

Huom: Voit saada tästä harjoituskerrasta max. 6 pistettä.

Tehtävä 1. Lähimmän naapurin luokitin.

Kokeile lähimmän naapurin luokittelijaa käsinkirjoitettujen numeroiden tunnistamiseen. Tiedosto `'mnist-x.data'` sisältää yhteensä 6000 kuvaa, yksi kuva jokaisella tiedoston rivillä, ja jokainen kuva on 28×28 pikseliä (eli jokaisella rivillä siis $28 \times 28 = 784$ numeroa). Jokainen pikseli on joko musta (0) tai valkoinen (1). Tiedosto `'mnist-y.data'` sisältää näitä kuvia vastaavat luokat (0-9). Tiedostossa `'MakeBMP.java'` löytyy datan lukemiseen ja kuvien esittämiseen tarkoitettut apufunktiot. Voit ajaa esimerkkikoodin komennolla

```
java MakeBMP mnist-x.data mnist-y.data
```

- (1 piste). Näytä ruudulla ensimmäiset 100 merkkiä ja printtaa vastaavat 100 luokkaa. Nämä pitäisi vastata toisiaan. Verifioi tällä tavoin että datan lukeminen on onnistunut.
- (1 piste). Olkoon ensimmäiset 5000 merkkiä opetusdataa, ja loput 1000 merkkiä testidataa. Toteuta lähimmän naapurin luokittelija, ja mittaa testivirhe.
- (1 piste). Kokeile toimiiko k -lähimmän naapurin luokittelija (jossa $k > 1$) paremmin vain huomommin kuin (b)-kohdassa käytetty $k = 1$ -lähimmän naapurin luokittelija.

Vinkki: Jos ohjelma toimii hitaasti, kannattaa testausvaiheessa kokeilla pienempää, esim. 500-1000 kuvan otosta aineistosta ja käyttää koko 6000 kuvan aineistoa vasta lopuksi.

Tehtävä 2. Robotiikkaa.

Tämä tehtävä suoritetaan osin “pajamalla” soveltaen, eli ensimmäinen osa (1 piste) tehdään normaaliin tapaan kotona ja se merkitään tehdyksi laskuharjoitusten aluksi. Toinen osa (3 pistettä) suoritetaan osallistumalla yhteen tai useampaan laskuharjoitussessioon. Huomaa että toista osaa on silti syytä valmistella mahdollisimman paljon etukäteen, jotta itse sessiosta saa enemmän irti.

Tehtävässä käytetään sitä varten valmiiksi rakennettua Lego Mindstorms -robottia, jonka rakennetta ei ole tarkoitus muuttaa. Robotissa on pyörät, valosensori, joka osoittaa kohti lattiaa pyörien akselista n. 10cm edempänä sekä ultraäänisensori, joka osoittaa suoraan vasemmalle n. 10cm robotin keulasta taaksepäin. Robotin rakenteesta saa paremman käsityksen katsomalla oheista kuvaa (kuva 1).

- a) (1 piste) Toteuta valmista runkoa (`HelloWorld.java`; kuten nimi kertoo) muokkaamalla ohjelma, jonka avulla robotti seuraa lattiaan merkittyä viivaa, kunnes viiva loppuu.
- b) (3 pistettä) Toteuta ohjelma, jonka avulla robotti etsii ultraäänisensorien avulla “taskun” eli esim. oviaukon, jonka koko on vähintään $40\text{cm} \times 40\text{cm}$, minkä löydettyään robotti “pysäköi” taskuun: ajaa taskun ohi, peruuttaa, kääntyy, ajaa taskuun ja pysähtyy siten, että se on kokonaisuudessaan taskussa.

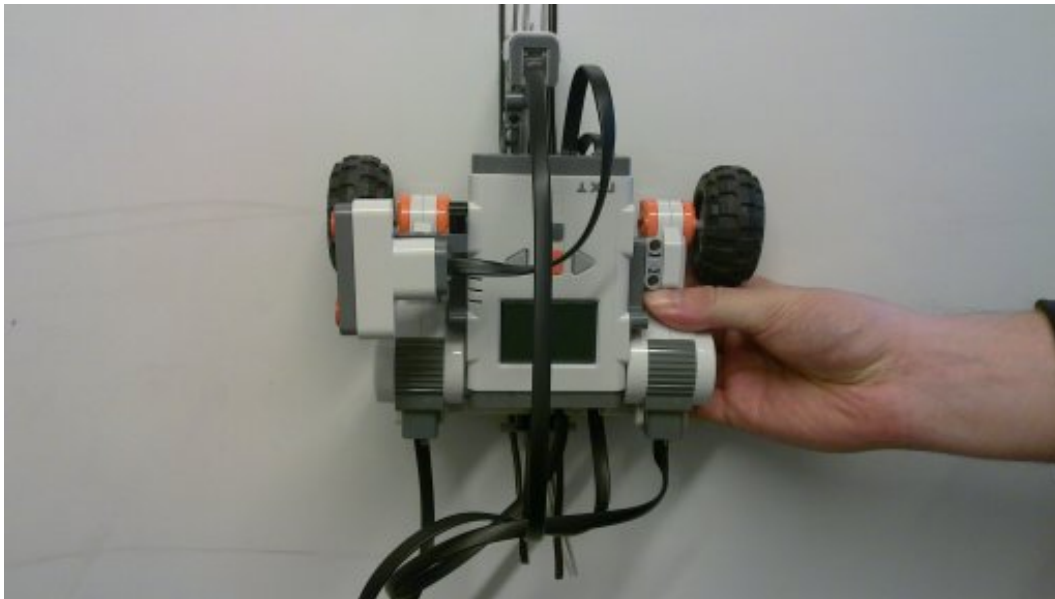


Figure 1: Tehtävässä käytettävä robotti.