

# ImageRecognition testausdokumentaatio

Simo Korkolainen

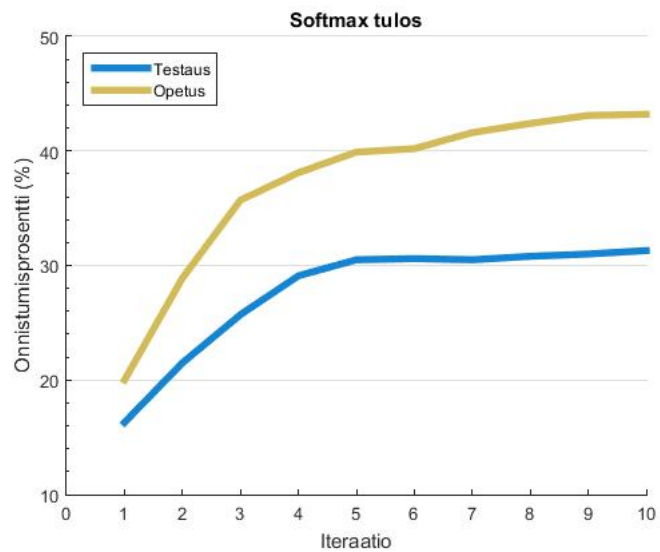
27. kesäkuuta 2016

Tässä dokumentaatiossa kerrotaan neuroverkon toiminnan testaamisen tuloksista. Testeihin liittyvä koodi löytyy projektin testdocumentation-kansiosta. Ohjelman toimintaa testattiin myös PIT-testeillä, joista löytyy tarkempaa tietoa dokumentaation PIT-raportista. Neuroverkko opetettiin tunnistamaan kuvia kymmenestä eri luokasta, kuten koira tai auto. Ensimmäiseksi oppimisalgoritmia testattiin hyvin yksinkertaisella verkon rakenteella. Verkko koostui vain yhdestä softmax-kerroksesta. Neuroverkon syötteenä oli  $32 \times 32 \times 3$  värikuva, ja neuroverkko antoi vastauksena kuvan todennäköisyyden kuulua kuhunkin kymmenestä eri luokasta.

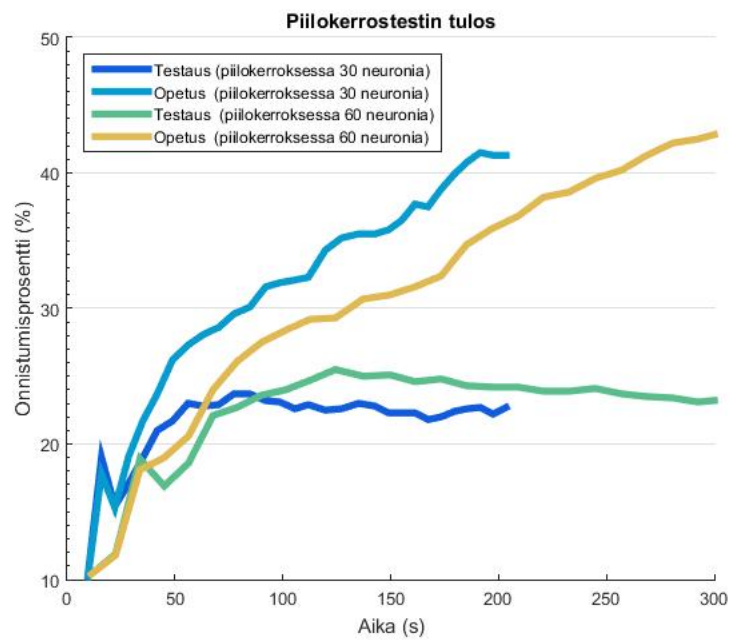
Yhden iteraation aikana neuroverkolle näytettiin tuhat kuvaa opetusdatasta, ja verkon painoja päivitettiin luokittelun onnistumisen perusteella. Iteraation loputtua laskettiin kuinka monta kuvaa neuroverkko tunnistaa opetus- ja testidatasta. Testidataa ei käytetty neuroverkon opettamiseen, jotta voitaisiin havaita yleistyykö neuroverkon oppima luokittelufunktion oppimisdatan ulkopuolelle. Kuvat valittiin tasaisesti eri luokista, joten täysin satunnaisesti kuvalle luokan arvaava luokittelija olisi odotusarvoisesti oikeassa kymmenessä prosentissa tapauksista.

Kuvasta 1 näkee, että opetusdatan luokittelutulos on noin 10% parempi kuin testidatan tulos. Tämä tarkoittaa sitä, että neuroverkko on ylisovittanut painonsa opetusdataan soveltuviksi, eikä neuroverkko pystynyt yleistämään oppimaansa ennalta näkemättömiin kuviin.

Ohjelmassa matriisilaskenta oli aluksi toteutettu siten, että aina jokaisen operaation yhteydessä varattiin uutta muistia. Kuitenkin tämä hidasti ohjelman toimintaa merkittävästi. Ohjelmaa muutettiin siten, että matriisilaskun tulos on mahdollista tallentaa jo aikasemmin luotuun matriisiin, jolloin uutta muistia ei tarvitse varata. Optimoinnin aikaansaamaa parannusta testattiin luokassa OptimizationTest. Sama ohjelma ajettiin optimoidulla ja optimoimattomalla koodilla. Ohjelmassa luokiteltiin samoja kuvia kuin edellisessä testissä. Testissä neuroverkon ensimmäiseksi piilokerrokseksi valittiin softplus-kerros, jossa oli 200 neuronua. Tulostekerros oli softmax-kerros, jossa oli 10 neuronua. Optimoidun koodin ajamisessa kului 31.59 sekuntia ja optimoimattoman koodin ajamisessa kului 200.12 sekuntia. Optimoitu koodi oli yli kuusi kertaa nopeampaa kuin optimoimaton koodi.



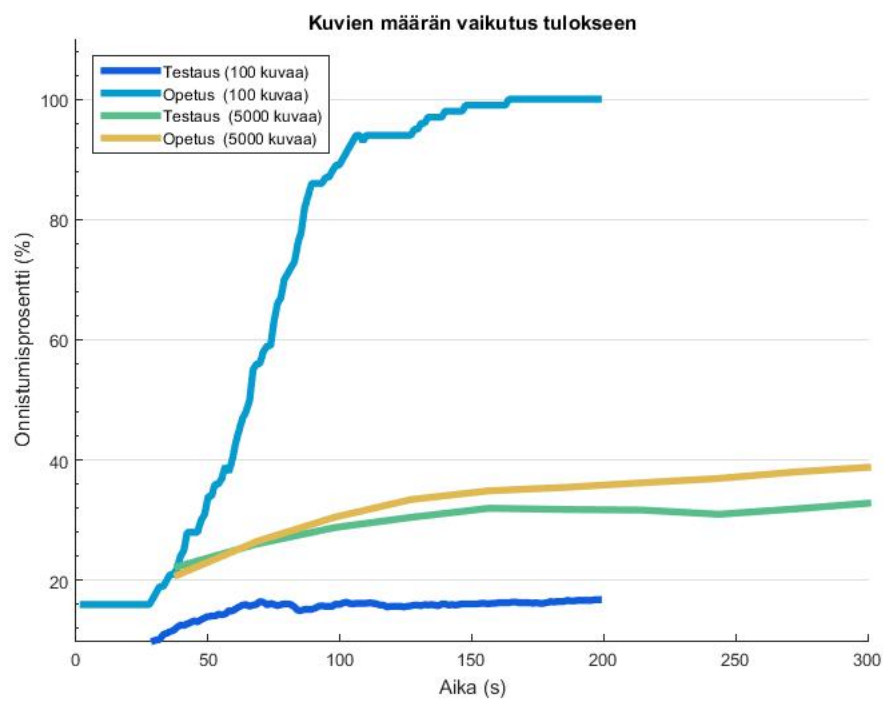
Kuva 1:



Kuva 2:

Kuvassa 2 on kuvattu piilokerrosten lukumäärän vaikutusta oppimistulokseen. Testissä neuroverkossa oli kolme kerrosta syötekerros, piilokerros ja softmax-kerros. Piilokerrokseen valittiin ensiksi 30 ja sitten 60 neuronia. Kuvasta nähdään, että neuronien lukumäärän kasvattaminen pidensi opetusdatan oppimiseen tarvittavaa aikaa. Testidatassa ero on pienempi. Neuronien lukumäärän kasvattaminen paransi hiukan oppimistulosta testidatassa, kunnes neuroverkko alkoi ylisovittamaan ja testidatan luokittelutulos alkoi heikkenemään.

Kuvassa 3 näkyy opetusdatan kuvien määrän vaikutus neuroverkon oppimistulokseen. Testauksessa neuroverkon rakenne oli samankaltainen kuin piilokerroksessa tapauksessa. Piilokerroksessa oli 30 neuronia. Opetusdataan valittiin ensiksi 100 kuvaa ja sitten 5000 kuvaa. Kuvien määrän lisääminen hidasti odotettavasti oppimista, koska algoritmin aikavaativuus on suoraan verrannollinen kuvien määrään. Kun kuvia on vähän, neuroverkko oppii luokittelemaan opetusdatan sadan prosentin tarkuudella, mutta testidatan kuvia neuroverkko ei luokittele kovinkaan hyvin. Kuvien määrän lisääminen saa neuroverkon tunnistamaan aiemmin näkemättömät kuvat suuremmalla tarkuudella. Lisäksi opetusdatan oppimistulos on luottettavampi mittari neuroverkon kyvystä tunnistaa kuvia, kun kuvia on enemmän.



Kuva 3: