Tomáš Oberhuber

Faculty of Nuclear Sciences and Physical Engineering
Czech Technical University in Prague

Biologická motivace

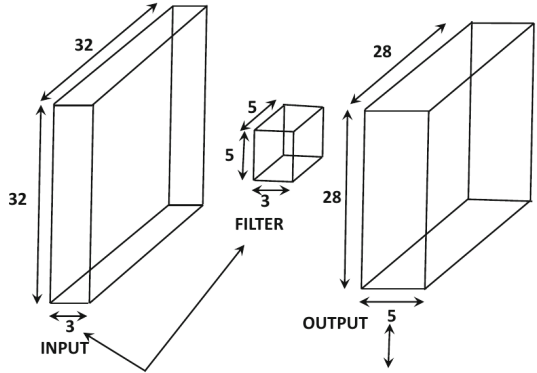
- konvoluční n. sítě (CNN) výrazně napomohly úspěchu hlubokých NN (Deep NN = DNN) sítí, který zaznamenáváme v posledních letech
- používají se zejména pro zpracování obrazových dat
- jsou inspirovány prací Hubela a Wiesela a jejich pokusy na vizuálním kortexu koček
- ukázalo se, že buňky této části mozku reagují na vizuální podněty
- reagují na podněty v různých částech zorného pole a navíc na různé pozorované tvary

CNN

- na základě těchto pozorování vznikla architektura CNN
- provádí regularizaci pomocí sdílených vah
- jde o dobrý příklad specializované sítě pro určitou doménu
- díky tomu je možné sítě rychleji trénovat a konstruovat větší sítě
- podstatou CNN je fakt, že přechod na další vrstvu se děje pomocí operace konvoluce
- konvoluční jádro není známé a právě to je předmětem učení sítě

CNN

- na vstupu síť dostává obrázek s předem daným fixním rozlišením
- podstatnou vlastností obrazových dat je to, že sousední pixely nebývají nezávislé, ale korelují spolu
- dále se opíráme o fakt, že translace obrazu nemění jeho význam



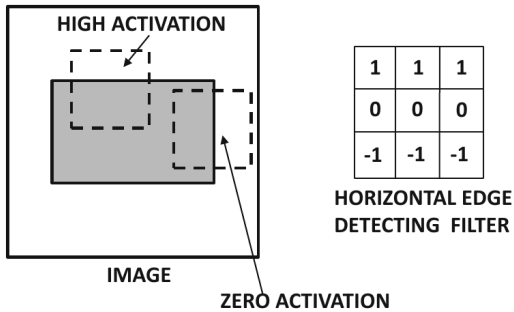
(a)

Ch.C.Aggarwal, Neural Networks and Deep Learning.

CNN

- na vstupu vrstvy q máme data o rozměrech $L_q \times B_q \times d_q$
  - L_q je výška
  - B_q je šířka
  - d_q je hloubka
- na vstupu d_q odpovídá počtu kanálů obrazových dat, např. RGB, multimodální snímky apod.
- na dalších vrstvách pracujeme s více tzv. *feature/activation maps*
  - ty nám umožňují extrahovat více rysů/příznaků (=feature) z obrázku

- na vrstvě q použijeme konvoluční jádro (filtr) o rozměrech $F_q \times F_q \times d_q$
- filtr přikládáme k obrázku všude, kde dojde ke kompletnímu překryvu a vlastně počítáme konvoluci



(b)

Ch.C.Aggarwal, Neural Networks and Deep Learning.

- na výstupu dostaneme aktivační mapu, tj. jakým způsobem reagoval daný filtr/konvoluční jádro na vstupní data
- pokud chceme na vrstvě $q+1$ více aktivačních map, použijeme více filtrů
- filtr je možné umístit na

$$\begin{aligned} L_{q+1} &= L_q - F_q + 1 \text{ pozic na výšku,} \\ B_{q+1} &= B_q - F_q + 1 \text{ pozic na šířku,} \end{aligned}$$

  což jsou zároveň rozměry aktivačních map na vrstvě $q+1$
- CNN se navrhují většinou tak, že s rostoucím q se zmenšuje L_q a B_q a roste d_q
- to je proto, že potřebujeme z obrazových dat těžit jejich různé rysy (=features)

- označme p-tý filtr na q-té vrstvě jako

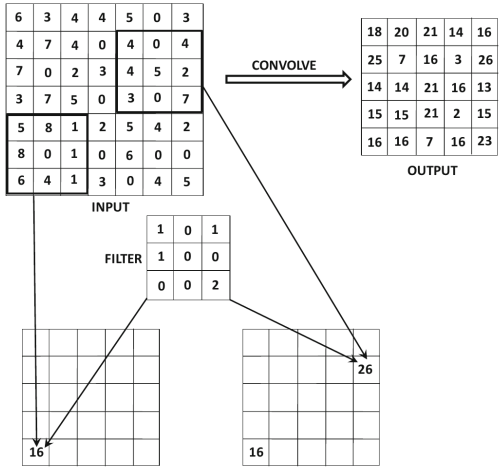
$$\mathbb{W}^{(p,q)} := w_{ijk}^{(p,q)},$$

kde $w_{ijk}^{(p,q)}$ jsou jeho váhy

- pak aplikace filtru lze popsat jako

$$a_{ijp}^{(q+1)} = \sum_{r=1}^{F_q} \sum_{s=1}^{F^q} \sum_{k=1}^{d_q} w_{rsk}^{(p,q)} h_{i+r-1,j+s-1,k}^{(q)}$$

pro všechna $i = 1, \ldots, L_q - F_q + 1, j = 1, \ldots, B_q - F_q + 1, p = 1, \ldots, d_{q+1}$



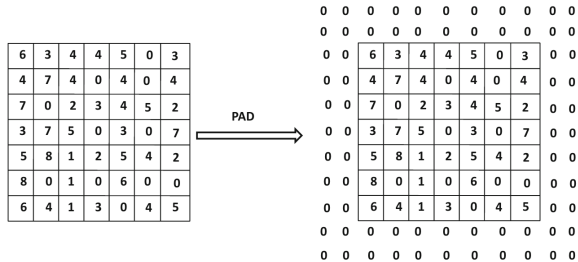
Ch.C.Aggarwal, Neural Networks and Deep Learning.

CNN

- v souladu s Hubelovým a Wieselovým experimentem platí, že malé oblasti zorného pole aktivují různé neurony
- dále platí, že filtry na nižších úrovních reagují na jednoduché tvary, jako hrany
  - bylo ukázáno, že CNN se učí filtry podobné Gaborovým filtrům
- vyšší vrstvy reagují na stále komplexnější tvary
- konvoluce také splňují tzv. equivarianci vůči translaci = pokud obraz posuneme o pár pixelů, stejně tak se změní i aktivační mapy

CNN - padding

- pokud chceme zachovat rozměry aktivační masky mezi jednotlivými vrstvami, použijeme obložení předchozí vrstvy pomocí nul



- pro zachování rozměru je potřeba obložit vrstvu z obou stran o polovinu velikosti filtru - *half padding*
- někdy se používá i obložení na celou velikost - jinak se totiž příspěvek krajních pixelů do následující vrstvy redukuje

CNN - strides

- standardně se filtr posouvá vždy o jeden pixel
- někdy se ale může volit i větší krok - S_q, pak platí

$$L_{q+1} = \frac{L_q - F_q}{S_q} + 1,$$
$$B_{q+1} = \frac{B_q - F_q}{S_q} + 1,$$

- $S_q = 1$ je nejčastější volba, ale lze se setkat i s $S_q = 2$

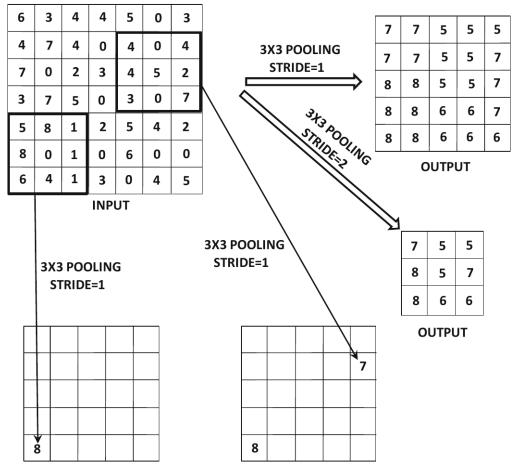
CNN - typická nastavení

- vrstvy mívají nejčastěji čtvercové rozměry, tj. $L_q = B_q$
- často se volí rozměry rovné mocnině dvou
- počet filtrů na jedné vrstvě se také často volí jako mocnina 2
- jako nelinearita je v případě CNN téměř výhradně používána funkce ReLU
- bylo ukázano, že s ReLU CNN fungují výrazně lépe

CNN - pooling

- kromě konvoluce a ReLU nelinearity s u CNN používá i tzv. pooling
- narozdíl od konvolučních filtrů se pooling provadí na jednotlivých aktivačních mapách, tj.
  - konvoluční filtr z d_q map udělá jednu novou
  - pooling udělá z každé mapy jednu novou
- pooling nejčastěji používá operace max, občas průměr
- pooling neobsahuje parametry k naučení

CNN - pooling



Max pooling

Ch.C.Aggarwal, Neural Networks and Deep Learning.

CNN - pooling

- lze opět používat různě velké masky a různý *stride*
- doporučuje se, aby velikost masky byla větší než *stride*, tj. aby docházelo k překryvům při posouvání masky
- údajně to pomáhá proti přetrénování sítě
- poslední trend je ale spíš ústup od poolingu, tj. vynechávat ho z návrhu sítí

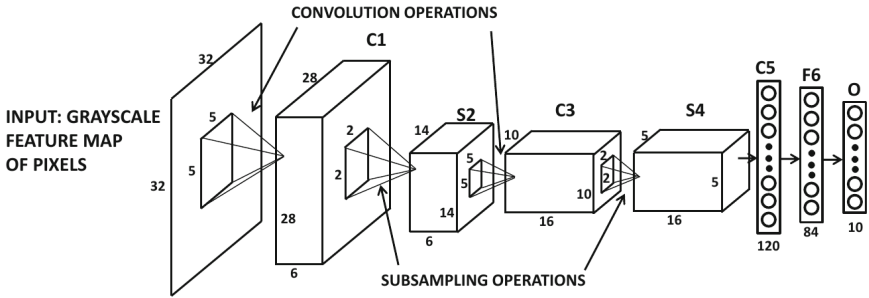
CNN - plně propojené sítě

- CNN téměř vždy obsahují několik plně propojených vrstev před svým výstupem
- ty vlastně operují nad rysy (=features) extrahovanými v předchozích konvolučních vrstvách
- tyto vrstvy většinou obsahují výraznou většinu parametrů k učení z celé CNN
- konvoluční vrstvy mají velký počet aktivačních proměnných, ale málo parametrů
- plně propojené vrstvy mají málo aktivačních proměnných, ale hodně parametrů

- v celé síti se pak střídají tyto vrstvy
  - *C* - konvoluční vrstva
  - *R* - ReLU nelinearita
  - *P* - pooling - pokud je použit
  - *F* - plně propojená vrstva
- celou síť pak lze popsat třeba takto

$$CRCRPCRCRPCRCRPF$$

- doplněním rozměrů jednotlivých vrstev, filtrů a velikost kroků (=*stride*) je pak architektura sítě určena jednoznačně
- jelikož *R* následuje prakticky vždy po *C*, často se při popisu sítě ani neuvádí

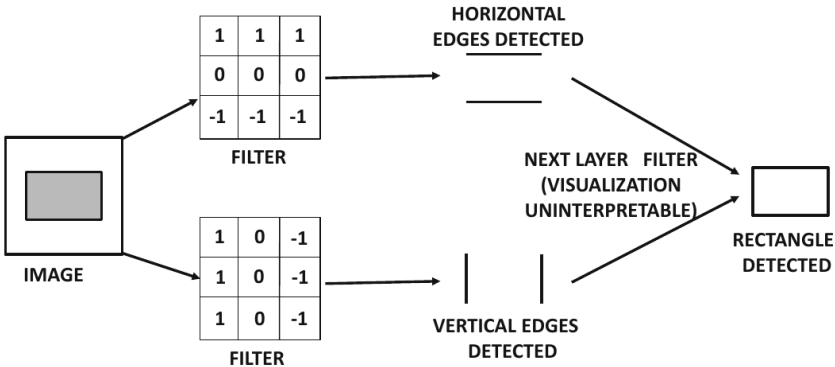


LeNet-5

Ch.C.Aggarwal, Neural Networks and Deep Learning.

CNN

- filtry na nižších vrstvách se učí reagovat na jednoduché tvary jako hrany (Gaborovy filtry)
- filtr na první vrstvě vidí pouze $F_1 \times F_1$ pixelů
- nemůže se naučit reagovat na něco většího
- jednotlivé vrstvy tak budují vyšší míru abstrakce
  - hrany
  - rohy, oblouky
  - polygony
  - ...
  - kola, okna
  - typy automobilů

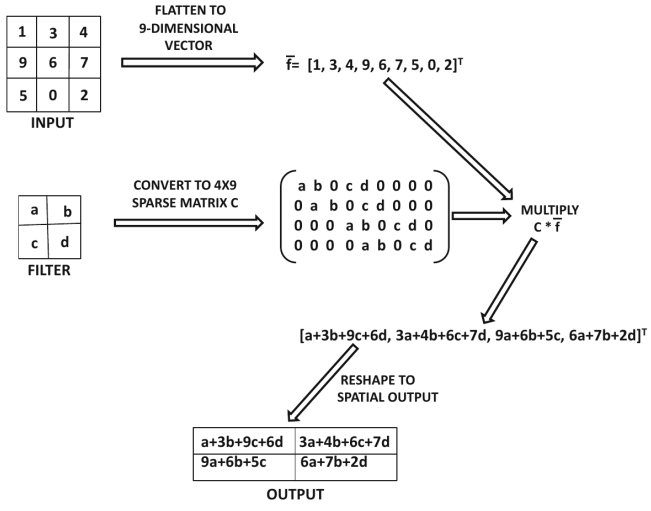


Ch.C.Aggarwal, Neural Networks and Deep Learning.

CNN - trénování

- za účelem odvození algorittmu backpropagation si přeformulujeme CNN do maticového tvaru

CNN - trénování



Ch.C.Aggarwal, Neural Networks and Deep Learning.

CNN - trénování

- pak se na CNN můžeme dívat stejně jako na úplně propojenou síť pouze s tím, že místo husté matice je zde matice řídká
- navíc každý řádek obsahuje stejné parametry, tj. každý řádek přispívá ke stejným složkám gradientu
- tento přístup by nebyl efektivní pro samotnou implementaci, ale nám pro představu postačí

CNN - rozšiřování trénovací sady

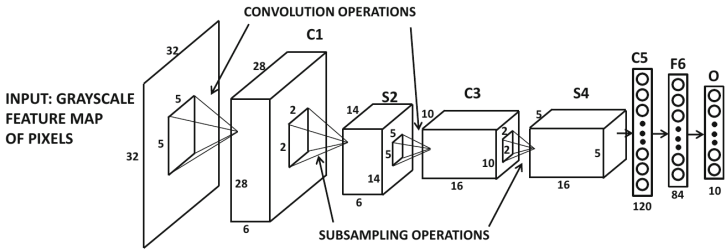
Data augmentation

- jde o postup, kdy si pomocí transformací zachovávajících význam dat generujeme nová data pro trénování
- například různé rotace nebo translace nemění význam obrázku
  - pokud nejde například o rozpoznávání číslic 6 a 9
- tím lze snadno významně zvětšit množinu trénovacích dat zamezit přetrénování sítě
- transformace lze často provádět *on-the-fly*

CNN architektury

- návrh konvolučních sítí je o něco složitější, než pokud jde o plně propojené sítě
- ukážeme si příklady některých úspěšných sítí, které výrazně posunuly schopnosti rozpoznávání obrazu
- jde často o vítěze soutěže ImageNet Large Scale Visual Recognition Challenge (ILSVRC)
  - https://www.image-net.org/challenges/LSVRC
- zřejmě první CNN byl Neocognitron - 1988
  - K. Fukushima, Neocognitron:A Hierarchical Neural Network Capable of Visual Pattern Recognition

CNN architektury - LeNet-5



LeNet-5

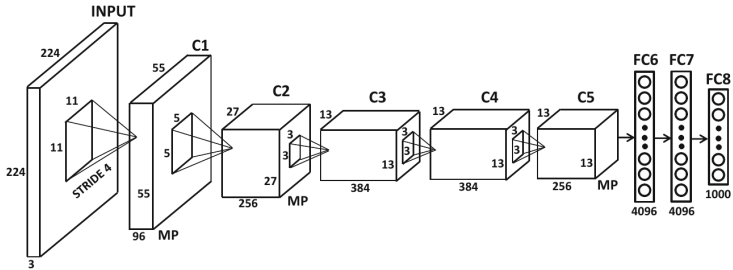
Ch.C.Aggarwal, Neural Networks and Deep Learning.

Y. LeCun, L. Bottou, Y.Bengio, P.Haffner, Gradient-Based Learning Applied to Document Recognition, Proc. of the IEEE, 1998

CNN architektury - LeNet-5

- na poměry pozdějších CNN jde o velice mělkou síť
- vstup je jednokanálový obrázek ve stupních šedi
- síť obsahuje
  - 2 konvoluční vrstvy - C1 a C3
  - 2 vrstvy pro pooling - S2 a S4 - subsampling pomocí průměrování
  - 3 plně propojené vrstvy - C5, F6, O
  - jako aktivační funkci používá sigmoid
- LeNet-5 byl využíván např. v bankách pro čtení šeků

CNN architektury - AlexNet



AlexNet

Ch.C.Aggarwal, Neural Networks and Deep Learning.

A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, 2012.

CNN architektury - AlexNet

- AlexNet vyhrála ILSVRC v roce 2012
- má pět kovolunčních vrstev
- vrstvy označené MP mají za sebou max-pooling
- za každou vrstvou se aplikue ReLU
  - AlexNet ukázala výhodu ReLU, do té doby je ReLU dominanntí v CNN
- za konvolučníma vrstvama následují tři plně propojené vrstvy
- na výstupu je použitý softmax
- vrstva FC7 se často požívá pro obecnou reprezentaci obrázku vektorem o velikosti 4096
- výstup na FC7 se často bere jako vstup pro různé DNN pro zpracování obrazu
- obecně se takový vektor pak označuje jako FC7 features

CNN architektury - AlexNet

AlexNet přinesla několik zásadních novinek

- použití ReLU
- *data augmentation* pro učení sítě
- dropout

ILSVRC top-5 error

- CNN dá ke každému obrázku pravděpodobnosti, že obrázek patří do určité třídy
- testujeme, jestli se správná třída nachází mezi 5 třídami, které CNN predikovala jako nejrpavděpodobnější
- procento případů, kdy tomu tak není, se označuje jako *top-5 error rate*
- AlexNet měla top-5 error rate 15.4%, zatímco předchůdci 25%
- AlexNet zaznamenala nejvýraznější posun v kvalitě rozpoznávání obrazu a právě ona odstartovala velký zájem o CNN, ale i DNN obecněji

CNN architektury - ZFNet

	AlexNet	*ZFNet*
Volume: Operations:	$224 \times 224 \times 3$ Conv 11×11 (stride 4)	$224 \times 224 \times 3$ Conv 7×7 (stride 2), MP
Volume: Operations:	$55 \times 55 \times 96$ Conv 5×5, MP	$55 \times 55 \times 96$ Conv 5×5 (stride 2), MP
Volume: Operations:	$27 \times 27 \times 256$ Conv 3×3, MP	$13 \times 13 \times 256$ Conv 3×3
Volume: Operations:	$13 \times 13 \times 384$ Conv 3×3	$13 \times 13 \times 512$ Conv 3×3
Volume: Operations:	$13 \times 13 \times 384$ Conv 3×3	$13 \times 13 \times 1024$ Conv 3×3
Volume: Operations:	$13 \times 13 \times 256$ MP, Fully connect	$13 \times 13 \times 512$ MP, Fully connect
FC6: Operations:	4096 Fully connect	4096 Fully connect
FC7: Operations:	4096 Fully connect	4096 Fully connect
FC8: Operations:	1000 Softmax	1000 Softmax

Ch.C.Aggarwal, Neural Networks and Deep Learning.

M. D. Zeiler, R. Ferguson, Visualizing and Understanding Convolutional Networks,European Conference on Computer Vision, pp. 818-833, 2014.

CNN architektury - ZFNet

- ZFNet vyhrála ILSVRC v roce 2013
- je velmi podobná síti AlexNet
- top-5 error zmenšila na 14.8% a později i na 11.1%
- tato síť ukazuje, jak malé změny architektury mohou výrazně zlepšit výsledek
- to je vlastně důvod, proč se u CNN tak často používají standardní architektury

CNN architektury - VGG

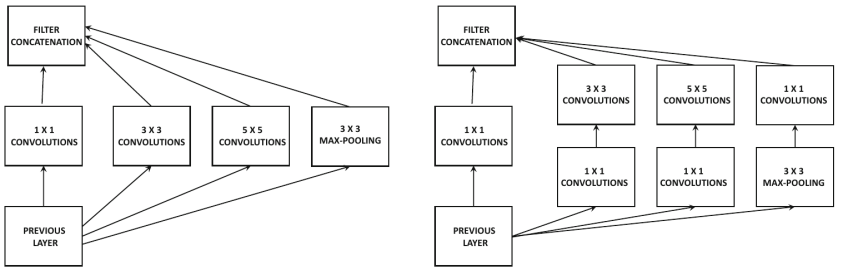
- u této sítě autoři navyšovali počet vrstev na 11 až 19, nelepší výsledky dosáhli s 16
- tato síť se umístila jako druhá na ILSVRC v roce 2014
- dosáhla s top-5 error rate na hodnotu 7.3%
- VGG zmenšuje velikost konvolučních filtrů a kompenzuje to vetším počtem vrstev
  - když po sobě aplikuju dva konluční filtry 2x2, ten druhý má vlastně efektivní suport 4x4
- větší hloubka sítě umožňuje více nelinearit
- trénování hlubokých sítí je velmi citlivé na počáteční nastavení
- proto se nejprve předtrénovalo několik prvních vrstev a potom se dotrénovala celá síť

CNN architektury - GoogLeNet

- tato síť využívá tzv. *inception architecture*
- základní myšlenka je taková, že pro úspěšné klasifikování potřebujeme informace na různých úrovních abstrakce
  - drobnější detaily lépe zachytíme větším množstvím menších filtrů
  - na vyšší abstrakci potřebujeme větší filtry
- pro různé typy obrázků se toto ale liší a my nevíme dopředu, jaká úroveň detailů/abstrakce je výhodnější
- GoogLeNet používá inception moduly, které v jedné vrstvě kombinují filtry různé velikosti

Ch. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going Deeper with Convolutions, 2014.

CNN architektury - GoogLeNet



(a) Basic inception module (b) Implementation with 1×1 bottlenecks

Ch.C.Aggarwal, Neural Networks and Deep Learning.

CNN architektury - GoogLeNet

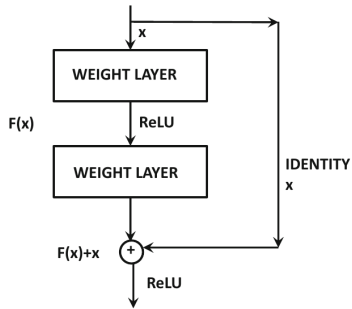
- GoogLeNet se skládá s 9 inception bloků jdoucích po sobě
- ceste skrze ně může vést přes různě velké filtry
  - pokud jdu přes filtry velikosti 1x1, extrahuji drobné detaily
  - pokud jdu přes několik filtrů 5x5 extrahuji vyšší abstrakci
- za účelem redukce počtu parametrů se používá redukční 1x1 konvoluce - snižuje počet features
- celkově obsahuje GoogLeNet o jeden řád méně parametrů než VGG
- GoogLeNet vyhrála ISLVRC 2014 a top-5 rate error 6.7%

CNN architektury - ResNet

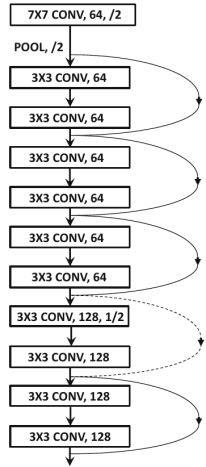
- tato síť vyhrála ISLVRC 2015 a top-5 rate error 3.6%
- šlo o první síť, která se přesností vyrovnala člověku
- má 152 vrstev
- autoři opět řešili problém adaptivního zachycení různé úrovně detailů/abstrakce
- zde se rozhodli umožnit adaptivní hloubku sítě
- k tomu se používají *skip connections*, tj. možnost přeskočit některé vrstvy

K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, Microsoft Research, 2016.

CNN architektury - ResNet

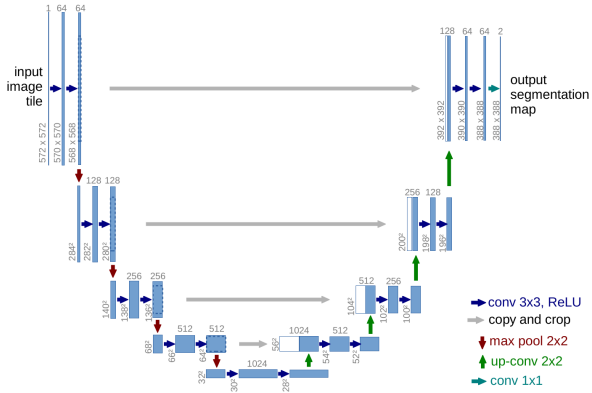


(a) Skip-connections in residual module



(b) Partial architecture of *ResNet*

CNN architektury - U-Net



O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, arXiv:1505.04597.

CNN architektury - U-Net

- hlavní výhoda sítě U-net je, že se dokáže trénovat rychle na relativně malém počtu obrázků
- ve své druhé půlce dělá tzv. upsampling pro přesnější segmentaci
- trochu se tam možná podobá autoeknodéru, i když autoři nic takového nezmiňují
- síť navíc používá efektivně skip-connections

CNN architektury - porovnani

Name	Year	Number of Layers	Top-5 Error
–	Before 2012	≤ 5	$> 25\%$
AlexNet	2012	8	15.4%
ZfNet/Clarifai	2013	$8/>8$	14.8% / 11.1%
VGG	2014	19	7.3%
GoogLeNet	2014	22	6.7%
ResNet	2015	152	3.6%

- roky 2012-2015 přinesly nevídané zlepšení v oblasti rozpoznávání obrazu
- bylo to možné zejména díky nárůstu hloubky NN
- proto se začal používat termín *deep learning*
- CNN patří mezi nejhlubší NN

Předtrénování sítí

- hlavním problémem CNN je potřeba obrovského počtu labelovaných dat
- v případě CNN se ale první vrstvy učí nizkou úroveň abstrakce, která se příliš neliší pro různé typy obrazových dat
- je tedy možné síť předtrénovat pomocí některé obecné databáze obrázků
  - ImageNet obsahuje více než milion obrázku klasifikovaných do 1000 kategorií
- na obecné databázi lze natrénovat základní vrstvy nebo i všechny konvoluční vrstvy
- s doménově specifickou databází se pak dotrénují jen FC vrstvy nebo pár vyšších konvolučních
- výstup z konvolučních vrstev se často označuje jako *FC7 features* a často se bere jako vstup pro plně propojené sítě
- například balík Caffe obsahuje "zoo" předtrénovaných sítí

Vizualizace CNN

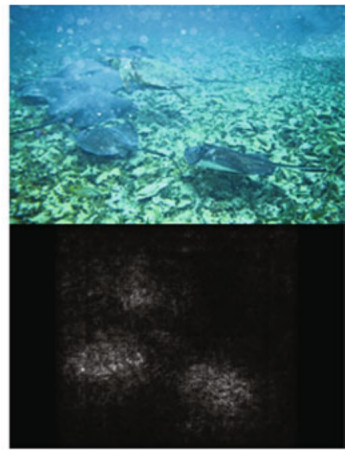
- u CNN lze poměrně úspěšně vizualizovat naučené parametry
- na nižších úrovních stačí pouze zobrazit naučené konvoluční filtry
- ukazuje se, že ty se často velice podobají Gaborovým filtrům 1, 2
  - některé CNN používají na prvních vrstvách parametrizované Gaborovy filtry
- na vyšších vrstvách s vyšší úrovní abstrakce již přímá vizualizace filtrů nedává dobrý smysl
- místo toho ale můžeme vizualizovat, na které pixely na vstupu nejvíce reagují některé skryté parametry nebo rysy (features)
- to lze snadno zjistit pomocí
  - citlivostní analýzy, tj. výpočtem gradientu dané váhy/feature vůči pixelům vstupního obrazu
  - natrénováním duální CNN pomocí autoenkodéru

Vizualizace CNN

Vizualizace rysů (features) natrénované sítě

- pro daný obrázek a aktivaci sítě, pro určitou feature uvnitř sítě, najdi ty části obrázku, na které daná feature nejvýraznější odezvu
- pro libovolnou feature uvnitř CNN, najdi obrázek, který jí aktivuje nejvíce

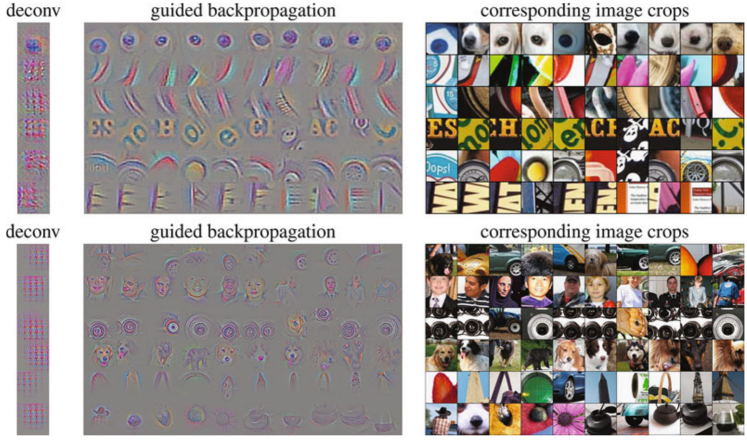
Vizualizace CNN



Ch.C.Aggarwal, Neural Networks and Deep Learning.

Vizualizace CNN

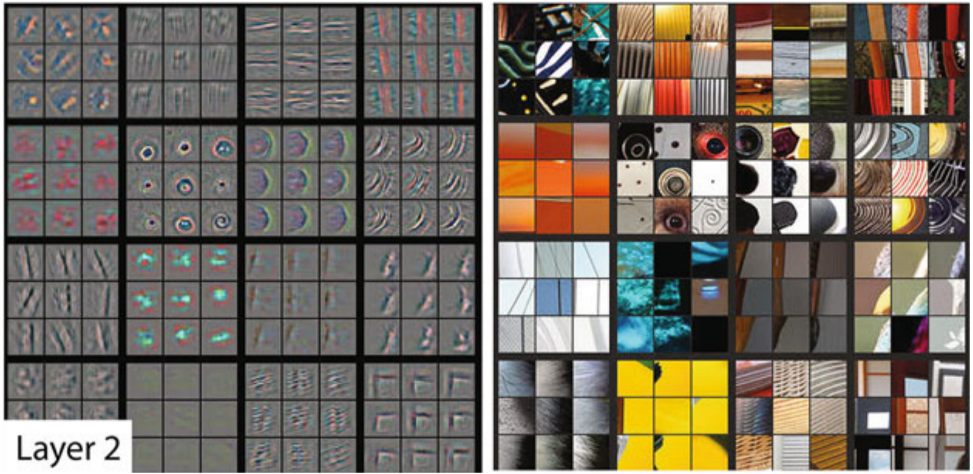
Výřezy obrázků, na které silně reagoval některý neuron (stejný v jednom řádku) uvnitř CNN. Šedá část zobrazuje senzitivitu na jednotlivé pixely.



Ch.C.Aggarwal, Neural Networks and Deep Learning.

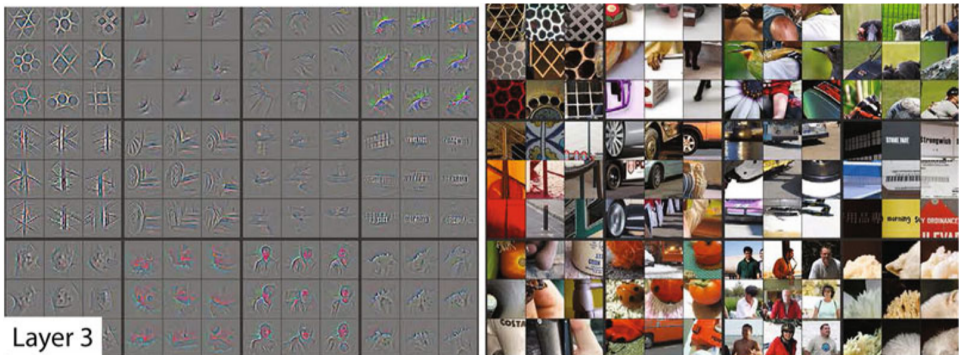
Vizualizace CNN

Vizualizace CNN



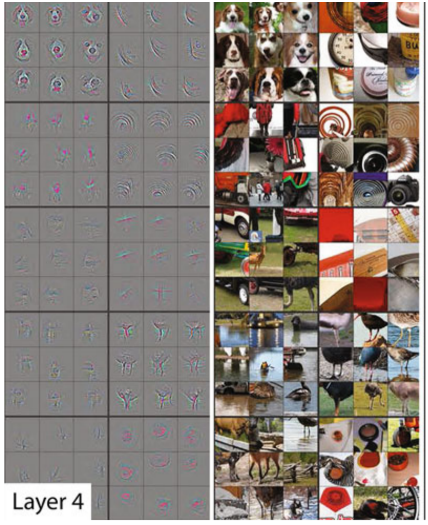
Ch.C.Aggarwal, Neural Networks and Deep Learning.

Vizualizace CNN

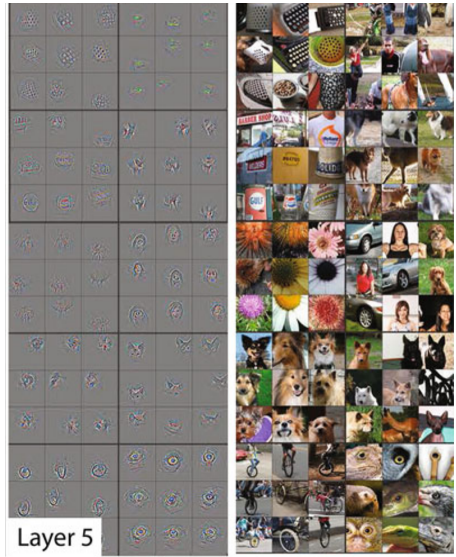


Ch.C.Aggarwal, Neural Networks and Deep Learning.

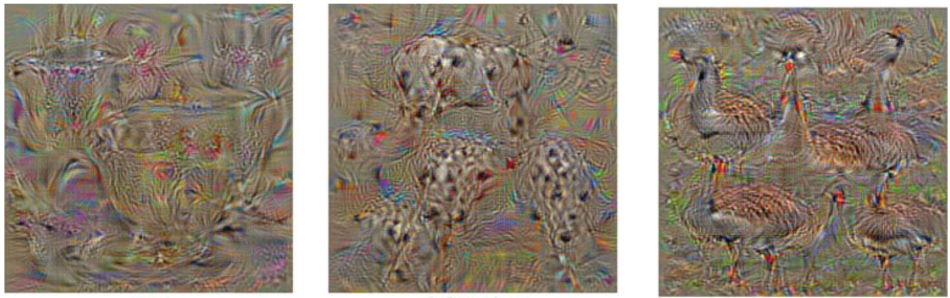
Vizualizace CNN



Vizualizace CNN



Vizualizace CNN



Ch.C.Aggarwal, Neural Networks and Deep Learning.

Vyhledávání obrázků

- cílem je najit obrázky se stejným obsahem jako zadaný obrázek
- lze použít *feature vectors*, které vystupují z prvních několika vrtstev konvolučních sítí
- lze použít např. i předtrénované sítě jako *AlexNet*
- pak už jde o porovnávání mezi *feature vectors*

A. Babenko, A. Slesarev, A. Chigorin, V. Lempitsky, *Neural Codes for Image Retrieval*, Computer Vision – ECCV 2014 pp 584–599, Springer, 2014.

Lokalizace objektů

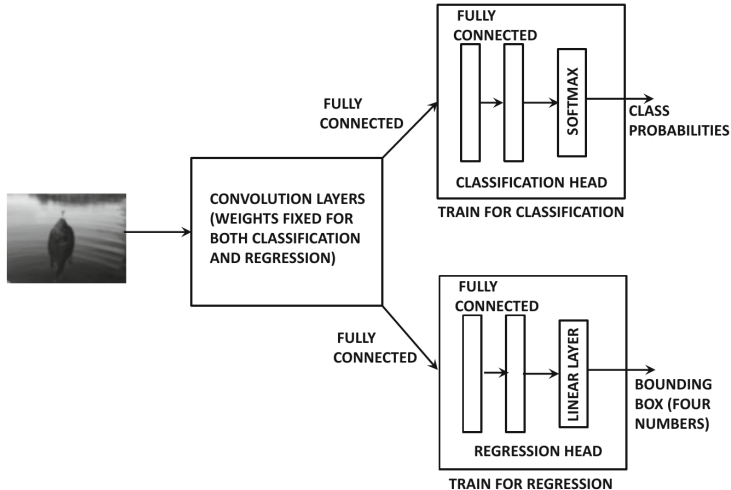
- máme obrázek a fixní počet objektů na obrázku
- ke každému objektu chceme najít nejmenší obdélník, který obsahuje daný objekt
- každou obálku lze popsat jejím **horním rohem a rozměry** $\equiv$ 4 integery
- výstupem je tedy $4k$ čísel, tj. jde o regresi
- jde tedy o podobnou úlohu jako je klasifikace jen s jiným výstupem

Lokalizace objektů

Lze postupovat takto:

- natránujeme klasifikátor jako *AlexNet* nebo vezmeme předtrénovanou síť
- poslední tři vrstvy u *AlexNetu* jsou zodpovědné právě za klasifikaci, jde o tzv. *classification head*
- tuto část odstraníme a nahradíme plně propojenými vrstvami bez posledního softmaxu
- tím dostaneme tzv. *regression head*
- natrénujeme pouze regression head na sadě obrázků s označenými boxy
- nyní do sítě zapojíme oba konce, tj. *classification head* i *regression head*
- případně je možné dotrénovat celou síť, tj. konvoluční vrstvy, klasifikační i regresní vrstvy na sadě obrázků, které obsahují jak třídy objektů tak obalující boxy
- následně je možné síť použít na lokalizaci objektů

Lokalizace objektů



Ch.C.Aggarwal, Neural Networks and Deep Learning.

Detekce objektů

- v obrázku máme detekovat přítomnost objektů z daných tříd - např. auto, kočka, ryba
- komplikace je v tom, že předem neznáme počet těchto objektů a neuronové sítě nedokážou vracet variabilní počet parametrů
- proto se nejprve použije tzv. *region proposal method*
- ta vytvoří množinu boxů jako kandidátu, kde by mohl být obsažen nějaký objekt
- to se často děje na základě podobnosti pixelů
- následně se zpracuje pomocí CNN každý kandidát
- jednotlivé výsledky je pak ještě nutné správně zakomponovat dohromady

Zpracování textu pomocí CNN

- existují i pokusy zpracovávat text pomocí CNN
- jednotlivá slova lze kódovat pomocí *one-hot-encoding*
- dostaneme tak 1D posloupnost vektorů o dimenzi rovné počtu slov ve slovníku
- na tuto posloupnost pak můžeme aplikovat konvoluční masky podobně jako u CNN pro zpracování obrazu
- nevýhodou je velká dimenze těchto vektorů - řádově 10^6
- používají se proto efektivnější postupy pro kódování slov do vektorů - *word2vec*, *GLoVe*

Klasifikace videa

- CNN lze také použít na video $\equiv$ sekvenci obrazových snímků
- místo 2D konvolučních mask tak používáme 3D masky, tj. 2 dimenze pro prostor a 1 dimenze pro čas
- nevýhodou je že, výrazně narůstá velikost konvolučních masek a je proto potřeba i více dat k natrénování
- analýza pohybu navíc často příliš nepomáhá k lepšímu pochopení obsahu, často stačí zpracovat několik i náhodně vybraných statických obrázků