Nevronske mreže

Ljupčo Todorovski

Univerza v Ljubljani, Fakulteta za upravo
Institut Jožef Stefan, Odsek za tehnologije znanja (E-8)

April 2018

Pregled predavanja

Osnovne definicije

- Nevroni, sinapse in topologija NM
- Funkcija nevrona
- Usmerjene nevronske mreže

Učni algoritmi

- Delta pravilo in vzvratno razširjanje napake
- Kompromis med predsodkom in varianco

Konvolucijske nevronske mreže

Nevroni in sinapse

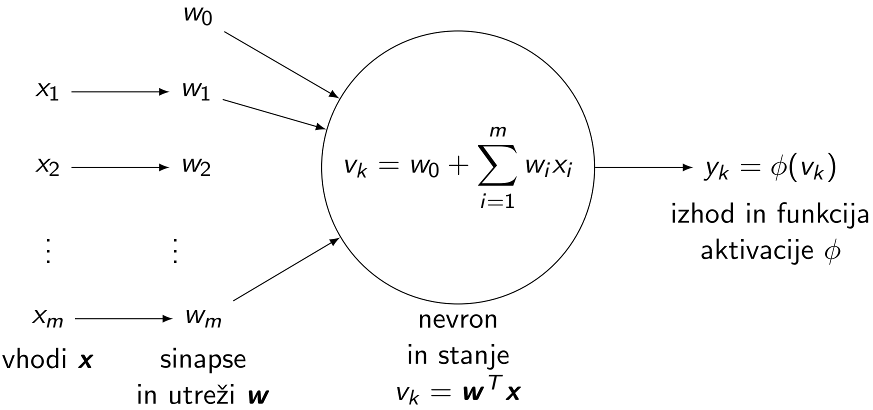
Gradnika nevronske mreže

- Nevroni: vsak ima stanje v in izhod y
- Sinapse: povezave med nevroni, ki določajo topologijo nevronske mreže; vsaka ima utež w

Izvajanje nevronske mreže

- Vhodni podatki spremenijo stanje in izhode izbranih nevronov
- Spremembe nevronov se prenašajo po sinapsah
- Nevron, ob spremembi vhodov na sinapsah, izračuna (spremeni) izhod
- Izračuna se nadaljuje dokler se nevroni (izhodi) ne ustalijo

Funkcija nevrona



Vektorska notacija: $\boldsymbol{x} = (1, x_1, x_2, \ldots x_m)^T$ in $\boldsymbol{w} = (w_0, w_1, \ldots w_m)^T$

Usmerjene (*feed-forward*) nevronske mreže

Vsaj dve ravni nevronov

- Raven vhodnih nevronov, en nevron za vsako vhodno spremenljivko
- Raven izhodnih nevronov
  - Regresija: en izhodni nevron
  - Klasifikacija: en izhodni nevron za vsak razred
- Nič ali več ravni skritih (*hidden*) nevronov

Topologija

- Nevroni v isti ravni nepovezani
- Sosednji ravni polno povezani: sinapsa med vsakim nevronom ravni $l-1$ in vsakim nevronom ravni l

Enostavni perceptron za binarno klasifikacijo

Dve ravni nevronov ($L = 0$)

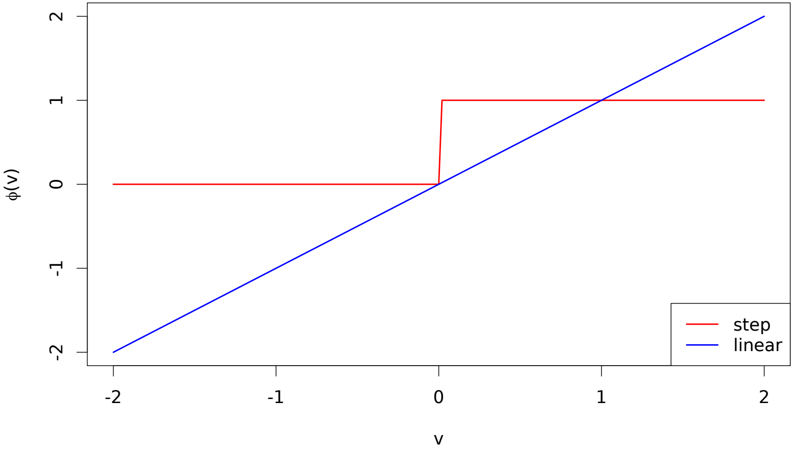
- Raven 0 vhodnih nevronov, funkcije aktivacije $\phi(v) = v$
  - Numerična spremenljivka: en nevron
  - Diskretna spremenljivka z zalogo D_X: $|D_X|$ nevronov (0 ali 1)
- Raven 1 izhodnih nevronov, funkcija aktivacije "stopnica"

$$\phi_{step}(v) = \begin{cases} 0 & ; v \leq 0 \\ 1 & ; v > 0 \end{cases}$$

Še ena verzija linearnega modela za klasifikacijo

$$\hat{y} = \phi_{step}(\boldsymbol{w}^T \boldsymbol{x})$$

Običajni funkciji aktivacije v perceptronu ϕ_{step} in $\phi(v) = v$



Iterativno pravilo za učenje uteži enostavnega perceptrona

$$\Delta \boldsymbol{w} = \eta \left(y - \phi_{step}(\boldsymbol{w}^T \boldsymbol{x}) \right) \boldsymbol{x}$$

- V vsaki iteraciji obravnavamo en primer $e = (\boldsymbol{x}, y)$
- $\Delta \boldsymbol{w}$ je sprememba uteži v iteraciji
- η je parameter, ki določa hitrost učenja (*learning rate*)
- Začetne vrednosti $\boldsymbol{w}$ nastavimo naključno

(Rosenblatt 1962)

Če sta razreda linearno ločljiva,
algoritem konvergira v končnem številu iteracij.

Pravilo delta za učenje uteži perceptrona

$$\Delta \boldsymbol{w} = \eta \, (y - \phi(\boldsymbol{w}^T \boldsymbol{x})) \, \phi'(v) \, \boldsymbol{x}$$

Izračun gradienta za $E = (y - \hat{y})^2$, $\hat{y} = \phi(v)$, $v = \boldsymbol{w}^T \boldsymbol{x} = \sum_k w_k x_k$

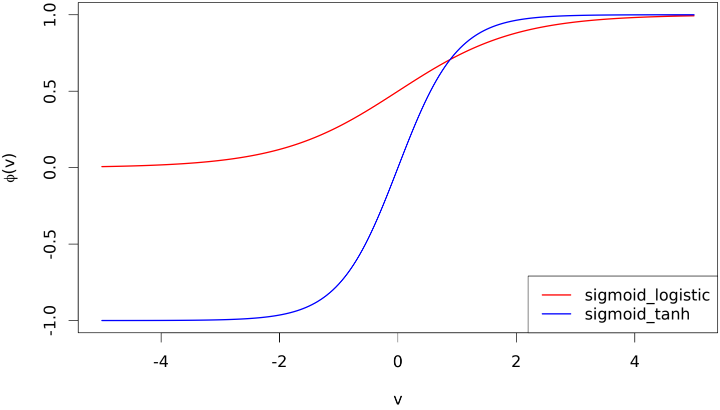
$$\begin{aligned} \frac{\partial E}{\partial w_i} &= \frac{\partial E}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial w_i} = \frac{\partial E}{\partial \hat{y}} \frac{\partial \phi(v)}{\partial v} \frac{\partial v}{\partial w_i} \\ &= -2 \, (y - \hat{y}) \, \phi'(v) \, \frac{\partial \sum_k w_k x_k}{\partial w_i} \\ &= -2 \, (y - \phi(\boldsymbol{w}^T \boldsymbol{x})) \, \phi'(v) \, x_i \end{aligned}$$

Posplošitev na učenje iz množice primerov

$$\Delta \boldsymbol{w} = \eta \sum_{(\boldsymbol{x}, y) \in S} (y - \phi(\boldsymbol{w}^T \boldsymbol{x})) \, \phi'(v) \, \boldsymbol{x}$$

Sigmoidni funkciji aktivacije v usmerjenih NM

$\phi_{logistic}(v) = 1/(1 + exp(-v))$ in $\phi_{tanh}(v) = \tanh v$



Funkcija aktivacije za izhodno raven klasifikacijske NM

$$\phi(v) = \frac{e^{v}}{\sum_k e^{v_k^{(L+1)}}}$$

Notacija

Indeksi ravni nevronov

- 0: raven vhodnih nevronov
- $L+1$: raven izhodnih nevronov
- $1, 2, \ldots L$: ravni skritih nevronov

Uteži w sinaps ter stanja v in izhodi y nevronov

- $w_{ji}^{(l)}$ utež sinapse med j-tim nevronom ravni $l-1$ in i-tim nevronom ravni l
- $v_i^{(l)}$ stanje i-tega nevrona ravni l, $v_i^{(l)} = \sum_j w_{ji}^{(l)} y_j^{(l-1)}$
- $y_i^{(l)}$ izhod i-tega nevrona ravni l, $y_i^{(l)} = \phi(v_i^{(l)})$

Posplošeno pravilo delta: raven izhodnih nevronov

$$E = \frac{1}{2}\sum_i (y_i - \hat{y}_i)^2,\ \hat{y}_i = y_i^{(L+1)} = \phi(v_i^{(L+1)}),\ v_i^{(L+1)} = \sum_j w_{ji}^{(L+1)} y_j^{(L)}$$

$$\begin{aligned}
\frac{\partial E}{\partial w_{ji}^{(L+1)}} &= \frac{\partial E}{\partial \hat{y}_i} \frac{\partial \hat{y}_i}{\partial v_i^{(L+1)}} \frac{\partial v_i^{(L+1)}}{\partial w_{ji}^{(L+1)}} \\
&= -(y_i - y_i^{(L+1)})\, \phi'(v_i^{(L+1)})\, \frac{\partial \sum_k w_{ki}^{(L+1)} y_k^{(L)}}{\partial w_{ji}^{(L+1)}} \\
&= -(y_i - y_i^{(L+1)})\, \phi'(v_i^{(L+1)})\, y_j^{(L)} \\
\Delta w_{ji}^{(L+1)} &= \eta\, (y_i - y_i^{(L+1)})\, \phi'(v_i^{(L+1)})\, y_j^{(L)}
\end{aligned}$$

Posplošeno pravilo delta: ravni skritih nevronov

$$\frac{\partial E}{\partial w_{ji}^{(l)}} = \frac{\partial E}{\partial y_i^{(l)}} \frac{\partial y_i^{(l)}}{\partial v_i^{(l)}} \frac{\partial v_i^{(l)}}{\partial w_{ji}^{(l)}} = \frac{\partial E}{\partial y_i^{(l)}} \phi'(v_i^{(l)})\, y_j^{(l-1)}$$

Trik za izračun $\frac{\partial E}{\partial y_i^{(l)}}$ iz $\frac{\partial E}{\partial y_k^{(l+1)}}$

$$\begin{aligned}
\frac{\partial E}{\partial y_i^{(l)}} &= \sum_k \frac{\partial E}{\partial v_k^{(l+1)}} \frac{\partial v_k^{(l+1)}}{\partial y_i^{(l)}} \\
&= \sum_k \frac{\partial E}{\partial y_k^{(l+1)}} \phi'(v_k^{(l+1)})\, w_{ik}^{(l+1)}
\end{aligned}$$

Pri prehodu iz prve v drugo vrstico še enkrat upoštevamo verižno pravilo.

Posplošeno pravilo delta: ravni skritih nevronov

$$\Delta w_{ji}^{(l)} = \eta\, \phi'(v_i^{(l)})\, y_j^{(l-1)} \sum_k \frac{\partial E}{\partial y_k^{(l+1)}}\, \phi'(v_k^{(l+1)})\, w_{ik}^{(l+1)}$$

- Iteracija pravila od ravni L do ravni 1
- Na ravni 1 upoštevamo $y_i = x_i$, t.j., vrednost spremenljivke X_i

Velikost in kontrola velikosti nevronskih mrež

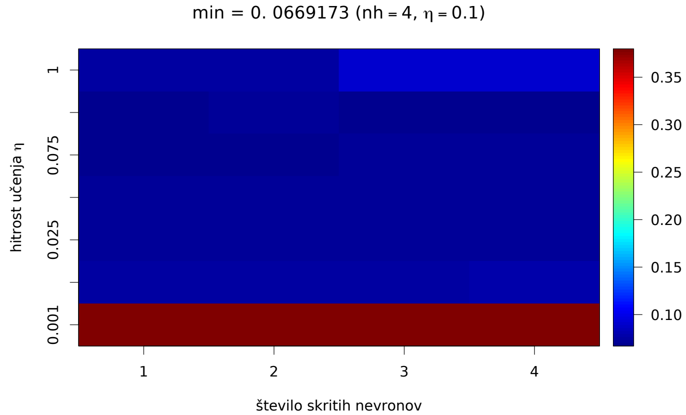
Velikost nevronske mreže

- Večanje nevronske mreže poveča število sinaps
- Torej število prostih parametrov napovednega modela
- Zato se zmanjša predsodek in poveča varianca

Kontrola velikosti NM

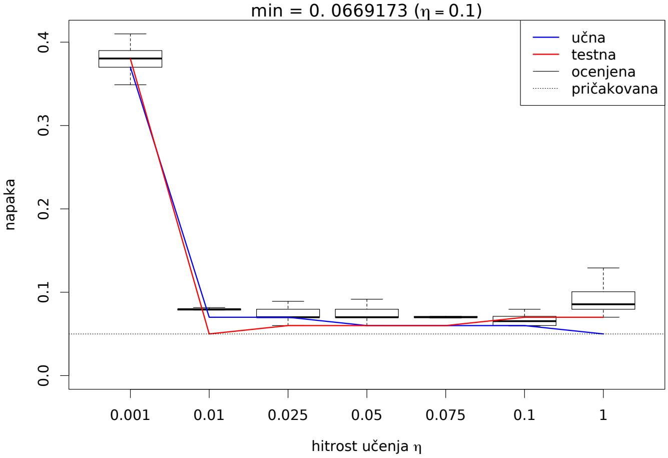
- Število ravni skritih nevronov (globina), globoke (*deep*) NM
- Število skritih nevronov v vsaki ravni

Klasifikacija z NM: število skritih nevronov (nh) in η

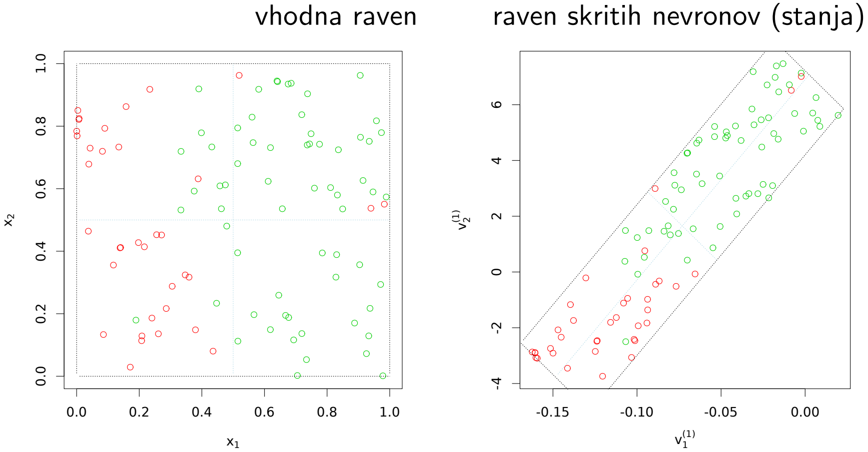


$Y = I((1 + X_1 + X_1X_2)/3 \geq 0.5), D_i = [0,1], i = 1..2, D_Y = \{0,1\}$
zamenjamo vrednost Y 0.05 naključno izbranim primerom

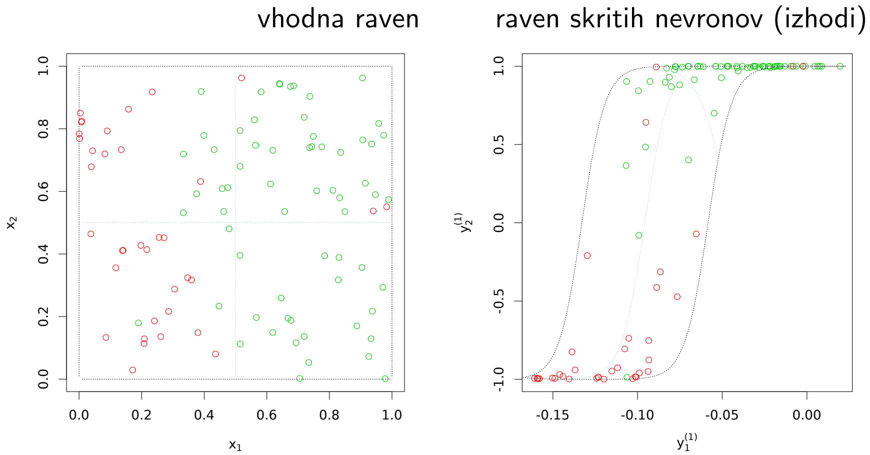
Klasifikacija z NM: $nh = 4$



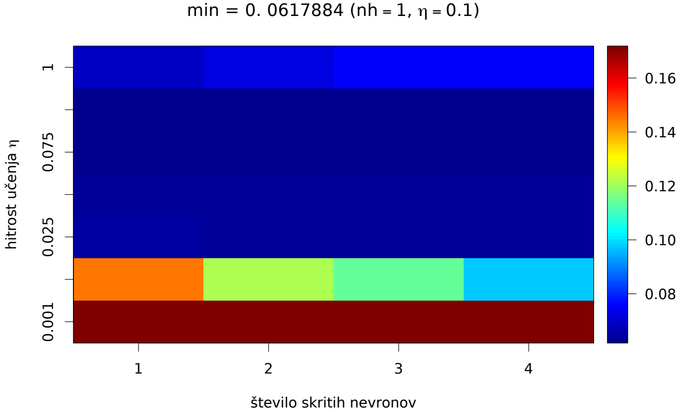
Raven skritih nevronov in nelinearnost pri klasifikaciji



Raven skritih nevronov in nelinearnost pri klasifikaciji

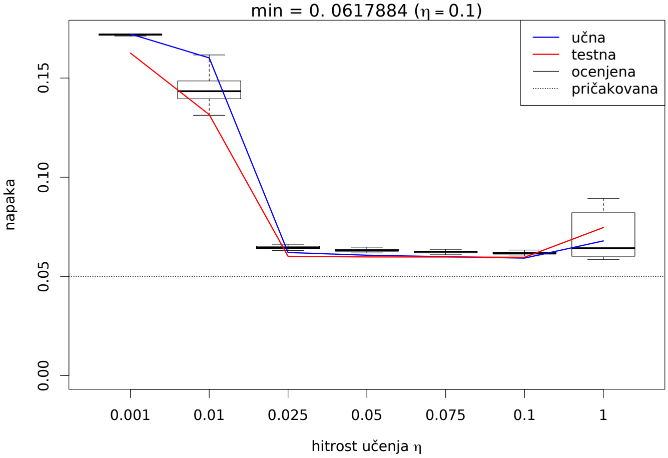


Regresija z NM: število skritih nevronov (nh) in η



$$Y = (1 + X_1 + X_1X_2)/3 + \mathcal{N}(0, 0.05),\ D_i = [0, 1], i = 1..2, D_Y = [0, 1]$$

Regresija z NM: $nh = 1$



Raven konvolucije (*convolution*)

Notacija

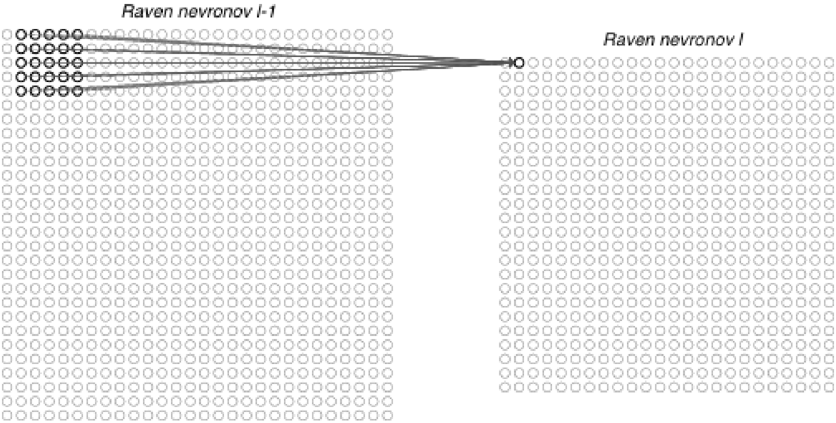
- Raven $l-1$ organizirana v obliki matrike nevronov dimenzij $x \times y$
- Uteži w_0 in konvolucijska matrika dimenzij $c \times c$, elementi $w_{i,j}$
- Raven l organizirana v obliki matrike nevronov dimenzij $(x-c+1) \times (y-c+1)$

Konvolucija, uporabljena namesto običajne obtežene vsote

$$v_{i,j}^{(l)} = w_0 + \sum_{k=1}^{c} \sum_{m=1}^{c} w_{k,m} y_{i+k-1,j+m-1}^{(l-1)}$$

- $y_{i,j}^{(l-1)}$ in $v_{i,j}^{(l)}$ so izhodi in stanja nevronov na ravni $l-1$ oz. l
- Običajna vrednost $c=5$

Raven konvolucije: grafični prikaz



Raven akumulacije (*pooling*)

Notacija

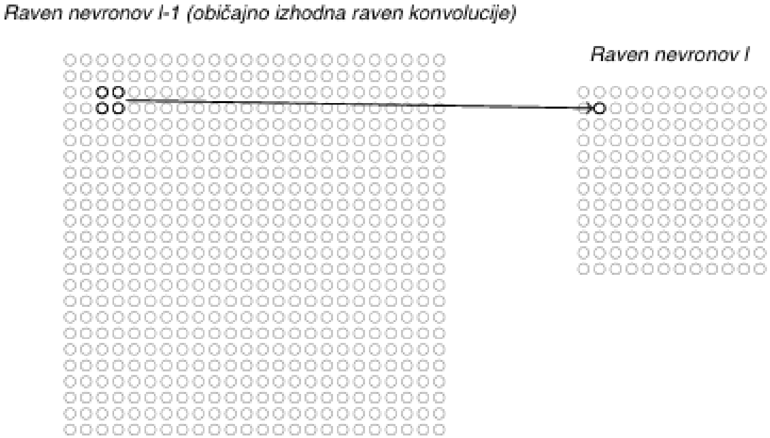
- Raven $l-1$ organizirana v obliki matrike nevronov dimenzij $x \times y$
- Raven l organizirana v obliki matrike nevronov dimenzij $x/a \times y/a$
- Predpostavka: $a|x$ in $a|y$

max-akumulacija, uporabljena namesto običajne obtežene vsote

$$v_{i,j}^{(l)} = \max \left\{ y_{(i-1)\cdot a+1,(j-1)\cdot a+1}^{(l-1)}, \ldots, y_{i\cdot a,j\cdot a}^{(l-1)} \right\}$$

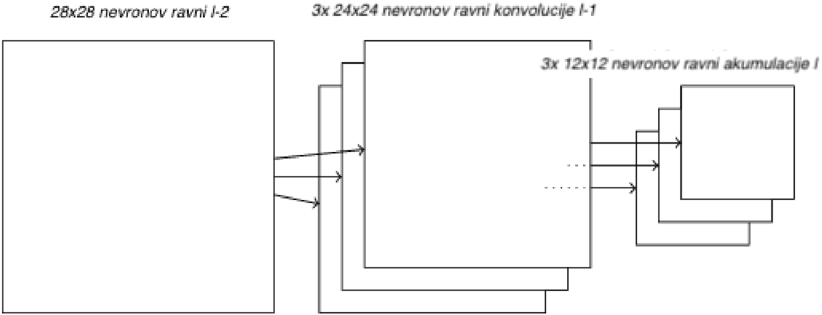
- $y_{i,j}^{(l-1)}$ in $v_{i,j}^{(l)}$ so izhodi in stanja nevronov na ravni $l-1$ oz. l
- Običajna vrednost $a = 2$

Raven akumulacije: grafični prikaz



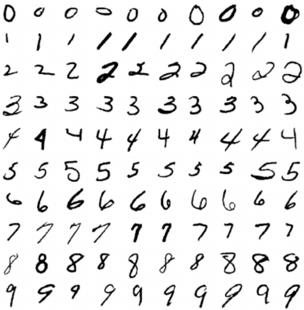
Običajna topologija: $r\times$konvolucija $\rightarrow$ $r\times$akumulacija

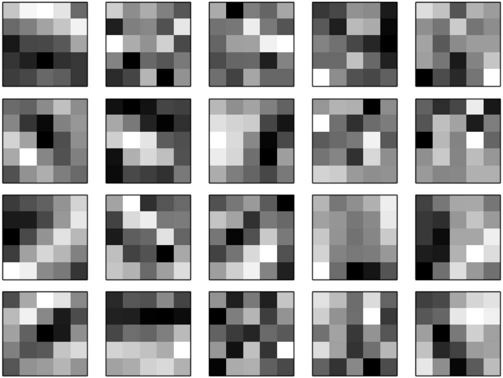
Široka uporaba za klasifikacijo slik in razpoznavanje z roko napisnih številk.



Razpoznavanje na roko napisanih števil

- Klasifikacijska napaka klasičnih metod: okoli 3%
- Klasifikacijska napaka konvolucijskih NM: manj kot 0.5%





Posebnosti učenja nevronskih mrež

- Običajno je opraviti normalizacijo podatkov
- Težave z gradientno metodo in lokalnimi ekstremi
- Večkratni izračuni, stohastične metode
- Funkcije aktivacije prilagojene problemu

Znani algoritmi in implementacije

Originalni predlog (Rosenblatt 1962)

Iterativno pravilo za klasifikacijo z enostavnim perceptronom.

Nadgradnja (Hinton 1986)

Vzvratno razširjanje napake, knjižnica nnet za R.

Globoko učenje (Hinton in ost. 2006)

Knjižnica TensorFlow, različne druge: tudi mxnet za R.