Neuronnät (neurala nätverk)

Göran Bergqvist, MAI

Neuronnät = neuralt nätverk =

= (flervariabel-)funktion $f(\bar{x})$ som är ...

... en sammansättning av flera "enkla" funktioner

$$y = f(\bar{x}) = f_L(\dots \bar{f}_2(\bar{f}_1(\bar{x}))\dots) = (f_L \circ \cdots \circ \bar{f}_2 \circ \bar{f}_1)(\bar{x})$$

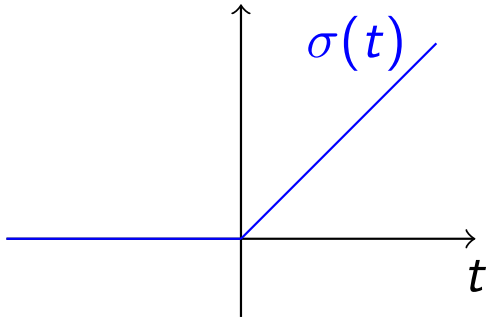
$$\bar{x} \mapsto \underbrace{\bar{x}_1}_{=\bar{f}_1(\bar{x})} \mapsto \underbrace{\bar{x}_2}_{=\bar{f}_2(\bar{x}_1)} \mapsto \dots \mapsto \underbrace{\bar{x}_{L-1}}_{=\bar{f}_{L-1}(\bar{x}_{L-2})} \mapsto \underbrace{x_L}_{=f_L(\bar{x}_{L-1})} = y$$

L = antal lager i neuronnätet

$\bar{f}_j$ "enkla": lager j oftast matrismultiplikation (linjär), ev addition av konstant vektor (affin), följt av olinjär men enkel $\bar{\sigma}$:

$$\bar{x}_j = \bar{f}_j(\bar{x}_{j-1}) = \bar{\sigma}(A_j\bar{x}_{j-1} + B_j)$$

En olinjär funktion och dess derivata:



$$\sigma(t) = (t)_+ = \begin{cases} t\ , & t \geq 0 \\ 0\ , & t < 0 \end{cases} \quad \text{och } H(t) = \sigma'(t) = \begin{cases} 1\ , & t > 0 \\ 0\ , & t < 0 \\ \textit{ej def}\,, & t = 0 \end{cases}$$

För vektorer:

$$\bar{\sigma}(\bar{t}) = (\bar{t})_+ = \begin{pmatrix} t_1 \\ \vdots \\ t_n \end{pmatrix}_+ = \begin{pmatrix} (t_1)_+ \\ \vdots \\ (t_n)_+ \end{pmatrix}, \text{ t ex } \quad \bar{\sigma} \begin{pmatrix} -1 \\ 2 \\ 3 \\ -4 \end{pmatrix} = \begin{pmatrix} -1 \\ 2 \\ 3 \\ -4 \end{pmatrix}_+ = \begin{pmatrix} 0 \\ 2 \\ 3 \\ 0 \end{pmatrix}$$

$\bar{\sigma}$ nollställer negativa komponenter

Exempel på neuronnät med 3 lager, $f(\bar{x}) = (f_3 \circ \bar{f}_2 \circ \bar{f}_1)(\bar{x})$:

$$A_1 = \begin{pmatrix} 2 & -1 \\ 1 & 1 \end{pmatrix}, B_1 = \begin{pmatrix} -1 \\ 2 \end{pmatrix}, A_2 = \begin{pmatrix} 1 & 1 \\ 2 & 0 \\ -3 & 1 \end{pmatrix}, B_2 = \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix}, A_3 = \begin{pmatrix} 1 & -1 & 1 \end{pmatrix}$$

$\bar{x} \in \mathbb{R}^2, \quad \bar{x}_1 = \bar{f}_1(\bar{x}) = \bar{\sigma}(A_1\bar{x} + B_1) = (A_1\bar{x} + B_1)_+ \in \mathbb{R}^2$
$\bar{x}_2 = \bar{f}_2(\bar{x}_1) = (A_2\bar{x}_1 + B_2)_+ \in \mathbb{R}^3, \quad y = x_3 = f_3(\bar{x}_2) = A_3\bar{x}_2 \in \mathbb{R}$

$$\text{T ex } \bar{x} = \begin{pmatrix} 2 \\ 1 \end{pmatrix} \Rightarrow \bar{x}_1 = (A_1 \begin{pmatrix} 2 \\ 1 \end{pmatrix} + B_1)_+ = \begin{pmatrix} 2 \\ 5 \end{pmatrix}_+ = \begin{pmatrix} 2 \\ 5 \end{pmatrix} \Rightarrow$$

$$\Rightarrow \bar{x}_2 = (A_2 \begin{pmatrix} 2 \\ 5 \end{pmatrix} + B_2)_+ = \begin{pmatrix} 7 \\ 5 \\ -2 \end{pmatrix}_+ = \begin{pmatrix} 7 \\ 5 \\ 0 \end{pmatrix} \Rightarrow y = x_3 = A_3 \begin{pmatrix} 7 \\ 5 \\ 0 \end{pmatrix} = 2$$

$$\text{Alltså, } f(\begin{pmatrix} 2 \\ 1 \end{pmatrix}) = (f_3 \circ \bar{f}_2 \circ \bar{f}_1)(\begin{pmatrix} 2 \\ 1 \end{pmatrix}) = (f_3 \circ \bar{f}_2)(\begin{pmatrix} 2 \\ 5 \end{pmatrix}) = f_3(\begin{pmatrix} 7 \\ 5 \\ 0 \end{pmatrix}) = 2$$

Neuronnät $\quad y = f(\bar{x}) = (f_L \circ \cdots \circ \bar{f}_2 \circ \bar{f}_1)(\bar{x})$

Lager j : $\quad \bar{f}_j(\bar{x}_{j-1}) = \bar{\sigma}(A_j \bar{x}_{j-1} + B_j) = (A_j \bar{x}_{j-1} + B_j)_+$

Hur bestämma ett neuronnät för en viss tillämpning?
För insatt $\bar{x}$ vill man få ut bra y. Hur bestämma alla matriser i nätet ?

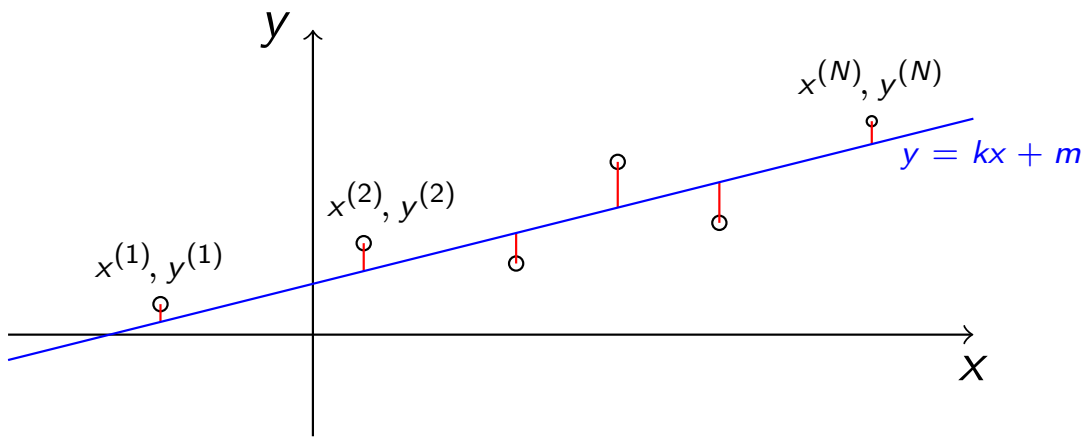
Utnyttja att man har träningsdata:

för N punkter $\bar{x}^{(1)}, \ldots, \bar{x}^{(N)}$ (indata) vet man
vad värdena $y^{(1)}, \ldots, y^{(N)}$ (utdata) ska vara

Anpassa matriserna A_j och B_j i alla $\bar{f}_j$ till dessa träningsdata.

Kallas övervakad inlärning. Man "lär" eller "tränar upp" nätet.
Djupinlärning om många lager.

Kom ihåg: Minstakvadrat-anpassning av linje $y = kx + m$ (dvs k och m) i $\mathbb{R}^2$ till N datapunkter :



Sök k och m så att $\sum_{r=1}^{N}((kx^{(r)} + m) - y^{(r)})^2 = \ell(k, m)$ minimeras $\Rightarrow$ Lös

$$\nabla\ell(k,m) = (\ell'_k, \ell'_m) \underset{\substack{\nabla(g^2)\\ =2g\nabla g}}{=} 2\sum_{r=1}^{N}(kx^{(r)}+m-y^{(r)}) \underbrace{\nabla(kx^{(r)} + m - y^{(r)})}_{=(x^{(r)},1)} = (0,0)$$

Alt.: $\ell(k, m) = ||A\bar{x} - \bar{b}||^2$, där $A = \begin{pmatrix} x^{(1)} & 1 \\ \vdots & \vdots \\ x^{(N)} & 1 \end{pmatrix}, \bar{x} = \begin{pmatrix} k \\ m \end{pmatrix}, \bar{b} = \begin{pmatrix} y^{(1)} \\ \vdots \\ y^{(N)} \end{pmatrix}$

Lin.alg.: k och m fås genom att lösa normalekvationerna $A^T A\bar{x} = A^T\bar{b}$ (minstakvadrat-lösning till olösbara systemet $A\bar{x} = \bar{b}$).

$$y = f(\bar{x}) = (f_L \circ \cdots \circ \bar{f}_2 \circ \bar{f}_1)(\bar{x}) \; ; \quad \bar{f}_j(\bar{x}_{j-1}) = (A_j \bar{x}_{j-1} + B_j)_+$$

Sök A_j, B_j så att $f(\bar{x}^{(k)}) \approx y^{(k)}$ för träningsdata $\{\bar{x}^{(k)}, y^{(k)}\}, 1 \leq k \leq N$

Beteckna alla element i $A_1, B_1, \ldots, A_L, B_L$ med $w_1, \ldots, w_M$.

Minstakvadrat-anpassa matriserna till träningsdata:

Sök $\bar{w} = (w_1, \ldots, w_M)$ så att $\ell(\bar{w}) = \frac{1}{N} \sum_{k=1}^{N} (f(\bar{x}^{(k)}; \bar{w}) - y^{(k)})^2$ minimeras.

Skriver $f(\bar{x}^{(k)}; \bar{w})$ eftersom f beror av matriselementen $\bar{w}$.

Minproblem i $\mathbb{R}^M$: $\nabla \ell(\bar{w}) = \frac{2}{N} \sum_{k=1}^{N} (f(\bar{x}^{(k)}; \bar{w}) - y^{(k)}) \nabla_w f(\bar{x}^{(k)}; \bar{w}) = \bar{0}$

ger stationära punkter, i praktiken omöjligt lösa exakt (M mycket stort).

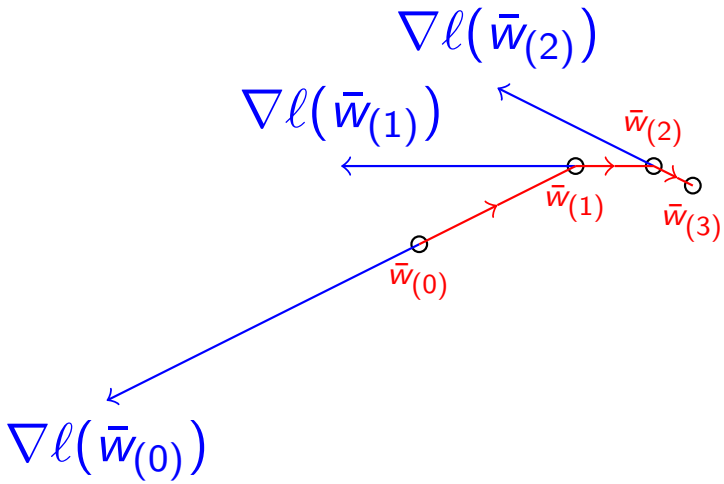
Hitta algoritm för minimering av $\ell(\bar{w})$.

Obs: ∇_w betyder gradient på $\bar{w}$-variablerna (inte på $\bar{x}$). Konstanta faktorn $\frac{1}{N}$ i $\ell(\bar{w})$ påverkar ej resultat men brukar finnas med.

Minimera $\quad \ell(\bar{w}) = \frac{1}{N}\sum_{k=1}^{N}(f(\bar{x}^{(k)};\bar{w}) - y^{(k)})^2$ på $\mathbb{R}^M$

$$\nabla\ell(\bar{w}) = \frac{2}{N}\sum_{k=1}^{N}(f(\bar{x}^{(k)};\bar{w}) - y^{(k)})\nabla_w f(\bar{x}^{(k)};\bar{w})$$

Gradient-algoritm (kallad brantaste lutning i optimering):
Ta startpunkt $\bar{w}_{(0)} \in \mathbb{R}^M$, $\ell(\bar{w})$ avtar snabbast i riktning $-\nabla\ell(\bar{w}_{(0)}) \Rightarrow$ ta steg i den riktningen till ny punkt $\bar{w}_{(1)} = \bar{w}_{(0)} - \eta\nabla\ell(\bar{w}_{(0)}) \in \mathbb{R}^M$. Fortsätt i riktning $-\nabla\ell(\bar{w}_{(1)})$ till ny punkt $\bar{w}_{(2)}$ etc etc. Kontrollera $\ell(\bar{w}_{(k)})$ i varje steg och stanna iterationen när det blivit acceptabelt litet. Då är $\bar{w}$ bestämd, och alltså alla matriser bestämda, och neuronnätet klart att användas.



Hur långa steg η?

Hur beräkna $\nabla\ell(\bar{w})$?

Hur beräkna $\nabla_w f(\bar{x}^{(k)};\bar{w})$?

Kedjeregeln för $\bar{f}(\bar{x}) = (\bar{g} \circ \bar{h})(\bar{x}) = \bar{g}(\bar{h}(\bar{x}))$. Om allt i $\mathbb{R}^2$:

$$\bar{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \mapsto \bar{h}(\bar{x}) = \begin{pmatrix} h_1(x_1, x_2) \\ h_2(x_1, x_2) \end{pmatrix} \mapsto \bar{g}(\bar{h}(\bar{x})) =$$

$$= \begin{pmatrix} g_1(h_1(x_1, x_2), h_2(x_1, x_2)) \\ g_2(h_1(x_1, x_2), h_2(x_1, x_2)) \end{pmatrix} = \begin{pmatrix} f_1(x_1, x_2) \\ f_2(x_1, x_2) \end{pmatrix} = \bar{f}(\bar{x})$$

Partiella derivator

$$\frac{\partial f_1}{\partial x_1} = \frac{\partial g_1}{\partial h_1}\frac{\partial h_1}{\partial x_1} + \frac{\partial g_1}{\partial h_2}\frac{\partial h_2}{\partial x_1}, \quad \frac{\partial f_1}{\partial x_2} = \frac{\partial g_1}{\partial h_1}\frac{\partial h_1}{\partial x_2} + \frac{\partial g_1}{\partial h_2}\frac{\partial h_2}{\partial x_2}$$

$$\frac{\partial f_2}{\partial x_1} = \frac{\partial g_2}{\partial h_1}\frac{\partial h_1}{\partial x_1} + \frac{\partial g_2}{\partial h_2}\frac{\partial h_2}{\partial x_1}, \quad \frac{\partial f_2}{\partial x_2} = \frac{\partial g_2}{\partial h_1}\frac{\partial h_1}{\partial x_2} + \frac{\partial g_2}{\partial h_2}\frac{\partial h_2}{\partial x_2}$$

På matrisform

$$\begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{pmatrix} = \begin{pmatrix} \frac{\partial g_1}{\partial h_1} & \frac{\partial g_1}{\partial h_2} \\ \frac{\partial g_2}{\partial h_1} & \frac{\partial g_2}{\partial h_2} \end{pmatrix} \begin{pmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} \end{pmatrix}$$

Med funktionalmatrisbeteckningar

$$\frac{\partial(f_1, f_2)}{\partial(x_1, x_2)} = \frac{\partial(g_1, g_2)}{\partial(h_1, h_2)}\frac{\partial(h_1, h_2)}{\partial(x_1, x_2)}$$

Eller mer kompakt

$$\frac{\partial(\bar{f})}{\partial(\bar{x})} = \frac{\partial(\bar{g} \circ \bar{h})}{\partial(\bar{x})} = \frac{\partial(\bar{g})}{\partial(\bar{h})}\frac{\partial(\bar{h})}{\partial(\bar{x})} \quad \Leftrightarrow \quad \bar{f}'(\bar{x}) = (\bar{g} \circ \bar{h})'(\bar{x}) = \bar{g}'(\bar{h}(\bar{x}))\,\bar{h}'(\bar{x})$$

som gäller i alla dimensioner (obs ordning viktig i matrismultiplikation).

Tre funktionalmatriser vi behöver

$$\bar{\sigma}(\bar{t}) = \begin{pmatrix} \sigma(t_1) \\ \vdots \\ \sigma(t_n) \end{pmatrix} = \begin{pmatrix} t_1 \\ \vdots \\ t_n \end{pmatrix}_+ \Rightarrow \bar{\sigma}'(\bar{t}) = \frac{\partial(\bar{\sigma})}{\partial(\bar{t})} = \begin{pmatrix} H(t_1) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & H(t_n) \end{pmatrix} = H(\bar{t})$$

(0 eller 1 på diagonalen)

$$\bar{g}(\bar{x}) = A\bar{x} + B \text{ (affin)} \Rightarrow \bar{g}'(\bar{x}) = \frac{\partial(\bar{g})}{\partial(\bar{x})} = A\,.$$

Explicit i $\mathbb{R}^2$: $\bar{g}(\bar{x}) = \begin{pmatrix} g_1(x_1, x_2) \\ g_2(x_1, x_2) \end{pmatrix} = A\bar{x} + B = \begin{pmatrix} w_1 & w_2 \\ w_3 & w_4 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} w_5 \\ w_6 \end{pmatrix} =$

$$= \begin{pmatrix} w_1x_1 + w_2x_2 + w_5 \\ w_3x_1 + w_4x_2 + w_6 \end{pmatrix} \Rightarrow \bar{g}'(\bar{x}) = \frac{\partial(g_1, g_2)}{\partial(x_1, x_2)} = \begin{pmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} \end{pmatrix} = \begin{pmatrix} w_1 & w_2 \\ w_3 & w_4 \end{pmatrix} = A$$

Om $\bar{x}$ konstant och elementen i A och B variabler i $A\bar{x} + B$, ex:

$$\bar{g}(w_1, \dots, w_6) = \begin{pmatrix} w_1 & w_2 \\ w_3 & w_4 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} w_5 \\ w_6 \end{pmatrix} = \begin{pmatrix} w_1x_1 + w_2x_2 + w_5 \\ w_3x_1 + w_4x_2 + w_6 \end{pmatrix} \Rightarrow$$

$$\frac{\partial(g_1, g_2)}{\partial(w_1, \dots, w_6)} = \underbrace{\begin{pmatrix} x_1 & x_2 & 0 & 0 \\ 0 & 0 & x_1 & x_2 \end{pmatrix}}_{\text{från } A} \underbrace{\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}}_{\text{från } B} = \begin{pmatrix} \bar{x}^T & \bar{0}^T & 1 & 0 \\ \bar{0}^T & \bar{x}^T & 0 & 1 \end{pmatrix} = M(\bar{x})$$

Neuronnät $y = f(\bar{x}; \bar{w}) = (f_L \circ \cdots \circ \bar{f}_2 \circ \bar{f}_1)(\bar{x}) = (f_L \circ \cdots \circ \bar{f}_{j+1})(\bar{x}_j) = (f_L \circ \cdots \circ \bar{f}_{j+1})[\bar{f}_j(\bar{x}_{j-1})] = (f_L \circ \cdots \circ \bar{f}_{j+1})[\bar{\sigma}(A_j \bar{x}_{j-1} + B_j)]$

Sökt: $\nabla_w f = \dfrac{\partial(y)}{\partial(w_1, \ldots, w_M)}$

Kedjeregeln mellan två lager: $\bar{f}_j(\bar{x}_{j-1}) = \bar{\sigma}(\overbrace{A_j \bar{x}_{j-1} + B_j}^{= \bar{g}_j(\bar{x}_{j-1})})$, med

$\bar{x}_{j-1} = \bar{f}_{j-1}(\bar{x}_{j-2})$, ger matris $\dfrac{\partial(\bar{f}_j)}{\partial(\bar{f}_{j-1})} = \dfrac{\partial(\bar{f}_j)}{\partial(\bar{g}_j)} \dfrac{\partial(\bar{g}_j)}{\partial(\bar{f}_{j-1})} = H(\bar{g}_j) A_j = H_j A_j$

(ej $\bar{\sigma}$ / H i sista f_L)

Om A_j och B_j innehåller elementen $w_p, \ldots, w_q$ ger kedjeregeln upprepad

$$\frac{\partial(y)}{\partial(w_p, \ldots, w_q)} = \frac{\partial(f_L)}{\partial(\bar{f}_{L-1})} \frac{\partial(\bar{f}_{L-1})}{\partial(\bar{f}_{L-2})} \cdots \frac{\partial(\bar{f}_{j+1})}{\partial(\bar{f}_j)} \frac{\partial(\bar{f}_j)}{\partial(w_p, \ldots, w_q)} =$$

$= A_L (H_{L-1} A_{L-1}) \ldots (H_{j+1} A_{j+1})(H_j M_j)$ där $M_j = M(\bar{x}_{j-1})$

Nu kan vi för varje träningspunkt $\bar{x}^{(k)}$ beräkna

$$\nabla_w f(\bar{x}^{(k)}; \bar{w}) = \frac{\partial(y)}{\partial(w_1, \ldots, w_M)} =$$

$$= (\underbrace{A_L H_{L-1} \ldots A_2 H_1 M_1}_{\text{derivator på element i } A_1, B_1}, \underbrace{A_L H_{L-1} \ldots H_2 M_2}_{\text{på element i } A_2, B_2}, \ldots, \underbrace{A_L H_{L-1} M_{L-1}}_{\text{i } A_{L-1}, B_{L-1}}, \underbrace{M_L}_{\text{i } A_L}),$$

och även differensen $f(\bar{x}^{(k)}; \bar{w}) - y^{(k)}$. Addera för alla träningspunkter $\Rightarrow$

$$\nabla \ell(\bar{w}) = \frac{2}{N} \sum_{k=1}^{N} (f(\bar{x}^{(k)}; \bar{w}) - y^{(k)}) \nabla_w f(\bar{x}^{(k)}; \bar{w})$$

som nu kan användas för brantaste lutningsmetoden.

Exempel

Vi söker att litet neuronnät $f(\bar{x}) = (f_3 \circ \bar{f}_2 \circ \bar{f}_1)(\bar{x}) = (f_3(\bar{f}_2(\bar{f}_1(\bar{x})))$ som ska anpassas till $N = 5$ träningspunkter $(\bar{x}^{(1)}, y^{(1)}), \ldots, (\bar{x}^{(5)}, y^{(5)})$:

$$\left(\begin{pmatrix}-0.5\\1.1\end{pmatrix}, 3\right),\ \left(\begin{pmatrix}0.6\\-0.7\end{pmatrix}, 1.5\right),\ \left(\begin{pmatrix}1.4\\0.6\end{pmatrix}, 2\right),\ \left(\begin{pmatrix}1.1\\-0.4\end{pmatrix}, 1.7\right),\ \left(\begin{pmatrix}-0.3\\-1.1\end{pmatrix}, 2.5\right)$$

Nätets struktur är

$$\bar{x}_1 = \bar{f}_1(\bar{x}) = (A_1\bar{x} + B_1)_+ \ , \quad A_1 = \begin{pmatrix} w_1 & w_2 \\ w_3 & w_4 \end{pmatrix}, \ B_1 = \begin{pmatrix} w_5 \\ w_6 \end{pmatrix}$$

$$\bar{x}_2 = \bar{f}_2(\bar{x}_1) = (A_2\bar{x}_1 + B_2)_+ \ , \quad A_2 = \begin{pmatrix} w_7 & w_8 \\ w_9 & w_{10} \end{pmatrix}, \ B_2 = \begin{pmatrix} w_{11} \\ w_{12} \end{pmatrix}$$

$$y = f_3(\bar{x}_2) = A_3\bar{x}_2 \ , \quad A_3 = \begin{pmatrix} w_{13} & w_{14} \end{pmatrix}$$

Minimera $\quad \ell(\bar{w}) = \frac{1}{5}\sum_{k=1}^{5}(f(\bar{x}^{(k)}; \bar{w}) - y^{(k)})^2$ på $\mathbb{R}^{14}$

Stega längs minus $\nabla\ell(\bar{w}) = \frac{2}{5}\sum_{k=1}^{5}(f(\bar{x}^{(k)}; \bar{w}) - y^{(k)})\nabla_w f(\bar{x}^{(k)}; \bar{w})$

$$\nabla_w f(\bar{x}^{(k)}; \bar{w}) = \frac{\partial(y)}{\partial(w_1, \ldots, w_{14})} = (\underbrace{A_3H_2A_2H_1M_1}_{\text{derivator på } w_1,..,w_6}, \underbrace{A_3H_2M_2}_{\text{på } w_7,..,w_{12}}, \underbrace{M_3}_{w_{13},w_{14}})$$

Chansa på att starta i $\bar{w}_{(0)} = (1, 2, 0, -1, -1, 1, 0, 2, 1, -1, 1, 2, 1, 1)$, och ta steglängd $\eta = 0.05$. Matriserna är då initialt

$$A_1 = \begin{pmatrix} 1 & 2 \\ 0 & -1 \end{pmatrix}, B_1 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, A_2 = \begin{pmatrix} 0 & 2 \\ 1 & -1 \end{pmatrix}, B_2 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, A_3 = \begin{pmatrix} 1 & 1 \end{pmatrix}$$

Träningspunkt $\bar{x}^{(1)} = \begin{pmatrix} -0.5 \\ 1.1 \end{pmatrix}$, $y^{(1)} = 3$ ger

$$\bar{x}_1^{(1)} = (A_1\bar{x}^{(1)} + B_1)_+ = \begin{pmatrix} 0.7 \\ -0.1 \end{pmatrix}_+ = \begin{pmatrix} 0.7 \\ 0 \end{pmatrix} \Rightarrow \bar{x}_2^{(1)} = (A_2\bar{x}_1^{(1)} + B_2)_+ = \begin{pmatrix} 1 \\ 2.7 \end{pmatrix}_+ = \begin{pmatrix} 1 \\ 2.7 \end{pmatrix}$$

$$\Rightarrow \quad f(\bar{x}^{(1)}; \bar{w}) = A_3\bar{x}_2^{(1)} = 3.7 \quad \text{och} \quad f(\bar{x}^{(1)}; \bar{w}) - y^{(1)} = 3.7 - 3 = 0.7,$$

samt $H_1 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, H_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, M_1 = \begin{pmatrix} -0.5 & 1.1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -0.5 & 1.1 & 0 & 1 \end{pmatrix}$,

$M_2 = \begin{pmatrix} 0.7 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0.7 & 0 & 0 & 1 \end{pmatrix}, M_3 = \begin{pmatrix} 1 & 2.7 \end{pmatrix}$ som kan läsas av och ger

$A_3H_2A_2H_1M_1 = (-0.5, 1.1, 0, 0, 1, 0)$, $A_3H_2M_2 = (0.7, 0, 0.7, 0, 1, 1)$ så

$\nabla_w f(\bar{x}^{(1)}; \bar{w}_{(0)}) = (-0.5, 1.1, 0, 0, 1, 0, 0.7, 0, 0.7, 0, 1, 1, 1, 2.7)$ $\circledast$

Nu är term 1 i $\nabla\ell(\bar{w}) = \frac{2}{5}\sum_{k=1}^{5}(\underbrace{f(\bar{x}^{(k)}; \bar{w}) - y^{(k)}}_{=0.7 \text{ för } k=1})\underbrace{\nabla_w f(\bar{x}^{(k)}; \bar{w})}_{=\circledast \text{ för } k=1}$ beräknad

$$A_1 = \begin{pmatrix} 1 & 2 \\ 0 & -1 \end{pmatrix}, B_1 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}, A_2 = \begin{pmatrix} 0 & 2 \\ 1 & -1 \end{pmatrix}, B_2 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, A_3 = \begin{pmatrix} 1 & 1 \end{pmatrix}$$

Likadant för de 4 andra träningspunkterna $\bar{x}^{(2)}, \ldots, \bar{x}^{(5)}$ ger ihop att

$$\ell(\bar{w}_{(0)}) = \frac{1}{5}\sum_{k=1}^{5}(f(\bar{x}^{(k)}; \bar{w}_{(0)}) - y^{(k)})^2 = 6.862 \quad \text{och}$$

$$\nabla\ell(\bar{w}_{(0)}) = \frac{2}{5}\sum_{k=1}^{5}(f(\bar{x}^{(k)}; \bar{w}_{(0)}) - y^{(k)})\nabla_w f(\bar{x}^{(k)}; \bar{w}_{(0)}) \approx$$

$$\approx (1.54, 1.03, 2.99, -2.98, 1.48, 5.72, 2.12, 6.44, 2.12, 4.17, 4.92, 3.84, 17.8, 5.63)$$

Stega till ny punkt, $\bar{w}_{(1)} = \bar{w}_{(0)} - 0.05\nabla\ell(\bar{w}_{(0)}) \approx$

$(0.92, 1.95, -0.15, -0.85, -1.07, 0.71, -0.11, 1.68, 0.89, -1.21, 0.75, 1.81, 0.11, 0.72)$

Nya matrisvärden är alltså

$$A_1 \approx \begin{pmatrix} 0.92 & 1.95 \\ -0.15 & -0.85 \end{pmatrix}, B_1 \approx \begin{pmatrix} -1.07 \\ 0.71 \end{pmatrix}, A_2 \approx \begin{pmatrix} -0.11 & 1.68 \\ 0.89 & -1.21 \end{pmatrix}, B_2 \approx \begin{pmatrix} 0.75 \\ 1.81 \end{pmatrix}, A_3 \approx \begin{pmatrix} 0.11 & 0.72 \end{pmatrix}$$

Upprepa processen med dessa matriser. Ger $\ell(\bar{w}_{(1)}) \approx 1.55$, $\nabla\ell(\bar{w}_{(1)}) \approx$

$(0.25, -0.30, 0.48, -0.11, -0.25, 0.36, -0.02, -0.24, -0.11, -0.57, -0.22, -0.82, -5.20, -1.24)$

och $\bar{w}_{(2)} = \bar{w}_{(1)} - 0.05\nabla\ell(\bar{w}_{(1)}) \approx$

$(0.91, 1.96, -0.17, -0.85, -1.06, 0.70, -0.10, 1.69, 0.90, -1.18, 0.76, 1.85, 0.37, 0.78)$

Steg med gradientmetoden ger (approximativa värden)

Punkt	A_1	B_1	A_2	B_2	A_3	fel $\ell(\bar{w}_{(k)})$
$\bar{w}_{(0)}$	$\begin{pmatrix} 1 & 2 \\ 0 & -1 \end{pmatrix}$	$\begin{pmatrix} -1 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 2 \\ 1 & -1 \end{pmatrix}$	$\begin{pmatrix} 1 \\ 2 \end{pmatrix}$	$(1 \quad 1)$	6.86
$\bar{w}_{(1)}$	$\begin{pmatrix} 0.92 & 1.95 \\ -0.15 & -0.85 \end{pmatrix}$	$\begin{pmatrix} -1.07 \\ 0.71 \end{pmatrix}$	$\begin{pmatrix} -0.11 & 1.68 \\ 0.89 & -1.21 \end{pmatrix}$	$\begin{pmatrix} 0.75 \\ 1.81 \end{pmatrix}$	$(0.11 \quad 0.72)$	1.55
$\bar{w}_{(2)}$	$\begin{pmatrix} 0.91 & 1.96 \\ -0.17 & -0.85 \end{pmatrix}$	$\begin{pmatrix} -1.06 \\ 0.70 \end{pmatrix}$	$\begin{pmatrix} -0.10 & 1.69 \\ 0.90 & -1.18 \end{pmatrix}$	$\begin{pmatrix} 0.76 \\ 1.85 \end{pmatrix}$	$(0.37 \quad 0.78)$	0.51
...						
...						
$\bar{w}_{(5)}$	$\begin{pmatrix} 0.86 & 1.98 \\ -0.19 & -0.88 \end{pmatrix}$	$\begin{pmatrix} -1.06 \\ 0.73 \end{pmatrix}$	$\begin{pmatrix} -0.12 & 1.72 \\ 0.88 & -1.19 \end{pmatrix}$	$\begin{pmatrix} 0.78 \\ 1.85 \end{pmatrix}$	$(0.54 \quad 0.76)$	0.26
...						
...						
$\bar{w}_{(10)}$	$\begin{pmatrix} 0.79 & 2.01 \\ -0.20 & -0.92 \end{pmatrix}$	$\begin{pmatrix} -1.05 \\ 0.76 \end{pmatrix}$	$\begin{pmatrix} -0.13 & 1.73 \\ 0.86 & -1.23 \end{pmatrix}$	$\begin{pmatrix} 0.79 \\ 1.82 \end{pmatrix}$	$(0.55 \quad 0.75)$	0.22
...						
...						
$\bar{w}_{(20)}$	$\begin{pmatrix} 0.66 & 2.06 \\ -0.21 & -0.98 \end{pmatrix}$	$\begin{pmatrix} -1.03 \\ 0.81 \end{pmatrix}$	$\begin{pmatrix} -0.13 & 1.72 \\ 0.85 & -1.31 \end{pmatrix}$	$\begin{pmatrix} 0.79 \\ 1.79 \end{pmatrix}$	$(0.55 \quad 0.78)$	0.15
...						
...						
$\bar{w}_{(40)}$	$\begin{pmatrix} 0.43 & 2.13 \\ -0.28 & -1.00 \end{pmatrix}$	$\begin{pmatrix} -1.02 \\ 0.81 \end{pmatrix}$	$\begin{pmatrix} -0.13 & 1.73 \\ 0.87 & -1.33 \end{pmatrix}$	$\begin{pmatrix} 0.78 \\ 1.78 \end{pmatrix}$	$(0.55 \quad 0.81)$	0.087
...						

När vi anser ℓ tillräckligt liten har vi hittat vårt neuronnät och det är klart för användning!

Mycket enkelt göra iterationerna i t.ex. Matlab:

$xt = [-0.5\ 1.1; 0.6\ -0.7; 1.4\ 0.6; 1.1\ -0.4; -0.3\ -1.1], yt = [3; 1.5; 2; 1.7; 2.5]$
$N = 5, s = 50, q = 0.05, w = [1\ 2\ 0\ -1\ -1\ 1\ 0\ 2\ 1\ -1\ 1\ 2\ 1\ 1]$
for i=1:s
$A1 = [w(1)\ w(2); w(3)\ w(4)], B1 = [w(5); w(6)], A2 = [w(7)\ w(8); w(9)\ w(10)],$
$B2 = [w(11); w(12)], A3 = [w(13)\ w(14)]$
$L = 0; DL = zeros(1, 14);$
for j=1:N
$x = transpose(xt(j, :));$
$y = yt(j, 1);$
$t1 = A1 * x + B1; H1 = diag(heaviside(t1)); x1 = H1 * t1;$
$t2 = A2 * x1 + B2; H2 = diag(heaviside(t2)); x2 = H2 * t2;$
$x3 = A3 * x2;$
$L = L + (1/N) * (x3 - y)^2;$
$m1 = [transpose(x), zeros(1, 2); zeros(1, 2), transpose(x)]; M1 = [m1, eye(2)];$
$m2 = [transpose(x1), zeros(1, 2); zeros(1, 2), transpose(x1)]; M2 = [m2, eye(2)];$
$M3 = transpose(x2);$
$Dy = [A3 * H2 * A2 * H1 * M1, A3 * H2 * M2, M3];$
$DL = DL + (2/N) * (x3 - y) * Dy;$
end
$i - 1, L$
$w = w - q * DL;$

end

Kommentarer

Antal träningspunkter kan vara $N \sim 10^3 - 10^6$
Antal element i alla matriser (längd på $\bar{w}$) kan vara $M \sim 10^6 - 10^9$

Det tar mycket datorkraft och tid att beräkna stegriktningen $\nabla \ell(\bar{w})$ som summa från N träningspunkter, oftast delar man slumpmässigt in dem i grupper om $\sim 10 - 10^2$ (?) och approximerar $\nabla \ell(\bar{w})$ med summan över dem (kör igenom alla grupperna, många hyfsade steg på samma tid som ett bättre steg), kallas stokastisk gradientmetod och sparar mycket tid. Iterationerna med gradientmetoden kallas epoker. Stora neuronnät kan ta veckor att träna upp på superdatorer.

Steglängden η kallas "learning rate". I mindre optimeringsproblem finns metoder för att hitta bästa η. För neuronnät tar detta oftast för lång tid, man fastslår η från början (ibland minskande η i varje iteration). Mer avancerade gradientmetoder är vanliga.

Val av startpunkt $\bar{w}_{(0)}$ är svårt, olika tekniker för bra val finns. Man kan också testa flera olika, men tar tid.

Olinjära funktionen $\sigma(t)$ kallas aktiveringsfunktion. Olika val kan göras men $(t)_+$ vi valt är idag den vanligaste, den kallas ReLU(t) av AI-folk.

Fler kommentarer

Sista f_L oftast annorlunda beroende på vilken typ av utdata man vill ha (skalär/vektor, kontinuerlig/diskret etc).

Olika val av "loss function" $\ell(\bar{w})$ kan göras, ofta med regularisering, beror på vilket felmått man vill ska vara litet för träningsdata.

Elementen i $\bar{w}$ (alla matriselement) kallas vikter. Antal lager, storleken på matriser, val av η mm kallas nätets hyperparametrar. Man kan testa olika val av hyperparametrar men det är kostsamt. Näten kallas djupa om de har många lager, exempel med $\sim 10^2 - 10^3$ lager finns.

En del träningspunkter (ofta 10-30 %) används inte vid uppträningen av nätet utan sparas som testdata för att kunna testa om det färdiga nätet är OK. Detta är fundamentalt för att se att man inte över- eller underanpassat storleken på nätet. Över $\Rightarrow$ generaliserar dåligt för nya indata $\bar{x}$. Under $\Rightarrow$ klarar inte av att beskriva den funktion man söker. Olika tekniker för validering finns.

Beräkningen av $\nabla\ell(\bar{w})$ med kedjeregeln kallas "back propagation"-algoritmen och upptäcktes enligt vissa på 1980-talet. Andra källor hävdar att kedjeregeln var känd innan dess ....

Ännu fler kommentarer

Min-problemet i $\mathbb{R}^M$ är svårt att analysera analytiskt. Det har ofta antagits att det finns enormt många isolerade stationära punkter, men nya rön antyder att det kan finnas globalt optimum på delmångfald/ varietet av hög dimension. Att metoderna ovan konvergerar mot nät som fungerar otroligt bra i tillämpningar kan inte sägas ha matematiskt bra förklaring, säkert är att det finns många olika nät som fungerar ungefär lika bra för samma tillämpning. Att designa och träna upp neuronnät har blivit en konst för ingenjörer, matematiker, datavetare.

Många tillämpningar utnyttjar matriser (och "tensorer") med speciell struktur. Även om matriserna är stora är många element 0 och övriga element bara har ett fåtal olika värden $\Rightarrow$ färre vikter $\Rightarrow$ mycket snabbare att träna upp. De som används med stor framgång i speciellt bildanalys, där indata $\bar{x}$ egentligen är tre matriser (eller en 3-tensor), kallas faltningsnät, "convolutional neural networks" (CNN's), och är den vanligaste typen av riktigt djupa neuronnät.

Chat GPT, GPT-x modeller (generative pre-trained transformer). Neuronnät där vanliga steget $(A\bar{x} + \bar{b})_+$ föregås av "(masked) self attention". För att från indata (en följd av ord) skapa $\bar{x}$ från ett ord så behandlas först ordet i ett par steg som gör att hänsyn tas till ordets placering i följden. Matematisk görs matrismultiplikationer, matris/vektor-multiplikationer, (standard-)skalärproduktberäkningar, projektioner. I vissa transformer-modeller görs återkopplingar i nätet (krånglar till uppträningen) men inte i GPT-x modeller (endast "feed-forward").

Mycket färdig mjukvara finns att utnyttja för att skapa neuronnät, man behöver knappt veta hur de fungerar eller hur de tränas upp för att skapa dem, som en svart låda. Testa själva på playground.tensorflow.org !

Allt bra ?

Bättre än mänskliga experter på att analysera bilder och spela spel, men har inget med logiskt tänkande eller förmåga att följa regler att göra.

Neuronnät ger inga förklaringar, svårt veta vad i indata $\bar{x}$ som avgör $f(\bar{x}) \rightarrow$ svart låda. Har man köpt färdig mjukvara som fixar neuronnät från träningsdata $\rightarrow$ svart låda kring svart lådan. Har man inte fixat träningsdata själv eller ber någon annan fixa hela neuronnätet $\rightarrow$ ...

Etiska och juridiska problem:

"Bias" (partiskhet, fördomar) byggs in i näten då det finns i träningsdata, många exempel på diskriminering finns. Vem är ansvarig då det blir fel inom rättssystem, medicin, för självkörande bilar eller då banker bedömer kunder? Hur kan "fairness" garanteras?

Individer kan ha laglig rätt att få beslut om dem förklarade, svårt då neuronnät används. Upprepbarhet? Stabilitet över tid?

Massövervakning med ansiktsigenkänning - integritet?

Program som Google Translate (ett neuronnät), kan ha diskriminerande översättningar inbyggda. Chat GPT ?!?

Neuronnät kan manipuleras av dem som vet hur de är konstruerade, och träningsdata kan manipuleras.

Till sist

Liknar neuronnät det nät med neuroner vi har i hjärnan? Hjärnans neuroner är otroligt många, och länkstrukturen mellan dem komplicerad.

Man har börjat konstruera hårdvara med speciell arkitektur som ska imitera hjärnan och leda till mycket snabbare träning av (artificiella!) neuronnät.

Neuronnät används nu överallt, inom många tekniska och vetenskapliga områden ökar användningen dramatiskt. De kan användas till mycket positivt, men man måste vara medveten om risker. Då är det bra att veta hur de (”algoritmerna”) fungerar, vilket är er uppgift!

Kom ihåg:

Ett neuronnät är en funktion $y = f(\bar{x})$ som är en sammansättning av flera enkla funktioner.
Det lärs upp genom att minimera en ”loss function” för en stor mängd träningsdata, då används gradientmetoder och kedjeregeln.