T-61.231 Hahmontunnistuksen perusteet

Laskuharjoitus 7: 11.11.2002

1. Kun MLP:tä käytetään luokittelutehtävään, ulostuloneuronien lukumäärä on yleensä sama kuin luokkien lukumäärä. Toivottu ulostulo on nolla kaikille paitsi yhdelle neuronille kerrallaan ja jokainen ulostuloneuroni vastaa yhtä luokkaa. Syöte luokitellaan siihen luokkaan, jonka neuroni on aktiivisin.

   Otetaan yksi ulostuloneuroni, jonka ulostulo on $y(x)$ kun verkon syöte on x ja toivottu vaste on d. Tälle yksittäiselle neuronille kustannusfunktio, joka minimoidaan back propagation -algoritmia käyttäen, on seuraavan muotoinen:

   $$J = \frac{1}{N} \sum_{k=1}^{N} \left(y(x^k) - d^k \right)^2 \ ,$$

   missä N on opetusnäytteiden lukumäärä. Jos N on hyvin suuri, kustannusfunktio approksimoi seuraavaa odotusarvoa:

   $$J = E_{x,d}[(y(x) - d)^2] \ .$$

   Osoita, että kustannusfunktion minimoiva ratkaisu on optimaalinen diskriminanttifunktio Bayesin luokittimelle:

   $$y(x) = P(d = 1|x) \ .$$

2. Perceptronin ulostulo on y ja sen syötteet $x_1, ..., x_n$ ovat jatkuva-arvoisia. Neuroni laskee ulostulonsa seuraavan funktion mukaisesti:

   $$y = \tanh(\sum_{i=1}^{n} w_i x_i - \theta) \ .$$

   Neuroni yrittää oppia antamaan halutun ulostulon d syötteillä $x_1, ..., x_n$. Yksi menetelmä tämän aikaansaamiseksi on minimoida funktiota $(y - d)^2$. Kun minimointiin käytetään gradient descent -menetelmää, voidaan osoittaa että gradienttiaskel on muotoa

   $$\Delta w_i = f(y, d) x_i \ .$$

   Johda funktio $f(y, d)$ tässä tapauksessa.

3. Ajatellaan back propagation -algoritmia 2-kerros MLP:ssä, jolla on kaksi neuronia sekä ulostulo, piilo että syötekerroksissa. W_{ij} ovat ulostulokerroksen painot ja Θ_j ovat offset-parametrit missä $j = 1, 2$ on neuronin indeksi ja $i = 1, 2$ piiloneuronin, josta syöte tulee, indeksi. Vastaavasti w_{kl} ja θ_l ovat painot ja offsetit piilokerrokselle. Kaikilla neuroneilla on 'logsig'-aktivaatiofunktio.

   Johda back propagation -algoritmi kaikkien parametrien päivittämiseksi. Oletetaan käyettävän on-line-oppimista, mikä tarkoittaa että verkko oppii parametrit välittömästi jokaisen input-output -parin jälkeen.

4. Osoita että jos kustannusfunktio, jonka monikerros-perseptroni optimoi, on ristikkäisentropia (cross-entropy)

$$J = -\sum_{i=1}^{N}\sum_{k=1}^{k_L} y_k(i)\ln\frac{\hat{y}_k(i)}{y_k(i)}$$

ja aktivaatifunktio on sigmoidi $f(x) = \frac{1}{1+\exp(-ax)}$, silloin gradientista

$$\delta_j^L(i) = \frac{\partial\mathcal{E}(i)}{\partial v_j^L(i)}$$

tulee $\delta_j^L(u) = a(1-\hat{y}_j(i))y_j(i)$. *(Theodorodis 4.6, p. 130)*

5. Toista edellinen ongelma softmax-aktivaatiofunktiolle

$$\hat{y}_k(i) = \frac{e^{v_k^L}}{\sum_{k'=1}^{L} e^{v_{k'}^L}} \ .$$

(Theodorodis 4.7, p. 130 ; huomaa että kirjan tehtävässä on (todennäköisesti) virhe (vastaus $\hat{y}_j(i)y_j(i) - y_j(i)$))

6. Seuraava malli oppimisparametri μ:n adaptointiin on esitetty C. Darken ja J. Moody (1991):n toimesta:

$$\mu = \mu_0\frac{1}{1+\frac{t}{t_0}}$$

Varmista, että riittävän suurille t_0:n arvoille (eg. $300 \leq t_0 \leq 500$) opetusparametri on lähes vakio varhaisissa vaiheessa (pienillä t:n arvoilla) ja vähenee kääntäen verrannollisesti t:hen verrattuna suurilla arvoilla. Ensimmäistä vaihetta kutsutaan “etsintävaiheeksi” (*search phase*) ja jälkimmäistä “konvergenssivaiheeksi” (*convergence phase*). Pohdi tällaisen järjestelyn perusteluja. *(Theodorodis 4.16, p. 131)*