1 perceptron, mnist set

Ensimmäisenä tehtävä oli luoda perceptron algoritmi, jota sitten käytetään käsinkirjoitettujen numeroiden luokitteluun. Tässä tapauksessa tarkoituksena oli luokitella ainoastaan 1 ja 0. Tein toteutuksen Matlabilla. Tein numeroiden lataamiseen oman scriptin, joka valitsee numeroista vain 1 ja 0, ja jakaa pikselidatan sekä oikeat luokat omiin muuttujiinsa. Tämän lisäksi on itse perceptron funktio omassa tiedostossaan, joka hoitaa luokittelijan opettamisen. Sitten on vielä classifier funktio, jolla voi luokitella käyttäen aiemmin luotua luokittelijaa.

Algoritmi on kaikessa yksinkertaisuudessaan seuraava. Valitaan luokittelijaksi alustusvaiheessa ensimmäinen luokiteltava alkio. Tämän jälkeen käydään yksi kerrallaan alkioita läpi ja verrataan niitä luokittelijaan. Jos luokittelija luokittelee alkion oikein, ei tehdä mitään. Jos luokittelija luokittelee alkion väärin, lisätään kyseinen alkio luokittelijaan. Luokittelijan vektori tällöin kääntyy kohti sellaista vektroia joka olisi luokitellut kyseisen alkion oikein. Tätä jatketaan niin kauan, kunnes käydään kaikki luokiteltavat alkiot läpi ilman että luokittelijaa tarvitsee muuttaa. Jos aineisto on lineaarisesti eroteltavissa, niin sopiva luokittelija löytyy aina.

Testatessa ensimmäisenä tein 2-uloitteisia testiaineistoja, jotta tiedän toimiiko luokittelija. Kaikissa seteissä pisteitä oli senverran vähän että jos luokat olivat eroteltavissa, niin algoritmi konvergoi parissa iteaatiossa. Seuraavalla sivulla on muutamia kuvia, joista näkee minkälaisia ryhmiä luokittelija sai luotua. Kohtasin pieniä ongelmia siinä miten olisin saanut MatLabissa esitettyä pisteet omina luokkinaan, mutta tässä tapauksessa onneksi pisteitä on niin vähän, että on melko selvää mitkä pisteet kuuluvat mihinkin luokkaan. Tarkastin kuitenkin käsin, että saadut tulokset olivat järkeviä.

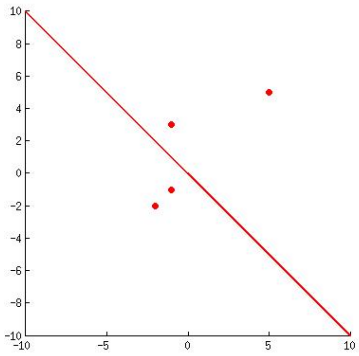


Figure 1: Luokittelijan testi 1

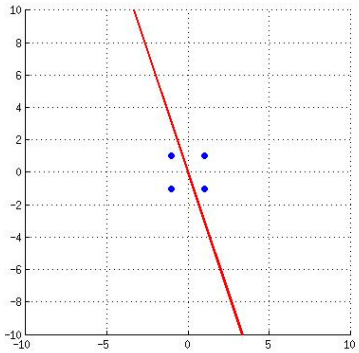


Figure 2: Luokittelijan testi 2

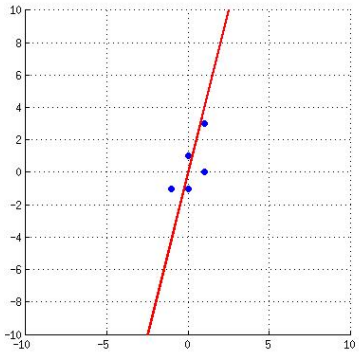
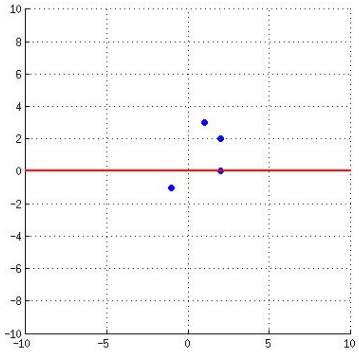


Figure 3: Luokittelijan testi 3, Kaikista toimivista luokista tässä oli pienin marginaali.



2

Figure 4: Pisteet eivät olleet luokiteltavissa, algoritmi ei konvergoinut

Seuraavaksi latasin apufunktiolla 5000 datapistettä mnist aineistosta ja otin sieltä vain 1 ja 0. Opetusaineistoon ja testiaineistoon tuli kumpaankin noin 500 alkiota. Perceptron algoritmi konvergoi toisella iteraatiolla. Tämän jälkeen kun luokittelijan antoi syötteenä testiaineistolle, oli virheprosentti 0%. Tulos ei sinällään ollut kovin yllättävä, koska aineiston dimensio oli todella suuri, sekä 1 ja 0 ovat melko kaukana toisistaan jos asiaa miettii pikselitasolla. Visualisaatio paketti oli joko jotenki rikki tai en osannut vain käyttää sitä, mutta en onnistunut luomaan kuvaa esimerkin mukaisesti yksittäisistä numeroista, enkä luokittelijasta,niinkuin tehtävänannossa pyydettiin. Uskoisin kuitenkin että luokittelija ei juurikaan näytä kummaltakaan numerolta, vain jonkinlaiselta sekoitukselta kummastakin.

Koodin testaus onnistuu esimerkiksi seuraavanlaisilla komennoilla.

- - [testNumbers, trainNumbers, testclasses, trainclasses] = helpLoadingNumbers()
- classifier = perceptron(trainNumbers,testClasses)
- classify(classifier, testNumbers, testClasses)

2 Bayes classifier, news set

Toisessa tehtävässä oli tarkoitus luoda yksinkertainen Bayes luokittelija, joka luokittelee dokumentteja kuulumaan erilaisiin uutisryhmiin perustuen dokumentin sisältöön. Ensimmäiseksi luotiin harjoitus setti, jonka avulla bayes luokittelijaa lähdetään opettamaan. Settiin valittiin jokaisesta uutisryhmästä 90% dokumenteista. Tämän jälkeen jokaiselle sanalle laskettiin todennäköisyys kuulua tiettyyn uutisryhmään sen perusteella kuinka monessa uutisryhmän dokumentissa kyseinen sana esiintyi. Kun kaikki dokumentit oli käyty läpi, oli luokittelija valmis. Luokittelijaa voi käyttää siten, että katsoo dokumentin sanojen todennäköisyydet kuulua uutisryhmiin ja se uutisryhmä joka saa suurimman todennäköisyyden on valittu luokka.

Yllätykseksi Luokittalija toimi todella huonosti. En osaa sanoa johtuiko se jostain implementointivirheestä tai muuta vastaavaa, mutta luokittelija onnistui luokittelemaan oikein kohtalaisella prosentilla vain muutamat luokat: 11,12, 16,17, 18 ja 19. Lopuilla oikein luokiteltujen dokumenttien määrä oli lähellä 0%.

Jos implementaatiossani ei ollut mitään vikaa ja tämä oli odotettava tulos, niin olen vähintäänkin hämmentynyt. Oletin että luokittelija olisi ollut parempi tällaisen aineiston kanssa.

3 Hierarial clustering, movie dataset

Kolmantena tehtävänä oli tehdä clusterointi algoritmi, joka luokittelee elokuvia samoihin luokkiin katsojien antamien suositusten perusteella. Tein ohjelman kotona matlabilla, jossa minulla ei ollut saatavilla statistical toolboxia, joten dendrogrammien tekeminen ei onnistunut, joten jouduin tulkitsemaan kaikki luomani esimerkit ruutupaperille.

Elokuva 1	Elokuva 2	Distance
'Alien (1979)'	Aliens	0,671511628
'Star Trek VI: The Undiscovered Country (1991)'	'Star Trek III: The Search for Spock (1984)'	0,651741294
'Star Trek: The Wrath of Khan (1982)'	20	0,646825397
'Star Trek IV: The Voyage Home (1986)'	21	0,643835616
'Lion King, The (1994)'	'Aladdin (1992)'	0,613970588
'Silence of the Lambs, The (1991)'	19	0,552995392
22	24	0,548387097
'Monty Python and the Holy Grail (1974)'	25	0,538126362
'Young Frankenstein (1974)'	26	0,474285714
'Psycho (1960)'	27	0,469626168
'Fantasia (1940)'	23	0,455555556
28	29	0,451410658
'Babe (1995)'	30	0,416129032
'Godfather: Part II, The (1974)'	31	0,41260745
'Toy Story (1995)'	32	0,408026756
'Muppet Treasure Island (1996)'	33	0,202380952
'Free Willy 2: The Adventure Home (1995)'	34	0,045248869

Figure 5: Complete-link klusteroidut elokuvat

Ensimmäisenä kokeilin algoritmia 5x5 matriisiin, johon olin laskenut käsin sopivia pisteiden etäisyyksiä, niin että niiden oikeellisuus olisi helppo tarkistaa suoraa ruutupaperilta. Tein esimerkin myös siten, että klustereista tulisi tarpeeksi erilaisia riippuen siitä pyydetäänkö algoritmia toimimaan single link vai multilink mallilla. Klusterit näyttivät toimivan juuri niinkuin pitääkin, joten siirryin elokuvien pariin. Etsin elokuvista muutaman selkeästi lasten elokuvan kuten Toy Storyn, 101 dalmatialaista ja niin edelleen. Toisena selkeänä ryhmänä valitsin mukaan Star Trek elokuvia, joiden kuvittelin olevan Jaccrd Coefficient mielessä melko lähellä toisiaan. Samaa genreä mukaillen otin mukaan myös Aliens elokuvat. Muuten valitsin mukaan satunnaisia elokuvia, kuitenkin niin että ne olisivat hieman erilaisia kuin jo valitut elokuvat.

Tälle setille kummallakin tavalla tuotettu tulos oli melko samannäköinen. Vain pienehköjä eroja miten klusterit olivat muodostuneet. Tämä oli toisaalta melko odotettu tulos, koska erot eri elokuvien välillä ovat melko samansuuruisia, joten mitään kovin isoja eroja ei voi odottaa muuttamalla klusterointitaktiikkaa.

'Alien (1979)'	Aliens	0,671511628
'Star Trek VI: The Undiscovered Country (1991)'	'Star Trek III: The Search for Spock (1984)'	0,651741294
'Star Trek IV: The Voyage Home (1986)'	20	0,637168142
'Lion King, The (1994)'	'Aladdin (1992)'	0,613970588
'Star Trek: The Wrath of Khan (1982)'	21	0,575875486
'Silence of the Lambs, The (1991)'	'Monty Python and the Holy Grail (1974)'	0,538126362
19	24	0,469733656
'Fantasia (1940)'	22	0,443223443
'Psycho (1960)'	'Godfather: Part II, The (1974)'	0,391304348
'Young Frankenstein (1974)'	25	0,353211009
26	28	0,323943662
'Babe (1995)'	29	0,301324503
'Toy Story (1995)'	30	0,272357724
23	31	0,25
27	32	0,22181146
'Muppet Treasure Island (1996)'	33	0,093283582
'Free Willy 2: The Adventure Home (1995)'	34	0,013824885

Figure 6: Single-link klusteroidut elokuvat