkaa $\mathcal{O}(N^3)$ (van Amersfoort et al., 2021b). Tämä estää eksaktin päättelyn, kun aineisto on liian suuri (Rasmussen & Williams, 2006). Keskeisimmistä suurille aineistoille esitetyistä approksimaatioista on kerrottu luvussa 2.4.

2.1 Määritelmiä

Muistetaan, että *mitallisella avaruudella* tarkoitetaan paria $(\Omega, \mathcal{F})$, missä Ω on avaruus ja $\mathcal{F}$ tämän σ-algebra (Kallenberg, 1997). *Mitalla* puolestaan tarkoitetaan mitallisen avaruuden $\Omega, \mathcal{F}$ kuvausta $\mu : \mathcal{F} \to \mathbb{R}_+$, jolle $\mu(\emptyset) = 0$ ja

$$\mu(\cup_{k\geq 1} F_k) = \sum_{k\geq 1} \mu(F_k)$$

kaikilla erillisillä $F_1, F_2, \cdots \in \mathcal{B}$. *Todennäköisyysmitta* on mitta P, jolle $P(\Omega) = 1$.

Satunnaismuuttuja määritellään mitallisessa avaruudessa $(\Omega, \mathcal{F})$ määriteltynä kuvauksena $X : \Omega \to \mathbb{R}$, jolle $X^{-1}(B) := \{\omega \in \Omega : X(\omega) \in B\} \in \mathcal{F}$. Ehto $X^{-1}(B) \in \mathcal{F}$ määrittelee, että satunnaismuuttuja X on *mitallinen* (Kallenberg, 1997). Mikäli tila-avaruuden $\mathbb{R}$ sallitaan olla mikä tahansa mitallinen avaruus S, voidaan puhua yleisemmin avaruuden S *satunnaiselementistä*.

Olkoon nyt $T = \{t_i\}_{i=1}^N$ indeksijoukko, mikä voi olla myös ääretön ja merkitään $\mathbb{R}^T = \{f : T \to \mathbb{R}\,;\, f \, on \, kuvaus\}$ eli funktiot indeksijoukosta T reaalilukujen joukkoon $\mathbb{R}$ käsittävä joukko. Olkoon edelleen $\pi_t : \mathbb{R}^T \to \mathbb{R}$, $\pi_t f = f(t)$ *evaluaatiokuvaus* eli kuvaus, mikä määrää säännön, kuinka mielivaltaisen funktion $f \in \mathbb{R}^T$ indeksijoukon alkiota t vastaava arvo määrätään.

Määritellään seuraavaksi funktioavaruuden $U \subset \mathbb{R}^T$ satunnaiselementti F eli $F : \Omega \to U \subset \mathbb{R}^T$ ja edelleen satunnaismuuttuja $F_t = \pi_t \circ F$. Satunnaismuuttuja F_t on siis yhdistetty kuvaus $\pi_t \circ F$, missä ensin F kuvaa otosavaruuden alkion $\omega' \in \Omega$ satunnaiseksi funktioksi f joukossa $U \subset \mathbb{R}^T$, minkä jälkeen evaluaatiokuvauksella π_t funktioon f liitetään arvo $f(t) \in \mathbb{R}$ eli $F_t : \Omega \to \mathbb{R}$. Mikäli F_t on mitallinen kaikilla $t \in T$, satunnaiselementti F on (reaaliarvoinen) *stokastinen prosessi* joukossa T ja voidaan ekvivalentisti määritellä kokoelmana satunnaismuuttujia F_t. Edellinen on täydentäen esitetty, kuten Kallenberg (1997)

Määritelmä 2.1 (Stokastinen prosessi)**.** Olkoon $(S, \mathcal{S})$ mitallinen avaruus, T indeksijoukko ja $F_t : \Omega \to S$ mitallinen kaikilla $t \in T$. Kokoelmaa $F = \{F_T\}_{t \in T}$ kutsutaan stokastiseksi prosessiksi.

Oletetaan jatkossa tarkasteltavan nimenomaan reaaliarvoista stokastista prosessia eli määritelmässä 2.1 yllä, $S = \mathbb{R}$, kuten aiemmin. Oletetaan lisäksi, että $T = \mathbb{R}^M$ ja kuten sovelluksissa yleisesti $M \geq 2$. Indeksijoukon alkioita merkitään tutummin kirjaimella $\mathbf{x}$ eli $\mathbf{x} = (x_1, \ldots, x_M)^T$. Omaksutaan lisäksi käytäntö, missä satunnaismuuttujaa $F_{\mathbf{x}} = \pi_{\mathbf{x}} \circ F$ voidaan merkitä myös lyhyesti $F_{\mathbf{x}} = f(\mathbf{x})$. Stokastisen prosessin F jakauman määrää sen äärellisten jakaumien joukko (Kallenberg, 1997).

Määritelmä 2.2 (Gaussinen prosessi)**.** Stokastinen prosessi $F = \{F_{\mathbf{x}}\}_{\mathbf{x} \in \mathbb{R}^M}$ on *Gaussinen prosessi*, jos millä tahansa äärellisellä kokoelmalla indeksejä $\mathbf{x}_i \in \mathbb{R}^M, i = 1, \ldots, N$, satunnaisvektori $(F_{\mathbf{x}_1}, \ldots, F_{\mathbf{x}_N})^T$ noudattaa moniulotteista normaalijakaumaa.

Gaussinen prosessi $\{F_{\mathbf{x}}\}_{\mathbf{x} \in \mathbb{R}^M}$ voidaan määrätä yksikäsitteisesti keskiarvofunktiolla $m : \mathbb{R}^M \to \mathbb{R}$ ja kovarianssifunktiolla $k : \mathbb{R}^M \times \mathbb{R}^M \to \mathbb{R}$, joille (Rasmussen & Williams, 2006)

$$\begin{aligned} m(\mathbf{x}) &= \mathbb{E}[f(\mathbf{x})] \\ k(\mathbf{x}, \mathbf{x}') &= \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}' - m(\mathbf{x}'))]. \end{aligned}$$

Kovarianssifunktio määrää äärelliselle vektorille $\mathbf{f} = (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N))^T, \mathbf{x}_i \in \mathbb{R}^M$ symmetrisen ja positiivisesti semidefiniitin $N \times N$ kovarianssimatriisin

$$\mathbf{K}_{XX} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & k(\mathbf{x}_1, \mathbf{x}_2) & \ldots & k(\mathbf{x}_1, \mathbf{x}_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(\mathbf{x}_N, \mathbf{x}_1) & k(\mathbf{x}_N, \mathbf{x}_1) & \ldots & k(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix}.$$

Käytännössä oletetaan, että $m(\mathbf{x}) = 0$ (Rasmussen & Williams, 2006), jolloin Gaussista prosessia voidaan merkitä $\mathbf{f} \sim N(\mathbf{0}, \mathbf{K}_{XX})$ tai $f \sim \mathcal{GP}(0, k(\mathbf{x}, \mathbf{x}'))$ (Rasmussen & Williams, 2006).

Kovarianssifunktioista esitellään tässä vain tässä tutkielmassa käytetty neliöity eksponenttifunktio.

Määritelmä 2.3 (Neliöity eksponenttifunktio)**.**

$$k(\mathbf{x}, \mathbf{x}') = \tau^2 \exp\left(-\frac{|\mathbf{x} - \mathbf{x}'|^2}{2\ell^2}\right).$$

Neliöidyssä eksponenttifunktiossa hyperparametri τ säätelee funktion arvojen vaihtelun suuruutta, skaalaparametri ℓ funktion tasaisuutta ja $|\mathbf{x} - \mathbf{x}'|^2$ on pisteiden (indeksien) $\mathbf{x}$ ja $\mathbf{x}'$ neliöity euklidinen etäisyys (Gelman et al., 2014). Määritelmä on

esitetty, kuten Gelman et al. (2014), mutta se esiintyy kirjallisuudessa myös ilman hyperparametreja τ ja ℓ, jolloin efektiivisesti $\tau = 1$ ja $\ell = 1$. Kovarianssifunktion hyperparametrit estimoidaan aineistosta suurimman uskottavuuden menetelmällä maksimoimalla marginaalista uskottavuutta $p(\mathbf{y} \mid \mathbf{X})$ (Rasmussen & Williams, 2006) ja jätetään yleisesti jakaumiin merkitsemättä, kuten tässäkin työssä.

Määritelmä 2.4 (Stationäärinen kovarianssifunktio)**.** Kovarianssifunktio on *stationäärinen*, jos $k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x} - \mathbf{x}')$ eli kovarianssifunktio on invariantti siirron suhteen.

Stationäärisen kovarianssifunktion määritelmää tarvitaan luvussa 2.4.1. Aiemmin määritelty neliöity eksponenttifunktio on yksi esimerkki stationäärisestä kovarianssifunktiosta.

Voidaan osoittaa, että Gaussista prosessia odotusarvolla $m(\mathbf{x})$ ja kovarianssifunktiolla $k(\mathbf{x}, \mathbf{x}')$ vastaa ekvivalentti ääretöntä määrää kantafunktioita ϕ_j käyttävä parametrinen esitys (Gelman et al., 2014)

$$f(\mathbf{x}) = \sum_{j=1}^{\infty} \beta_j \, \phi_j(\mathbf{x}).$$

Äärellisellä määrällä kantafunktioita, kun $\boldsymbol{\beta} \sim N(\boldsymbol{\mu}_\beta, \boldsymbol{\Sigma}_\beta)$, parametrista muotoa vastaa Gaussinen prosessi keskiarvofunktiolla $m(\mathbf{x}) = \boldsymbol{\mu}_\beta^T \boldsymbol{\phi}(\mathbf{x})$ ja kovarianssifunktiolla $k(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}(\mathbf{x})^T \boldsymbol{\Sigma}_\beta \boldsymbol{\phi}(\mathbf{x}')$, missä $\boldsymbol{\phi}(\mathbf{x}) = (\phi_1(\mathbf{x}), \ldots, \phi_M(\mathbf{x}))^T$ (Gelman et al., 2014).

2.2 Latentin Gaussisen prosessin malli

Latentin Gaussisen prosessin malleissa priori (GP-priori) asetetaan latentille funktiolle f, mikä vastaavasti kuin yleistettyjen lineaaristen mallien kohdalla, linkkifunktion g kautta määrää uskottavuuden $p(\mathbf{y} \mid \mathbf{f}, \mathbf{X})$ (Gelman et al., 2014). Kuten luvun johdannossa todettiin, latentin Gaussisen prosessin malleja käytetään, kun uskottavuus ei ole normaalijakauma (Gelman et al., 2014), eikä posteriorijakaumia siten saada johdettua normaalijakauman ominaisuuksilla. Regressio jatkuvalla vasteella saadaan kuitenkin myös latentin Gaussisen prosessi mallista, kun valitaan $g(x) = x$. Jatkossa riippuvuutta muuttujista $\mathbf{x}_i$ ei enää merkitä jakaumiin näkyviin.

Dikotomiselle vasteelle oletetaan, että $y_i \sim Bin(1, \pi_i)$, missä $\pi_i = g^{-1}(f_i)$. Edeltä nähdään, kuinka latentin Gaussisen prosessin mallissa latentti funktio f korvaa linkkifunktiossa g yleistettyjen lineaaristen mallien lineaarisen prediktorin $\eta = \boldsymbol{\beta}^T \mathbf{x}$ vastaavasti, kuin selostettiin luvussa 1.1, missä f oli neuroverkon parametrisoima funktio.

Koska GP-priori on aina normaalijakauma ja konjugaattipriori binomijakaumalle olisi betajakauma, on posteriorijakaumaa joko approksimoitava analyyttisesti tai simuloitava Monte Carlo Markov Chain -menetelmään perustuvilla menetelmillä (Rasmussen & Williams, 2006). Seuraavassa posteriorin $p(\mathbf{f} \mid \mathbf{y})$ tilalla on käytetty tarkemmin yksilöimätöntä approksimaatiota $q(\mathbf{f} \mid \mathbf{y})$.

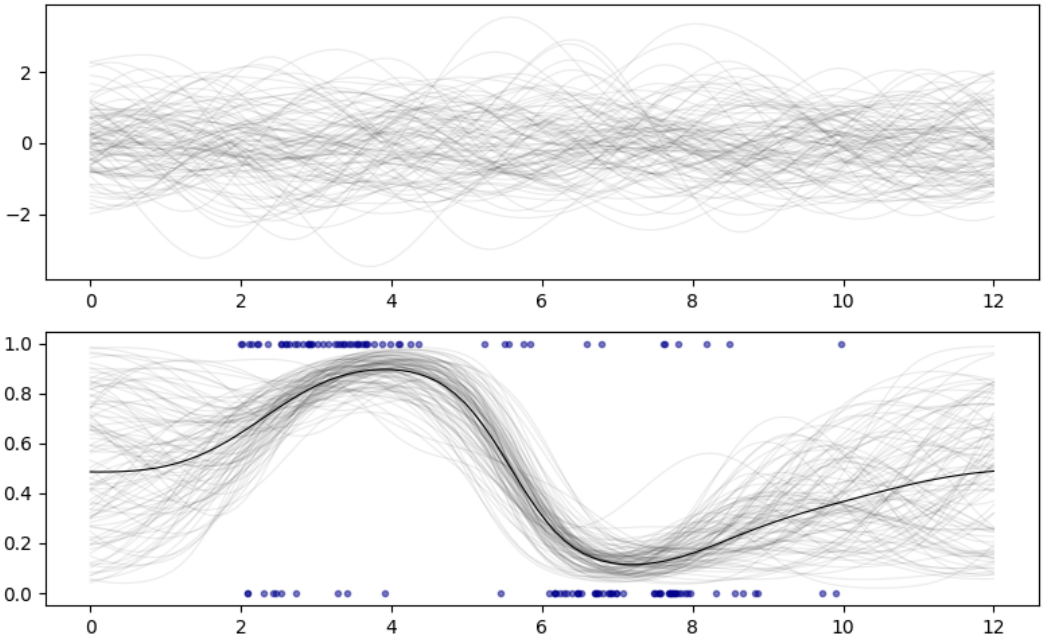
Posteriorijakauman approksimoinnista jakaumalla q seuraa, että myös posterioriennustejakauma on approksimaatio (Rasmussen & Williams, 2006). Merkitään

$$q(f^* \,|\, \mathbf{y}) = \int p(f^* \,|\, \mathbf{f}, \mathbf{y})\, q(\mathbf{f} \,|\, \mathbf{y})\, d\mathbf{f}.$$

Luokittelusäännössä käytetään muunnoksen $\pi^\star = g^{-1}(f^*)$ odotusarvoa $E[\pi^* \,|\, \mathbf{y}]$ (Rasmussen & Williams, 2006)

$$E[\pi^* \,|\, \mathbf{y}] \approx \int g^{-1}(f^*)\, q(f^* \,|\, \mathbf{y}) df^*. \tag{3}$$

Menettelyä dikotomisella vasteella on havainnollistettu kuvassa 2.1, missä ylhäällä on $n = 100$ otos latentin f priorista $\mathcal{GP}(0, \exp\left(-(\mathbf{x} - \mathbf{x}')^2/2\right))$ ja alhaalla vastaava otos posterioriennustejakaumasta parametrille $\pi = g^{-1}(f^\star)$ sekä mustalla värillä kuvattuna tämän posteriorikeskiarvo.



Kuva 2.1: Otokset (n=100) GP-priorista $f \sim \mathcal{GP}(0, \exp\left(-(\mathbf{x} - \mathbf{x}')^2/2\right))$ (ylh.) ja kuvaan lisätyillä havainnoilla parametrille $\pi = g^{-1}(f^\star)$ määrätystä posterioriennustejakaumasta (posteriorikeskiarvo mustalla).

Gaussista prosessia ei käytännössä kyetä arvioimaan koko määrittelyjoukossa. Sovelluksissa kuitenkin riittää tarkastella indeksijoukon äärellisiä kokoelmia $\mathbf{x}_i \in \mathcal{D}$, missä $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ eli havaittu aineisto sekä uusia arvoja $\mathbf{x}_j^*, j = 1, \ldots, P$, joiden ennusteista ollaan kiinnostuneita (Gelman et al., 2014). Kuvan 2.1 kuvaajissa Gaussisen prosessin eli funktion f arvoja on tarkasteltu sadassa väliltä $[0, 12]$ tasavälein määrätyssä pisteessä, mutta kuvaajat piirretty jatkuvina.

2.3 Posteriorijakauman approksimointi

Edellisessä luvussa posteriorijakauman $p(\mathbf{f} \mid \mathbf{y})$ approksimaatio $q(\mathbf{f} \mid \mathbf{y})$ jätettiin yksilöimättä. Tässä luvussa esitellään tämän tutkielman sovelluksissa käytetyt analyyttiset approksimaatiot eli Laplace -menetelmä sekä variationaalinen approksimointi. Kattavampi kuvaus erilaisista jakaumallisista approksimaatioista löytyy esimerkiksi teoksesta Gelman et al. (2014).

2.3.1 Laplace -menetelmä

Laplace -menetelmä (Tierney & Kadane, 1986) perustuu vektorin $\mathbf{f}$ posteriorimoodin ympärille sovitettavaan normaalijakaumaan. Koska $\mathbf{f}$ a priori noudattaa moniulotteista normaalijakaumaa, on myös posteriorijakauma usein lähellä normaalijakaumaa (Gelman et al., 2014). Koska tässä tutkielmassa Laplace -approksimaatiota tullaan soveltamaan luvussa 2.1 esitettyyn Gaussisen prosessin parametriseen esitykseen

$$f(\mathbf{x}) = \sum_j \beta_j \, \phi_j(\mathbf{x}),$$

on Laplace -menetelmä seuraavassa kuvattu posteriorijakaumalle $p(\boldsymbol{\beta} \mid \mathbf{y})$ ja latentin Gaussisen prosessin mallia koskevat tulokset eli approksimaatiot jakaumille $p(\mathbf{f} \mid \mathbf{y})$ ja $p(f^\star \mid \mathbf{y})$ sivuutettu. Sivuutettujen tulosten osalta viitataan tässä teokseen Rasmussen ja Williams (2006).

Laplace -approksimaatio johdetaan posteriorijakauman logaritmin toisen asteen Taylorin sarjakehitelmästä posteriorimoodia $\hat{\boldsymbol{\beta}}$ vastaassa pisteessä (Rasmussen & Williams, 2006). Approksimaatio on muotoa

$$\begin{aligned} \log p(\boldsymbol{\beta} \mid \mathbf{y}) &\approx \log p(\hat{\boldsymbol{\beta}} \mid \mathbf{y}) + \frac{1}{2}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^T \nabla^2 \log p(\hat{\boldsymbol{\beta}} \mid \mathbf{y}) \, (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}) \\ &\propto \exp\left(-\frac{1}{2}(\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})^T \boldsymbol{\Sigma}_\beta^{-1} (\boldsymbol{\beta} - \hat{\boldsymbol{\beta}})\right) \end{aligned}$$

sillä $\nabla \log p(\hat{\boldsymbol{\beta}} \mid \mathbf{y}) = \mathbf{0}$ ja merkitään $\boldsymbol{\Sigma}_\beta^{-1} = -\nabla^2 \log p(\hat{\boldsymbol{\beta}} \mid \mathbf{y})$. Edellinen voidaan tunnistaa normaalijakauman $N(\hat{\boldsymbol{\beta}}, \boldsymbol{\Sigma}_\beta)$ tiheysfunktioksi eli Laplace -approksimaation alla $p(\boldsymbol{\beta} \mid \mathbf{y}) \approx q(\boldsymbol{\beta})$, missä $q \sim N(\hat{\boldsymbol{\beta}}, \boldsymbol{\Sigma}_\beta)$. Posteriorimoodi $\hat{\boldsymbol{\beta}}$ voidaan estimoida Newtonin menetelmällä (Rasmussen & Williams, 2006).

2.3.2 Variationaaliset menetelmät

Variationaalisissa menetelmissä (Jordan et al., 1999) parametrisoidaan jakaumaperhe Q_ϕ ja etsitään tästä jakauma $\hat{q}(\mathbf{f} \mid \boldsymbol{\phi})$ siten, että

$$\hat{q}(\mathbf{f} \mid \boldsymbol{\phi}) = \underset{q \in Q}{\operatorname{argmin}} \, \mathrm{KL}(q(\mathbf{f} \mid \boldsymbol{\phi}) \,||\, p(\mathbf{f} \mid \mathbf{y})).$$

missä $\mathrm{KL}(q \,||\, p) = \mathbb{E}_q\,[\log q - \log p]$ on jakauman p ja jakauman $q \in Q$ välinen *Kullback-Leibler* -poikkeavuus (Gelman et al., 2014). Minimointi suoritetaan jakaumaperheen Q_ϕ indusoimien hyperparametrien $\boldsymbol{\phi}$ eli *variationaalisten parametrien* suhteen. Jakauman q riippuvuutta variationaalisista parametreista ei jatkossa merkitä erikseen näkyviin, vaan oletetaan tunnetuksi, että $q(\mathbf{f}) = q(\mathbf{f} \mid \boldsymbol{\phi})$. Ellei erikseen toisin ole merkitty, lähteenä seuraavassa on käytetty esitystä Blei et al. (2017).

KL -poikkeavuus ei sellaisenaan sovi optimointiongelman tavoitteeksi, koska siihen sisältyy posteriorijakauman $p(\mathbf{f} \mid \mathbf{y})$ kautta marginaalisen uskottavuuden sisältävä termi $\log p(\mathbf{y})$, mikä nimenomaan on syy siihen, että posterijakaumaa joudutaan approksimoimaan. Edellinen voidaan todeta laajentamalla posteriorijakauma $p(\mathbf{f} \mid \mathbf{y})$ KL -poikkeavuuden määritelmän sisällä

$$\begin{aligned}
\mathrm{KL}(q(\mathbf{f}) \,||\, p(\mathbf{f} \mid \mathbf{y})) &= \mathbb{E}_q\,[\log q(\mathbf{f}) - \log p(\mathbf{f} \mid \mathbf{y})] \\
&= \mathbb{E}_q\,[\log q(\mathbf{f})] - \mathbb{E}_q\left[\log\left[\frac{p(\mathbf{f}, \mathbf{y})}{p(\mathbf{y})}\right]\right] \\
&= \mathbb{E}_q\,[\log q(\mathbf{f})] - \mathbb{E}_q\,[\log p(\mathbf{f}, \mathbf{y})] + \log p(\mathbf{y}),
\end{aligned}$$

missä viimeinen rivi seuraa siitä, ettei marginaalinen uskottavuus $p(\mathbf{y})$ riipu muuttujasta $\mathbf{f}$ eli $\mathbb{E}_q[\log p(\mathbf{y})] = \log p(\mathbf{y})$. KL -poikkeavuuden sijaan tavoitefunktioksi variationaalisissa menetelmissä asetetaankin ns. ELBO -funktio (*evidence lower bound*)

$$\begin{aligned}
\mathrm{ELBO}(q) &= \mathbb{E}_q\,[\log p(\mathbf{f}, \mathbf{y})] - \mathbb{E}_q\,[\log q(\mathbf{f})] && (4) \\
&= -\mathrm{KL}(q(\mathbf{f}) \,||\, p(\mathbf{f} \mid \mathbf{y})) + \log p(\mathbf{y}) && (5) \\
&= \mathbb{E}_q\,[\log p(\mathbf{y} \mid \mathbf{f})] - \mathrm{KL}(q(\mathbf{f}) \,||\, p(\mathbf{f})), && (6)
\end{aligned}$$

missä rivin (6) muoto on saatu laajentamalla yhteisjakauma $p(\mathbf{f}, \mathbf{y}) = p(\mathbf{f})\, p(\mathbf{y} \mid \mathbf{f})$ rivillä (4). Rivin (6) esityksessä ensimmäinen termi on odotettu log-uskottavuus ja jälkimmäinen termi approksimaation $q(\mathbf{f})$ ja priorin $p(\mathbf{f})$ välinen KL-poikkeavuus eli ELBO -funktion suosimissa jakaumissa tulisi ilmetä vastaava tasapaino uskottavuuden ja priorin suhteen, kuin posteriorijakaumassa $p(\mathbf{f} \mid \mathbf{y}) \propto p(\mathbf{y} \mid \mathbf{f})\, p(\mathbf{f})$.

Edelleen koska marginaalinen uskottavuus $p(\mathbf{y})$ on jakauman q suhteen vakio, riviltä (5) voidaan nähdä, että ELBO-funktion maksimointi on tavoitteena ekvivalentti KL-poikkeavuuden minimoinnin kanssa. Koska lisäksi KL-poikkeavuudelle pätee $\mathrm{KL}(q \,||\, p) \geq 0$ ja $\mathrm{KL}(q \,||\, p) = 0$ vain, kun $q = p$ (Kullback & Leibler, 1951), riviltä (5) seuraa myös, että

$$\log p(\mathbf{y}) \geq \mathrm{ELBO}(q), \quad q \in Q.$$

Edellinen tarkoittaa sitä, että ELBO asettaa alarajan (*lower bound*) marginaaliselle uskottavuudelle (*evidenssille*).

2.4 Approksimaatiot suurille aineistoille

Edellisessä luvussa esitetyillä menetelmillä voidaan approksimoida Gaussisen prosessin posteriorijakaumaa, kun sitä ei saada ratkaistua analyyttisesti. Tässä luvus-

sa käsiteltäviä menetelmiä puolestaan on esitetty approksimaatioiksi silloin, kun aineisto on suuri. Kuten luvun johdannossa todettiin, $N \times N$ kovarianssimatriisin kääntämisen vaativuusluokka $\mathcal{O}(N^3)$ muodostuu esteeksi ekstaktille päättelylle, kun aineisto on liian suuri.

Rasmussen ja Williams (2006) mukaan aineiston koko muodostuu esteeksi, kun $N > 10000$. Hensman et al. (2013) kuitenkin toteaa, että tarve approksimaatioille nousee jo, kun aineisto sisältää enemmän kuin muutaman tuhatta havaintoa, mitä perinteisesti on pidetty Gaussisille prosesseille suurena. Keskeisimpinä suurille aineistoille esitettyinä approksimaatioina voidaan pitää seuraavaksi esiteltäviä satunnaisiin Fourier piirteisiin ja indusoiviin muuttujiin perustuvia approksimaatioita (van Amersfoort et al., 2021b). Molemmilla menetelmillä kovarianssimatriisin kääntämisen vaativuusluokaksi tulee $\mathcal{O}(NM^2)$, missä M voidaan valita siten, että $M < N$.

2.4.1 Satunnaiset Fourier -piirteet

Luvussa 2.1 todettiin, että Gaussista prosessia $f \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$ vastaa ekvivalentti esitys parametrisena kantafunktioregressiona $f(\mathbf{x}) = \sum_k \beta_k \, \phi_k(\mathbf{x})$. Satunnaisilla Fourier -piirteillä Gaussinen prosessi voidaan muuntaa tätä approksimoivaksi kantafunktioregressioksi ja soveltaa tässä laskennallisesti kevyempiä lineaarisia menetelmiä (Rahimi & Recht, 2007). Kovarianssimatriisin approksimaation kääntämisen vaativuusluokkaa koskee, mitä luvun johdannossa todettiin(Ton et al., 2018).

Satunnaiset Fourier -piirteet perustuvat alkuperäisten piirteiden dimensiota alentavaan muunnokseen $\boldsymbol{\phi} : \mathbb{R}^P \to \mathbb{R}^M$, missä $M < P$ ja tyypillisesti myös $M < N$ (Rahimi & Recht, 2007). Piirteet on seuraavassa johdettu vastaavalla periaatteella, kuin Hensman et al. (2017), mitä tapaa on pidetty alkuperäisessä julkaisussa Rahimi ja Recht (2007) esitettyä intuitiivisempana. Approksimaation antavan muunnoksen $\boldsymbol{\phi}$ johtaminen perustuu Fourier -muunnokseen ja Bochnerin lauseeseen (Bochner, 1959). Fourier -muunnos on määritelty kompleksiluvuille, mihin liittyvää teoriaa on seuraavaan esitykseen sisällytetty vain sen verran, kuin esityksen seuraamiseksi on tarpeellista.

Määritelmä 2.5 (Fourier -muunnos)**.** Funktion $f(\mathbf{x})$ Fourier -muunnos $\hat{f}(\omega)$ määritellään

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(x) \, e^{-i\omega x} \, dx.$$

Määritelmä on esitetty kuten (Herman, 2016). Määritelmässä i on ns. *imaginääriyksikkö*, jolle $i^2 = -1$. Kompleksiselle eksponenttifunktiolle e^{iy} pätee $e^{iy} = cos\, y + i \, sin\, y$ (Freitag, 2009).

Lause 2.6 (Bochnerin lause)**.** *Jatkuva ydinfunktio $k(x, x') = k(x - x'), x \in \mathbb{R}^n$ on positiivisesti definiitti, jos ja vain jos $k(\delta) = k(x - x')$ on ei-negatiivisen mitan $p(\omega)$ Fourier -muunnos.*

Mikäli $k(0) = 1$, mitta $p(\omega)$ on todennäköisyysmitta (Sutherland & Schneider, 2015). Huomautetaan ensin, että kovarinssifunktio on ydinfunktio ja määritelmänsä nojalla positiivisesti definiitti. Nyt Fourier -muunnoksen määritelmästä ja Bochnerin lauseesta seuraa, että stationääriselle kovarianssifunktiolle

$$\begin{aligned} k(\mathbf{x} - \mathbf{x}') &= \int_{-\infty}^{\infty} p(\boldsymbol{\omega}^T)\, e^{-i\boldsymbol{\omega}(\mathbf{x}-\mathbf{x}')}\, d\boldsymbol{\omega} \\ &= \int_{-\infty}^{\infty} p(\boldsymbol{\omega})\, (cos(\boldsymbol{\omega}^T(\mathbf{x} - \mathbf{x}')) - i\, sin(\boldsymbol{\omega}^T(\mathbf{x} - \mathbf{x}')))\, d\boldsymbol{\omega} \qquad (7) \\ &= \int_{-\infty}^{\infty} p(\boldsymbol{\omega})\, cos(\boldsymbol{\omega}^T(\mathbf{x} - \mathbf{x}'))\, d\boldsymbol{\omega} \qquad (8) \\ &= \mathbb{E}_{\omega}\left[cos(\boldsymbol{\omega}^T(\mathbf{x} - \mathbf{x}'))\right], \qquad (9) \end{aligned}$$

missä rivi (7) seuraa kompleksisen eksponenttifunktion ominaisuuksista ja rivillä (8) on käytetty tietoa, että sekä kovarianssifunktio k että todennäköisyysmitta $p(\boldsymbol{\omega})$ ovat reaaliarvoisia (Rahimi & Recht, 2007). Kovarianssifunktion approksimaationa voidaan nyt käyttää odotusarvon (9) Monte Carlo -estimaattia

$$\mathbb{E}_{\omega}\left[cos(\boldsymbol{\omega}^T(\mathbf{x} - \mathbf{x}'))\right] \approx \frac{1}{M}\sum_{k=1}^{M} cos(\boldsymbol{\omega}_k^T(\mathbf{x} - \mathbf{x}')), \qquad (10)$$

kun poimitaan otoksia $\boldsymbol{\omega}_k \sim p(\boldsymbol{\omega})$. Alkuperäisessä julkaisussa (Rahimi & Recht, 2007) käytetään tulosta $\mathbb{E}_{\omega}\left[cos(\boldsymbol{\omega}^T(\mathbf{x} - \mathbf{x}'))\right] = \mathbb{E}_{\omega}\left[\sqrt{2}\, cos(\boldsymbol{\omega}^T\mathbf{x} + b)\sqrt{2}\, cos(\boldsymbol{\omega}^T\mathbf{x}' + b)\right]$, kun määrätään $b \sim U(0, 2\pi)$. Tällöin merkitsemällä $\phi_k(\mathbf{x}) = \sqrt{2}\, cos(\boldsymbol{\omega}_k^T\mathbf{x} + b)$ ja $\boldsymbol{\Sigma}_\beta = {}^1\!/_M\, \mathbf{I}$ voidaan kirjoittaa

$$\begin{aligned} k(\mathbf{x} - \mathbf{x}') &= \mathbb{E}_{\omega}\left[cos(\boldsymbol{\omega}^T(\mathbf{x} - \mathbf{x}'))\right] \\ &\approx \frac{1}{M}\sum_{k=1}^{M} \sqrt{2}\, cos(\boldsymbol{\omega}_k^T\mathbf{x} + b_k)\sqrt{2}\, cos(\boldsymbol{\omega}_k^T\mathbf{x}' + b_k) \\ &= \boldsymbol{\phi}(\mathbf{x})^T\, \boldsymbol{\Sigma}_\beta\, \boldsymbol{\phi}(\mathbf{x}). \end{aligned}$$

Kun muistetaan kovarianssifunktiolle kantafunktioesityksen yhteydessä luvussa 2.1 esitetty muoto, voidaan edellisestä lukea Gaussiselle prosessille approksimaatio kantafunktioregressiona

$$\begin{aligned} f(\mathbf{x}) = \sum_k \beta_k\, \phi_k(\mathbf{x}) = \boldsymbol{\beta}^T\, \boldsymbol{\phi}(\mathbf{x}), \qquad (11) \\ \boldsymbol{\beta} \sim N\left(\mathbf{0}, \frac{1}{M}\mathbf{I}\right). \end{aligned}$$

Kantafunktioiden määrämät satunnaiset piirteet $\boldsymbol{\phi}(\mathbf{x}_i) = (\phi_1(\mathbf{x}_i), \ldots, \phi_M(\mathbf{x}_i))^T$ saadaan siis poimilla otoksia $\boldsymbol{\omega}_k \sim p(\boldsymbol{\omega})$ ja $b_k \sim U(0, 2\pi)$ ja määräämällä $\phi_k(\mathbf{x}_i)$. Tarvittava todennäköisyysmitta $p(\boldsymbol{\omega})$ saadaan Fourier -muunoksen käänteismuunnoksella (Herman, 2016). Neliöidylle eksponenttifunktiolle, kun määritelmässä 2.3 $\tau = 1$ ja $\ell = 1$ (Rahimi & Recht, 2007)

$$p(\boldsymbol{\omega}) = (2\pi)^{-\frac{M}{2}} e^{-\frac{\boldsymbol{\omega}^T\boldsymbol{\omega}}{2}},$$

mikä voidaan tunnistaa jakauman $N(\mathbf{0}, \mathbf{I})$) tiheysfunktioksi. Hyperparametri τ voidaan huomioida approksimaatiossa (10) huomaamalla, että jos k' on neliöity eksponenttifunktio, missä $\tau' = 1$, tällöin $k(\mathbf{x}, \mathbf{x}') = \tau^2\, k'(\mathbf{x}, \mathbf{x}')$ ja esityksessä (11),

$$\boldsymbol{\beta} \sim N\left(\mathbf{0}, \frac{\tau^2}{M}\,\mathbf{I}\right),$$

mikä vastaisi Hensman et al. (2017) käytettyä muotoa.

Merkitään nyt $\boldsymbol{\Phi}$ $N \times M$ matriisia, minkä riveinä ovat satunnaiset Fourier -piirteet $\boldsymbol{\phi}(\mathbf{x}_i)$ havainnoille $\mathbf{x}_i$ eli

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi_{11}(\mathbf{x}_1) & \cdots & \phi_{1M}(\mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ \phi_{1N}(\mathbf{x}_N) & \cdots & \phi_{NM}(\mathbf{x}_N) \end{bmatrix}.$$

Approksimaatiossa kovarianssimatriisille $\mathbf{K}$ voidaan nyt käyttää matriisituloa $\boldsymbol{\Phi}\boldsymbol{\Phi}^T$, minkä asteelle pätee $\text{Rank}(\boldsymbol{\Phi}\boldsymbol{\Phi}^T) \leq \min(\text{Rank}(\boldsymbol{\Phi}), \text{Rank}(\boldsymbol{\Phi}^T)) \leq M$ (Ton et al., 2018). Kovarianssimatriisin kääntämisen vaativuusluokasta luvun alussa mainittu tulos seuraa nyt Woodburyn identiteetistä (Woodbury, 1950), minkä tarkempi esittäminen tässä sivuutetaan.

2.4.2 Indusoivat muuttujat

Indusoiviin muuttujiin perustuvien approksimaatioiden perusajatus on käsitellä vain osaa latenteista muuttujista eksaktisti ja käyttää muiden kohdalla jotain laskennallisesti kevyempää approksimaatiota (Quinonero-Candela & Rasmussen, 2005). Yhteisenä tekijänä approksimaatioissa on mallin augmentointi *indusoivilla muuttujilla* $\mathbf{u} = (u_1, \ldots, u_M)^T$, mitkä ovat latentin funktion $\mathbf{f}$ arvoja $f(\mathbf{z_1}), \ldots, f(\mathbf{z_M})$ *indusoivissa pisteissä* $\mathbf{z}_i \in \mathbb{R}^M$. Indusoiville muuttujille oletetaan sama GP-priori, kuin muuttujille $\mathbf{f}$ eli $\mathbf{f}, \mathbf{u} \sim \mathcal{GP}(0, k(\mathbf{x}_i, \mathbf{x}_j))$ ja yhteisjakaumalle $p(\mathbf{f}, \mathbf{u})$ (Hensman et al., 2015)

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{u} \end{bmatrix} \sim N\left(\mathbf{0}, \begin{bmatrix} \mathbf{K}_{nn} & \mathbf{K}_{nm} \\ \mathbf{K}_{mn} & \mathbf{K}_{mm} \end{bmatrix}\right).$$

Edellä lohkomatriisien alaindekseillä on nyt viitattu matriisien dimensiohin.

Lähes kaikissa approksimaatioissa oletetaan lisäksi, että $\mathbf{f}$ ja $f^\star$ ovat riippumattomia ehdolla $\mathbf{u}$ (Quinonero-Candela & Rasmussen, 2005). Tällöin augmentoidussa mallissa $p(\mathbf{y}, \mathbf{f}, \mathbf{u})$ posterioriennustejakaumalle (Titsias, 2009)

$$
\begin{aligned}
p(f^\star \,|\, \mathbf{y}) &= \int p(f^\star, \mathbf{f}, \mathbf{u} \,|\, \mathbf{y})\, d\mathbf{f}\, d\mathbf{u} \\
&= \int p(f^\star \,|\, \mathbf{u})\, p(\mathbf{f} \,|\, \mathbf{u})\, p(\mathbf{u} \,|\, \mathbf{y})\, d\mathbf{f}\, d\mathbf{u}. \qquad (12)
\end{aligned}
$$

Edellisestä nähdään, kuinka muuttujat $\mathbf{u}$ *indusoivat* riippuvuuden opetushavaintoja ja uusia havaintoja vastaavien muuttujien $\mathbf{f}$ ja $f^\star$ välille (Quinonero-Candela & Rasmussen, 2005).

Indusoivina pisteinä voidaan käyttää esimerkiksi osajoukkoa opetusaineistosta. Tässä luvussa esitettävässä menetelmässä (Titsias, 2009) indusoivia muuttujia ja pisteitä pidetään variationaalisina parametreina ja estimoidaan vastaavasti, kuin luvussa 2.3.2 kuvattiin. Variationaalinen approksimaatio $q(\mathbf{u})$ asetetaan kaavassa (12) indusoivien muuttujien posteriorijakaumalle $p(\mathbf{u} \,|\, \mathbf{y})$, mitä ei-konjugaattisen uskottavuuden tapauksessa muutoinkin jouduttaisiin approksimoimaan. Jakaumalle q oletetaan $q(\mathbf{u}) = N(\mathbf{m}, \mathbf{S})$ (Hensman et al., 2015) eli variationaalisina parametreina on odotusarvovektori $\mathbf{m}$ ja kovarianssimatriisi $\mathbf{S}$.

Johdetaan seuraavaksi marginaaliselle uskottavuudelle uusi alaraja aloittaen luvun 2.3.2 kaavasta (6). Seuraava esitys perustuu julkaisuun Hensman et al. (2015). Todetaan ensin, että

$$
p(\mathbf{y}) = \int p(\mathbf{y} \,|\, \mathbf{u})\, q(\mathbf{u}) = \mathbb{E}_{q(\mathbf{u})}\left[p(\mathbf{y} \,|\, \mathbf{u})\right], \qquad (13)
$$

$$
p(\mathbf{y} \,|\, \mathbf{u}) = \int p(\mathbf{y} \,|\, \mathbf{f})\, p(\mathbf{f} \,|\, \mathbf{u}) d\mathbf{f} = \mathbb{E}_{p(\mathbf{f} \,|\, \mathbf{u})}\left[p(\mathbf{y} \,|\, \mathbf{f})\right] \qquad (14)
$$

ja muistetaan Jensenin epäyhtälö (Jensen, 1906)

$$
\mathbb{E}\varphi(X) \geq \varphi(\mathbb{E}(X)), \qquad (15)
$$

missä φ on konveksi funktio, kuten logaritmi. Nyt

$$
\begin{aligned}
\log p(\mathbf{y}) &\geq \log p(\mathbf{y}) - \mathrm{KL}(q(\mathbf{u}) \,||\, p(\mathbf{u})) \\
&\geq \mathbb{E}_{q(\mathbf{u})}\left[\log p(\mathbf{y} \,|\, \mathbf{u})\right] - \mathrm{KL}(q(\mathbf{u}) \,||\, p(\mathbf{u})) \qquad (16) \\
&\geq \mathbb{E}_{q(\mathbf{u})}\left[\mathbb{E}_{p(\mathbf{f} \,|\, \mathbf{u})}\left[\log p(\mathbf{y} \,|\, \mathbf{f})\right]\right] - \mathrm{KL}(q(\mathbf{u}) \,||\, p(\mathbf{u})) \qquad (17) \\
&= \mathbb{E}_{q(\mathbf{f})}\left[\log p(\mathbf{y} \,|\, \mathbf{f})\right] - \mathrm{KL}(q(\mathbf{u}) \,||\, p(\mathbf{u})) \qquad (18) \\
&= \sum_i \mathbb{E}_{q(f_i)}\left[\log p(y_i \,|\, f_i)\right] - \mathrm{KL}(q(\mathbf{u}) \,||\, p(\mathbf{u})),
\end{aligned}
$$

missä rivit (16) ja (17) seuraavat kaavoista (13) ja (14) ja näillä saatuihin odotusarvoihin sovelletusta Jensenin epäyhtälöstä (15) ja rivillä (18) on määrätty $q(\mathbf{f}) = \int p(\mathbf{f} \,|\, \mathbf{u})\, q(\mathbf{u})\, d\mathbf{u}$.

Edeltä voidaan huomata, että vaikka variationaaliset parametrit $\mathbf{m}$ ja $\mathbf{S}$ eivät riipu indusoivien pisteiden $\mathbf{z}$ sijainnista, tulevat myös nämä estimoitaviksi parametreiksi marginaalisen priorin $p(\mathbf{u})$ ja ehdollisen priorin $p(\mathbf{f} \,|\, \mathbf{u})$ kautta (Titsias, 2009). Tämä johtuu siitä, että jakaumassa p kuten alussa todettiin, $u_i = f(\mathbf{z}_i)$.

Jakaumalle $q(\mathbf{f})$ odotusarvossa (18) saadaan (Hensman et al., 2015)

$$q(\mathbf{f}) = N(\mathbf{Am}, \mathbf{K}_{nn} + \mathbf{A}(\mathbf{S} - \mathbf{K}_{mm})\mathbf{A}^T,$$

missä $\mathbf{A} = \mathbf{K}_{nm}\mathbf{K}_{mm}^{-1}$ ja odotusarvo voidaan laskea käyttäen marginaalijakaumia $q(f_i)$. Viimeisellä rivillä saatu faktoroituva muoto mahdollistaa parametrien estimoinnin stokastisilla osajoukkoja käyttävillä optimointialgoritmeilla, kuten luvussa 1.1.

3 Aineistot ja menetelmät

Tässä luvussa esitellään kaksi verrattain uutta Gaussisen prosessin neuroverkkoon yhdistävää syväoppimismenetelmää. Gaussinen prosessi voidaan yleisesti kuvattuna yhdistää neuroverkkoon käyttämällä Gaussisen prosessin kovarianssifunktiossa $k(\mathbf{x}, \mathbf{x}')$ syötteinä alkuperäisten piirteiden $\mathbf{x}_i$ sijaan neuroverkolla johdettuja piirteitä $\mathbf{h}_i = f^{(L-1)}(\mathbf{x}_i)$ (Wilson et al., 2016), missä merkinnät ovat, kuten luvussa 1.1. Gaussinen prosessi implementoidaan mallin ulostuloskerrokseen, missä kovarianssifunktioksi nyt tulee $k(\mathbf{h}, \mathbf{h}')$. Jatkossa tähän kerrokseen viitataan termillä GP-kerros. Tavoitteena mallien yhdistämisessä on konstruoida malli, missä yhdistyisi Gaussisten prosessien kyky luotettavasti kuvata epävarmuutta ja neuroverkkojen kyky oppia monimutkaisia esityksiä alkuperäisille piirteille $\mathbf{x}_i$ (Ober et al., 2021).

Seuraavassa kuvattuja menetelmiä on aikaisemmin vertailtu kuvantunnistukseen liittyvissä tehtävissä (van Amersfoort et al., 2021b) konvoluutioarkkitehtuuriin perustuvilla malleilla (*Wide Residual Networks*; Zagoruyko & Komodakis, 2016). Tässä tutkielmassa menetelmien implementaatioita verrataan luonnollisen kielen käsittelyyn kehitetyllä, luvussa 1.2 kuvattua transformer -arkkitehtuuria soveltavalla BERT -mallilla sentimenttianalyysiin liittyvässä tehtävässä.

Menetelmien kuvaukset on esitetty luvussa 3.1. Tutkimuskysymykset on muotoiltu luvussa 3.2, missä on kerrottu myös menetelmien arvioinnista sekä arviointiin liittyvistä käsitteistä. Vertailuissa käytetyt aineistot on kuvattu luvussa 3.3.

3.1 Implementoidut mallit

Seuraavassa on ensin kuvattu BERT -malli ja samalla kerrottu tekstin muodossa olevien syötteiden muuntamisesta numeeriseen muotoon. BERT -mallia käytettiin vertailuissa sekä itsenäisenä mallina ilman GP-kerrosta implementoidun mallin suorituskykyä kuvaamaan että GP-kerroksen implementoineissa malleissa johdetut piirteet $\mathbf{h}$ GP-kerrokselle antavana osana (BERT-enkooderi).

GP-kerroksen implementoineet mallit on kuvattu luvuissa 3.1.2 ja 3.1.3. Menetelmät on tässä kuvattu siten, kuin ne on alkuperäisissä julkaisuissa esitetty. Yksityiskohtaisemmat tiedot menetelmien implementoinneista on annettu liitteessä A. Teorian osalta tarvittavat tiedot on annettu luvuissa 1 ja 2.

3.1.1 BERT-malli

Vastaavasti kuin luvussa 1.2 esitettiin, BERT-malli ottaa syötteenä matriisimuotoisen esityksen tekstistä. Luvussa 1.2 matriisin $\mathbf{X}$ riveiksi oletettiin esityksen yksinkertaistamiseksi yksittäiset sanat. Tosiasiassa teksti muutetaan numeeriseen muotoon *saneiden* avulla ja kukin matriisin rivi on vektorimuotoinen esitys yksittäisestä saneesta. Saneet (*token*) voivat olla kokonaisia sanoja tai sanan osia, jolloin kokonaisia sanoja saadaan saneita sopivasti yhdistämällä. Saneet määrätään tähän tarkoitukseen kehitetyillä algoritmeilla kohdekieltä edustavasta tekstikokoelmasta ja kootaan sanakirjaksi, missä kutakin sanetta vastaa järjestysluku $n_s = 1, \ldots, N_s$. Numeerinen esitys tekstistä saadaan nyt *saneistamalla* teksti ja korvaamalla kukin sane tämän järjestysluvulla.

Edellä kuvattua menettelyä on havainnollistettu taulukolla 1 käyttäen sanakirjana tässä tutkielmassa käytettyä Turun yliopistossa BERT-mallille rakennettua suomenkielistä sanakirjaa (Virtanen et al., 2019). Sanakirja käsittää 50000 sanetta ja se on rakennettu käyttäen tekstikokoelmina uutisartikkeleita sisältävää Yle -tekstikokoelmaa, kirjoituksia keskustelupalstoilta sisältävää Suomi24 -tekstikokoel- maa ja suomenkielisiä internetin ryöminnällä (*crawl*) saatuja tekstejä siten, että yhdistetty tekstikokoelma käsitti 234 miljoonaa virkettä. Yksityiskohtien osalta tässä viitataan julkaisuun Virtanen et al. (2019) ja sanakirjan rakentamisessa käytettyjen algoritmien osalta julkaisuihin Sennrich et al. (2015) sekä Kudo ja Richardson (2018).

Taulukko 1: Esimerkki tekstin saneistamisesta ja muuntamisesta numeeriseen muotoon. Esimerkkivirke ensimmäisellä rivillä, toisella rivillä virkkeen esitys saneiden avulla ja kolmannella rivillä saneita vastaavat sanakirjan järjestysnumerot.

Teksti	Kaikki	meni	erittäin	ketterästi			ja	asiat	hoituivat		hienosti	.
Saneet	[Kaikki]	[meni]	[erittäin]	[ket]	[##terä]	[##sti]	[ja]	[asiat]	[hoit]	[##uivat]	[hienosti]	[.]
Vektori	1975	2142	1912	2817	26548	1024	142	2336	1859	12446	8734	111

Edellisessä esimerkissä saneita "ketterästi" ja "hoituivat" ei ollut käytetyssä sanakirjassa, joten ne muodostettiin saneita yhdistämällä. Merkintä ## saneen edellä tarkoittaa sitä, että kyseistä sanetta tulee edeltää jokin toinen sane eli esimerkiksi saneet [terä] ja [##terä] ovat eri saneita. BERT-mallissa ensimmäiseksi saneeksi jokaisessa tekstissä asetetaan erityissane [CLS], minkä merkityksestä on kerrottu jäljempänä. Teksti päätetään erityissaneeseen [SEP]. Kaikki tekstit muutetaan tasamittaisiksi, joko täyttämällä teksti haluttuun mittaan erityissaneilla [PAD] tai katkaisemalla teksti siten, että pituudeksi [CLS] ja [SEP] -saneet mukaanlukien tulee asetettu mitta.

Merkitään nyt $\mathbf{x}$ edellä kuvatulla tavalla tekstille saatua esitystä saneiden järjestysluvuista muodostettuna vektorina. Saneiden vektoriesitykseen päästään muunnoksella $\mathbf{E} = \mathbf{W}_t\mathbf{x}$, missä $D_t \times D_s$ matriisin $\mathbf{W}_t$ alkiot ovat opittavia parametreja. Matriisin $\mathbf{E}$ riviä i, eli vektoria $\mathbf{e}_i$, kutsutaan sanan (saneen) upotukseksi (*embedding*). Jatkossa oletetaan, että $D_t = 128$ eli tekstien mitta on 128 sanetta. Edelleen oletetaan, että $D_s = 768$ eli kukin sane esitetään 768-ulotteisena vektorina, kuten myös BERT-mallissa on oletuksena.

Saneiden sijaintia koskeva informaatio tuodaan syötteeseen lisäämällä saneiden upotusvektoreihin $\mathbf{e}_i$ näiden sijaintia kuvaavat positiovektorit $\mathbf{p}_i$ (*positional embedding*),

mitkä BERT-mallissa opitaan aineistosta mallia sovitettaessa. Koska samaa mallia voidaan käyttää myös esimerkiksi kysymys-vastaus -tehtävissä, joissa syötteenä käytetään tekstiparia eli

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_A \\ \mathbf{X}_B \end{bmatrix},$$

missä A ja B viittaavat tekstin segmentteihin, lisätään upotusvektoreihin $\mathbf{e}_i$ vielä opittavat segmenttivektorit $\mathbf{s}_i$, mitkä kuvaavat saneiden kuulumista segmenttiin A tai B. Tekstiparista muodostuvia syötteitä ja segmenttivektoreita ei käsitellä tässä tutkielmassa tämän tarkemmin, koska sovelluksena tässä on luokittelutehtävä, missä syötteitä käsitellään yhtenä tekstinä. Mallille annettavaksi syötteeksi $\mathbf{X}_{in}$ saadaan edellisistä $\mathbf{X}_{in} = \mathbf{E} + \mathbf{P} + \mathbf{S}$, missä matriisin $\mathbf{P}$ riveinä ovat positiovektorit $\mathbf{p}_i$, matriisin $\mathbf{S}$ riveinä vastaavasti segmenttivektorit $\mathbf{s}_i$ ja kaikkien matriisien dimensiot $D_t \times D_s$.

Varsinainen BERT-malli muodostuu 12 peräkkäin kytketystä Transfor- mer-lohkosta sekä näiden perään luokittelutehtävissä liitettävästä tiheästä ulostulokerroksesta, missä yksiköiden lukumäärä riippuu luokkien lukumäärästä. Luokittelutehtävissä käytetään ainoastaan jo aiemmin mainittua [CLS] -sanetta, minkä on tarkoitus tiivistää koko tekstiä kuvaavaa informaatiota. Käytännössä viimeisen Tranformer-lohkon ja ulostulokerroksen välissä käytetään [CLS] -saneelle vielä tiheää ns. koontikerrosta (*pooled output / pooler layer*), missä yksiköiden lukumäärä on Transformer -kerrosten ulostuloa vastaavasti 768 ja missä aktivointifunktiona käytetään hyperbolista tangenttia

$$\tanh(z) = \frac{e^{2z} - 1}{e^{2z} + 1}.$$

BERT-malli on esiopetettu malli, mikä tarkoittaa sitä, että mallin kertoimet voidaan alustaa esiopetetuista arvoista ja tämän jälkeen hienosäätää käsillä olevaan tehtävään sopiviksi. Esiopetuksessa käytetään ohjaamattomaan oppimiseen liittyviä tehtäviä ja esimerkiksi tässä tutkielmassa käytettyä Turun yliopistossa suomen kielellä esiopetettua FinBERT-mallia on esiopetettu noin 12 päivää ja yli miljoona opetusaskelta. Esiopetetuista malleista saatavien sanojen vektoriesitysten, eli tässä viimeisestä Transformer -lohkosta saatavien vektoreiden, on voitu osoittaa ilmentävän erittäin hyvin sanojen semanttisia piirteitä ja sanojen välisiä suhteita (Mikolov et al., 2013).

Tässä tutkielmassa itsenäisenä mallina käytetyssä BERT-mallissa koontikerroksen perään liitettiin yhden yksikön ulostulokerros (ks. Liite A). GP-kerroksen implementoineissa malleissa viimeiseksi kerrokseksi jätettiin koontikerros, miltä [CLS] -saneelle johdetut piirteet $\mathbf{h}_{[CLS]}$ syötettiin edelleen GP-kerrokselle. Molemmissa GP-kerroksen implementoineissa malleissa enkooderin koontikerroksessa käytettiin myös *spektraalinormalisointia* (Miyato et al., 2018), kuten Liu et al. (2020). Spektraalinormalisoinnin tarkoituksena on rajoittaa mallia siten, että malli olisi sekä herkkä että tasainen (van Amersfoort et al., 2021b). Herkkyydellä tarkoitetaan sitä, että muutoksen syötteessä tulisi ilmetä myös muutoksena mallin kuvaamissa piirteissä

ja tasaisuudella sitä, ettei piirteissä ilmenevän muutoksen tulisi olla liian suuri suhteessa muutokseen syötteessä. Spektraalinormalisointi toteutetaan normalisoimalla painokertoimet $\mathbf{W}$ siten, että

$$\mathbf{W} = \begin{cases} c\mathbf{W}/\hat{\lambda}, & \hat{\lambda} > c \\ \mathbf{W}, & \text{muutoin.} \end{cases}$$

Edellä $\hat{\lambda}$ on approksimaatio matriisin $\mathbf{W}$ *singulaariarvolle* $\lambda = \max(\sqrt{\lambda_j})$, missä $\lambda_j, j = 1, \ldots, N_\lambda$ ovat matriisin $\mathbf{W}^T\mathbf{W}$ ominaisarvot ja c on vakio, jolle tässä työssä asetettiin arvo 0.95, kuten Liu et al. (2020).

3.1.2 Satunnaisten Fourier -piirteiden Laplace -approksimaatio

Menetelmä, jota tässä tutkielmassa kutsutaan satunnaisten Fourier -piirteiden Laplace -approksimaatioksi, on ensin esitelty julkaisussa Liu et al. (2020) ja myöhemmin osin laajennettuna julkaisussa Liu et al. (2022). Menetelmä on tässä kuvattu, kuten ensimmäisessä julkaisussa. Menetelmässä latentille funktiolle f oletetaan priori $f \sim N(\mathbf{0}, \mathbf{K})$ ja Gaussista prosessia approksimoidaan neuroverkon kerroksena määrättävillä satunnaisilla Fourier -piirteillä (Luku 2.4.1)

$$\begin{aligned} \boldsymbol{\phi}(\mathbf{x}) &= \sqrt{2/M}\cos(\mathbf{W}\mathbf{h} + \mathbf{b}) \\ &= \sqrt{2/M}\left(\cos(\mathbf{w}_1^T\mathbf{h} + b_1), \cdots, \cos(\mathbf{w}_M^T\mathbf{h} + b_M)\right)^T, \end{aligned}$$

missä M on satunnaisten piirteiden lukumäärä, $\mathbf{W} = \mathbf{W}^{(GP)}$ on GP-kerrosta vastaava $M \times P$ kerroinmatriisi, minkä alkiot w_{ji} poimitaan otoksina jakaumasta $N(0, 1)$ ja vakiotermit b_j otoksina jakaumasta $U(0, 2\pi)$. Sovitettaessa mallia käsitellään GP-kerroksessa parametrisessa muodossa (Luku 2.1), jolloin muunnoksen alla a priori

$$\begin{aligned} f(\mathbf{x}) &= \boldsymbol{\beta}^T\boldsymbol{\phi}(\mathbf{x}) \\ \boldsymbol{\beta} &\sim N(\mathbf{0}, \mathbf{I}), \end{aligned}$$

eli Gaussisen prosessin mallissa $f \sim N(\mathbf{0}, \boldsymbol{\Phi}\boldsymbol{\Phi}^T)$. Edellinen voidaan ajatella kahdesta kerroksesta muodostettuna neuroverkon lohkona, missä ensimmäisessä tiheässä kerroksessa yksiköiden lukumäärä on M ja aktivointifunktiona $\varphi_1(z) = \sqrt{2/M}\cos(z)$ ja jälkimmäinen tiheä kerros käsittää ainoastaan yhden yksikön, missä aktivointifunktiona on identiteettifunktio $\varphi_2(z) = z$ tai suoraan estimaatiksi $\pi = g^{-1}(f)$ muunnettuna, sigmoid -funktio. Edellä z on, kuten luvussa 1.1 (Kuva 1.1).

Posterioriennustejakauma $p(f^* \mid \mathbf{y})$ voidaan kirjoittaa parametrisessa muodossa

$$p(f^* \mid \mathbf{y}) = \int p(f^* \mid \boldsymbol{\beta})\, p(\boldsymbol{\beta} \mid \mathbf{y})\, d\boldsymbol{\beta},$$

missä posterijakauman $p(\boldsymbol{\beta} \,|\, \mathbf{y})$ tilalla käytetään tämän Laplace -approksimaatiota $q(\boldsymbol{\beta} \,|\, \mathbf{y})$ (Luku 2.3.1). Tällöin $\boldsymbol{\beta} \,|\, \mathbf{y} \sim N(\hat{\boldsymbol{\beta}}, \boldsymbol{\Sigma}_{\hat{\beta}})$, missä $\hat{\boldsymbol{\beta}} = \operatorname{argmax}_{\beta} p(\boldsymbol{\beta} \,|\, \mathbf{y})$ ja $\boldsymbol{\Sigma}_{\beta} = -\nabla^2 \log p(\hat{\boldsymbol{\beta}} \,|\, \mathbf{y})$.

Posteriorimoodi $\hat{\boldsymbol{\beta}}$ ja neuroverkko-osan parametrit $\boldsymbol{\theta}$ estimoidaan minimoimalla stokastisella gradienttimenetelmällä normalisoimatonta logaritmista posteriorijakaumaa

$$\log p(\boldsymbol{\beta} \,|\, \mathbf{y}) \propto \log p(\mathbf{y} \,|\, \boldsymbol{\beta}) + \frac{1}{2} \boldsymbol{\beta}^T \boldsymbol{\beta}.$$

Estimaatti Laplace -approksimaatiota vastaavalle kovarianssimatriisille $\hat{\boldsymbol{\Sigma}}_{\beta}$ saadaan tämän käänteismatriisilla, jolle pätee

$$\hat{\boldsymbol{\Sigma}}_{\beta}^{-1} = \mathbf{I} + \sum_i \hat{\pi}_i (1 - \hat{\pi}_i) \, \boldsymbol{\phi}(\mathbf{x}_i) \boldsymbol{\phi}(\mathbf{x}_i)^T,$$

missä $\hat{\pi}_i = g^{-1}(\hat{\boldsymbol{\beta}}^T \mathbf{x}_i)$. Kovarianssimatriisin päivitys voidaan siten tehdä osajoukko kerrallaan sovituksen viimeisellä epookilla. Posterioriennustejakaumaksi uudelle havainnolle Gaussisen prosessin mallissa saadaan

$$f^* \,|\, \mathbf{y} \sim N(\hat{\boldsymbol{\beta}}^T \boldsymbol{\phi}(\mathbf{x}^*), \boldsymbol{\phi}(\mathbf{x}^*)^T \boldsymbol{\Sigma}_{\hat{\beta}} \boldsymbol{\phi}(\mathbf{x}^*)).$$

Luokittelu tehdään, kuten luvussa 2.2. Odotusarvon $\mathbb{E}\left[\pi^* \,|\, \mathbf{y}\right]$ tilalla voidaan käyttää tämän Monte Carlo -estimaattia tai sigmoid -funktion probit -approksimaatioon perustuvaa approksimaatiota

$$\int g^{-1}(f^*) \, q(f^* \,|\, \mathbf{y}) df^* \approx g^{-1}\left(\frac{\mu_{f^*}}{\sqrt{1 + \lambda s_{f^*}^2}} \right), \tag{19}$$

missä $\mu_{f^*} = \hat{\boldsymbol{\beta}}^T \boldsymbol{\phi}(\mathbf{x}^*)$ eli mallin ennuste latentille f uudelle havainnolle $\mathbf{x}^*$, $s_{f^*}^2$ posterioriennustejakauman varianssi ja tässä tutkielmassa $\lambda = \pi/8$, kuten alunperin Bishop (2006) esitetty.

3.1.3 Variationaalisesti approksimoitavat indusoivat muuttujat

Indusoivia muuttujia soveltanut GP-kerroksen implementaatio jäljittelee van Amersfoort et al. (2021b) esittelemää menetelmää, mikä perustuu pitkälle luvussa 2.4.2 esiteltyyn indusoivien pisteiden ja muuttujien variationaaliseen approksimaatioon. Menetelmässä latentille funktiolle f oletetaan priori $f \sim \mathcal{GP}(m(\mathbf{h}), k(\mathbf{h}, \mathbf{h}'))$, missä nyt $m(\mathbf{h}) = \mu_0$ on estimoitava hyperparametri.

Indusoiville pisteille $\mathbf{z}_j$ oletetaan, että $\mathbf{z}_j = f^{(L-1)}(\mathbf{x}_j)$ jollekin $x_j \in \mathbb{R}^P$, toisin sanoen pisteiden sijainnit estimoidaan johdettujen piirteiden $\mathbf{h}$ määräämästä avaruudesta $\mathbb{R}^{D_h}$. Indusoivien pisteiden lukumääräksi j tämän tutkielman implementaatiossa

asetettiin 10, mitä myös alkuperäisessä julkaisussa van Amersfoort et al. (2021b) oli käytetty.

Indusoivien pisteiden sijainnit ja kovarianssifunktion skaalaparametri ℓ (Luku 2.1, Määritelmä 2.3) alustetaan poimimalla ensin opetusdatasta $n = 1000$ otos $\mathcal{S} \subset \mathcal{D}$ ja määräämällä otokselle johdetut piirteet $\mathbf{h}_{s,i} = f^{(L-1)}(\mathbf{x}_i), \mathbf{x}_i \in \mathcal{S}$. Sijainnit alustetaan lähinaapurimenetelmällä johdetuille piirteille $\mathbf{h}_{s,i}$ saaduilla klustereiden keskiarvovektoreilla ja skaalaparametri johdettujen piirteiden parittaisten etäisyyksien keskiarvolla (van Amersfoort et al., 2021a). Variationaalisen posteriorin $q(\mathbf{u})$ (Luku 2.4.2) odotusarvovektori $\mathbf{m}$ ja kovarianssimatriisi $\mathbf{S}$ puolestaan alustetaan arvoihin $\mathbf{m} = \mathbf{0}$ ja $\mathbf{S} = \mathbf{I}$. Variationaaliset parametrit, kovarianssifunktion hyperparametrit ja neuroverkon parametrit estimoidaan maksimoimalla luvussa 2.4.2 marginaaliselle uskottavuudelle johdettua alarajaa.

Ennusteina käytetään odotusarvon $\mathbb{E}[\pi^* \mid \mathbf{y}]$ Monte Carlo -approksimaatiota. Ennustettaessa havainnot oletetaan riippumattomaksi, eli otanta posterioriennustejakaumasta tehdään käyttäen latentille f approksimoidusta kovarianssimatriisista vain diagonaalin variansseja.

3.2 Menetelmien arvioinnista

Kuten johdannossa todettiin, neuroverkkojen robustisuutta ja epävarmuuden kvantifioinnin luotettavuutta arvioitaessa ollaan kiinnostuneita siitä, onko malli yleistettävissä myös opetusaineistosta opitun jakauman ulkopuolelle sekä ilmentääkö malli luotettavasti epävarmuutta tehdessään luokitteluvirheen tai kohdatessaan poikkeavan havainnon. Esiopetettujen Transformer -arkkitehtuuria käyttävien mallien on todettu parantavan robustisuutta sekä luonnollisen kielen käsittelyyn liittyvissä tehtävissä (Hendrycks et al., 2020) että kuvantunnistukseen liittyvissä tehtävissä (Bai et al., 2021), joihin Transformer -arkkitehtuuria myös on sovellettu (*ViT, Vision Transformers*). Robustisuutta on osin selitetty mallien esiopettamisella. On myös saatu viitteitä siitä, että esiopetuksessa käytetyn aineiston määrällä voi olla merkitystä robustisuuden kannalta (Hendrycks et al., 2020). Edelleen, ViT -malleja tutkittaessa on saatu viitteitä siitä, että ajurina paremmalle robustisuudelle voi olla myös nimenomaan Transformer -arkkitehtuuri ja tässä rinnakkain sovellettu attention -funktio (Bai et al., 2021).

Tässä tutkielmassa vertaillut menetelmät jakavat saman Transformer -arkkitehtuurin ja ne alustetaan samoista esiopetetuista kertoimista. Arvioinnissa erityisen kiinnostuksen kohteena on se, voidaanko Gaussista prosessia soveltamalla parantaa mallin robustisuutta tai epävarmuuden kvantifioinnin luotettavuutta. Edelleen arvioitavaksi on otettu varianssin soveltuvuus epävarmuutta kuvaavaksi mitaksi, kuten luvussa 2 perusteltiin ja luvussa 4 tarkemmin kuvataan. Varianssia on aikaisemmin käytetty epävarmuutta kuvaavana mittana pääasiassa jatkuvalle vasteelle (ks. esim. van Amersfoort et al., 2021b.

Tutkimusasetelmat, joissa vertailut toteutettiin, on kuvattu yksityiskohtaisemmin tulosten yhteydessä luvussa 4. Tutkimuskysymykset johdettiin julkaisuista Hendrycks et al. (2020) sekä Hendrycks ja Gimpel (2016) ja muotoiltiin täsmällisemmin seuraavasti:

(1) Kuinka paljon mallin yleistettävyys kärsii, kun ennustetaan opetusaineistosta poikkeavia havaintoja?

(2) Voidaanko mallilla ennustaa, milloin tämä tekee virheen?

(3) Voidaanko mallilla ennustaa, milloin tämä kohtaa poikkeavan havainnon?

Yleistettävyys liittyy mallin kykyyn ennustaa oikea luokka havainnoille, joita malli ei ole sovitettaessa kohdannut (Hastie et al., 2009). Poikkeavilla havainnoilla (*OOD*, *Out-of-Domain* tai *Out-of-Distribution*) tarkoitetaan tässä erityisesti ennustettassa kohdattavia havaintoja, joita ei ole kohdattu sovitettaessa ja jotka lisäksi poikkeavat sovitettaessa kohdatusta opetusjakaumasta. Opetusjakaumalla tarkoitetaan tässä tutkielmassa sitä jakaumaa, jota kerätyistä havainnoista eli opetusaineistosta saatavalla empiirisellä jakaumalla saadaan kuvattua. Hyvän mallin tulisi yleistyä opetusjakaumasta poikkeaville havainnoille, silloin kun mahdollista ja toisaalta tunnistaa poikkeavat havainnot, silloin kun nämä eivät kuulu mihinkään tunnettuun luokkaan (Hendrycks et al., 2020).

Edelliseen liittyy keskeisesti jakauman siirtymän (*distribution shift*) sekä erityisesti jakauman taustan siirtymän (*background shift*) ja semanttisen siirtymän (*semantic shift*) käsitteet. Jakauman siirtymällä tarkoitetaan tilannetta, missä esimerkiksi reaalimaailman ilmiöiden muutoksista johtuen sovitettaessa opittu ennustavien muuttujien $\mathbf{x}_i$ ja vastemuuttujien y_i välinen suhde ei välttämättä enää päde (Quinonero-Candela et al., 2008). Kyse on siten em. suhteesta mallia sovitettaessa implisiittisesti tehdyistä oletuksista, joita Draper (1987) kutsuu *skenaarioksi* ja liittää mallin rakennetta koskevaksi epävarmuudeksi. Tunnistamatta jäädessä jakauman siirtymällä voi olla suuria vaikutuksia mallin suorituskykyyn käytäntöön sovellettaessa (Paleyes et al., 2022)

Jaotellessa jakauman siirtymä edelleen taustan siirtymäksi ja semanttiseksi siirtymäksi, piirrevektorin oletetaan muodostuvan kahdesta komponentista (Ren et al., 2019). Taustamallin generoima taustakomponentti $\mathbf{x}_B$ kuvaa yleisiä populaatiotason piirteitä ja semanttisen mallin generoima semanttinen komponentti $\mathbf{x}_S$ opetusjakaumalle spesifejä piirteitä. Esimerkiksi tekstin voidaan ajatella muodostuvan vähämerkityksisistä sanoista, kuten partikkeleista sekä semanttisen merkityksen tuovista sanoista (Ren et al., 2019). Taustakomponentti oletetaan riippumattomaksi vastemuuttujasta, mutta semanttiselle komponentille sen sijaan $p(\mathbf{x}_S \mid y) \neq p(\mathbf{x}_S)$ (Arora et al., 2021). Edellinen voidaan karakterisoida myös siten, että jos $p(\mathbf{x}, y)$ on opetusjakauma, taustalta siirtynyt jakauma on jakauma $p_t(\mathbf{x}, y)$, kun taas semanttisesti siirtynyt jakauma on jakauma $p_S(\mathbf{x}, \overline{y})$, missä nyt $y \cap \overline{y} = \emptyset$ (Hsu et al., 2020).

Luokittelun kannalta relevanttien piirteiden oletetaan kuvautuvan semanttiseen komponenttiin, jolloin taustakomponentin siirtymästä huolimatta luokittelu on mahdollista tehdä, ja robusti malli kykenee yleistymään opetusjakaumasta poikkeaville havainnoille. Koska semanttisen komponentin siirtymistä kuvaavassa jakaumassa sen sijaan $y \cap \overline{y} = \emptyset$, hyvän mallin tulisi tässä joko kyetä pidättäytymään ennustamiseta tai ilmaista epävarmuutta siten, että poikkeava havainto kyetään tällä perusteella tunnistamaan. Neuroverkkojen on kuitenkin todettu kuvaavan suurta varmuut-

ta myös esimerkiksi täysin tunnistamattomille kuville (Nguyen et al., 2015). Edellä poikkeavien havaintojen tunnistamisesta todettu pätee myös taustakomponentin siirtymään, mitä voidaan perustella ainakin reaalimaailman muutosten ja trendien tunnistamisen näkökulmalla.

Jakauman siirtymää, sekä taustakomponentin että semanttisen komponentin osalta, voidaan simuloida ennustamalla mallilla kokonaan eri aineistoa kuin, mistä opetus- ja testiaineistot on eroteltu (Hendrycks et al., 2020; Ovadia et al., 2019). Menettelyä hyödynnettiin myös tässä tutkielmassa aineistoilla, jotka on tarkemmin kuvattu seuraavassa luvussa.

3.3 Aineistot

Opetusjakaumaa kuvaamaan käytettiin ScandiSent -tekstiaineiston (Isbister et al., 2021) suomenkielistä osa-aineistoa. ScandiSent -tekstiaineisto sisältää Trustpilot.com -palvelusta koottuja positiivisiksi ja negatiivisiksi luokiteltuja kuluttaja-arvioita (Taulukko 2). Palvelusta kootut arviot, mitkä annettaessa on pisteytetty asteikolla 1-5, on luokiteltu positiivisiksi (luokka 1) ja negatiivisiksi (luokka 0) siten, että positiivisiksi on luokiteltu 4 tai 5 pistettä saaneet arviot ja negatiivisiksi 1 tai 2 pistettä saaneet arviot. Kolme pistettä saaneet arviot on jätetty aineistosta pois. Suomenkielinen osa-aineisto käsitti 10000 arviota tasaisesti molemmista luokista. Saneistettujan tekstien (Luku 3.1.1) pituudet vaihtelivat välillä 1-731 siten, että keskipituus oli 32.4 sanetta ja syötteiden enimmäispituudeksi asetetulla 128:lla saneella 96.7 % teksteistä saatiin esitettyä kokonaan.

Aineisto jaettiin opetus-, validointi- ja testiaineistoiksi siten, että testiaineistoksi erotettiin 2000 havaintoa ja jäljelle jääneet 8000 havaintoa jaettiin opetus- ja validointiaineistoiksi suhteessa 80 : 20. Opetusaineiston kooksi tuli näin 6400 havaintoa ja validointiaineiston kooksi 1600 havaintoa. Validointiaineistoa käytettiin hyperparametreille (Liite A) käytettäviä arvoja valittaessa.

Taulukko 2: Sentimenttiannotoituja esimerkkiarvosteluita ScandiSent -aineistosta. Luokka 1 kuvaa positiivista sentimenttiä ja luokka 0 negatiivista sentimenttiä.

Arvostelu	**Luokka**
Tilasin tuotteen uskoen sivulla luvattuun nopeaan toimitukseen. Kas, sitten tulikin viesti, että toimitus viivästyy usealla viikolla. Jos olisin tämän tiennyt, olisin hankkinut tuotteen muualta!	0
Olen taysin tyytyvainen kaikkeen,tosi nopeaa ja sujuvaa kaupantekoa.	1
Loisto asiakaspalvelu, hyviä vinkkejä yrittäjyyteen, Helppo käyttää! Tykkään!	1
Aika kauan saa odotella tilausta. Kyselin tilauksen perään niin automaattiviesti vastasi vain että "kiitos yhteydenotosta".	0
Hyvät tuotteet, hyvä hinta-laatusuhde, ystävällinen asiakaspalvelu, toimitukset tulevat aina oikein, toimituksen nopeudessa vähän parantamisen varaa. (Jos ei käytetä Postin palveluita, saattaisi toimia paremmin.)	1

Semanttisen komponentin siirtymää kuvattiin Finlex -tekstiaineistosta (Tiedemann, 2012) muodostetulla aineistolla. Finlex -tekstiaineisto sisältää suomen- ja ruotsinkielisiä säädöstekstejä Finlex -säädöskokoelmasta. Ilmaisultaan neutraalien säädösteks-

tien ajateltiin ilmentävän vain vähän tekstin sentimenttiä määritettäessä hyödyllisiä semanttisia piirteitä ja sopivan siten kuvaamaan jakauman semanttista siirtymää.

Tutkielmassa käytetty aineisto muodostettiin suomenkielisistä säädösteksteistä (Taulukko 3), mitkä rajattiin edelleen annetuiksi välillä 2010-2018. Lopullinen aineisto poimittiin $n = 2000$ otoksena edellä mainitusti rajatusta osa-aineistosta. Lopullisen aineiston saneistettujen tekstien keskipituus oli 28.0 ja 99.7 % kaikista teksteistä saatiin esitettyä kokonaan 128:lla saneella.

Taulukko 3: Esimerkkitekstejä Finlex -säädöskokoelmasta muodostetusta tekstiaineistosta.

Teksti
Postiyrityksellä on oikeus säilyttää kysymyksessä olevalle postin saajalle tulevat kirjelähetykset noudettavina postin saajan osoitteen mukaan määräytyvässä toimipaikassa tai muussa vastaavassa noutopisteessä:
Tukihakemuksen hylkäämisestä valvonnan estyessä säädetään soveltamisasetuksen 26 artiklan 2 kohdassa.
Koulutusta järjestettäessä tulee olla yhteistyössä alle 18-vuotiaiden opiskelijoiden huoltajien kanssa.
Turvamiehen on tunnettava kantamiensa voimankäyttövälineiden vaikutukset ja niiden käyttöön liittyvät säännökset sekä osattava käyttää niitä asianmukaisesti.
Oikaisuvaatimukseen annettuun päätökseen saa hakea muutosta valittamalla hallintooikeuteen siten kuin hallintolainkäyttölaissa säädetään.

Taustan siirtymää kuvattiin osa-aineistolla FinnSentiment -tekstiaineistosta (Lindén et al., 2023), mikä sisältää kirjoituksia Suomi24 -keskustelupalstalta ($n = 27000$). Tekstit on ensin kolme ihmistä toisistaan riippumattomasti luokitellut positiivisiksi, negatiivisiksi tai neutraaleiksi. Koska kyse oli sentimenttiannotoidusta aineistosta, oletettiin teksien sisältävän samankaltaisia semanttisia piirteitä, kuin opetusaineistona käytetyn ScandiSent -aineiston, mutta ainakin jossain määrin erilaisissa asiayhteyksissä. Koska tässä vastemuuttuja vastaa opetusaineiston vastemuuttujaa, voidaan aineistoa pyrkiä luokittelemaan, kuten testiaineistoa.

Tekstit oli luokiteltu viiteen luokkaan pisteyttämällä ihmisten teksteille arvioimat luokat siten, että positiivinen arvio sai pisteluvun 1, negatiivinen pisteluvun -1 ja neutraali pisteluvun 0 ja määräämällä luokat yhteispisteiden perusteella. Yhteispisteitä -3 vastasi luokka 1, pisteitä -2 ja -1 luokka 2, pisteitä 0 luokka 0, pisteitä 1 ja 2 luokka 4 ja yhteispisteitä 3 luokka 5.

Taustan siirtymää kuvaavassa aineistossa palattiin ScandiSent -aineistoa vastaten kahteen luokkaan luokittelemalla luokat 1 ja 2 negatiiviseksi, luokat 4 ja 5 positiivisiksi ja jättämällä luokka 3 aineiston ulkopuolelle (Taulukko 4). Lopullinen aineisto poimittiin $n = 2000$ otoksena tasaisesti molemmista luokista. Saneistettujen tekstien keskipituudeksi tuli 31.3 sanetta ja 99.5 % teksteistä saatiin katettua kokonaan 128:lla saneella.

Taulukko 4: Sentimenttiannotoituja esimerkkitekstejä FinnSentiment -aineistosta. Luokka 1 kuvaa positiivista sentimenttiä ja luokka 0 negatiivista sentimenttiä.

Teksti	**Luokka**
Googlasin myyjän nimellä ja löysin tämän keskustelun, joka oli hyödyllinen taustoittamaan myyjän toimintaa.	1
En ymmärrä miksi cheerleading ei muka olisi urheilulaji!?	0
Eipä vaivuta synkkyyteen, mielestäni nämä palstat tällaisenaan tuottavat monille piristystä päivään, sitä varmaankaan monet eivät tiedä.	1
Jokaisella poliitikolla on oltava oikeus tehdä omat virheensä.....veronmaksajien rahoilla.	0
Ei kannata luottaa, etenkin kun kertomansa mukaan aloittaja on jo joutunut työpaikalla jonkinlaisen eristämisen ja negatiivisen kohtelun kohteeksi.	0

4 Tulokset

Luvussa 3.2 esitettyjä tutkimuskysymyksiä tarkasteltiin käyttäen kolmea erilaista asetelmaa. Asetelmat on kuvattu ja tulokset raportoitu seuraavassa tutkimuskysymyksittäin. Ellei toisin ole mainittu, tulokset on raportoitu kymmenen sovituksen antamien mallien keskiarvoina ja keskivirheinä. Useamman sovituksen käyttäminen on kirjallisuudessa verrattain yleistä ja perusteltavissa satunnaisuudella, mikä liittyy ei-konveksin funktion optimointiin stokastisella menetelmällä (Luvut 1.1 ja 1.3). Raportoidut suureet on kuvattu luvussa 4.1.

Epävarmuuden kvantifiointia koskevissa asetelmissa (Luvut 4.3 ja 4.4) ennusteen epävarmuutta kuvaavina mittoina käytettiin suurinta ennustettua luokkatodennäköisyyttä $\pi_{max} = \max(\pi, 1-\pi)$, minkä skaala käännettiin muunnoksella $1-\pi_{max}$, jolloin suuremmat arvot saatiin kuvaamaan suurempaa epävarmuutta. Suurinta luokkatodennäköisyyttä on ehdotettu perustasoa kuvaavaksi mitaksi esimerkiksi poikkeavien havaintojen tunnistamista koskevissa tarkasteluissa (Hendrycks & Gimpel, 2016). Gaussisen prosessin implementoineissa malleissa epävarmuutta kuvattiin myös ennusteelle π estimoidulla posteriorivarianssilla. Variansseja estimoitiin poimimalla $n = 1000$ otos latentin f posterioriennustejakaumasta ja määräämällä tästä otosvarianssi muunnokselle $\pi = g^{-1}(f)$.

4.1 Raportoidut suureet

Seuraavassa on kuvattu tässä luvussa raportoiduista suureista ne, joita ei ole oletettu entuudestaan tunnetuiksi. Tällä rajauksella erikseen käsittelemättä on jätetty luokittelutarkkuus ja negatiivinen logaritminen uskottavuus. Koska molemmat ensin esiteltävät AUC -suureet (*Area Under Curve*) AUROC (*Area Under Receiver Operating Characteristic Curve*) ja AUPR (*Area Under Precision-Recall Curve*) sekä FPR95 (*False Positive Rate at 0.95 Recall*) ovat johdettavissa dikotomisen vasteen sekaannusmatriisista, esitellään tässä ensin sekaannusmatriisista johdettavat suureet ja näistä käytetyt merkinnät:

$$\text{TPR}\,(\textit{true positive rate}) = \frac{\text{TP}}{\text{TP} + \text{FN}},$$
$$\text{FPR}\,(\textit{false positive rate}) = \frac{\text{FP}}{\text{FP} + \text{TN}},$$
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}.$$

Edellä TP (*true positive*) on positiivisiksi (luokka 1) luokiteltujen positiivisten, FP (*false positive*) positiivisiksi luokiteltujen negatiivisten (luokka 0), TN (*true negative*) negatiivisiksi luokiteltujen negatiivisten ja FN (*false negative*) negatiivisiksi luokitelujen positiivisten lukumäärät.

TPR (myös *Recall* ja *sensitiivisyys*) kuvaa todennäköisyyttä, että malli tunnistaa todellisuudessa positiivisen tapauksen (Zhu et al., 2010) ja FPR vastaavasti tulkiten todennäköisyyttä, että negatiivinen tapaus luokitellaan väärin positiiviseksi. Precision puolestaan voidaan tulkita todennäköisyytenä, että positiiviseksi luokiteltu havainto on positiivinen. Täydellisessä luokittelussa TPR = 1, FPR = 0 ja Precision = 1.

4.1.1 AUC -suureet

AUROC-suure saadaan tätä vastaavan ROC -kuvaajan alle jäävänä pinta-alana. ROC-kuvaaja kuvaa luokittelusäännössä $y_i = \mathbb{I}(\pi_i > \gamma)$ käytetyn kynnysarvon γ koordinaatiston pisteeksi (FPR_γ, TPR_γ). Kynnysarvoa 0 vastaa piste (1,1) eli kaikki tapaukset luokitellaan positiiviseksi ja vastaavasti kynnysarvoa 1 piste (0,0), kun kaikki tapaukset luokitellaan negatiiviseksi. Mitä lähempänä pisteet muutoin ovat täydellistä luokittelua vastaavaa pistettä $(0, 1)$, sitä lähempänä arvoa 1 on kuvaajan alle jäävä pinta-ala. Satunnaista luokittelijaa vastaa diagonaalille asettuvat pisteet (Zhu et al., 2010) eli satunnaisella luokittelijalla AUROC = 0.5.

AUPR-suure saadaan vastaavasti PR -kuvaajan alle jäävänä pinta-alana. PR-kuvaaja kuvaa vastaavasti kuin ROC-kuvaaja luokittelusäännössä käytetyn kynnysarvon γ koordinaatiston pisteeksi $(Recall_\gamma, Precision_\gamma)$. X-akselin pisteessä 0 eli kynnysarvolla 1 Precision ei ole määritelty, koska tällöin kaikki tapaukset luokitellaan negatiiviseksi ja nimittäjässä $TP + FP = 0$. Käytäntö kuitenkin on, että kuvaaja aloitetaan pisteestä (0,1). Täydellistä luokittelua vastaa piste (1,1), mikä on saavutettavissa vain, jos kaikki luokiteltavat tapaukset ovat positiivisia. Muutoin AUPR-suureen arvo riippuu positiivisten (P) ja negatiivisten (N) osuuksista aineistossa, koska kynnysarvolla 0 eli X-akselin pisteessä 1, Precision saa arvon P/(P+N).

Perustasona pidettävälle satunnaiselle luokittelijalle AUPR = P/(P+N) (Saito & Rehmsmeier, 2015). Koska AUPR huomioi luokkien osuudet aineistossa, pidetään sitä epätasaisesti jakautuneessa aineostossa informatiivisempana, kuin AUROC -suuretta. Edelleen koska kuvaajat piirretään kaikille mallista saataville luokittelijoille eli kaikilla kynnysarvoilla γ, kuvaavat ne mallien yleistä, käytetystä kynnysarvosta riippumatonta suorituskykyä.

4.1.2 FPR95

FPR95-suure saadaan yksinkertaisesti TPR:n arvoa 0.95 vastaavana FPR:n arvona. FPR95 ilmoittaa näin negatiivisten osuuden positiivisiksi luokitelluista, kun 95% positiivisista on luokiteltu oikein.

4.1.3 OC-Tarkkuus ja arviointitehokkuus

OC-Tarkkuutta (*OC-Acc, Oracle Collaborative Accuracy*) ja arviointitehokkuutta (*Review Efficiency*) laskettaessa oletetaan, että rajattu määrä tapauksia tulee ohjatuksi oraakkelin, esimerkiksi ihmisen, arvioitavaksi ja luokiteltavaksi (Kivlichan et al., 2021). OC-Tarkkuus ilmoittaa luokittelutarkkuuden, kun oraakkelille ohjatut tapaukset oletetaan tulevan oikein luokitelluksi ja arviointitehokkuus sen osuuden, mikä oraakkelille ohjatuista tapauksista tosiasiassa oli väärin luokiteltu. OC-Tarkkuus lasketaan kaavalla

$$\text{OC-Acc}(\alpha) = \frac{1}{N}\sum_{i=1}^{N} \text{OC-Acc}(\mathbf{x}_i \mid \alpha),$$

$$\text{OC-Acc}(\mathbf{x}_i \mid \alpha) = \begin{cases} 1, & u(\mathbf{x}_i) > q_{1-\alpha} \\ \mathbb{I}(f(\mathbf{x}_i) = y_i), & \text{muutoin.} \end{cases}$$

missä u on mitta, minkä perusteella määrätään ihmisluokittelijalle osoitettavat tapaukset, α oraakkelille ohjattavaksi sallittavien tapausten osuus ja $q_{1-\alpha}$ tätä vastaava yläkvantiili. Arviointitehokkuus (AT) puolestaan saadaan kaavalla

$$AT(\alpha) = \frac{TP_\alpha}{TP_\alpha + FP_\alpha},$$

missä taustalla olevassa sekaannusmatriisissa positiiviseksi luokaksi nyt on oletettu luokitteluvirhettä kuvaava luokka ja luokittelussa käytetään epävarmuutta kuvaavaa mittaa siten, että tapaukset i, joille $u(\mathbf{x}_i) > q_{1-\alpha}$ luokitellaan positiivisiksi eli virheellisesti luokitelluiksi.

4.2 Yleistettävyys ennustettaessa poikkeavia havaintoja

Ensimmäistä tutkimuskysymystä tarkasteltiin asetelmalla, missä opetusaineistolla sovitetuilla malleilla luokiteltiin sekä testiaineisto että taustakomponentin siirtymää simuloinut FinnSentiment -aineiston osa-aineisto. Molemmista luokitteluista määrättiin talukossa 5 raportoidut suureet. Malleihin viitataan jatkossa taulukoissa ja kuvissa lyhenteillä siten, että BERT tarkoittaa ilman Gaussista prosessia impementoitua mallia, RFFL satunnaisten Fourier -piirteiden Laplace -approksimaatiota käyttänyttä mallia ja IPVI indusoivien muuttujien variationaalista approksimaatiota käyttänyttä mallia.

Taulukko 5: Testi- ja FinnSentiment -aineiston luokittelua kuvaavat suureet. Vastemuuttujana tekstin positiivinen sentimentti.

Malli	Tarkkuus		-log L		AUROC		AUPR	
	Testi	FinnSent	Testi	FinnSent	Testi	FinnSent	Testi	FinnSent
BERT	0.958 (0.000)	0.855 (0.002)	0.121 (0.000)	**0.335 (0.002)**	**0.991 (0.000)**	**0.933 (0.001)**	**0.991 (0.000)**	**0.938 (0.001)**
RFFL	0.959 (0.001)	**0.857 (0.002)**	**0.118 (0.000)**	0.344 (0.003)	**0.991 (0.000)**	0.930 (0.001)	**0.991 (0.000)**	0.936 (0.001)
IPVI	**0.961 (0.001)**	0.852 (0.003)	**0.118 (0.000)**	0.347 (0.002)	0.990 (0.000)	0.929 (0.001)	**0.991 (0.000)**	0.936 (0.002)

Paras luokittelutarkkuus testiaineistolla saavutettiin indusoivien muuttujien variationaalista approksimaatiota käyttäneellä mallilla ja FinnSentiment -aineistolla satunnaisten Fourier -piirteiden Laplace -approksimaatiota käyttäneellä mallilla. Erot luokittelutarkkuudessa parhaimman ja huonoimman mallin välillä jäivät kuitenkin molemmilla aineistoilla verrattain vähäisiksi (≤ 0.005). Parhaan sovitteen negatiivisella log-uskottavuudella arvioituna FinnSentiment -aineistolle kuitenkin alhaisemmasta luokittelutarkkuudesta huolimatta antoi ilman Gaussista prosessia implementoitu malli.

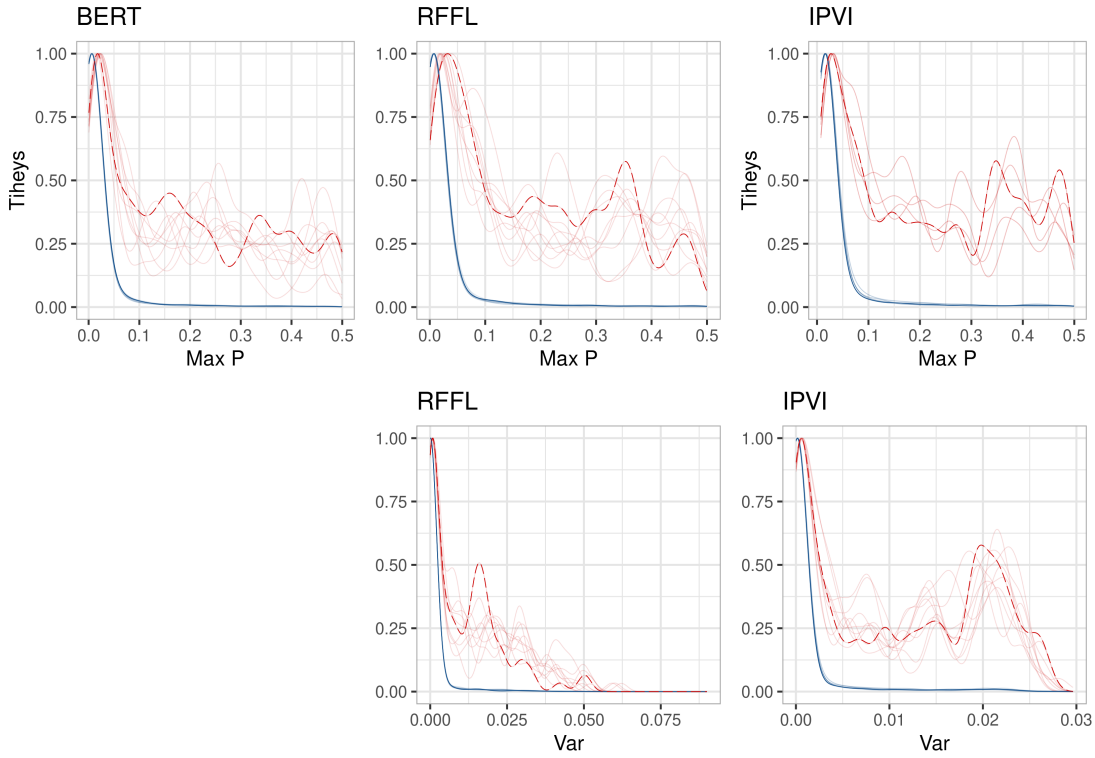
AUROC- ja AUPR-suureilla tarkasteltuna testiaineiston luokittelussa eroja ei käytännössä todettu. FinnSentiment -aineistossa marginaalisella erolla luokittelukykyisimmäksi malliksi osoittautui ilman Gaussista prosessia implementoitu malli.

4.3 Luokitteluvirheen ennustaminen

Luokitteluvirheen ennustamista tarkasteltiin, kuten edellä luokittelemalla opetusaineistolla sovitetuilla malleilla sekä testiaineisto että taustakomponentin siirtymää kuvannut FinnSentiment -aineisto. Ensimmäisessä vaiheessa arvioitiin mallien kykyä erotella oikein- ja väärinluokiteltuja tapauksia tarkastelemalla suurimman luokkatodennäköisyyden ja varianssin jakaumia oikein- ja väärinluokiteltujen ryhmissä (Kuvat 4.1 ja 4.2), kuten esim. Ovadia et al. (2019). Tässä epävarmuuden tulisi kuvautua suuremmaksi väärin luokiteltujen tapausten kohdalla ja edelleen mitä paremmin jakaumat erottuvat, sitä paremmin mallilla tai mitalla lähtökohtaisesti kyetään tunnistamaan väärin luokiteltuja tapauksia esimerkiksi ihmisluokittelijan arvioitavaksi (Ovadia et al., 2019).

Testiaineiston kuvaajista (Kuva 4.1) voidaan ensinnäkin todeta, että oikein luokiteltujen ryhmässä ennusteet on keskimäärin kuvattu selvästi varmemmiksi kuin väärin luokiteltujen ryhmässä. Samaan aikaan on kuitenkin tehty luokitteluvirheitä siten, että myös tässä ryhmässä jakaumien huiput sijoittuvat alakvantiileille, eivätkä jakaumat siten tältä osin käytännössä erotu.

Tarkasteltaessa väärin luokiteltujen ryhmää voidaan todeta, että kaikkien mallien kohdalla yleisesti virheelliseen ennusteeseen kuvautuu suurempi epävarmuus. Parhaiten huippua yläkvantiileille vaikuttaa saavan indusoivien pisteiden variationaalista approksimaatiota soveltanut malli erityisesti suurimmalla todennäköisyydellä. Sovitteissa on kuitenkin vaihtelua siten, että myös muilla malleilla voidaan todeta sovitteita, joilla huippua syntyy vielä aivan yläkvantiileille. Kuvaajista on myös syytä huomata, että koska kyse on ryhmittäin skaalatuista jakaumista, ei ryhmien välisistä osuuksista voida päätellä kuvaajien perusteella.

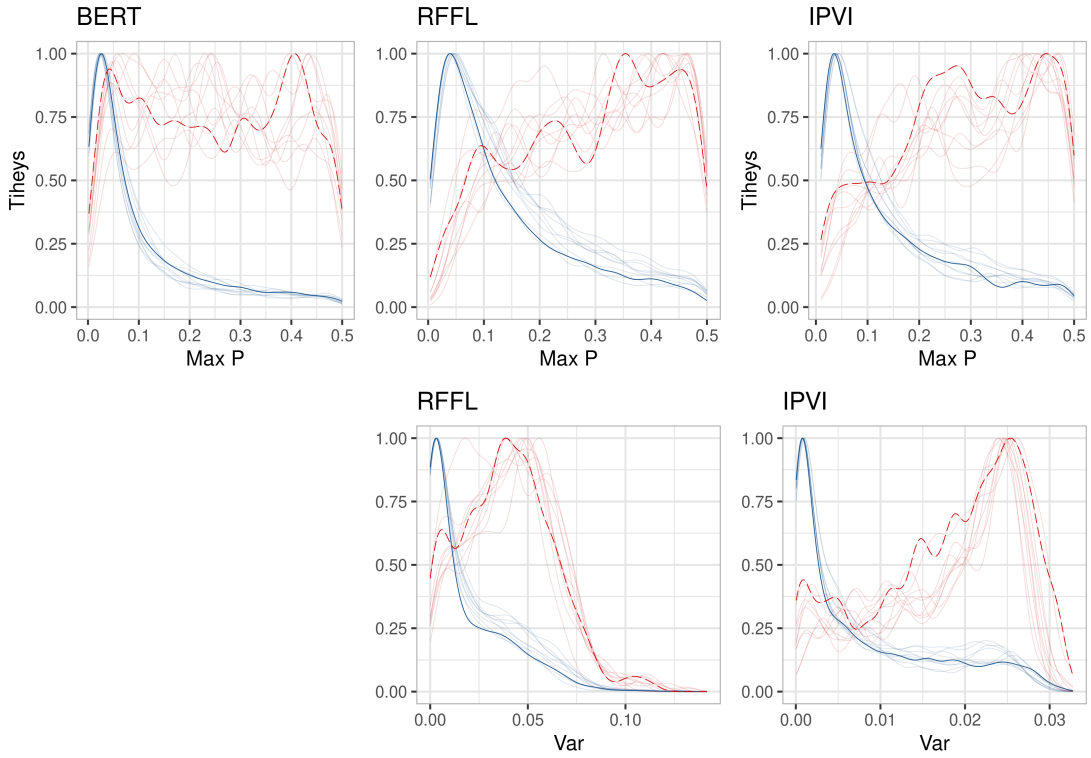


Kuva 4.1: Suurimman todennäköisyyden (ylh.) ja varianssin (alh.) skaalatut jakaumat testiaineistossa väärin- (pun.) ja oikeinluokiteltujen (sin.) ryhmissä kaikilla sovituksilla. Alhaisimman validointitappion antanut malli korostettuna.

Taustakomponentin siirtymää kuvanneella aineistolla (Kuva 4.2) suurimman todennäköisyyden jakaumat oikein ja väärin luokiteltujen ryhmissä vaikuttavat erottuvan Gaussista prosessia soveltaneilla malleilla paremmin, kuin ilman Gaussista prosessia määrätyllä mallilla. Parempi erottuvuus voidaan havaita lähinnä jakaumien alakvantiileilla, joilla Gaussisen prosessin soveltaneilla malleilla väärin luokiteltujen ryhmässä tiheys yleisesti vaikuttaa estimoituvan alhaisemmaksi. Erottuvuutta yläkvantiileilla tosin saattaa sotkea tässä oikein luokitelluille hiukan suuremmaksi estimoituva tiheys, kuin ilman Gaussista prosessia määrätyllä mallilla. Ilman Gaussista prosessia määrätyllä mallilla todennäköisyydet väärin luokiteltujen ryhmässä vaikuttavat jakautuneen tasaisemmin, kuin Gaussista prosessiä käyttäneillä malleilla ja myös tasaisemmin, kuin testiaineistolle määrättyinä.

Edelleen verrattaessa Gaussista prosessia soveltaneiden mallien jakaumia testiaineistolla saatuihin jakaumiin, ryhmien erottuvuus vaikuttaa paremmalta taustakomponentin siirtymää kuvanneella aineistolla, minkä luokittelu on luokittelutarkkuuksien perusteella ollut malleille selvästi testiaineiston luokittelua vaikeampaa. Yksittäisiä sovituksia tarkasteltaessa voidaan todeta jopa jakaumia, joissa väärinluokiteltujen ryhmän jakaumien tiheys estimoituu lähes nollaksi vaihteluvälin alarajalla. Silmämääräisesti tarkasteltuna paras erottuvuus keskimäärin vaikutetaan saavutettavan Gaussista prosessia käyttäneillä malleilla suurinta todennäköisyyttä käytettäessä. Selkein huippu yläkvantiileilla saadaan kuitenkin indusoivin pisteiden variationaalista approksimaatiota soveltaneella mallilla.

Taulukossa 6 raportoiduilla suureilla tarkasteluna yleisesti luokittelukykyisimmäksi



Kuva 4.2: Suurimman todennäköisyyden (ylh.) ja varianssin (alh.) skaalatut jakaumat taustan siirtymää kuvanneessa aineistossa väärin- (pun.) ja oikeinluokiteltujen (sin.) ryhmissä kaikilla sovituksilla. Alhaisimman validointitappion antanut malli korostettuna.

malliksi luokitteluvirhettä ennustettaessa osoittautuu johdonmukaisesti kaikilla suureilla ja molemmissa aineistoissa ilman Gaussista prosessia määrätty malli. Gaussisen prosessin soveltaneista malleista voidaan edelleen todeta suurimman todennäköisyyden vaikuttavan yleisesti ennustavan luokitteluvirhettä paremmin kuin varianssin erityisesti taustakomponentin siirtymää kuvanneessa aineistossa.

Taulukko 6: Luokitteluvirheen ennustamista koskevat suureet testiaineistossa ja taustakomponentin siirtymää kuvanneessa FinnSentiment -aineistossa, kun vastemuuttujana on luokitteluvirhe ja luokittelu tehdään suurimman todennäköisyyden (max tn) tai posteriorivarianssin (var) perusteella.

Malli	Mitta	Testiaineisto			FinnSentiment		
		AUROC	AUPR	FPR95	AUROC	AUPR	FPR95
BERT	max tn	**0.908 (0.002)**	**0.386 (0.011)**	**0.439 (0.017)**	**0.811 (0.003)**	**0.392 (0.008)**	**0.627 (0.012)**
RFFL	max tn	0.903 (0.001)	0.332 (0.014)	0.450 (0.019)	0.794 (0.002)	0.368 (0.007)	0.648 (0.010)
RFFL	var	0.902 (0.001)	0.334 (0.014)	0.481 (0.016)	0.758 (0.002)	0.297 (0.002)	0.683 (0.009)
IPVI	max tn	0.889 (0.003)	0.295 (0.009)	0.581 (0.037)	0.797 (0.004)	0.386 (0.006)	0.654 (0.014)
IPVI	var	0.887 (0.003)	0.304 (0.010)	0.602 (0.037)	0.791 (0.004)	0.356 (0.012)	0.657 (0.015)

Kuten luvussa 4.1 todettiin, AUROC ja AUPR -suureet kuvaavat mallin luokittelukynnyksestä riippumatonta luokittelukykyä, eikä näistä suoraan voida päätellä tiettyyn luokittelukynnykseen liittyvää luokittelukykyä. Skaalatuista jakaumista tehtäviä tulkintoja puolestaan haittaa huomattava epätasapaino oikein- ja väärinluokiteltujen tapausten lukumäärissä. Edellisten huomioiden johdosta asetelmaa täydennettiin tältä osin olettamalla lopuksi malli osaksi systeemiä, missä sen on

mahdollista tehdä yhteistyötä ns. *oraakkelin* kanssa eli ohjata osa havainnoista esimerkiksi ihmisluokittelijalle, jonka oletetaan varmuudella kykenevän luokittelemaan tälle ohjatut tapaukset oikein (Luku 4.1). Tässä oraakkelille ohjattaessa käytettävä luokittelukynnys määräytyy sen perusteella, millainen osuus α oraakkelille sallitaan ohjattavaksi. Käytännössä aineisto järjestetään epävarmuutta kuvaavan mitan perusteella laskevaan järjestykseen ja tarkastellaan oraakkelille ohjattavaksi sallittua osuutta vastaavaa yläkvantiilia.

Oraakkelin kanssa yhteistyötä tekevää systeemiä kuvaavat suureet, eli arviointitehokkuus ja OC-tarkkuus, on raportoitu taulukoissa 7 ja 8. Suureet on raportoitu alhaisimman validointitappion antaneilla sovituksilla vastaten kuvainnollisesti mallinvalinnan jälkeistä tilannetta.

Taulukko 7: Arviointitehokkuudet osuuksille $\alpha \in \{0.01, 0.05, 0.1, 0.2\}$ testiaineistolle ja taustakomponentin siirtymää kuvanneelle FinnSentiment -aineistolle.

Malli	Mitta	Testiaineisto				FinnSentiment			
		0.01	0.05	0.1	0.2	0.01	0.05	0.1	0.2
BERT	max tn	**0.500**	**0.440**	**0.305**	**0.175**	0.500	0.470	**0.445**	0.375
RFFL	max tn	0.350	0.390	0.275	0.168	0.650	**0.530**	0.435	**0.385**
RFFL	var	0.250	0.380	0.275	0.173	0.300	0.360	0.335	0.315
IPVI	max tn	0.450	0.360	0.265	0.163	**0.700**	0.440	0.430	0.375
IPVI	var	0.450	0.360	0.260	0.163	0.500	0.450	0.435	0.373

Testiaineistossa tehokkaimmin virheellisiä luokitteluita kykenee erottelemaan johdonmukaisesti ilman Gaussista prosessia määrätty malli. Taustakomponentin siirtymää kuvanneessa aineistossa vastaavaa johdonmukaisuutta ei voida todeta yhtä selvänä minkään mallin osalta. Kahden parhaan mallin joukkoon kaikilla osuuksilla tässä yltää satunnaisten Fourier -piirteiden Laplace -approksimaatiota ja suurinta todennäköisyyttä käyttänyt malli, mikä on joko tehokkain ($\alpha \in \{0.05, 0.2\}$) tai toiseksi tehokkain marginaalilla ≤ 0.05. Ylimmässä tarkastellussa kvantiilissa ($\alpha = 0.01$) parhaiten menestyvät Gaussisen prosessin soveltaneet mallit suurimmalla todennäköisyydellä, missä indusoivien muuttujien variationaalista approksimaatiota soveltaneen mallin oraakkelille ohjaamista tapauksista jopa 70% oli väärin luokiteltuja. Edellinen voidaan tulkita kuvautuvan myös vastaaviin jakaumiin (Kuva 4.2, ylh. kesk. ja oik.) siten, että molemmissa kuvissa väärin luokiteltujen jakaumassa voidaan havaita huipukkuutta vielä aivan yläkvantiileilla toisin kuin BERT-mallissa, missä tiheys alkaa laskea jo aikaisemmin.

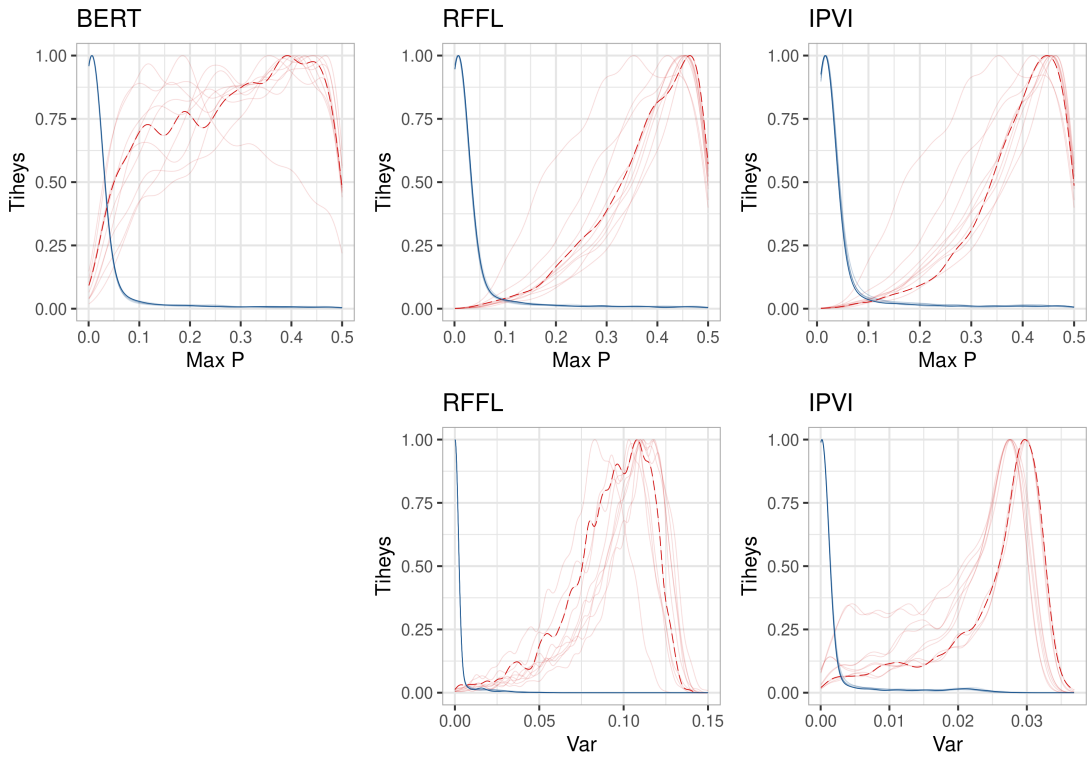
Taulukko 8: Luokittelutarkkuudet testiaineistolle ja taustakomponentin siirtymää kuvanneelle FinnSentiment -aineistolle, kun osuuksia $\alpha \in \{0.01, 0.05, 0.1, 0.2\}$ vastaavat määrät tapauksista voidaan ohjata oraakkelille, minkä oletetaan korjaavan luokittelut oikeiksi.

Malli	Mitta	Testiaineisto				FinnSentiment			
		0.01	0.05	0.1	0.2	0.01	0.05	0.1	0.2
BERT	max tn	0.963	0.980	0.988	0.993	0.864	0.882	0.903	0.934
RFFL	max tn	0.964	0.980	0.988	0.994	**0.868**	**0.888**	**0.905**	**0.938**
RFFL	var	0.963	0.980	0.988	**0.995**	0.864	0.879	0.895	0.924
IPVI	max tn	**0.966**	0.980	0.988	0.994	0.860	0.875	0.896	0.928
IPVI	var	**0.966**	0.980	0.988	0.994	0.858	0.876	0.897	0.928

Oraakkelin kanssa yhteistyöllä saatavissa luokittelutarkkuuksissa testiaineistossa pieniä eroja voidaan todeta vain osuuksilla 0.01 ja 0.2. Parhaat luokittelutarkkuudet ylimmässä kvantiilissa saavutetaan indusoivien muuttujien variationaalista approksimaatiota soveltaneella mallilla. Luokittelutarkkuuden mahdollinen parannus ylimmässä kvantiilissa on kaikilla malleilla sama 0.5 -prosenttiyksikköä. Taustakomponentin siirtymää kuvanneessa aineistossa johdonmukaisesti paras luokittelutarkkuus saavutetaan satunnaisten Fourier -piirteiden Laplace -approksimaatiota suurimmalla todennäköisyydellä soveltaneella mallilla.

4.4 Poikkeavien havaintojen ennustaminen

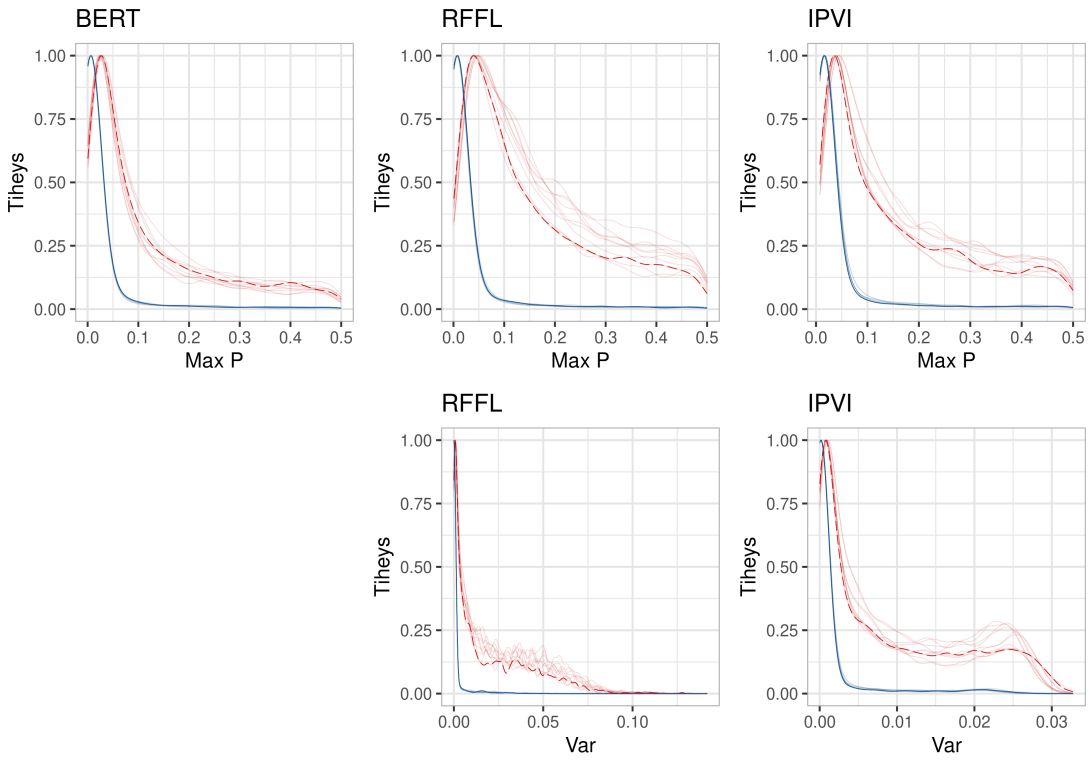
Poikkeavien havaintojen ennustamista tarkasteltiin käyttämällä sekä taustakomponentin siirtymää kuvannutta FinnSentiment -aineistoa että semanttisen komponentin siirtymää kuvannutta Finlex -aineistoa. Molemmat aineistot yhdistettiin testiaineiston kanssa ja luokiteltiin opetusaineistolla sovitetulla mallilla. Vastaavasti kuin edellä luokitteluvirhettä ennustettaessa ensimmäisessä vaiheessa tarkasteltiin suurimman todennäköisyyden ja varianssin jakaumia, missä nyt ryhmät muodostettiin aineistojen perusteella. Jakaumien vertailut testiaineistoon ja Finlex -aineistoon kuuluvien havaintojen ryhmissä on esitetty kuvassa 4.3 ja testiaineistoon ja FinnSentiment -aineistoon kuuluvien havaintojen ryhmissä kuvassa 4.4.



Kuva 4.3: Suurimman todennäköisyyden (ylh.) ja varianssin (alh.) skaalatut jakaumat testiaineistolle (sin.) ja semanttisen komponentin siirtymää kuvanneelle Finlex -aineistolle (pun.) kaikilla sovituksilla. Alhaisimman validointitappion antanut malli korostettuna.

Testiaineiston ja semanttisen komponentin siirtymää kuvaavan Finlex -aineiston jakaumien voitiin todeta erottuvan verrattain hyvin kaikkien mallien kohdalla. Gaus-

sista prosessia soveltaneilla malleilla päällekkäisyyttä ei juuri ole havaittavissa. Merkittäviä eroja suurimman todennäköisyyden ja varianssin jakaumissa ei Gaussisia prosesseja käyttäneillä havaittu. Joillakin sovituksilla indusoivien muuttujien variationaalista approksimointia käyttänyt malli kuitenkin kuvasi varianssien jakaumat vaihteluvälin alarajalla hiukan muita enemmän päällekkäisiksi.



Kuva 4.4: Suurimman todennäköisyyden (ylh.) ja varianssin (alh.) skaalatut jakaumat testiaineistolle (sin.) ja taustakomponentin siirtymää kuvanneelle FinnSentiment -aineistolle (pun.) kaikilla sovituksilla. Alhaisimman validointitappion antanut malli korostettuna.

Verrattaessa jakaumia testiaineiston ja taustan siirtymää kuvanneen aineiston ryhmissä sen sijaan voidaan todeta jakaumien erottuvan selvästi heikommin. Kaikilla malleilla jakaumien huiput ovat hyvin lähellä tai käytännössä päällekkäin. Erottuvuutta voidaan todeta vain siirryttäessä vaihteluvälin alarajalta kohti yläkvantiileja, jolloin testiaineistolle estimoidut tiheydet estimoituvat nopeasti nollaan tai hyvin lähellä.

Toisessa vaiheessa menetelmiä vertailtiin myös edellisessä luvussa raportoiduilla suureilla käsitellen nyt vastemuuttujana havainnon jakaumaa eli kuulumista siirtynyttä jakaumaa kuvanneeseen aineistoon. Tulokset on raportoitu taulukossa 9.

Tuloksista voidaan todeta satunnaisia Fourier -piirteitä ja varianssia käyttäneen mallin tunnistavan yleisesti poikkeavat havainnot vertailluista malleista parhaiten jopa niin, että semanttista siirtymää kuvanneen aineiston kohdalla parannettavaaa ei käytännössä jää. Eroja satunnaisia Fourier -piirteitä ja varianssia käyttäneen mallin hyväksi erityisesti taustakomponentin siirtymää kuvanneessa FinnSentiment -aineistossa voidaan pitää merkittävinä. Edelleen voidaan todeta ennusteen varianssilla yleisesti kyettävän paremmin tunnistamaan poikkeavia havaintoja kuin suurimmalla todennäköisyydellä.

Taulukko 9: Semanttisen komponentin siirtymän (Finlex) ja taustakomponentin siirtymän (FinnSentiment) tunnistamista koskevat suureet, kun vastemuuttujana pidetään havainnon kuulumista siirtyneeseen jakaumaan ja luokittelu tehdään suurimman todennäköisyyden (max_tn) tai posteriorivarianssin (var) perusteella.

Malli	Mitta	Finlex			FinnSentiment		
		AUROC	AUPR	FPR95	AUROC	AUPR	FPR95
BERT	max tn	0.968 (0.001)	0.947 (0.003)	0.085 (0.004)	0.885 (0.004)	0.832 (0.005)	0.309 (0.009)
RFFL	max tn	0.982 (0.001)	0.969 (0.001)	0.045 (0.002)	0.919 (0.002)	0.869 (0.002)	0.213 (0.007)
RFFL	var	**0.999 (0.000)**	**0.999 (0.000)**	**0.002 (0.001)**	**0.943 (0.001)**	**0.930 (0.002)**	**0.187 (0.005)**
IPVI	max tn	0.967 (0.001)	0.945 (0.002)	0.092 (0.005)	0.884 (0.004)	0.833 (0.004)	0.335 (0.013)
IPVI	var	0.982 (0.002)	0.980 (0.002)	0.079 (0.005)	0.896 (0.004)	0.865 (0.004)	0.306 (0.010)

5 Yhteenveto ja johtopäätökset

Tässä tutkielmassa tarkasteltiin, voidaanko Transformer -arkkitehtuuria käyttävien neuroverkkojen robustisuutta ja epävarmuuden kvantifioinnin luotettavuutta parantaa Gaussisen prosessin ja neuroverkon yhdistävillä menetelmillä. Robustisuutta arvioitiin mallien yleistettävyydellä opetusjakaumasta poikkeavasta jakaumasta tuleville havainnoille ja epävarmuuden kvantifioinnin luotettavuutta mallien kyvyllä ennustaa luokitteluvirheitä ja poikkeavia havaintoja.

Tulosten perusteella voidaan todeta, että tarkasteluissa käytetty BERT-malli on varsin robusti poikkeaville havainnoille. Gaussisilla prosesseilla kyettiin marginaalisesti parantamaan luokittelutarkkuutta testiaineistossa ja satunnaisten Fourier -piirteiden Laplace -approksimaatiolla myös semanttisen komponentin siirtymää kuvanneessa aineistossa. Kun vertailtiin siirtynyttä jakaumaa parhaiten luokitelleita malleja, suorituskyvyn laskussa aineistojen välillä ei havaittu suuria eroja. Yleistettävyyttä arvioitaessa Suomen kielellä saatujen tulosten voidaan katsoa olevan linjassa aikaisempien englantia käyttäneiden tutkimusten kanssa (Hendrycks et al., 2020).

Edelleen voidaan todeta, ettei mikään malli johdonmukaisesti osoittautunut muita paremmaksi kaikissa tarkastelluissa tehtävissä, vaan paras malli, kuten myös epävarmuutta kuvaava mitta riippui tehtävästä ja tavoitteesta. Luokitteluvirhettä ennustettassa BERT -mallin luokittelukyky ei parantunut Gaussista prosessia tämän lisäksi soveltamalla. Satunnaisten Fourier -piirteiden Laplace -approksimaatiolla kyettiin kuitenkin paremmin erottelemaan väärin luokiteltuja ylimpiin epävarmuutta kuvaaviin kvantiileihin, kun epävarmuutta mitattiin suurimmalla todennäköisyydellä. Selvimmät erot Gaussisen prosessin soveltaneiden mallien eduksi, erityisesti satunnaisten Fourier -piirteiden Laplace -approksimaatiolla, voitiin todeta poikkeavia havaintoja ennustettaessa, kun epävarmuutta kuvattiin varianssilla.

Yhteenvetona tulosten käytännön implikaatioista voidaan todeta, että jos ollaan ensisijaisesti kiinnostuneita luokittelun oikeellisuudesta ja on realistista odottaa, että sovelluksessa kohdattavat tapaukset on saatu kattavasti edustetuksi opetusaineistossa, eikä toimintaympäristössä ole odotettavissa tapahtuvan muutoksia, Gaussisilla prosesseilla ei saada lisäarvoa virheellisten luokitteluiden tunnistamisessa. Gaussisella prosessilla voidaan mahdollisesti saavuttaa marginaalisesti parempi luokittelutarkkuus, mutta tämän käytännön merkitys riippunee virheellisten luokitteluiden seurauksesta tai sovelluksessa kohdattavasta volyymistä. Luokittelutarkkuutta voi-

daan yleisesti parantaa varsin niukillakin resursseilla jo sillä, että huomioidaan myös ennustettujen todennäköisyyksien suuruudet.

Jos on syytä odottaa, että sovelluksessa kohdattavat tapaukset voivat tulla siirtyneistä jakaumista, mitä voidaan pitää realistisena, satunnaisten Fourier -piirteiden soveltamisella voidaan yleisesti ottaen saada hieman parempia tuloksia etenkin, jos rinnalla voidaan käyttää ihmisluokittelijaa. Poikkeavia havaintoja tunnistettaessa herkin luokittelija vertailluista on Gaussisen prosessin antama varianssi erityisesti satunnaisten Fourier -piirteiden Laplace -approksimaatiolla. Edellistä havaintoa voidaan hyödyntää esimerkiksi sovelluksessa kohdattavassa jakaumassa tapahtuvien muutosten ja mallin suorituskyvyn seurannassa ilman merkittäviä laskennallisia lisäkustannuksia. Estimaatti varianssille voidaan määrätä samalla, kun havaintoja luokitellaan. Mikäli luokiteltavaksi tulee yhä enemmän opetusaineistosta poikkeavia havaintoja, varianssin jakauman voidaan tässä tutkielmassa saatujen tulosten perusteella odottaa vastaavasti siirtyvän. Edellistä voidaan mahdollisesti hyödyntää esimerkiksi arvioitaessa, milloin käytäntöön sovellettua mallia tulisi päivittää.

Tutkielmassa sovelluksena käytettyä tekstin sentimenttiin perustuvaa luokittelua voidaan pitää luokittelutarkkuuksillakin arvioituna nykyaikaisille malleille jokseenkin helppona tehtävänä. Tässä Gaussista prosessia soveltamalla saavutettu marginaalinen parannus luokittelutarkkuudessa vastaa suuruudeltaan sitä parannusta, mitä alkuperäisessä julkaisussa Liu et al. (2020) oli satunnaisten Fourier -piirteiden Laplace -approksimaatiota soveltaneelle menetelmälle raportoitu, vaikka implementointi tässä työssä ei täysin alkuperäistä vastannutkaan. Indusoivien muuttujien variationaaliselle approksimaatiolle vastaavaa vertailukohtaa ei ole käytettävissä. Jakaumatarkasteluissa saatiin kuitenkin viitteitä siitä, että ennustettaessa vaikeampaa aineistoa Gaussisen prosessin soveltaneet mallit kykenivät saamaan parempaa erottuvuutta oikein ja väärinluokiteltujen ryhmien välille, mikä myös jossain määrin realisoitui arviointitehokkuuksissa. Avoimeksi jää, millä tavalla tähän ilmiöön vaikuttaisi se, että tehtävä jo sovitettaessa olisi vaikeampi.

Lähteet

Arora, U., Huang, W., & He, H. (2021). Types of Out-of-Distribution texts and how to detect them. *arXiv preprint arXiv:2109.06827*.

Ba, J. L., Kiros, J. R., & Hinton, G. E. (2016). Layer normalization. *arXiv preprint arXiv:1607.06450*.

Bai, Y., Mei, J., Yuille, A. L., & Xie, C. (2021). Are Transformers more robust than CNNs? *Advances in Neural Information Processing Systems*, *34*, 26831–26843.

Bengio, Y. (2013). Deep learning of representations: Looking forward. Teoksessa *Statistical Language and Speech Processing: First International Conference, SLSP 2013, Tarragona, Spain, July 29-31, 2013. Proceedings 1* (s. 1–37).

Bishop, N. M., Christopher M & Nasrabadi. (2006). *Pattern Recognition and Machine Learning* (osa 4). Springer.

Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American statistical Association*, *112*(518), 859–877.

Bochner, S. (1959). *Lectures on Fourier Integrals* (osa 42). Princeton University Press.

Britz, D., Goldie, A., Luong, M.-T., & Le, Q. (2017). Massive exploration of neural machine translation architectures. *arXiv preprint arXiv:1703.03906*.

Castro, J. L., Mantas, C. J., & Benıtez, J. (2000). Neural networks with a continuous squashing function in the output are universal approximators. *Neural Networks*, *13*(6), 561–563.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, *2*(4), 303–314.

Dauphin, Y. N., Pascanu, R., Gulcehre, C., Cho, K., Ganguli, S., & Bengio, Y. (2014). Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *Advances in Neural Information Processing Systems*, *27*.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Draper, D. (1987). A research agenda for assessment and propagation of model uncertainty. *The RAND Publication Series*.

Draper, D. (1995). Assessment and propagation of model uncertainty. *Journal of the Royal Statistical Society: Series B (Methodological)*, *57*(1), 45–70.

Eldan, R., & Shamir, O. (2016). The power of depth for feedforward neural networks. Teoksessa *Conference on learning theory* (s. 907–940).

Freitag, E. . B. R. (2009). *Complex Analysis*. Springer.

Gawlikowski, J., Tassi, C. R. N., Ali, M., Lee, J., Humt, M., Feng, J., ... Zhu, X. X. (2021). A survey of uncertainty in deep neural networks. *arXiv preprint arXiv:2107.03342*.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2014). *Bayesian Data Analysis*. Chapman and Hall/CRC.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. Teoksessa *International Conference on Machine Learning* (s. 1321–1330).

Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (osa 2). Springer.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Teoksessa *Proceedings of the IEEE conference on computer vision and pattern recognition* (s. 770–778).

Hendrycks, D., & Gimpel, K. (2016). A baseline for detecting misclassified and Out-of-Distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

Hendrycks, D., Liu, X., Wallace, E., Dziedzic, A., Krishnan, R., & Song, D. (2020). Pretrained Transformers improve Out-of-Distribution robustness. *arXiv preprint arXiv:2004.06100*.

Hensman, J., Durrande, N., & Solin, A. (2017). Variational Fourier features for Gaussian processes. *Journal of Machine Learning Research*, *18*(1), (s. 5537–5588).

Hensman, J., Fusi, N., & Lawrence, N. D. (2013). Gaussian processes for big data. *arXiv preprint arXiv:1309.6835*.

Hensman, J., Matthews, A., & Ghahramani, Z. (2015). Scalable variational Gaussian process classification. *Artificial Intelligence and Statistics*, 351–360.

Herman, R. L. (2016). *An Introduction to Fourier Analysis*. Chapman and Hall/CRC.

Hsu, Y.-C., Shen, Y., Jin, H., & Kira, Z. (2020). Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. Teoksessa *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (s. 10951–10960).

Huber, P. J. (2011). Robust statistics. Teoksessa *International Encyclopedia of Statistical Science* (s. 1248–1251). Springer.

Isbister, T., Carlsson, F., & Sahlgren, M. (2021). Should we stop training more monolingual models, and simply use machine translation instead? *arXiv preprint arXiv:2104.10441*.

Jensen, J. L. W. V. (1906). Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta Mathematica*, *30*(1), (s. 175–193).

Jordan, M. I., Ghahramani, Z., Jaakkola, T., & Saul, L. (1999). Introduction to variational methods for graphical models. *Machine Learning*, *37*, 183–223.

Kallenberg, O. (1997). *Foundations of Modern Probability* (osa 2). Springer.

Kivlichan, I. D., Lin, Z., Liu, J., & Vasserman, L. (2021). Measuring and improving model-moderator collaboration using uncertainty estimation. *arXiv preprint arXiv:2107.04212*.

Kudo, T., & Richardson, J. (2018). Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*.

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, *22*(1), 79–86.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Leshno, M., Lin, V. Y., Pinkus, A., & Schocken, S. (1993). Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Networks*, *6*(6), 861–867.

Lin, H., & Jegelka, S. (2018). ResNet with one-neuron hidden layers is a universal approximator. *Advances in Neural Information Processing Systems*, *31*.

Lindén, K., Jauhiainen, T., & Hardwick, S. (2023). Finnsentiment: a Finnish social media corpus for sentiment polarity annotation. *Language Resources and Evaluation*, 1–29.

Liu, J., Lin, Z., Padhy, S., Tran, D., Bedrax Weiss, T., & Lakshminarayanan, B. (2020). Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in Neural Information Processing Systems*, *33*, 7498–7512.

Liu, J., Padhy, S., Ren, J., Lin, Z., Wen, Y., Jerfel, G., ... Lakshminarayanan, B. (2022). A simple approach to improve single-model deep uncertainty via distance-awareness. *Journal of Machine Learning Research*, *23*, 1–63.

Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Lu, Z., Pu, H., Wang, F., Hu, Z., & Wang, L. (2017). The expressive power of neural networks: A view from the width. *Advances in Neural Information Processing Systems*, *30*.

Marcus, G. (2018). Deep learning: A critical appraisal. *arXiv preprint arXiv:1801.00631*.

Mikolov, T., Yih, W.-t., & Zweig, G. (2013). Linguistic regularities in continuous space word representations. Teoksessa *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (s. 746–751).

Miyato, T., Kataoka, T., Koyama, M., & Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*.

Neal, R. M. (2012). *Bayesian Learning for Neural Networks* (osa 118). Springer Science & Business Media.

Nguyen, A., Yosinski, J., & Clune, J. (2015). Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. Teoksessa *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (s. 427–436).

Ober, S. W., Rasmussen, C. E., & van der Wilk, M. (2021). The promises and pitfalls of deep kernel learning. *Uncertainty in Artificial Intelligence*, 1206–1216.

OpenAI. (2023). GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., ... Snoek, J. (2019). Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. *Advances in Neural Information Processing Systems*, *32*.

Paleyes, A., Urma, R.-G., & Lawrence, N. D. (2022). Challenges in deploying machine learning: a survey of case studies. *ACM Computing Surveys*, *55*(6), 1–29.

Pang, G., Shen, C., Cao, L., & Hengel, A. V. D. (2021). Deep learning for anomaly detection: A Review. *ACM Computing Surveys (CSUR)*, *54*(2), 1–38.

Park, S., Yun, C., Lee, J., & Shin, J. (2020). Minimum width for universal approximation. *arXiv preprint arXiv:2006.08859*.

Quinonero-Candela, J., & Rasmussen, C. E. (2005). A unifying view of sparse approximate Gaussian process regression. *The Journal of Machine Learning Research*, *6*, 1939–1959.

Quinonero-Candela, J., Sugiyama, M., Schwaighofer, A., & Lawrence, N. D. (2008). *Dataset shift in machine learning*. MIT Press.

Rahimi, A., & Recht, B. (2007). Random features for large-scale kernel machines. *Advances in Neural Information Processing Systems*, *20*.

Rangwani, H., Aithal, S. K., Mishra, M., et al. (2022). Escaping saddle points for effective generalization on class-imbalanced data. *Advances in Neural Information Processing Systems*, *35*, 22791–22805.

Rasmussen, C. E., & Williams, C. K. (2006). *Gaussian Processes for Machine Learning* (osa 2). MIT Press Cambridge, MA.

Ren, J., Liu, P. J., Fertig, E., Snoek, J., Poplin, R., Depristo, M., ... Lakshminarayanan, B. (2019). Likelihood ratios for Out-of-Distribution detection. *Advances in Neural Information Processing Systems*, *32*.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PloS one*, *10*(3).

Seeger, M. (2004). Gaussian processes for machine learning. *International Journal of Neural Systems*, *14*(02), 69–106.

Sennrich, R., Haddow, B., & Birch, A. (2015). Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929–1958.

Sutherland, D. J., & Schneider, J. (2015). On the error of random Fourier features. *arXiv preprint arXiv:1506.02785*.

Tiedemann, J. (2012). Parallel data, tools and interfaces in opus. Teoksessa *Language Resources and Evaluation Conference* (osa 2012, s. 2214–2218).

Tierney, L., & Kadane, J. B. (1986). Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association*, *81*(393), 82–86.

Titsias, M. (2009). Variational learning of inducing variables in sparse Gaussian processes. Teoksessa *Artificial Intelligence and Statistics* (s. 567–574).

Ton, J.-F., Flaxman, S., Sejdinovic, D., & Bhatt, S. (2018). Spatial mapping with Gaussian processes and nonstationary Fourier features. *Spatial Statistics*, *28*, 59–78.

Torralba, A., & Efros, A. A. (2011). Unbiased look at dataset bias. Teoksessa *Computer Vision and Pattern Recognition 2011* (s. 1521–1528).

van Amersfoort, J., Smith, L., Jesson, A., Key, O., & Gal, Y. (2021a). *Deterministic uncertainty estimation.* https://github.com/y0ast/DUE/tree/main. GitHub.

van Amersfoort, J., Smith, L., Jesson, A., Key, O., & Gal, Y. (2021b). On feature collapse and deep kernel learning for single forward pass uncertainty. *arXiv preprint arXiv:2102.11409*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

Virtanen, A., Kanerva, J., Ilo, R., Luoma, J., Luotolahti, J., Salakoski, T., ... Pyysalo, S. (2019). Multilingual is not enough: BERT for Finnish. *arXiv preprint arXiv:1912.07076*.

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2018). Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Wilson, A., Hu, Z., Salakhutdinov, R., & Xing, E. P. (2016). Deep kernel learning. Teoksessa *Artificial Intelligence and Statistics* (s. 370–378).

Woodbury, M. A. (1950). *Inverting Modified Matrices*. Department of Statistics, Princeton University.

Zagoruyko, S., & Komodakis, N. (2016). Wide residual networks. *arXiv preprint arXiv:1605.07146*.

Zhu, W., Zeng, N., Wang, N., et al. (2010). Sensitivity, specificity, accuracy, associated confidence interval and ROC analysis with practical SAS implementations. *NESUG Proceedings: Health Care and Life Sciences, Baltimore, Maryland*, *19*, 67.

Liite A Mallien kuvaukset ja koodit

Tässä liitteessä on annettu tarkemmat tiedot luvussa 3 selostettujen menetelmien implementoinneista sekä julkaistu mallien määrittämiseen ja ennusteiden määräämiseen liittyvät koodit keskeisimmiltä osin. Ennusteiden määräämisen osalta koodi on sisällytetty vain testiaineiston osalta, koska menettely on tässä kaikilla aineistoilla identtinen. Tarvittavat ohjelmointikirjastot ja -moduulit käyvät ilmi kunkin koodia esittelevän osuuden alusta.

Hyperparametreina kaikilla malleilla pidettiin opetusaskeleella käsiteltävän osajoukon kokoa (`_BATCH_SIZE`), opetustahtia (`_LR`) ja epookkien lukumäärää (`_EPOCHS`). Kussakin mallissa käytettiin alhaisimman validointitappion antanutta hyperparametrien konfiguraatiota. Kokeiluissa käytetyt hyperparametrien arvot otettiin julkaisusta Devlin et al. (2018), mistä kuitenkin poikettiin epookkien lukumäärälle ilmoitetuista (2,3,4), koska ylisovittumista todettiin alkavan tapahtua jo toisen epookin aikana. Kokeillut hyperparametrien arvot on ilmoitettu taulukossa 10 ja sovitettaessa käytetyt arvot mallikohtaisesti koodien yhteydessä.

Taulukko 10: Malleille kokeillut hyperparametrien arvot.

Hyperparametri	Arvot
Osajoukon koko	16, 32
Opetustahti	2e-5, 3e-5, 5e-5
Epookkien lukumäärä	1, 2

Indusoivien muuttujien variationaalista approksimaatiota soveltaneessa mallissa hyperparametreina käsiteltiin lisäksi kovarianssifunktion hyperparametreille käytettävää opetustahtia, koska alustavissa kokeiluissa tämän todettiin vaikuttavan mallin suorituskykyyn. Variationaalisille parametreille käytetyn opetustahdin suhteen vastaavaa herkkyyttä ei todettu. Kovarianssifunktion hyperparametreille kokeiltiin arvoja 0.1, 0.01 ja 0.001. Variationaalisten parametrien kohdalla sen sijaan käytettiin lähdejulkaisun mukaisesti opetustahtia 0.1.

Optimointialgoritmina kaikilla malleilla käytettiin AdamW -algoritmia (Loshchilov & Hutter, 2017), mikä porrastettiin siten, että opetustahti nostettiin ensimmäisen opetusaskeleiden kymmenyksen aikana mallille asetettuun opetustahtiin ja häivytettiin tämän jälkeen lineaarisesti kohti nollaa, kuten Devlin et al. (2018). Optimointialgoritmin yksityiskohtia ei käsitellä tässä tutkielmassa. Myöskään koodissa käytettyjen luokkien ottamia argumentteja ei tässä käsitellä tarkemmin muutoin, kuin lähdejulkaisuissa kuvatuilta osin. Muutoin argumenttien selitysten osalta viitataan ohjelmointikirjastojen dokumentaatioihin.

Kaikki mallit määritettiin vastaamaan logistista regressiota eli dikotomista vastetta käsiteltiin yksiulotteisena.

A.1 BERT-malli

BERT-malli muodostettiin TensorFlow Model Garden -kirjastosta ladatusta Bert -enkooderista (`BertEncoder`) ja tämän perään liitetystä luokittelupäästä (`Classifi-`

`cationHead`), mille syötteenä annettiin [CLS] -saneen koontikerroksen (`pooled_output`) ulostulo (Luku 3.1.1). Luokittelupää on tässä yhden yksikön (`num_classes`) tiheä kerros, minkä aktivoinfunktio tulee tässä toteutuksessa määrätyksi tappiofunktiossa ja binääriselle ristientropialle (`BinaryCrossentropy`) on sigmoid -funktio.

Malli määritettiin siten, että argumentit voitiin antaa sanakirjana (`cfg`), mistä kuitenkin asetettiin vain FinBERT -mallin konfiguraatiotiedostossa oletusargumenteista poikenneet argumentit. Konfiguraatiotiedosto, esiopetetut kertoimet (`bert_model.ckpt`) ja aineistojen esikäsittelyssä tekstien saneistukseen tarvittu sanakirja on julkaistu Turun yliopiston Turku NLP -tutkimusryhmän internetsivuilla.

```
import tensorflow_models as tfm
import tensorflow as tf
from tfm.nlp.networks import BertEncoder
from tfm.nlp.layers import ClassificationHead
from official.nlp import optimization

def BERTBaseClassifier(cfg):

  encoder = BertEncoder(vocab_size=cfg['vocab_size'],
                        type_vocab_size=cfg['type_vocab_size'],
                        dict_outputs=True)
  _input = encoder.inputs
  encoder_output = encoder(_input)
  output = ClassificationHead(num_classes=1,
                              inner_dim=None)
                              (encoder_output['pooled_output'])

  model = tf.keras.Model(inputs = _input, outputs = output)

  return model, encoder

# Hyperparametrit
_BATCH_SIZE = 32
_LR = 3e-5
_EPOCHS = 1

model, encoder = BERTBaseClassifier(cfg)
encoder.load_weights('bert_model.ckpt'))

# Optimointialgoritmi
steps_per_epoch = X_train['input_word_ids'].shape[0]/_BATCH_SIZE
num_train_steps = steps_per_epoch * _EPOCHS
num_warmup_steps = int(0.1*num_train_steps)
optimizer = optimization.create_optimizer(
                              init_lr=_LR,
                              num_train_steps=num_train_steps,
                              num_warmup_steps=num_warmup_steps,
                              optimizer_type='adamw')

# Tappiofunktio
loss = tf.keras.losses.BinaryCrossentropy(from_logits = True)

# Kootaan malli
model.compile(optimizer=optimizer,
```

```
                    loss=loss,
                    metrics=['accuracy'])

# Sovitetaan malli
model.fit(X_train, Y_train,
          validation_data = (X_val, Y_val),
          epochs=_EPOCHS,
          batch_size=_BATCH_SIZE)

# Ennusteet
logits = model.predict(X_test)
```

A.2 Satunnaisten Fourier -piirteiden Laplace -approksimaatio

Satunnaisten Fourier -piirteiden Laplace -approksimaatiolle on olemassa valmis implementaatio TensorFlow Model Garden -kirjaston nlp -moduulissa (`GaussianProcessClassificationHead`), mutta koska tälle ei ollut mahdollista antaa argumenttina kovarianssifunktion skaalaparametria (`gp_kernel_scale`), mille lähdejulkaisussa Liu et al. (2020) oli ilmoitettu arvo 2.0, toteuttiin malli Tensorflow Model Garden -kirjaston nlp -moduulin `SpectralNormalization` ja `RandomFeatureGaussianProcess` -kerroksilla. Satunnaisten Fourier -piirteiden lukumääräksi (`num_inducing`) asetettiin lähdejulkaisussa ilmoitettu 2048.

Koska spektraalinormalisoinnin suorittava `SpectralNormalization` -kerros on ns. kääreluokka (*wrapper*), mille täytyy antaa argumenttina kerros -moduulin `tf.keras.layers.Layer` instanssi ja spektraalinormalisointi tuli toteuttaa koontikerrokselle, ei koontikerroksena tässä voitu käyttää Bert -enkooderiin sisältyvää koontikerrosta, kuten BERT-mallissa tehtiin. Edellisestä johtuen koontikerros implementointiin erikseen (`spec_norm_pooler`) ja enkooderin koontikerros ohitettiin ohjaamalla [CLS] -saneen ulostulo viimeiseltä Transformer -lohkolta (`sequence_output`) itse määritetylle koontikerrokselle.

FinBERT -mallin kertoimien lataaminen suoritettiin malliin tätä tarkoitusta varten luodulla metodilla `set_encoder_weights`. Edelleen koska FinBERT -mallin koontikerroksen kertoimet tulevat tässä ladatuiksi enkooderin koontikerrokselle, minkä sijaan mallissa käytetään itse määrättyä koontikerrosta, implementoitiin malliin vielä metodi `set_pooler_weights`, millä kertoimet voitiin kopioida enkooderin koontikerrokselta itse määrätylle koontikerrokselle.

Optimointialgoritmin, tappiofunktion ja mallin kokoamisen sekä sovittamisen osalta koodi on identtinen BERT-mallin kanssa, eikä sitä siten tältä osin ole sisällytetty seuraavaan koodiin. Mallilla tallennettiin latentin funktion f posteriennustejakauman odotusarvo ja (`f_mean`) ja posteriorivarianssi (`f_var`) sekä ennusteina käytetty luvussa 3.1.2 kuvattu sigmoid -muunnoksen posterioriodotusarvon probit -approksimaatio (`mf_logits`). Kovarianssimatriisia ei laskennallisista syistä kyetty määräämään koko testiaineistolle yhdellä kertaa, joten ennusteita määrättäessä aineistoa käsiteltiin 50:n havainnon osajoukoissa ja varianssit poimittiin osajoukoille määrättyjen kovarianssimatriisien diagonaaleilta.

```
import tensorflow as tf
import tensorflow_models as tfm
from tfm.nlp.networks import BertEncoder
from tfm.nlp.layers import SpectralNormalization
from tfm.nlp.layers import RandomFeatureGaussianProcess
from tf.keras.initializers import TruncatedNormal
from tf.keras.initializers import RandomNormal
from official.nlp import optimization
from official.nlp.modeling.layers.gaussian_process import
                                                    mean_field_logits

class BERTSNGPClassifier(tf.keras.Model):

  def __init__(self,
               encoder):

    spec_norm_pooler = tf.keras.layers.Dense(
                    units=768,
                    activation='tanh',
                    kernel_initializer=TruncatedNormal(stddev=0.02),
                    name='spec_norm_pooler')

    spec_norm_pooler = SpectralNormalization(spec_norm_pooler,
                                             iteration=1,
                                             norm_multiplier=0.95)

    gp_layer = RandomFeatureGaussianProcess(
            units=1,
            scale_random_features=False,
            use_custom_random_features=True,
            kernel_initializer=TruncatedNormal(stddev=0.02),
            custom_random_features_initializer=(RandomNormal(
                                                    mean=0.0,
                                                    stddev=0.05)),
            gp_kernel_scale=2.0,
            num_inducing=2048,
            normalize_input=True,
            gp_cov_ridge_penalty=1,
            gp_cov_momentum=-1)

    # Malli Functional API: lla
    _input = encoder.inputs
    encoder_output = encoder(_input)
    cls_token = encoder_output['sequence_output'][:, 0, :]
    pooled_output = spec_norm_pooler(cls_token)
    output = gp_layer(pooled_output)

    super(BERTSNGPClassifier, self).__init__(inputs = _input,
                                             outputs = output)

    # Tarvittavat attribuutit
    self._encoder = encoder
    self._pooler = spec_norm_pooler
    self._encoder_pooler = encoder.layers[-1]

  def call(self, inputs, training=False, return_covmat=False):
```

```
    logits, covmat = super().call(inputs)

    # Kovarianssimatriisin palautus vain ennustettaessa
    if not training and return_covmat:
      return logits, covmat

    return logits

  # Metodi FinBERTin kertoimien lataamista varten:
  def set_encoder_weights(self, ckpt_path):
    self._encoder.load_weights(ckpt_path)

  # Metodi kertoimien kopioimiseksi spec_norm_pooler -kerrokselle
  def set_pooler_weights(self):
    _pooler_weights = self._pooler.get_weights()
    _pooler_weights[0] = self._encoder_pooler.get_weights()[0]
    _pooler_weights[1] = self._encoder_pooler.get_weights()[1]
    self._pooler.set_weights(_pooler_weights)

# Hyperparametrit

_BATCH_SIZE = 32
_LR = 2e-5
_EPOCHS = 1

encoder = BertEncoder(vocab_size=cfg['vocab_size'],
                      type_vocab_size=cfg['type_vocab_size'],
                      dict_outputs=True)
model = BERTSNGPClassifier(encoder)
model.set_encoder_weights('bert_model.ckpt'))
model.set_pooler_weights()

...

# Poimitaan ennusteet ja varianssit
num_test_steps = int(len(Y_test)/50)

for j in range(num_test_steps):

  start = j*50
  end = start+50
  X_sample = {}
  for key in X_test:
    X_sample[key] = X_test[key][start:end]

  f_mean, covmat = model(X_sample, training=False, return_covmat = True)
  mf_logits = mean_field_logits(f_mean, covmat, math.pi/8)
  f_var(tf.linalg.diag_part(covmat).numpy().squeeze())
```

A.3 Indusoivien muuttujien variationaalinen approksimaatio

Indusoivien muuttujien variationaalista approksimaatiota soveltanut malli toteutettiin PyTorch -koneoppimiskirjastoa käyttävällä Gaussisen prosessin -malleille tarkoitetulla GPyTorch -kirjastolla. BERT-enkooderi voitiin ladata suoraan FinBERT -mallin kertoimilla Transformers -kirjastosta. Toteutus koontikerroksen ja spektraalinormalisoinnin osalta vastaa edellä TensorFlow -kirjastoa käyttäen kuvattua toteutusta sillä poikkeuksella, että tässä FinBERT -mallin koontikerroksen kertoimet kopioitiin jo mallista instanssia luotaessa.

Spektraalinormalisointiin käytettiin PyTorch -kirjaston `spectral_norm` -moduulia. Koska tässä normalisoiduille mallin kertoimille ei ollut mahdollista asettaa kerrointa c (Luku 3.1.1) alkuperäistä lähdekoodia muutettiin lisäämällä tämä moduuliin argumentiksi ja korjaamalla moduulin palauttamia kertoimia vastaavasti.

Gaussinen prosessi (`gp`) voitiin määrätä suoraan alkuperäisen julkaisun lähdekoodia kopioiden. Koodia tai tässä käytettyjä moduuleita ei tältä osin toisteta tässä tutkielmassa, vaan viitataan GitHub -repositorioon van Amersfoort et al., 2021a. Alkuperäisestä julkaisusta poiketen tämän tutkielman toteutuksessa malli toteutettiin logistista regressiota vastaten yhtenä Gaussisena prosessina (`num_outputs`), kun alkuperäisessä julkaisussa kutakin luokkaa $k = 1, \ldots, K$ vastasi oma Gaussinen prosessi. Myös indusoivien pisteiden sijainnin ja kovarianssifunktion skaalaparametrin alustuksissa käytettyjen funktioiden koodit on julkaistu mainitussa repositoriossa, eikä niitä siten toisteta tässä tutkielmassa.

GPyTorch -mallissa muunnos todennäköisyyksiksi tehdään uskottavuusfunktiossa, mille ei ollut valmista logistista regressiomallia vastaavaa toteutusta. Tätä vastaava uskottavuusfunktio (`LogisticLikelihood`) toteutettiin perimällä likelihood -moduulin `_OneDimensionalLikelihood` -luokkaa, kuten kirjaston dokumentaatiossa yksiulotteisten uskottavuuksien implementoinneista oli kerrottu.

Mallilla tallennettiin latentin funktion f posteriennustejakauman odotusarvo (`f_mean`) ja posteriorivarianssi (`f_var`) sekä ennusteina käytetty sigmoid -muunnoksen posterioriodotusarvo (`prob`), mikä nyt määrättiin $n = 32$ otoksesta latentin f posteriennustejakaumasta (Luku 3.1.3).

```
import torch
import gpytorch
from torch import distributions
from torch.optim import AdamW
from transformers import AutoModel
from transformers import get_linear_schedule_with_warmup
from gpytorch.likelihoods.likelihood import _OneDimensionalLikelihood
from gpytorch.mlls import VariationalELBO


class BERTSNFeatureExtractor(torch.nn.Module):

  def __init__(self):
    super(BERTSNFeatureExtractor, self).__init__()
    self.encoder = AutoModel.from_pretrained(
```

```
                                "TurkuNLP/bert-base-finnish-cased-v1",
                                vocab_size=50105,
                                return_dict=False)

        self.encoder_pooler = self.encoder.pooler

        pooler_layer = torch.nn.Linear(768, 768)

        with torch.no_grad():
          pooler_layer.weight.copy_(self.encoder_pooler.dense.weight)

        self.spec_norm_pooler = spectral_norm(pooler_layer,
                                              norm_multiplier=0.95)
        self.pooler_activation = torch.nn.Tanh()

      def forward(self, ids, mask, token_type_ids):
        encoder_output = self.encoder(ids, mask, token_type_ids)
        cls_last_hidden = encoder_output[0][:,0,:]
        pooled_out = self.spec_norm_pooler(cls_last_hidden)
        pooled_out = self.pooler_activation(pooled_out)
        return pooled_out

...

class DKL(gpytorch.Module):

    def __init__(self, feature_extractor, gp):
        super().__init__()

        self.feature_extractor = feature_extractor
        self.gp = gp

    def forward(self, ids, mask, token_type_ids):
        features = self.feature_extractor(ids, mask, token_type_ids)
        return self.gp(features)

class LogisticLikelihood(_OneDimensionalLikelihood):

  def forward(self, function_samples, **kwargs):

    output_probs = torch.sigmoid(function_samples)
    return distributions.Bernoulli(probs=output_probs)

# Hyperparametrit:
_BATCH_SIZE = 32
_LR = 3e-5
_LR_HYPER = 1e-3
_EPOCHS = 1

# Muut argumentit
INITIAL_INDUCING_POINTS = torch.load('due_init_points')
INITIAL_LENGTHSCALE = torch.tensor(4.2379, dtype=torch.float)

feature_extractor = BERTSNFeatureExtractor()
gp = GP(num_outputs=1,
        initial_lengthscale=INITIAL_LENGTHSCALE,
```

```
        initial_inducing_points=INITIAL_INDUCING_POINTS,
        kernel='RBF')
model = DKL(feature_extractor, gp)
model.to(device)

likelihood = LogisticLikelihood()
likelihood = likelihood.to(device)

# Asetetaan opetustahdit
param_groups = [
    {'params': model.feature_extractor.parameters(), 'lr': _LR},
    {'params': model.gp.hyperparameters(), 'lr': _LR_HYPER},
    {'params': model.gp.variational_parameters(), 'lr': 0.1},
    {'params': likelihood.parameters(), 'lr': 0.1} # Ei parametreja
    ]

# Optimointialgoritmi
optimizer = AdamW(param_groups)
steps_per_epoch = len(training_set) / _BATCH_SIZE
num_steps = _EPOCHS * steps_per_epoch
num_warmup_steps = 0.1 * num_steps
scheduler = get_linear_schedule_with_warmup(
                                    optimizer,
                                    num_warmup_steps=num_warmup_steps,
                                    num_training_steps=num_steps)

# Tappiofunktio
elbo_fn = VariationalELBO(likelihood,
                          model.gp,
                          num_data=len(training_set))

# Sovitetaan malli
model.train()
likelihood.train()
for epoch in range(_EPOCHS):
  for j ,data in enumerate(training_loader, 0):

    ...

    # Nollataan gradientit jokaiselle osajoukolle
    optimizer.zero_grad()

    output = model(ids, mask, token_type_ids)

    # Otoskooksi halutaan 32 (oletus = 10)
    with gpytorch.settings.num_likelihood_samples(32):

      # Lasketaan tappio ja gradientit
      loss = -elbo_fn(output, targets)
      loss.backward()

    # Opetusaskel kertoimille
    optimizer.step()

    # Opetustahti seuraavalle opetusaskeleelle
    scheduler.step()
```

```
# Ennusteet
model.eval()
likelihood.eval()

    with torch.no_grad():
        for j ,data in enumerate(test_loader, 0):

        ...

        output = model(ids, mask, token_type_ids)
        output = output.to_data_independent_dist()
        f_mean = output.loc
        f_var = output.scale
        f_sample = output.sample(sample_shape=([32]))
        prob_sample = likelihood(f_sample).probs
        prob = prob_sample.mean(0)
```