Hlboké Neuronové siete
super stručný úvod

Peter Sinčák

Február 2019

Kapitola 1

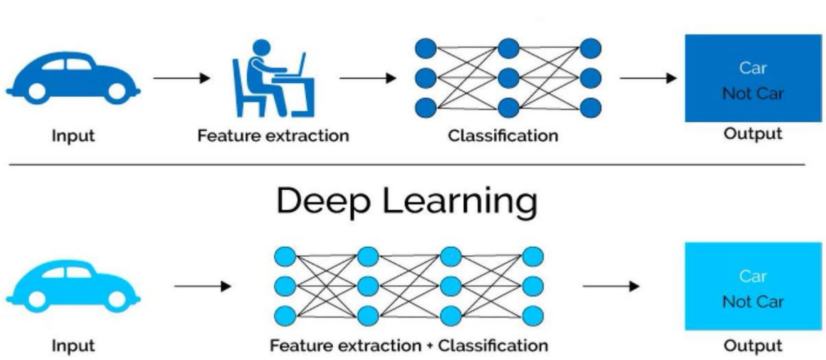
Konvolučné neurónové siete

1.1 Úvodné poznámky

Hlboké neuronové siete sú v súlade s základnou štruktúrou rozpoznávacieho procesu prostriedkom na

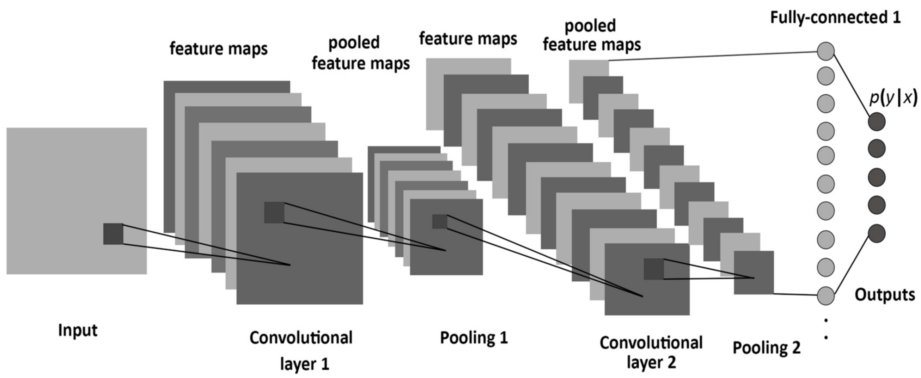
- meranie príznakov
- výber príznakov
- klasifikácia
- kategorizácia

Tieto 4 časti predstavujú základnú štruktúru rozpoznávania. Podčiarkujeme rozdiel medzi rozpoznávacím procesom a klasifikačným procesom resp. kategorizáciou. Viac o tejto štruktúre nájdete v ??.Na nasledujúcom obrázku demonštrujeme porovnanie plytkých a hlbokých neurónových sietí.



Obr. 1.1: Porovnanie plytkých a hlbokých neurónových sietí

Z časti o plytkých neurónových sieťach vieme, že tieto siete v princípe splňujú tzv. **Univerzálnu aproximačnú teorému** pre aproximáciu ľubovolnej všade hladkej funkcie. V tejto časti sa budeme venovať problematike využitia hlbokých neurónových sietí v spracovaní obrazovej informácie. Typická topológiou takejto konvolučnej neurónovej sieti je nasledovnom obrázku



Obr. 1.2: Príklad typickej topológie konvolučnej neurónovej siete

1.2 Základné pojmy

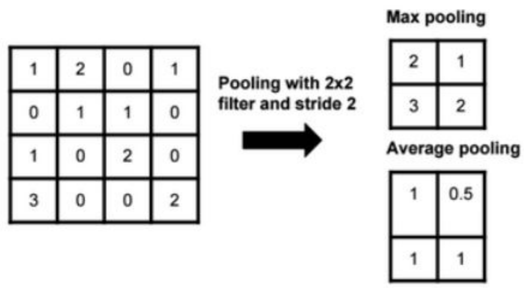
V tejto časti si priblížime základné pojmy a taktiež základný matematický aparát pre pochopenie práce s konvolučnými neurónovými sieťami. Taktiež je treba pripomenúť že sa sústredíme na aplikáciu CNN na spracovanie digitálneho obrazu.

1.2.1 Pojmy pri CNN

Medzi základné pojmy pri práci CNN narazíme na nasledovné pojmy

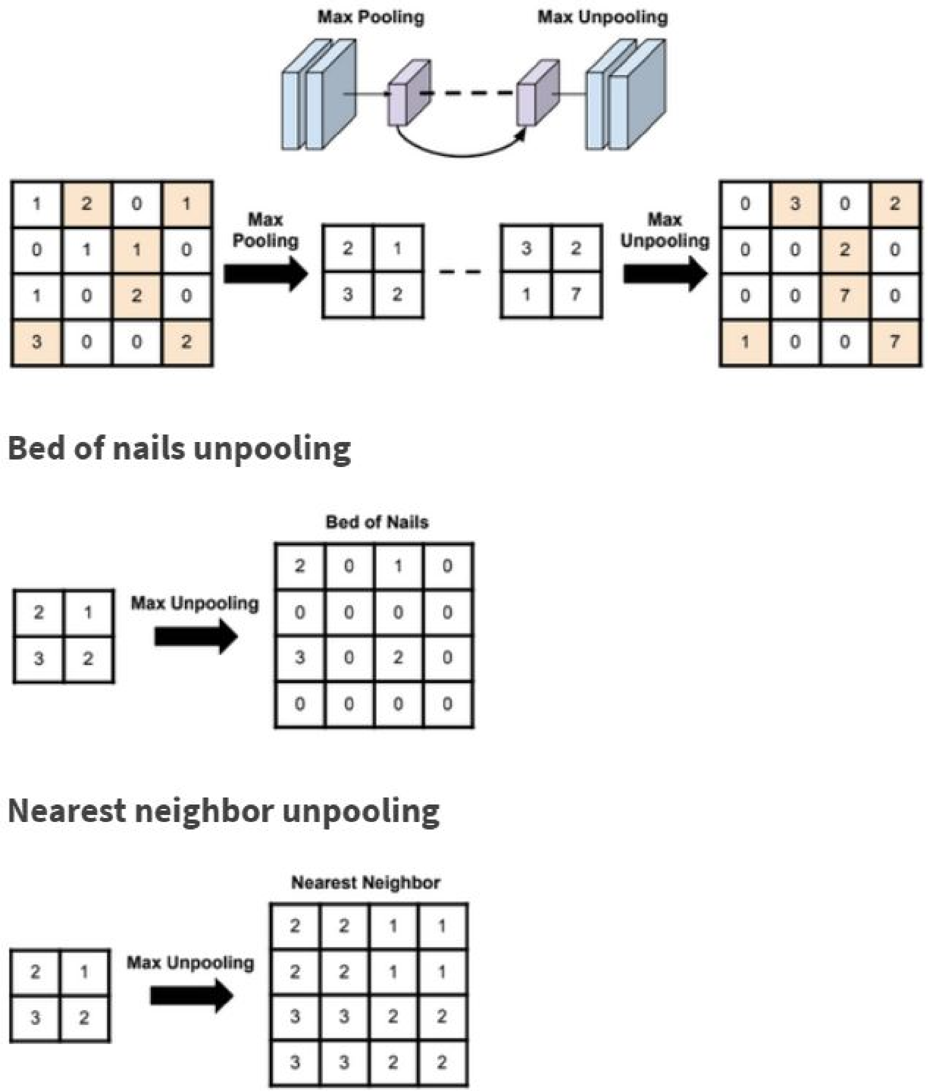
1. pojem vrstva konvolučná , operácia konvolucie a crosskoralacie
2. pojem vrstva pooling - jednoduchá matematická operácia z predchádzajúcej vrstvy (môže byť MAX pooling, AVERAGE pooling a iné

   a súčasne formy inverzného poolingu tzv. unpoolingu sú napr. nasledovne



Obr. 1.3: Ilustračný obrázok pre operáciu pooling na digitálnom obraze

3. pojem Vectorizácia a Concatizácia - priestorové pre usporiadanie neurónov do inej geometrickej formácie



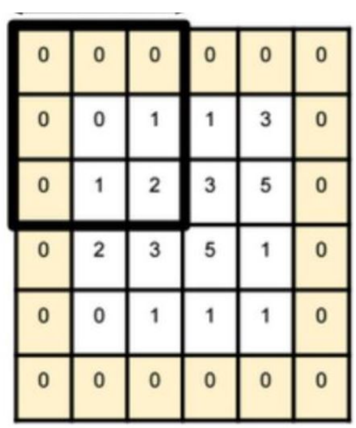
Obr. 1.4: Ilustračný obrázok operácie unpooling na digitálnom obraze

4. pojem "*padding*" - sposob aplikácie filtra na predošlú vrstvu. Ak padding je $= 0$ tak ak predošlá vrstva mala rozmer $nxmxch$ tak po aplikácii filtra o veľkosti $f1xf2xch$ bude výsledok konvolučnej vrstvy $(n+2p-f1+1, m+2p-f1+1, ch)$ teda ak mame $(n = m = 28, 1)$ a filter je $u = v = 5, 1$ tak

dostaneme na výstupe

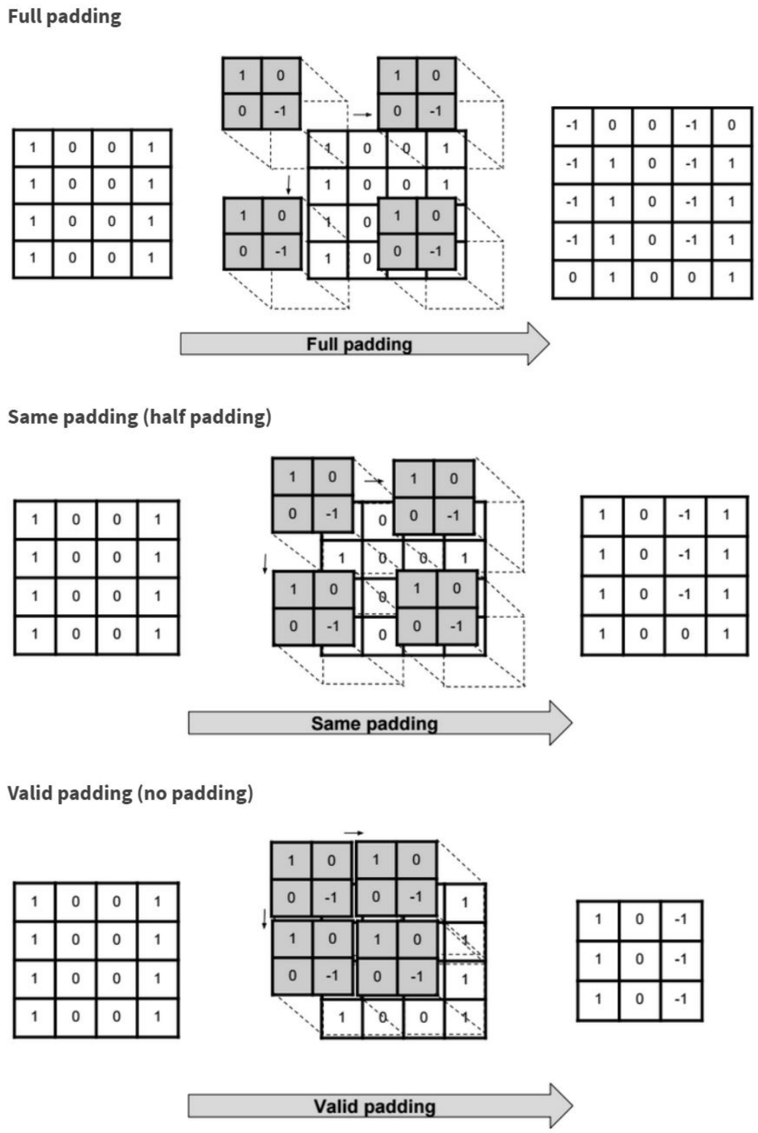
$$(28 + 2 * 0 - 5 + 1, 28 + 2 * 0 - 5 + 1, 1)$$

teda $24x24$. Ak je ”$padding = 0$” (tzv. valid padding) tak postupne sa obrázok zmenšuje a preto sa niekedy využíva aj tzv. "same padding"teda obrázok sa nezmenšuje lebo pôvodný sa rozšíri a padding bude $padding = (f1 - 1)/2$ teda ak je filter $3x3$ tak pri padding=1 sa obrázok neredukuje, pri $5x5$tak pri padding=2 sa obrázok taktiež neredukuje a podobne ďalej. Pri paddingu sa rozšírenie pôvodného obrázku realizuje číslom "0".



Obr. 1.5: Ilustračný obrázok operácie padding na digitálnom obraze

resp. jej rôzne formy



Obr. 1.6: Ilustračný obrázok rôznych typov operácie padding na digitálnom obraze

5. pojem "*strading*" hovorí o spôsobe posunu filtra na obrázku. Ak je $strading = 1$ filter sa kĺže po obraze a stále sa vypočítava referenčne nová hodnota centrálneho pixel,ak je $strading > 1$ filter sa aplikuje skokom po obraze nie

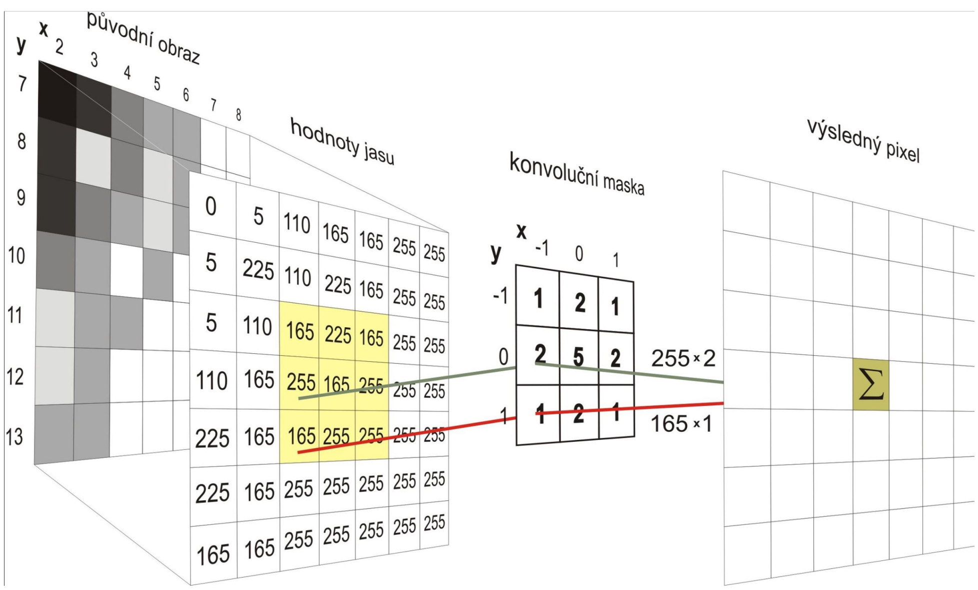
kĺzaním. Príklady na rôzne typy "*strading*" sú na nasledovnom obrázku

OBRAZOK	**STRIDING**

6. pojem konvolúcia dvoch funkcii v matematike je definovaná pre diskrétny prípad ako skalár a to nasledovne

$$(f * g)(x,y) = \sum_{i=-k}^{i=k} \sum_{j=-k}^{j=k} f(x-i, y-j) * g(i,j) \tag{1.1}$$

V princípe ide o lokálny operátor (kernel, filter) **g** aplikovaný napr. na digitálny obraz **f**. Výsledok tejto operácie bude hodnota pixela vo príznakovom poli na súradnici **x,y**. Dobre na tejto operácii je že sa dá vykonať rýchlo a pre celý obraz s vysokým stupňom paralelizácie. Ilustračne to zobrazuje nasledovný obrázok. ň



Obr. 1.7: Ilustračný obrázok konvolučnej operácie na digitálnom obraze

Súčasne je treba povedať, že parametre filtra môžeme považovať za považovať za synaptické váhy, ktoré budeme v ďalšom adaptovať.

1.2.2 Hyperparametre a parametre v CNN

V práci s CNN narazíme na hyperparametre a parametre CNN. Základný rozdiel medzi nimi je že

- hyperparametre sú definované užívateľom (ich voľba alebo nastavenie potrebujú určitý stupeň intuície a potrebujú skúsenosti s prácou na hlbokých neurónových sieťach)
- parametre - tieto sa adaptujú teda učia, v prípade kontrolovaného učenia podľa chybovej funkcie v rámci chybového priestoru tvoreného napr. W počtom parametrov a plus jeden rozmer chyba. Teda máme

  $$(W + 1)$$

  rozmerný chybový priestor, kde hľadáme globálne minimum pre $(W + 1)$. parameter práve hľadaním Woptimálnych parametrov v celom chybovom priestore (vo väčšine prípadov gradientovou metódou, parciálnej derivácie chyby podľa jednotlivých parametrov).

Medzi hyperparametre patria nasledovne údaje zvolené užívateľom a to :

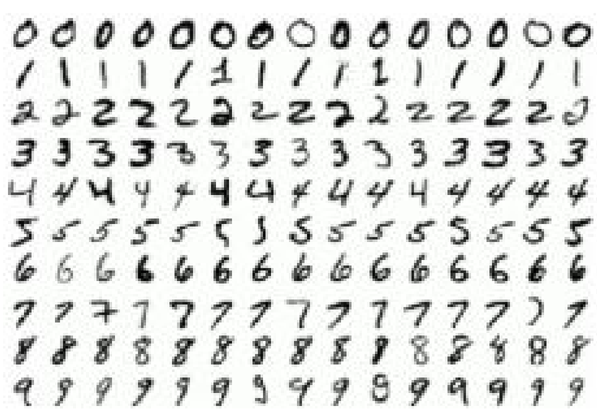
1. veľkosť filtra - veľká väčšina používa štvorcové filtre 3x3xch, 5x5xch, 7x7xch , kde ch je počet kanálov
2. počet filtrov p aplikovaných v jednej konvolučnej vrstve
3. počet konvolučných vrstiev
4. Výber aktivačnej funkcie konvolučnej vrstve
5. výber typu poolingovej vrstvy
6. výber typu vektorizácie a následnej integračnej vrstvy
7. definovanie aktivačnej funkcie na výstupe (pre klasifikáciu väčšinou softmax funkcia).

1.3 Jednoduchá CNN na spracovanie obrazu

V nasledujúcom prípade si popíšme topológiu jednoduchej konvolučnej neurónovej siete - nazvime ju pracovne **CNNSIMPLE**.

Na vstupe siete budeme mať binárny obrázok 28x28 kde budeme môcť predkladať číslice z intervalu prirodzených čísel $< 0, 9 >$ a na výstupe sa nám vybudí jeden z 10 výstupov indikujúci číslicu na vstupe. Takže popis samotnej topológie a logiky siete je nasledovný :

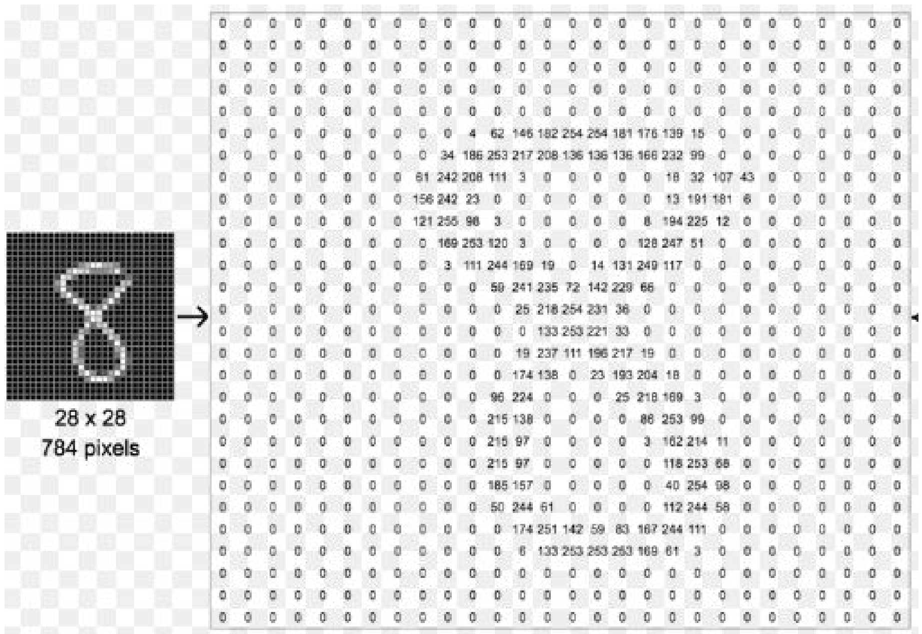
1. vstupom do siete je digitálny obraz o rozmeroch n=28 (šírka), m=28 (výška), ch=1 (počet kanálov) je to monochromatický obraz, kde hodnoty pixlov sú z intervalu prirodzených čísel $< 0, 255 >$. Na vstupnom obraze sú rôzne tvary čísel z intervalu prirodzených čísel $< 0, 9 >$. Príklady takýchto obrázkov sú na nasledujúcom obrázku 1.8. Každá jedna číslica na vstupnom obrázku má rozmer 28x28 pixlov.



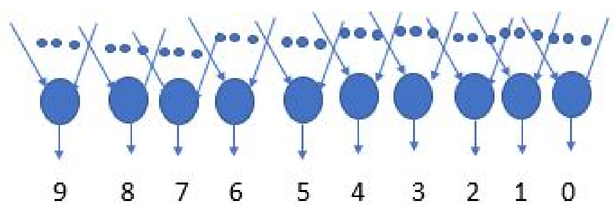
Obr. 1.8: Príklady vstupov do CNNSIMPLE neurónovej siete

teda každé jedno číslo na obrázku je reprezentované matici čísel 28x28x1 ako napríklad číslo 8 na obrázku 1.9

2. Výstupom zo CNNSIMPLE je vektor 10 hodnôt, ktorý prezentuje o ktorú číslicu sa jedna Ten výstupný neurón bude mať najvyššiu hodnotu ako je to na nasledovnom obrázku 1.10 načrtnuté.



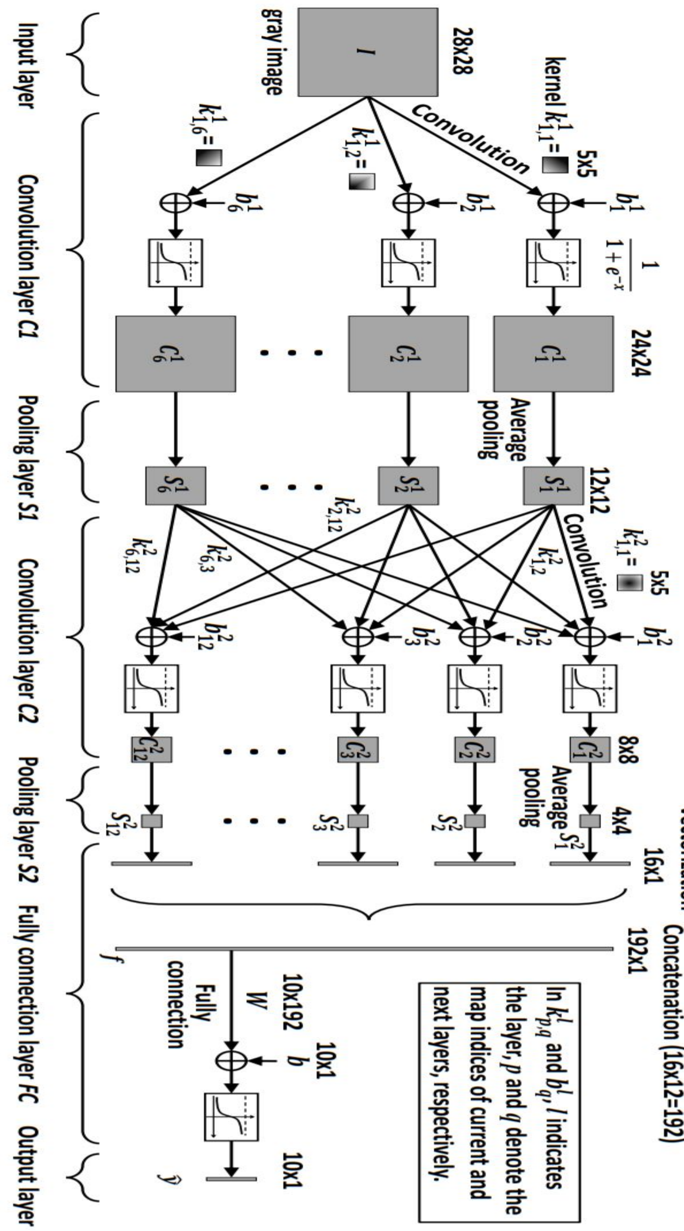
Obr. 1.9: Príklady vstupu čísla 8 ako matice čísel 28x28x1 do CNNSIMPLE neurónovej siete



Obr. 1.10: výstup z CNNSIMPLE (výstup z klasifikačnej časti CNN)

1.3.1 Topológia, parametre a dopredný prechod cez CNN-SIMPLE

V ďalšom si popíšeme topológiu CNNSIMPLE, jej topológiu a identifikujeme jej parametre resp. hyperparametre. Samotná neurónová sieť je na obrázku 1.11.

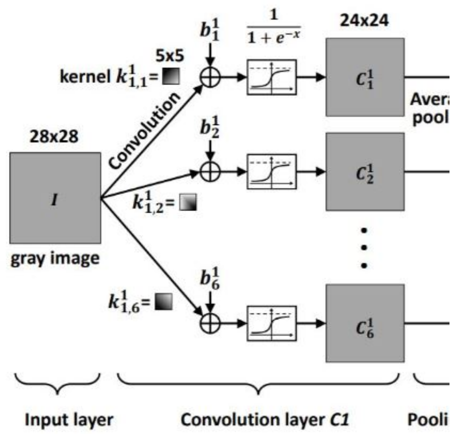


Obr. 1.11: Príklad jednoduchej konvolučnej neurónovej siete

Ako je jasné z obrázku 1.11 CNNSIMPLE sa skladá z nasledovných 4 častí :

1. Vstup pozostáva z digitalneho obrazu o rozmeroch $28x28$ pixelov teda matice o rovnakých rozmeroch.

2. Konvolučná vrstva C1, ktorá sa skladá



Obr. 1.12: Vstupná časť CNNSIMPLE

(a) 6 filtrov 5x5x1, ktoré a následne výsledok konvolúcie ide o aktivačnej funkcie sigmoidalnej. Po každej konvolúcii a výpočte sigmoidalnej funkcie sa na vytvorí hodnota vo všetkých 6 vrstvách a pripočíta baies. Teda výpočet hodnôt v pre C_1^1 teda prvá hodnota C^1 pre filter 1 bude nasledovná

$$C_1^1 = \sigma\left(I * k_{1,1}^1 + b_1^1\right) \tag{1.2}$$

kde σ je sigmoidálna funkcia

$$\sigma = \frac{1}{1 + exp^{-x}} \tag{1.3}$$

a súčasne platí že

$$x = \left(I * k_{1,1}^1 + b_1^1\right)$$

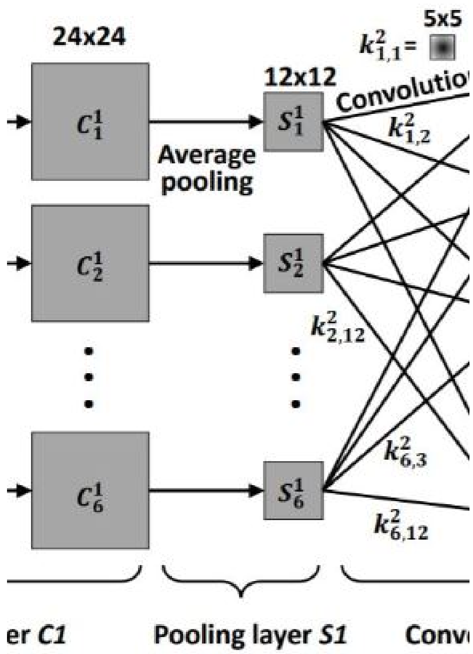
a teda celkový výsledok po konvolúcii je pre všetky filtre môžeme zapísať nasledovne

$$C_p^1(i,j) = \sigma\left(\sum_{u=-2}^{u=2}\sum_{v=-2}^{v=2} I(i-u, j-v).k_{1,p}^1(u,v) + b_p^1\right) \tag{1.4}$$

kde

- I je vstupný obraz $28x28$
- $k^j_{i,p}$ je hodnota jadra (kernelu) filtra "p"na pozícii $1,1$
- b^1_p je prah (baies) filtra "p v našom prípade je stále iba jedna hodnota na filter
- padding je nulový
- ...

3. Vrstva S1 - Pooling vrstva



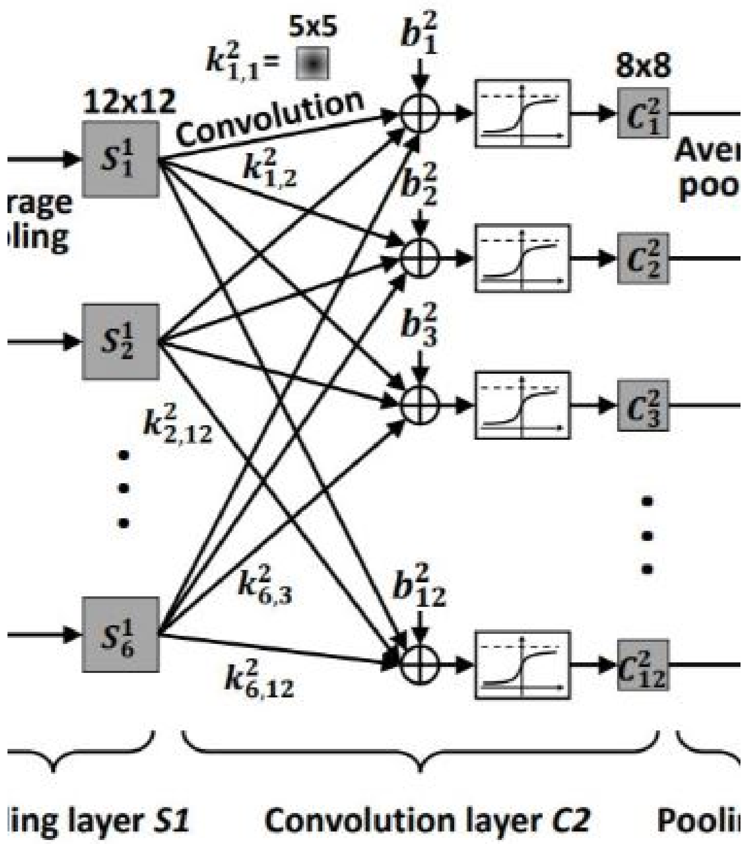
Obr. 1.13: Druhá časť CNNSIMPLE

Ak z predchádzajúcej časti vieme vypočítať hodnoty $C^1_p(i,j)$ zo vzorca 1.4 sa tzv. Average Pooling vypočíta jednoducho nasledovne

$$S^1_p = \frac{1}{4}\sigma\left(\sum_{u=0}^{v=1}\sum_{v=0}^{v=1} C^1_p(2i-u, 2j-v)\right) \tag{1.5}$$

kde $i,j-1,2,...,12$ a $p=1,...,6$ lebo C^1_p vrstva má rozmer $24x24$ a priemerovaným poolingom dostaneme rozmer $12x12$.

4. Konvolučná vrstva $C2$
   následne po vrstve $S1$ pozostáva z 12 filtrov $5x5$, s nulovým paddingom a jednotkovým stridom, aplikovaných na 6 obrazov po poolingu o rozmeroch $12x12$. Výsledok konvolúcie znova prechádza do sigmoidálnej aktivačnej funkcie σ_qa prahom b_q kde $q = 1, ...12$. Za takýchto podmienok dostaneme na výstupe vrstvy $C2$ 12 obrazov C_q o rozmeroch $8x8$ ako výsledok konvolúcie filtra $3x3$ na obraze $12x12$ s nulovým padingom a stridom o hodnote 1.



Obr. 1.14: Tretia časť CNNSIMPLE

Teda znova ak

$$\sigma = \frac{1}{1 + exp^{-x}} \tag{1.6}$$

a súčasne platí že

$$x = \left(S_p^1 * k_{p,q}^2 + b_q^2\right)$$

a teda celkový výsledok po konvolúcii je pre všetky filtre môžeme zapísať nasledovne

$$C_q^2(i,j) = \sigma\left(\sum_{p=1}^{p=6}\sum_{u=-2}^{u=2}\sum_{v=-2}^{v=2} S_p^1(i-u, j-v).k_{p,q}^2(u,v) + b_p^2\right) \tag{1.7}$$

teda v rovnice 1.7 je jasne že $q = 1, ...12$ pretože máme 12 výstupov z vrstvy $C2$ o veľkosti $8x8$. Teda kým v vrstve $C1$ sme mali 6 príznakových polí $24x24$ po poolingu a vrstve $C2$ máme 12 príznakových polí o rozmere $8x8$.

Z toho teda vyplýva, že ak na vstupe sme mali obrázok $28x28$ *to je* **784** *pixelov (príznakov), po prvej konvolučnej vrstve* $C1$ *sme mali* $6x24x24$ *to jest* **3456** *hodnôt príznakov po druhej konvolučnej vrstve* $C2$ *sme mali* $12x8x8$ *to jest* **768** *príznakov. Všetko toto následne smerovalo k vrstve* $S2$*, ktorá mala* **192 neurónov** *(príznakov). Týmto bola ukončená časť hľadania príznakového priestoru a nasleduje výstup z konvolučnej siete, kde sa realizuje klasifikácia dát.*

---

5. vrstva $S2$ a výstupná vrstva

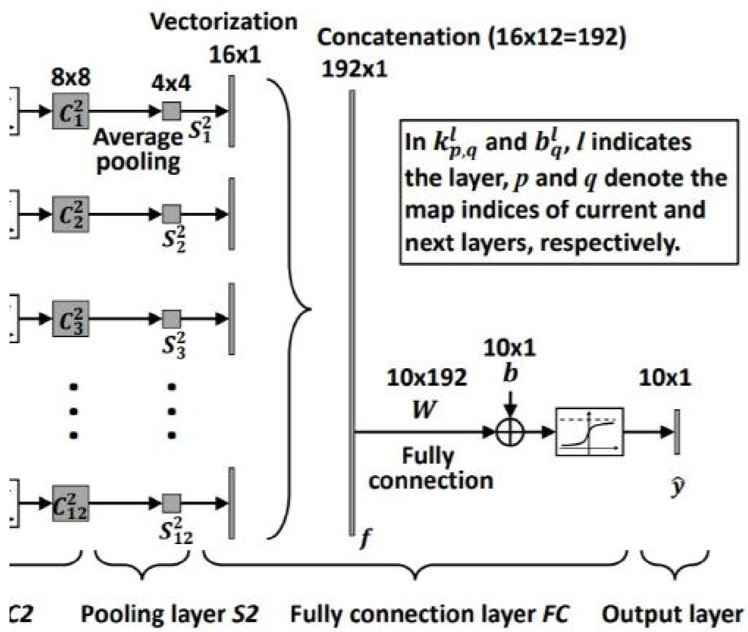
   Táto časť CNNSIMPLE sa skladá z

   (a) jednoduchej poolingovej vrstvy $S2$, ktorá priemerovaním dostane z 12 príznakových polí $8x8$ následne 12 príznakových polí $4x4$

   (b) Plne prepojenej vrstvy FC, ktorá sa samotná skladá z

   - vektorizačnej časti, ktorá príznakové pole $4x4$ iba dá do vektora $16x1$
   - integračnej časti (concatenation), ktorá z 12 vektorov $16x1$ vytvorí jeden vektor $196x1$, lebo $12x16 = 192$
   - plne prepojenej časti, ktorá 192 neurónov plne prepojí na 10 výstupných neurónov

   (c) výstupnej vrstvy ktorá 10 neurónov má sigmodálnu funkciu a prah a je plne napojená na predchádzajúcu $192x1$vrstvu neurónov



Obr. 1.15: Výstupná časť CNNSIMPLE

Teda samotné matematické vzťahy sú v celku jednoduché :

- vrstva priemerového poolingu $S2$

$$S_p^2 = \frac{1}{4}\left(\sum_{u=0}^{v=1}\sum_{v=0}^{v=1} C_p^2(2i-u, 2j-v)\right) \tag{1.8}$$

kde $i, j = 1, .., 4$ a $q = 1, ..., 12$ lebo C_p^2 vrstva má rozmer $8x8$ a priemerovaným poolingom dostaneme rozmer $4x4$ pre všetkých 12 príznakových polí.

- vektorizačná a integračná vrstva vlastne iba preorganizuje konfiguráciu siete z formácie $12x4x4$ zo vzorca 1.8 do jedného vektora neurónov o veľkosti $1x192$. tento proces označíme ako funkciu

$$f = F\left(\left\{S_q^2\right\}_{q=1,...12}\right) \tag{1.9}$$

kde f je vlastne vektor $f_{k=1,...192}$. Súčasne môžeme napísať pre tento jednoduchý vzťah 1.9 aj inverznú operáciu teda z $1x192$ späť na formáciu $12x4x4$ a to nasledovne

$$\left\{S_q^2\right\}_{q=1,...12} = F^{-1}(f) \tag{1.10}$$

- výstupná vrstva o veľkosti $1x10$ neurónov,ktorá má prepojenie z predchádzajúcej vrstvy o veľkosti $1x192$ neurónov cez sigmoidálne aktivačné funkcie σ a prahy b teda vypočítaný výstup $\hat{y}$ dostaneme ak

$$fc_l = \sum_{k=1}^{k=192} w_{l,k} * f_k + b_l \tag{1.11}$$

kde $l = 1, ..10$ a $w_{l,k}$ je hodnota váhy medzi k-tym neurónom v integračnej vrstve (zdrojovým) a l-tým neurónom (cieľovým) vo výstupnej vrstve a fc_l je suma súčinov pre jednotlivé výstupy pred aktivačnou funkciou výstupnej vrstvy.Následne teda

$$\sigma = \frac{1}{1 + exp^{-fc}} \tag{1.12}$$

Vo vektorovom tvare to môžeme napísať ako

$$\hat{y} = \sigma\left\{(Wxf) + b\right\} \tag{1.13}$$

Na záver ak predpokladáme vypočítanú hodnotu na výstupnej vrtsve $\hat{y}(l)$ kde $l-1, ...10$ a očakávanú hodnotu na výstupnej vrstve označme $y(l)$ na l-tom neuróne, tak potom môžeme definovať našu **chybovú funkciu** L (**Lost function**) nasledovne cez všetky výstupné neuróny na výstupnej vrstve CNNSIMPLE

$$L = \frac{1}{2}\sum_{l=1}^{l=10}(\hat{y}(l) - y(l))^2 \tag{1.14}$$

Na základe horeuvedeného teda môžeme uzavrieť **dopredný** prechod cez konvolučnú neurónovú sieť CNNSIMPLE.

1.3.2 Parametre CNNSIMPLE pre učenie

Veľmi dôležitá je otázka koľko parametrov budeme adaptovať resp. učiť na tejto konvolučnej sieti.

Teda ide o nasledovné parametre :

1. **parametre filtrov $\mathbf{k^1_{1,p}}$** v konvolučnej vrstve $C1$ ide o $5x5 + 1 = 26$ pre každý filter teda máme 6 filtrov **26x6 = 156 parametrov** na $C1$. Tieto parametre je potrebné na úvod učenie inicializovať. Na základe teórie je vhodné inicializovať tieto parametre pomocou generátora rovnomerného rozdelenia nasledovnými parametrami

$$k^1_{1,p} \sim U\left(\pm\sqrt{\frac{6}{(1+5)x5^2}}\right) \tag{1.15}$$

   kde $p = 1, ...6$ počet filtrov v $C1$. Táto inicializácia je dôležitá pre prvý výpočet vo vzorci 1.4. Súčasne b^1_p sú nastavené na hodnotu nula.

2. **parametre filtrov $\mathbf{k^2_{p,p}}$** lebo na 6 príznakových platní sa aplikuje konvolučná vrstva s filtrom $5x5$ teda pre jednu konvolúciu na 6 príznakových platní mame $5x5x6$ parametrov a ešte prah b teda spolu je to $5x5x6 + 1$ násobené počtom filtrov v konvolučnej vrstve $C2$ teda 12 a výsledok je **(5x5x6 + 1)x12 = 1812 parametrov** v vrstve $C2$.Tieto parametre je potrebné na úvod učenie inicializovať. Na základe teórie je vhodné inicializovať tieto parametre pomocou generátora rovnomerného rozdelenia nasledovnými parametrami

$$k^2_{p,q} \sim U\left(\pm\sqrt{\frac{6}{(6+12)x5^2}}\right) \tag{1.16}$$

   kde $q = 1, ...12$ počet filtrov v $C2$. Táto inicializácia je dôležitá pre prvý výpočet vo vzorci 1.7. Súčasne b^2_q sú nastavené na hodnotu nula.

3. na výstupnej vrstve máme **synaptické váhy W**, teda **192x10 + 10 = 1930 parametrov**.Tieto parametre je potrebné na úvod učenie inicializovať. Na základe teórie je vhodné inicializovať tieto parametre pomocou generátora rovnomerného rozdelenia nasledovnými parametrami

$$W \sim U\left(\pm\sqrt{\frac{6}{(192+10)}}\right) \tag{1.17}$$

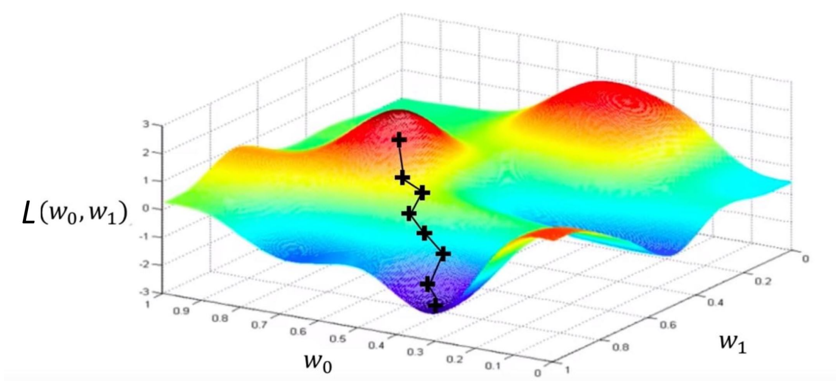
   Táto inicializácia je dôležitá pre prvý výpočet vo vzorci 1.11. Súčasne b zo vzorca 1.13 sú nastavené na počiatku výpočtov na hodnotu nula.

Výsledok je teda nasledovný, konvolučná neurónová sieť CNNSIMPLE má spolu

$$\mathbf{156 + 1812 + 1930 = 3898} \tag{1.18}$$

parametrov, ktoré musíme nastaviť.Tu však je treba upozorniť že pôvodný obrázok $28x28$ piselov bol pretrasponovaný do 192 rozmerného vektora. Teda vrstvy C1, S1, C2 a S2 hľadajú nové príznaky a v novom príznakovom priestore sa realizuje klasifikácia.

Globálne však my hľadáme aj transformáciu aj diskriminačné hyper-plochy v novom príznakovom priestore **SÚBEŽNE** a teda sa **jedna o** $3898+1 = 3899$ **rozmerný chybový priestor**, ktorý bude predmetom nášho záujmu s cieľom nájdenia tých **správnych resp. optimálnych** parametrov, ktoré budú dávať najmenšiu hodnotu chybovej funkcie L definovanej vo vzorci 1.14.



Obr. 1.16: ilustračný obrázok $2 + 1$ rozmerného chybového priestoru a výpočtu gradientu na chybovom povrchu s cieľom hľadania globálneho minima chyby L