ImageRecognition toteutus

Simo Korkolainen

12. kesäkuuta 2016

Projektin tarkoituksena on tehdä ohjelma, joka opettaa neuroverkon tunnistamaan kuvia backpropagation-algoritmin avulla. Neuroverkon opetuksessa verkon painoja muutetaan liikuttamalla niitä virhefunktion gradientin vastaiseen suuntaan, kunnes virhefunktio on minimoitunut ja neuroverkko on oppinut tunnistamaan kuvat. Derivoinnin ketjusääntöön perustuva backpropagation-algoritmi mahdollistaa gradientin nopean laskemisen.

1 Neuroverkko

Neuroverkko koostuu kerroksista, joissa on neuroneita. Olkoon L neuroverkon kerroksien lukumäärä. Olkoon l_k kerroksen $k = 1, ..., L$ neuronien lukumäärä. Merkitään kerroksen k aktivaatiota vektorina $z_k \in \mathbb{R}^{l_k}$. Ensimmäisen kerroksen aktivaatio z_1 on neuroverkon syöte ja viimeisen kerroksen aktivaation z_L on neuroverkon antama tuloste.

Jokaisen kerroksen $k > 1$ aktivaation voidaan ajatella laskettavan parametrisoidun funktion $f_k : \mathbb{R}^{l_{k-1}} \times A_k \to \mathbb{R}^{l_k}$ avulla. Tässä A_k on verkon kerroksien $k-1$ ja k yhteyksien painoina toimivien parametrien joukko. Kerroksen aktivaatio lasketaan rekursiivisesti kaavan

$$z_k = f(z_{k-1}, a_k)$$

avulla, missä $a_k \in A_k$. Parametrit $a_1, ..., a_L$ on tarkoitus oppia backpropagation-algoritmia käyttäen.

2 Aineisto ja kokonaisvirhe

Olkoon $x_1, x_2, \ldots, x_n \in \mathbb{R}^{l_1}$ neuroverkon opetussyötteitä ja $y_1, y_2, \ldots, y_n \in \mathbb{R}^{l_L}$ syötteitä vastaavia tavoitetuloksia. Olkoon $E : \mathbb{R}^{l_L} \times \mathbb{R}^{l_L} \to \mathbb{R}$ virhefunktio. Merkitään syötteen x_j aiheuttamaa aktivaatiota kerroksessa k merkinnällä z_k^j, jolloin syötekerroksessa pätee $z_1^j = x_j$. Määritellään neuroverkon kokonaisvirhe kaavalla

$$E_{tot} = \sum_{j=1}^{n} E(z_L^j, y_j).$$

Backprobagation-algoritmin tarkoituksena on valita parametrien $a_1, ..., a_L$ arvot siten, että kokonaisvirhe E_{tot} minimoituu.

Jos neuroverkkoa käytetään syötteiden luokittelemiseen erillisiin luokkiin $1, ..., S = l_L$, viimeisen kerroksen $z_L = (z_{L1}, \dots, z_{LS})$ aktivaation z_{Lm} tulkitaan tarkoittavan todennäköisyyttä, että syöte x_j kuuluu luokkaan m. Halutuksi tulokseksi valitaan $y_{jm} = 1$, kun x_j kuuluu luokkaan m ja $y_{jm} = 0$ muulloin. Luokittelun tapauksessa on luontevaa käyttää viimeisessä kerroksessa softmax-funktiota $f_L : \mathbb{R}^S \times A_L \to (0,1)^S$, missä

$$f_{Lm}(z_{L-1}) = \frac{e^{\sum_{i=1}^{l_{L-1}} a_{mi} z_{(L-1)i}}}{\sum_{k=1}^{S} e^{\sum_{i=1}^{l_{L-1}} a_{ki} z_{(L-1)i}}}.$$

Softmax-funktion kuva-alkion komponenttien summa on yksi.

Luokittelussa virhefunktiona käytettään yleensä logloss-virhettä. Tässä tapauksessa virhefunktio määritellään kaikilla $s \in \{0,1\}^S$ ja $t \in (0,1)^S$ kaavalla

$$E(s,t) = \sum_{m=1}^{S} s_m \log(t_m).$$

3 Derivaatta ja ketjusääntö

Tässä kappaleessä käytetään yleisiä differentiaalilaskennan merkintöjä aiemmista merkinnöistä huolimatta. Olkoon funktio $f = (f_1, \dots, f_n) : \mathbb{R}^m \to \mathbb{R}^n$ derivoituva. Derivaatalla pisteessä $x_0 \in \mathbb{R}^m$ tarkoitetaan matriisia

$$f'(x_0) = \begin{bmatrix} \frac{\partial f_1(x_0)}{\partial x_1} & \dots & \frac{\partial f_1(x_0)}{\partial x_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n(x_0)}{\partial x_1} & \dots & \frac{\partial f_n(x_0)}{\partial x_m} \end{bmatrix},$$

missä osittaisderivaatat $\frac{\partial f_i(x_0)}{\partial x_j}$ on määritelty kaavalla

$$\frac{\partial f_i(x_0)}{\partial x_j} = \lim_{h \to 0} \frac{f_i(x_0 + h e_j) - f_i(x_0)}{h}.$$

Olkoon lisäksi funktion $g : \mathbb{R}^n \to \mathbb{R}^p$ derivoituva. Tällöin yhdistetty kuvaus $g \circ f : \mathbb{R}^m \to \mathbb{R}^p$ on myös derivoituva ja yhdistetyn kuvauksen derivaatan pisteessä x_0 on mahdollista laskea kaavalla

$$(g \circ f)'(x_0) = g'(f(x_0))g'(x_0).$$

4 Backpropagation-algoritmi

Backpropagation algoritmi perustuu edellä mainittuun differentiaalilaskennan ketjusääntöön. Merkitään

$$\frac{\partial z_{i+1}}{\partial z_i} = \begin{bmatrix} \frac{\partial z_{(i+1)1}}{\partial z_{i1}} & \dots & \frac{\partial z_{(i+1)1}}{\partial z_{il_i}} \\ \vdots & \ddots & \vdots \\ \frac{\partial z_{(i+1)l_{i+1}}}{\partial z_{i1}} & \dots & \frac{\partial z_{(i+1)l_{i+1}}}{\partial z_{il_i}} \end{bmatrix},$$

$$\frac{\partial z_i}{\partial a_i} = \begin{bmatrix} \frac{\partial z_{i1}}{\partial a_i} \\ \vdots \\ \frac{\partial z_{il_{i+1}}}{\partial a_i} \end{bmatrix}, \text{ ja}$$

$$\frac{\partial E_{tot}}{\partial z_i} = \begin{bmatrix} \frac{\partial E_{tot}}{\partial z_{i1}} & \dots & \frac{\partial E_{tot}}{\partial z_{il_i}} \end{bmatrix}.$$

Ketjusäännön perusteella eri kerrosten aktivaatioiden virhegradientti voidaan laskea rekursiivisesti

$$\frac{\partial E_{tot}}{\partial z_i} = \frac{\partial E_{tot}}{\partial z_{i+1}} \frac{\partial z_{i+1}}{\partial z_i}$$

Eli, jos tiedämme ylemmän kerroksen virhegradientin $\frac{\partial E_{tot}}{\partial z_{i+1}}$ arvon, voimme laskea alemman kerroksen virhegradientin kertomalla gradienttia $\frac{\partial E_{tot}}{\partial z_{i+1}}$ oikealta matriisilla $\frac{\partial z_{i+1}}{\partial z_i}$. Parametrien virhegradientti voidaan laskea vastaavasti käyttäen aktivaatioiden virhegradienttia

$$\frac{\partial E_{tot}}{\partial a_i} = \frac{\partial E_{tot}}{\partial z_i} \frac{\partial z_i}{\partial a_i}$$

Seuraavaksi on esitettynä pseudokoodi backpropagation-algoritmille.

Algorithm 1 Backpropagation-algoritmi

procedure BACKPROPAGATION(x, y, α)

    for $j = 1, \ldots, n$ **do**

        FORWARD PROPAGATION(x_j)
        BACKWARD PROPAGATION(y_j, α)

    end for

end procedure

procedure FORWARD PROPAGATION(x)

    Aseta syöte $z_1 = x$.
    for jokaiselle kerrokselle $k = 2, \ldots, L$ **do**
        Laske aktivaatio $z_k := f_k(z_{k-1}, a_k)$
    end for

end procedure

procedure BACKWARD PROPAGATION(y, α)

    Laske virhegradientti $\frac{\partial E}{\partial z_L}$ arvojen y ja z_L avulla
    for jokaiselle kerrokselle $k = L - 1, \ldots, 2$ **do**

        Laske ensiksi $\frac{\partial z_{k+1}}{\partial z_k}$ arvon z_k avulla
        Laske toiseksi $\frac{\partial E}{\partial z_k} := \frac{\partial E}{\partial z_{k+1}} \frac{\partial z_{k+1}}{\partial z_k}$
        Laske kolmanneksi $\frac{\partial E_{tot}}{\partial a_i} := \frac{\partial E_{tot}}{\partial z_i} \frac{\partial z_i}{\partial a_i}$

        Päivitä painot $a_i := a_i - \alpha \frac{\partial E_{tot}}{\partial a_i}$

    end for

end procedure