

Hands on 3 (Konenäkö ja kuva-analyysi 2010)

Tarkoitus oli että yhdistettäisiin hermoverkot, Java ja ImageJ. Nyt kävi niin että yhdistetään lähinnä hermoverkot ja Java. ImageJ on kyllä nähtävästi helppo liittää oman Java-ohjelman osaksi. Sen päätoiminnot ovat `ij.jar` -paketissa, jossa on jopa niin kattava rajapinta etten ehtinyt siitä tähän hätään löytämään sopivia kaluja yhdisteltäviksi luokittelun demonstroimiseen. Niinpä laskeudumme tässä demossa suoraviivaisen ohjelmoinnin äärelle. Kirjoittelin tehtäviä aika paljon; tekee sitten minkä ehtii.

Vaihe 1.

Hae seuraavasta nettiosoitteesta esimerkkiohjelmat ja katso että saat käännettyä ja ajettua niitä mikroluokassa:

http://users.jyu.fi/~nieminen/konenako2010/java_example/

Kokeilin eilen että Maceissa ja Linuxeissa on Java-kääntäjät ja että Hello World toimii. Siinä vaiheessa minulla ei kuitenkaan ollut vielä MLP-neuroverkkokoodia kokeiltavana. Saatamme joutua säätämään.

Vaihe 2.

Simppli ja nopeasti pyörivä esimerkki, Fisherin Iirikset:

- Aja `DemoIris` luokka, tutki tulostetta. Mitä voit päätellä?

Ei muuta kuin häksäilemään koodia:

-Katso missä kohtaa hermoverkon arkkitehtuuri määritellään, missä taas parametroidaan opetusalgoritmi. Kokeile hiukan erilaisia parametreja ja tutki vaikutusta.

Teoriapuoli: Iris-datassa on kolme luokkaa ja mitattuja piirteitä on neljä. Hahmottele rakennekuva hermoverkosta, jossa olisi viisi "piiloneuronia" yhdellä kerroksella syötteen ja ulostulon välillä.

Vaihe 3.

Varsinainen kuvantunnistus:

- Hae (jos et vielä hakenut) em. nettisivulta tiedosto `mnist_reduced.zip` ja avaa se.

Kyseessä on 1000 ensimmäistä bittikarttakuvaa MNIST numerontunnistusdatasta. Rajasin dataa, koska mikroluokan koneissa on pienet quodat ja ison datan laskenta myös kestäisi pidempään.

- Aja `DemoMNIST` luokka. Se ei piirteytä kuvia millään tavalla, vaan käyttää pikseliarvoja. Tutki tulostetta. Havaintoja?

Esim. Onko tämä hyvä luokitin? Mitkä luokat menevät sekaisin keskenään? (Luokat 1-9 ovat numerot '1'-'9' ja luokka 10 on numero '0').

Saako luokittinta paremmaksi säätämällä parametreja?

Vaihe 4.

Piirteytys:

Mahdot havaita edellisessä kohdassa ainakin sen että 28x28 ulotteisessa avaruudessa (MNIST) laskeskelu on hitaampaa kuin 4-ulotteisessa. Itse asiassa MLP:n optimointi tapahtuu vielä luokittimen syöteavaruutta paljon isommassa paikassa.

- Laske käsin tai apuvälineellä tarkasti (tai häksää hermoverkko-ohjelmani tulostamaan lukumäärä) kuinka monta neuronien välistä synapsia eli painokerrointa on yhteensä hermoverkossa, jonka koko on 784-20-20-10 eli 748 syötettä, kaksi 20:n neuronin piilokerrosta ja 10-ulotteinen ulostulo. Huomioi bias-termit. Tämä on sen avaruuden ulottuvuus, jossa optimointialgoritmi koettaa kiivetä kustannusfunktiota alamäkeen.

Piirteytys suhteellisen pienellä lukumäärällä mahdollisimman kuvaavia piirteitä lienee motivoitu. Tutki koodia `DemoFeatureExtraction`. Olen siihen laittanut kolme erilaista varsin alkeellista piirrelaskua. Tavoite on, että itse toteuttaisit fiksumpia tapoja. Katsele kuitenkin ensin miten luokittelu sujuu, kun pääohjelmassa kommentoidaan päälle ja pois erilaisia piirrektoreita.

Vaihe 5.

Suuri numerontunnistuskilpa:

- Ohjelmoi uusi piirteenirroittajaluokka joka toimii kuten esimerkin luokat eli perii `AbstractDumbGrayscaleFeatureExtractor` -luokan ja toteuttaa `extractFromGrayscale` -metodin. Tyyli on muutoin vapaa. Piirteesi avulla on tavoitteena päästä parempaan tulokseen kuin pelkkää pikselidataa käyttämällä, tai ainakin parempaan kuin esimerkikoodi tai vieruskaverin tuotos :).
- Esim. luennoilla nähtyä Fourier-deskriptoria voi koettaa miettiä. LBP taitaa olla enemmänkin tekstuuripiirre, joka soveltuu sellaisten kuvien tunnistamiseen, joissa on pintamateriaaleja.

Vaihe 6.

Eteenpäin kohti uusia haasteita.

- Katso MNISTin kotisivulta luokittelutuloksia (koko datassa siis 60000 opetuskuva ja 10000 testikuva; tämän hands-on demon pieni datasetti ei ole suoraan vertailukelpoinen):
<http://yann.lecun.com/exdb/mnist/>
Mihin tarkkuuteen on parhaimmillaan päästy ja millä luokittimella?
- Teoriaa: Pohdi hermoverkkoa, jossa on kaksi syötettä ja yksi neuroni jolla yksi ulostulo (hyperbolinen tangenti aktivaatiofunktiona). Piirrä paperille koordinaatistoon kahdesta painokertoimesta muodostuva vektori ja raja, jonka eri puolilla ulostulon etumerkki vaihtuu. Muista bias-termi. Kuvaile kuvan geometriaa. Mieti, miten toisen ja useamman neuronin lisääminen vaikuttaisi.
- Etsi Internetistä testikuvajoukkoja, joihin olisi kiva kokeilla algoritmeja jatkossa.