

Projektin testikattavuusraportti näyttää tällä hetkellä (21.6.2025) seuraavalta:

Name	Stmts	Miss	Branch	BrPart	Cover	Missing
src/mnist_loader.py	29	0	6	0	100%	
src/neural_network.py	80	1	12	2	97%	116->113, 142
TOTAL	109	1	18	2	98%	

Olen kirjoittanut omat testinsä MNIST-datan luentaan keskittyvälle moduulille sekä omat testinsä varsinaiselle neuroverkko-luokalle.

MNIST-tietokannan käsittelevän moduulin testit testaavat sitä, että kuvia sekä niiden tunnisteita ladataan oikeat määrät ja että tunnisteet ovat verkolle järkevät, eli sisältävät yhteenä 10 alkiota, joista yksi on ykkönen ja loput nollia.

Varsinaiselle verkolle olen kirjoittanut jonkin verran yksikkötestejä: Testaan sigmoid-funktiota ja sen derivaattaa parilla eri arvolla minkä lisäksi testaan, että softmax-funktion ulostulo on vakio, jos sen parametriksi annettavaa vektorin jokaiseen komponenttiin lisätään sama vakio.

Näiden lisäksi on erinäisiä vastavirta-algoritmin testaukseen liittyviä testejä, jotka ovat aika lailla samoja kuin kurssimateriaalissa linkitetty neuroverkkojen testausta käsittelevässä artikkelissa kuvatut testit:

Ensimmäisessä testissä tarkistetaan, että verkko ylisovittuu pieneen määrään testidataa. Tässä testissä otetaan MNIST-tietokannasta neljä ensimmäistä kuvaa ja syötetään nämä verkolle niin monta kertaa, kunnes se luokittelee näistä jokaisen oikein. Iteraatioiden ylärajaksi on asetettu 30.

Toisessa testissä tarkastetaan koulutuksen aikana tapahtuvaa virhefunktion arvonn laskua. Testissä valitaan MNIST-datasta satunnaisotos (128 kuvaa), suoritetaan vastavirta-algoritmi tällä aineistolla 10 kertaa ja lopuksi tarkistetaan, että luokitteluvirhe todella pienenee. Tätä testiä varten käytän melko pientä parametrin `learn_rate` arvoa, jotta luokitteluvirhe ei lähde oskilloimaan, mikä on mahdollista pitkällä askelpituuksilla.

Kolmannessa testissä tarkistetaan, että kaikki verkon parametrit todella päivittyvät. Tätä varten otetaan jälleen 128 harjoituskuva satunnaisotos ja suoritetaan vastavirta-algoritmi yhteensä 10 kertaa. Jokaisen kierroksen jälkeen tarkistetaan, että gradientit ovat nolasta poikkeavia. Samankaltaisia testejä on yhteensä kaksi, joista toisessa käytetään `network`-luokan koulutukseen tarkoitettua korkeamman tason metodia. Toisessa testissä taas testi kutsuu suuremmin gradientteja laskevia metodeja. Testejä on kaksi testikattavuuden parantamiseksi.

Viimeisessä testissä testataan, että tällä neuroverkkototeutuksella on ylipäänsä mahdollista päästä hyvään tunnistustarkkuuteen kouluttamalla yhtä neuroverkkoa koko koulutusdatalla. Muista testeistä poiketen tässä testissä luodaan verkko hieman pienemmällä määrällä piilokerroksia, jotta testi olisi hieman nopeampi.

Testit voi suorittaa itse asentamalla ensin riippuvuudet, aktivoimalla virtuaaliympäristö ja suorittamalla sitten komento `coverage run --branch -m`

pytest src. Katso tarkemmat ohjeet readme.MD-tiedostosta.