Mesterséges Neuronhálók

1 Tipikus ZH hibák

1. feladat

$$J(Q,\underline{y}) = \frac{1}{2}(\underline{x}-Q\underline{y})^T(\underline{x}-Q\underline{y}) = \frac{1}{2}\|\underline{x}-Q\underline{y}\|_2^2$$

$$\frac{\partial J(Q,\underline{y})}{\partial \underline{y}} = \frac{1}{2}\cdot 2\,Q^T(\underline{x}-Q\underline{y})\cdot(-1)$$

$$= -Q^T(\underline{x}-Q\underline{y})$$

$$\frac{\partial J(Q,\underline{y})}{\partial Q} = \frac{1}{2}\cdot 2\,(\underline{x}-Q\underline{y})\,\underline{y}^T$$

$$\Delta \underline{y} = -\alpha \frac{\partial J(Q,\underline{y})}{\partial \underline{y}}$$

$$\Delta Q = -\alpha \frac{\partial J(Q,\underline{y})}{\partial Q}$$

4. feladat

$$f(\underline{w},\underline{x}) = \frac{1}{1+e^{-\underline{w}^T\underline{x}}}$$

$$\frac{\partial f(\underline{w},\underline{x})}{\partial \underline{w}} = \frac{-1\cdot\left(\frac{\partial 1+e^{-\underline{w}^T\underline{x}}}{\partial \underline{w}}\right)}{(1+e^{-\underline{w}^T\underline{x}})^2} =$$

$$\left(\frac{f}{g}\right)' = \frac{f'g - fg'}{g^2} \qquad \frac{-1\cdot(e^{-\underline{w}^T\underline{x}}\cdot(-\underline{x}))}{1+e^{-\underline{w}^T\underline{x}}}$$

$$\Delta \underline{y} = -\alpha \frac{\partial J(Q,\underline{y})}{\partial \underline{y}} = 0$$

$$Q^T(\underline{x}-Q\underline{y}) = 0$$

$$Q^T\underline{x} - Q^TQ\underline{y} = 0 \qquad \leftarrow \text{Lesz kérdés!}$$

$$Q^T\underline{x} = Q^TQ\underline{y}$$

$$\underbrace{(Q^TQ)^{-1}Q^T}_{\text{pszeudoinverz}}\underline{x} = \underline{y}$$

amit sokan nem tudtak:

a parciális derivált és a gradiens közötti különbség: a gradiens a (komponensenkénti) parc. derivált vektor konkatenáltja

2. [illegible]ha Autoencoder

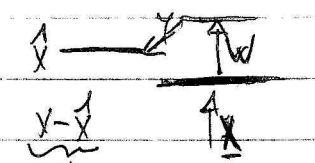
Lineáris - Nem lineáris autoencoder

hébbi tanulás

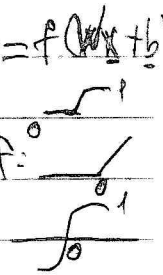
Q mátrix

reprezentáció

= v kissebb

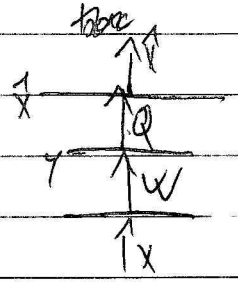


$y = f(Wx + b)$



hiba

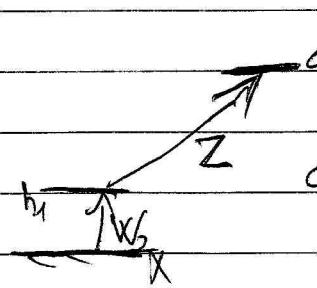
mély hálóban



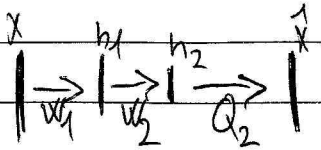
$\dim y < \dim x = \dim \hat{x}$

$x - \hat{x}$ bacu prop.

Stacked:

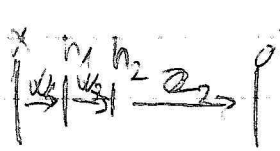


outp-outp hangolja W_2-t és 2-t.

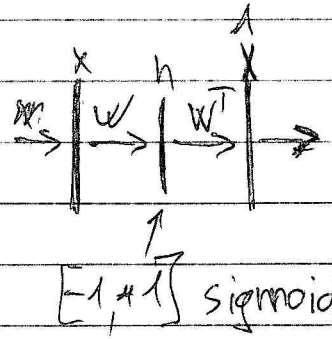


Így beállít 2 kiddjuk (rompresszál és rekonstruál)

$x - \hat{x}$ hangol W_1 W_2 Q_2-t



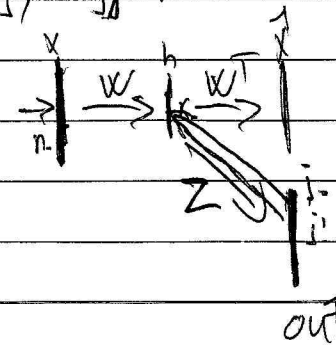
stacked autoencoder: eljárás nem egy háló



$[-1, +1]$ sigmoid $h = \sigma(Wx + b)$

a sigmoid nagyon vonda: a 0-tól távol nagyon kicsi. ha távol vagyunk a 0-tól akkor az a neuron rb elveszett.

mivel a dimenziót apránként csökkentjük így eggenként



mi a garancia, hogy W-t nem hangoljuk el durván outputt - outputt -al hangoljuk a vissza nyilak a hiba visszaterjesztést mutatják

A sok neuron miatt kb
véletlen számokkal haladunk vissza:
ekkor $\sqrt{n}$ ezek összege kb.
$\sqrt{n}$ így az összegzés nem
változtatja meg darabán a W-t
annál inkább igaz minél több
réteg van.
Ráadásul az újabb autoencoder lépés
korrigálja is ezt ugyanilyen mértékben

A kis lépések garantálják, hogy az érzékeny tartományban maradjunk.

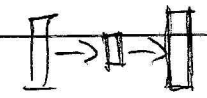
A világban vannak olyan adatbázisok amik emberileg annotálhatóak. A rekonstrukciós lépésben akár nagyobb adathalmazon is tanulhatunk.
Van amikor műszeres az annotáció:
pl robotkar: vesszük kamerával + mérjük belül -> adatok -> képből predikció.
(proprioceptikus)
emberekben is megfigyelhető, pl.: jóslat predikáljuk és utánozzuk a másik beszédét
recurrens neuronháló

ez egy fajta megerősítés a hallottak értelmezéséhez.

Konzultáció

felügyelt tanulás pl.: mély feedforward hálón
(x_i, y_i)
inp. outp mintapárok
$f(W, b, x) \approx y$.
felügyelet nélküli tanulás:
x_i: input output minta nincs
⇒ próbáljuk meg az inputot közelíteni
$f(W, b, x) \approx x$
probléma: ezt az identitás triviálisan megoldja.
⇒ nem tanulnánk semmit.
megoldás: f garantáltan ne lehessen identitás

valamilyen kompresszálást hajtunk végre: $x \to h \to \hat{x}$ $\dim(h) < \dim(x)$
nemlineáris „összenyomás" széthúzás



lépésenként ich 75 - 90% -ával nyomunk össze.

⇒ W oszlopainak száma = dim h
< W oszlopainak száma = dim x

Lehető legált. leírás:

$$\hat{x} = g(W_2 h + b_2) \qquad h = f(W_1 x + b_1)$$

paraméterek: W_1 W_2 b_1 b_2
mindegyik gradiens módszerrel optimalizálható
költség: $\sum_i \|x_i - \hat{x}_i\|_2^2$

⇒ költség minimalizálás.

A függvényeket az elején fixáljuk

Ez a klasszikus autoencoder.

Bonyolíthatjuk pl: több rejtett réteggel.

$g = I$
$W_2 = W_1^T$ $b_2 = 0$ } tied weights

erről szokt nincs beszélni.

ha kettes normát használunk akkor a W^T egyben a pszeudoinverz is.

W^T ⇒ kevesebb változó ⇒ kevésbé ragad el lokális minimumban.

Stacked autoencoder: Egyszerűbb, klasszikus félfelügyelt:

(x_i, y_i) inp outp. mintapárok és
x_j inp minták outp. nélkül.

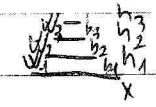
⇒ felváltva felügyelt és felügyeletlen tanulás

1, új réteg — felügyeletlen tanulás ahol az új réteg a rejtett az inputés az output az eggyel korábbi réteg. ⇒ W_j b_j -t kapjuk

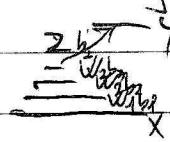
pl. tied weights-ed

x_i és x_j mintákkal. = az összes input mintával

2. Eldobjuk a 2 új réteg fölöttieket, a megfelelő mátrixokkal.



3. Felügyelt tanulás:
az (x_i, y_i) párokkal:
x_i-halmazon y_i-re hangoljuk az összes paramétert ami megmaradt és egy 2 mátrixot.



$\Rightarrow$ változik $W_1 W_2 W_3$
$b_1 b_2 b_3$

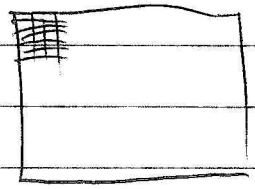
'4 goto 1

hány réteg stb.: Bayesian optimization

CNN: konvolúciós hálók:
eddig feedforward hálók voltak, mátrix-vektor szorzattal ($f(Wx+b)$)
most konvolúció: $f(W * x + b)$

$*$ művelet. konvolúció
x eddig bármilyen vektor lehetett, most szigorúan csak kép 2D-ben.

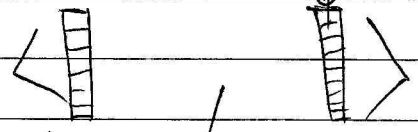
implementáció:

$x :=$  (oszlopfolytonosan)

$W =$  pl. 3x3, pici.

A nagy képet fel kell bontani a W méretű blokkokra.
= Minden pozícióba eltolom W-t.

$x^{(k)} =$  $W =$ 



skalárszorzatot veszek. (vektor mátrix)
Ezt minden pozícióban.
Több W: $\begin{bmatrix} \text{col}(W_1)^T \\ \vdots \\ \text{col}(W_2)^T \end{bmatrix} \cdot \text{im2col}(x)$

Ezzel egy kicsit kisebb képet kapunk

pooling réteg  dW már futott.

diszjunkt felbontás 2×2

minden blokkból maximális értéket

ez sokkal csökkenti a kép méretét.

durva alulmintavételezés.

pl felváltva konvolúciós réteggel.

pl. 3 ilyen váltás majd hangolás

y címkére.

f itt is tipikusan RELu

max -> lokális eltolás invariancia.

=> kis póz különbségek kiküszöbölése.