

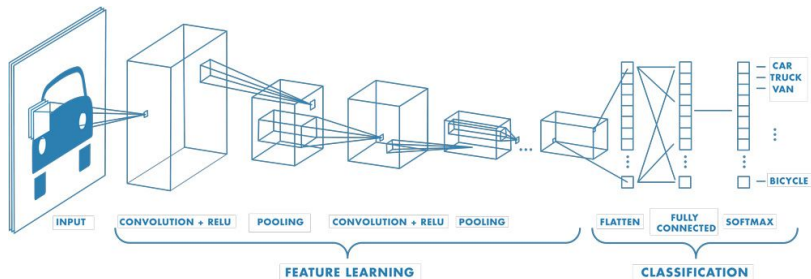
Co jsme probírali minule

- 1 Shlukování - prohloubení látky
 - algoritmus k středů
 - hierarchické shlukování
- 2 Hybridní modely umělých neuronových sítí (kombinace učení s učitelem a bez učitele):
 - LVQ (Učení vektorové kvantizace)
 - Sítě se vstřícným šířením (Counter-propagation)
 - RBF-sítě
 - ART (Adaptive Resonance Theory)
 - ...

Zbývá (minule jsme nestihli): modulární neuronové sítě

- ART (Adaptive Resonance Theory)
- Kaskádová korelace

Dnešní hodina: úvod do konvolučních neuronových sítí



Zdroj :

<https://matlabacademy.mathworks.com/details/deep-learning-onramp/deeplearning>

Motivační příklad: klasifikace obrázků

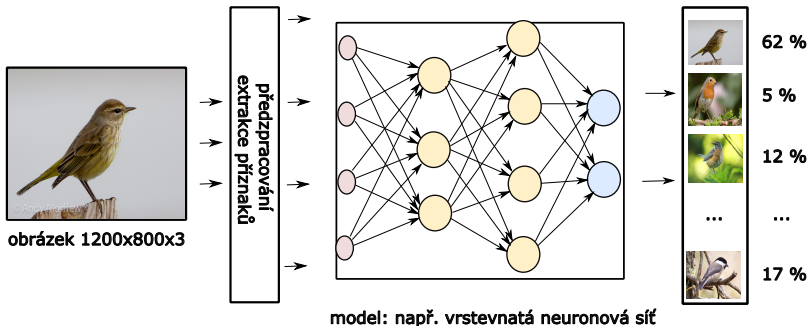
Rozpoznávání pěvců



Motivační příklad: klasifikace obrázků

Rozpoznávání pěvců

Klasický přístup ve strojovém učení



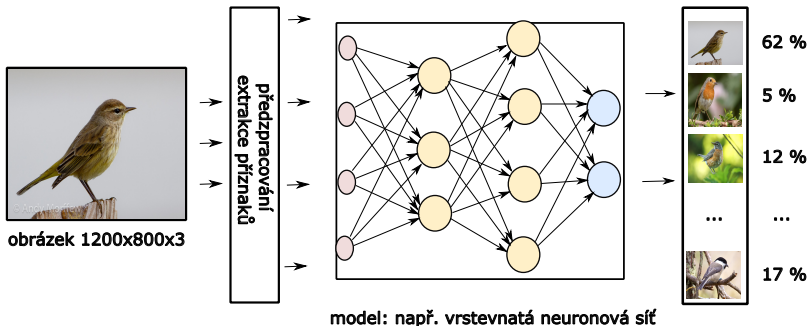
Extrakce příznaků

- detekce hran, LBP histogramy,...
- poměrně náročné na návrh, ztráta informace

Motivační příklad: klasifikace obrázků

Rozpoznávání pěvců

Co takhle učit neuronovou síť přímo?



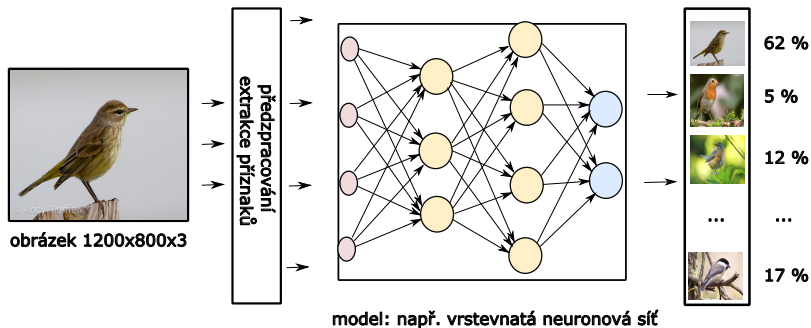
Princip hlubokého učení

- extrakci příznaků necháme na modelu

Motivační příklad: klasifikace obrázků

Rozpoznávání pěvců

Co takhle učit neuronovou síť přímo?



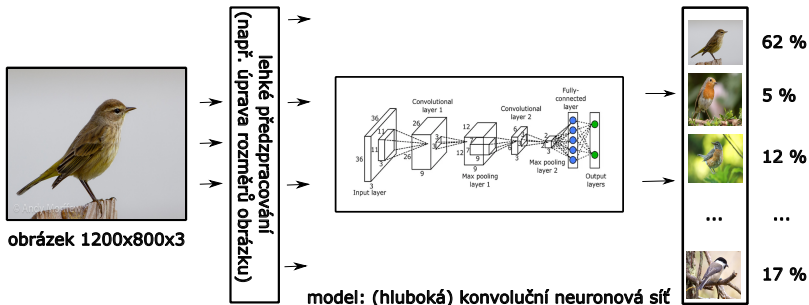
Nevýhody klasického přístupu (plně propojené vrstvy):

- velký počet příznaků
- ztráta informace o vzájemné poloze pixelů
- obtížné učení

Motivační příklad: klasifikace obrázků

Rozpoznávání pěvců

Jak by to šlo udělat lépe?

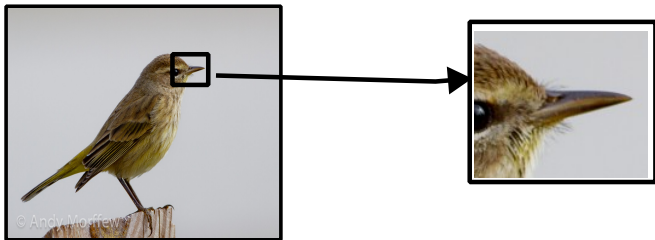


Konvoluční neuronová síť:

- neuronová síť s konvolučními vrstvami
- bere v potaz prostorové rozložení pixelů
- méně parametrů a snadnější učení než v případě plně

Motivační příklad: Rozpoznávání pěvců

V datech jsou vzory: např. zobáček



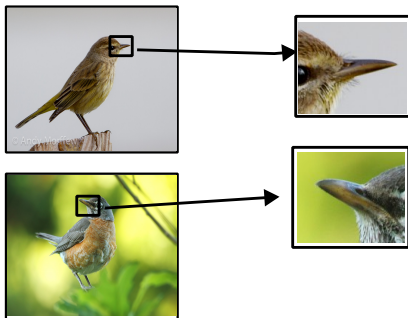
Co vytvořit detektor zobáčku:

- jednoduchý model (např. jednovrstvá neuronová síť), co najde na obrázku zobáček

→ zobáček může být na různých místech v obrázku

Motivační příklad: Rozpoznávání pěvců

V datech jsou vzory: např. zobáček

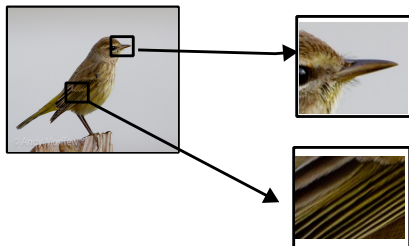


zobáček může být na různých místech v obrázku

- detektor by měl najít zobáček v libovolném obrázku a na libovolném místě v obrázku
→ detektor se bude pohybovat zpracovávaným obrázkem

Motivační příklad: Rozpoznávání pěvců

V datech jsou různé vzory:



Myšlenka

- vytvořím množinu detektorů pro různé příznaky (vzory)
- detektory by měly najít příznak v libovolném obrázku a na libovolném místě v obrázku → detektory se budou pohybovat zpracovávaným obrázkem
- detektory budou tvořit počáteční vrstvy konvoluční neuronové sítě

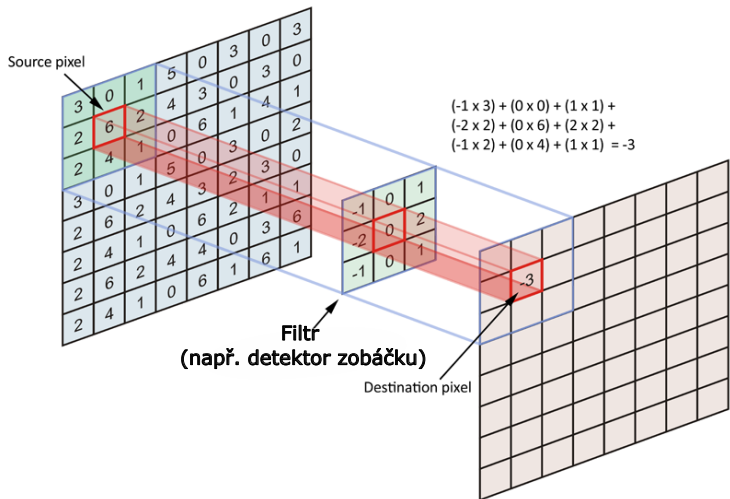
Konvoluční neuronová síť

- neuronová síť obsahující konvoluční vrstvy

Konvoluční vrstva

- tvořená skupinou filtrů (jader, detektorů)
- filtry provádí operaci konvoluce nad vstupním obrázkem
- do další vrstvy postupuje výsledek konvoluce

Konvoluce



Konvoluce

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

Obrázek 6x6 (zjednodušeně)

Konvoluční vrstva

1	-1	-1
-1	1	-1
-1	-1	1

Filtr 1 (3x3)

-1	1	-1
-1	1	-1
-1	1	-1

Filtr 2 (3x3)

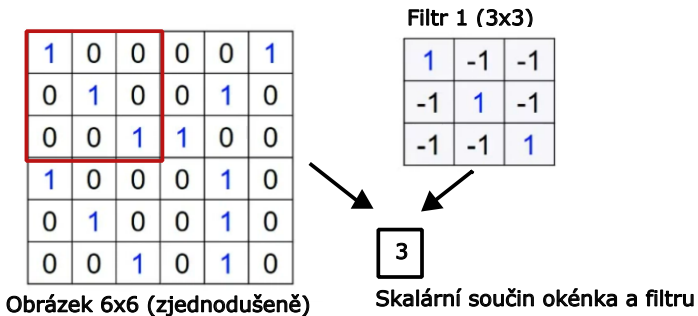
■ ■ ■

- každý filtr detekuje vzor (příznak) velikosti 3x3 pixely

zdroj příkladu na násl. slidech: Petr Doležel: Konvoluční neuronová síť,

<https://www.youtube.com/watch?v=-2vEi-Aa0FA>

Operace konvoluce



- $y = \sum_{i=1}^9 w_i x_i + b$ (pro linearizované matice)

Operace konvoluce

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

Obrázek 6x6 (zjednodušeně)

Filtr 1 (3x3)

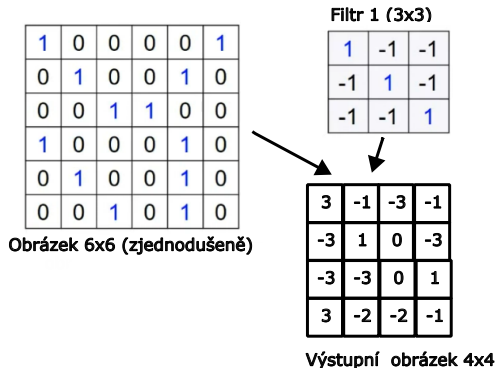
1	-1	-1
-1	1	-1
-1	-1	1

3
-1

Skalární součin okénka a filtru

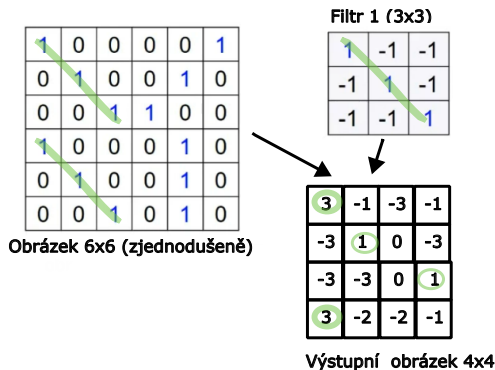
- okénko posunu a opět spočítám skalární součin

Operace konvoluce



- postupným posouváním okénka aplikuji filtr na celý obrázek
- získám matici příznaků (feature map)

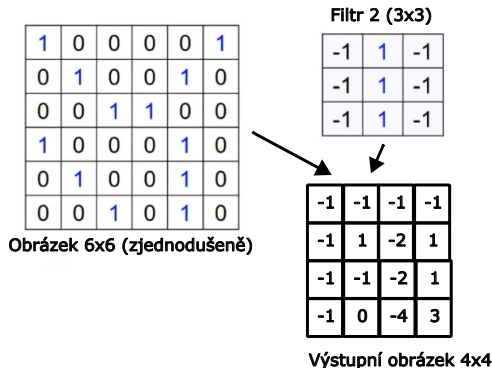
Operace konvoluce



Matice příznaků (feature map)

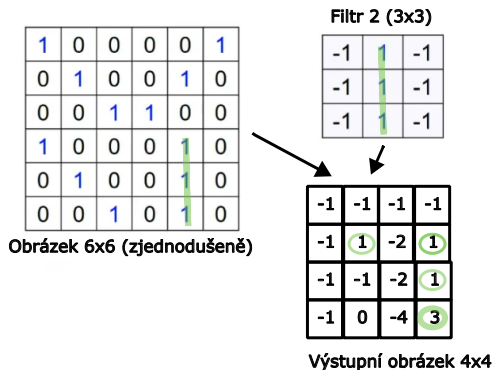
- určuje, kde se v původním obrázku vyskytuje vzor reprezentovaný filtrem
- zde: hranový filtr pro diagonální hranu

Operace konvoluce



- podobně mohou aplikovat druhý filtr

Operace konvoluce

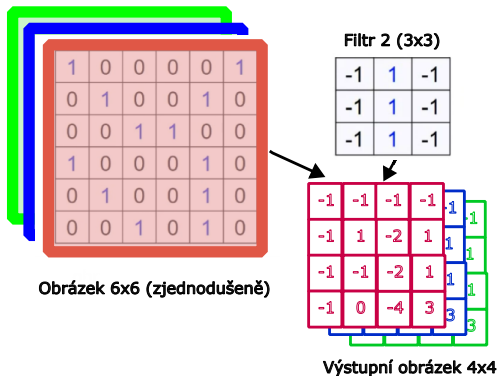


Druhá matice příznaků (feature map)

- určuje, kde se v původním obrázku vyskytuje vzor reprezentovaný filtrem
- zde: hranový filtr pro svislou hranu

Operace konvoluce

Barevný obrázek: 3 kanály (channels) R, G, B



- každý filtr aplikuji na každou barevnou složku (channel) zvlášť
→ získám větší množství matic příznaků

Operace konvoluce

Příklad: zebra

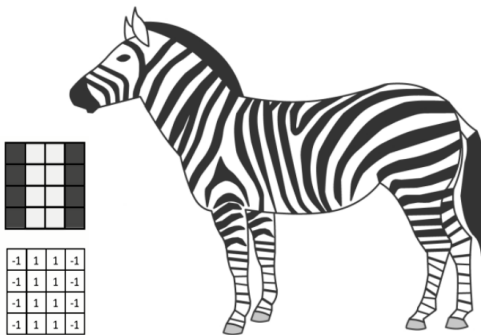


Zdroj příkladu:

<https://matlabacademy.mathworks.com/details/deep-learning-onramp/deeplearning>

Operace konvoluce

Příklad: zebra



- aplikujeme filtr pro detekci svislého proužku

Operace konvoluce

Příklad: zebra



-1	1	1	-1
-1	1	1	-1
-1	1	1	-1
-1	1	1	-1



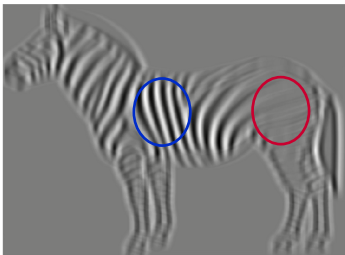
- takto vypadá výsledek aplikace filtru

Operace konvoluce

Příklad: zebra



-1	1	1	-1
-1	1	1	-1
-1	1	1	-1
-1	1	1	-1



- snadno rozeznáme oblasti, kde je vzor výrazně zastoupen a oblasti, kde není

Operace konvoluce

Příklad: zebra



Vertical stripes

Diagonal cross

Horizontal edge

White blob

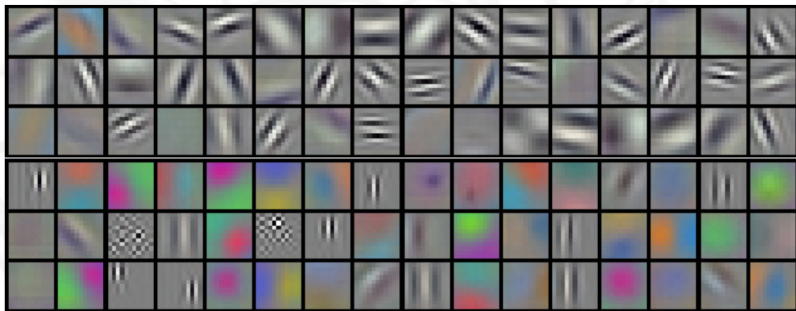
Diagonal edge



- ukázky dalších filtrů a výsledných matic příznaků

Operace konvoluce

- **Ukázka:** 96 filtrů $11 \times 11 \times 3$ v první konvoluční vrstvě u AlexNet



Zdroj obrázku: Alex Krizhevsky et al.: "ImageNet Classification with Deep Convolutional Neural Networks", Figure 3

Operace konvoluce

Poznámka: konvoluční operace, které se používají mimo konvoluční sítě

- identita

0	0	0
0	1	0
0	0	0

- blur (rozmazání)

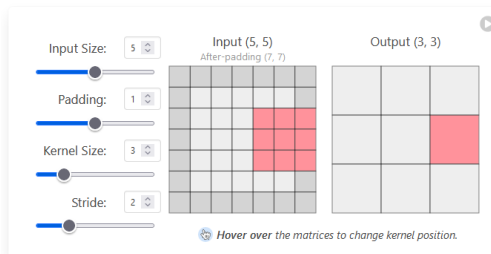
1/8	1/8	1/8
1/8	1/8	1/8
1/8	1/8	1/8

Operace konvoluce

Parametry konvoluční operace

- rozměry vstupního obrázku
- padding - okraje obrázku
- rozměry filtru
- stride = krok, pomocí kterého procházíme obrázek

Understanding Hyperparameters

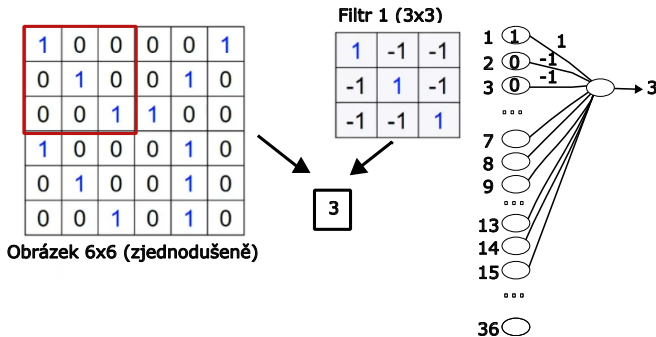


pěkná vizualizace: <https://poloclub.github.io/cnn-explainer/>

Konvoluční vrstva vs. plně propojená vrstva

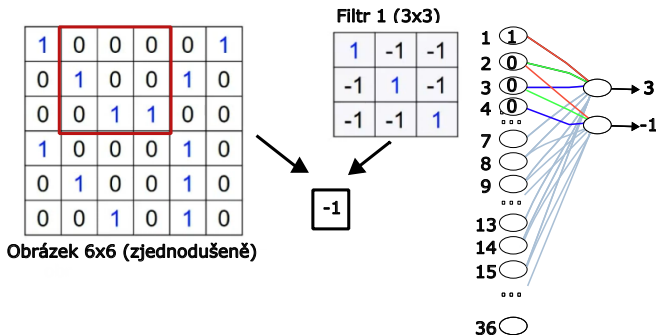
- na konvoluční vrstvu se můžeme dívat jako na klasickou vrstvu neuronové sítě
- neurony ale nejsou plně propojené

→ mnohem méně parametrů

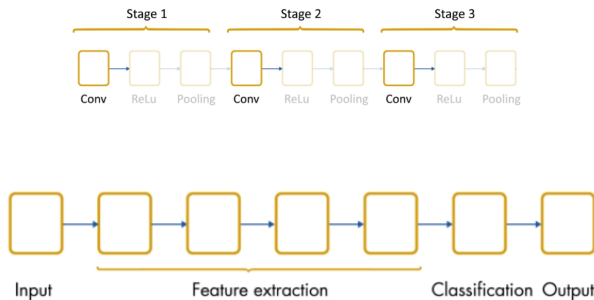


Konvoluční vrstva vs. plně propojená vrstva

- na konvoluční vrstvu se můžeme dívat jako na klasickou vrstvu neuronové sítě
 - „fiktivní“ neurony ve skryté vrstvě sdílejí stejné váhy
- ještě méně parametrů, efektivnější učení

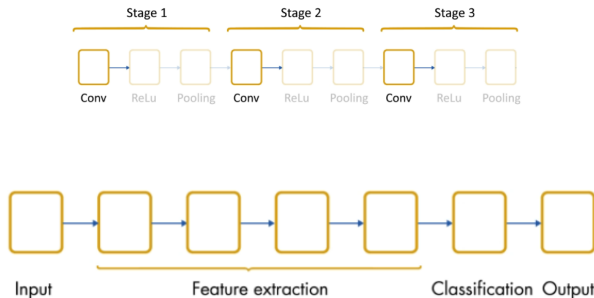


Typická architektura konvoluční neuronové sítě



- konvoluční vrstvy vrstvíme na sebe
- první vrstva detekuje jednoduché příznaky, např. hrany, bloby
- každá další vrstva extrahuje příznaky (vzory) vyšší úrovně

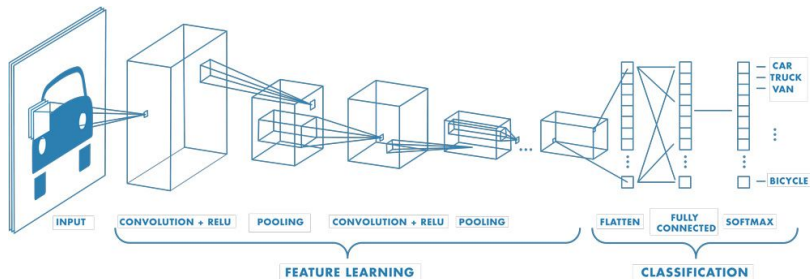
Typická architektura konvoluční neuronové sítě



Typická struktura konvolučního bloku:

- konvoluční vrstva
- aplikace nelineární přenosové funkce (např. ReLU)
- pooling vrstva

Typická architektura konvoluční neuronové sítě



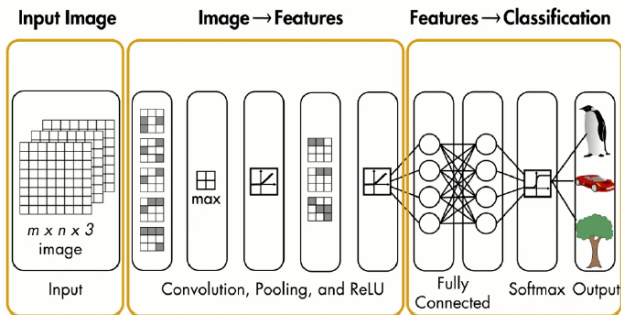
Části konvoluční neuronové sítě

- Konvoluční vrstvy pro extrakci příznaků
- Flattening vrstva - převede data na vektor čísel
- Vrstevnatá neuronová síť pro klasifikaci

Zdroj :

<https://matlabacademy.mathworks.com/details/deep-learning-onramp/deeplearning>

Typická architektura konvoluční neuronové sítě



Části konvoluční neuronové sítě

- Konvoluční vrstvy pro extrakci příznaků
- Flattening vrstva - převede data na vektor čísel
- Vrstevnatá neuronová síť pro klasifikaci

Zdroj :

<https://matlabacademy.mathworks.com/details/deep-learning-onramp/deeplearning>

Konvoluční blok

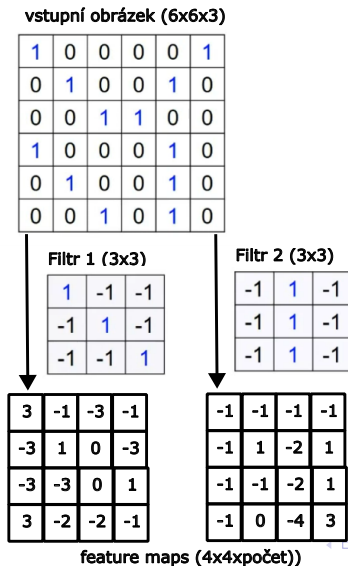
Typická struktura konvolučního bloku:

- konvoluční vrstva
- aplikace nelineární přenosové funkce (např. ReLU) - aby nebyla transformace dat jen lineární
- pooling vrstva

Konvoluční vrstvy

- Pokud je vrstvíme za sebe, s každou další vrstvou roste počet feature maps
- Nemění velikost obrázku (až na okraje, v závislosti na padding)
s každou další vrstvou roste počet parametrů

Konvoluční blok - ukázka I.



Konvoluční blok

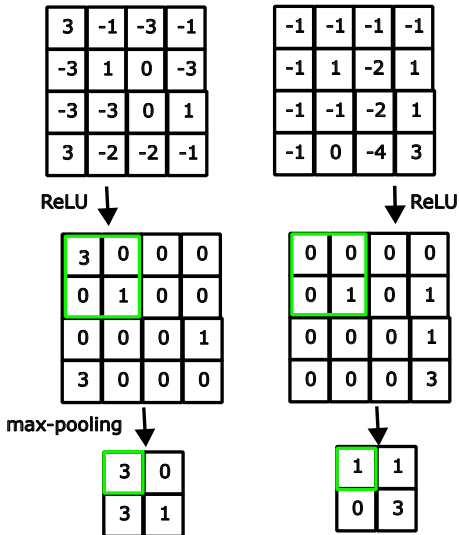
Typická struktura konvolučního bloku:

- konvoluční vrstva
- aplikace nelineární přenosové funkce (např. ReLU) - aby nebyla transformace dat jen lineární
- pooling vrstva

Pooling (subsampling) vrstva

- Slouží k zefektivnění výpočtu, zmenšuje rozlišení obrázku, aniž by se (příliš) zmenšila přenášená informace
- opět posouváme okénko, tentokrát např. velikosti 2×2 , bez překryvu (stride = 2)
- operace MAX (max-pooling) nebo AVERAGE (average-pooling), žádné váhy

Konvoluční blok - ukázka II.



Konvoluční blok

Pooling (subsampling) vrstva

- Zahušťuje obrázek při zachování informace (dat) - kde a jak silně se v obrázku vyskytuje příznak (vzor)
- zmenšuje data ($2 \times 2 \rightarrow$ na čtvrtinu)

Střídání konvoluční a pooling vrstvy: bipyramidální efekt

- postupně se zmenšují obrázky a zvětšuje se počet feature maps

Učení konvoluční neuronové sítě

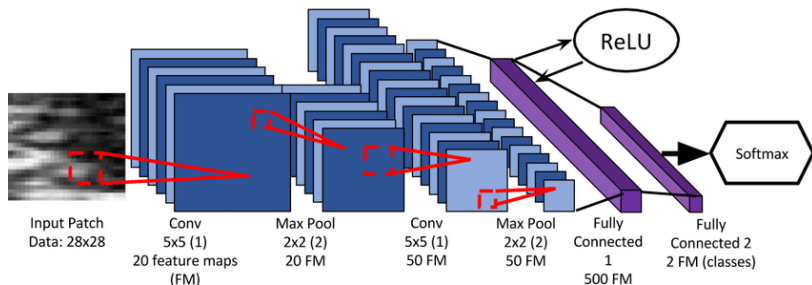
- nějaká varianta algoritmu zpětného šíření (např. SGD)
- mini-batch učení, model potřebuje k naučení větší množství dat
- velké množství parametrů

Jak zvolit vhodnou architekturu v praxi?

- neoptimalizujeme počet vrstev a neuronů
- vybereme z dostupné literatury topologii osvědčenou pro daný typ problému

LeNet 5

- jedna z původních architektur (Yann LeCun, 1998), poměrně jednoduchá



Zdroj obrázku: M. H. Yap et al., "Automated Breast Ultrasound Lesions Detection Using Convolutional Neural Networks," in IEEE Journal of Biomedical and Health Informatics, vol. 22, 2018.

LeNet 5

Datová sada MNIST

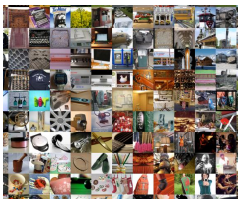
- Oblíbená benchmarková datová sada pro porovnávání různých modelů (přesnost klasifikace)
<http://yann.lecun.com/exdb/mnist/>
- 60000 označovaných trénovacích obrázků ručně psaných číslic, 10000 testovacích (od jiných lidí)
- obrázky 28x28, vycentrované a normalizované na danou velikost

0 0 0 0 0 0 0 0 0 0
 1 1 1 1 1 1 1 1 1 1
 2 2 2 2 2 2 2 2 2 2
 3 3 3 3 3 3 3 3 3 3
 4 4 4 4 4 4 4 4 4 4
 5 5 5 5 5 5 5 5 5 5
 6 6 6 6 6 6 6 6 6 6
 7 7 7 7 7 7 7 7 7 7
 8 8 8 8 8 8 8 8 8 8
 9 9 9 9 9 9 9 9 9 9

ImageNet Large-Scale Visual Recognition Challenge (ILSVRC, 2010-2017)

Datová sada ImageNet

- 16 milionů barevných obrázků z 20 tisíc kategorií
- soutěž odstartovala boom konvolučních neuronových sítí v rozpoznávání obrazu

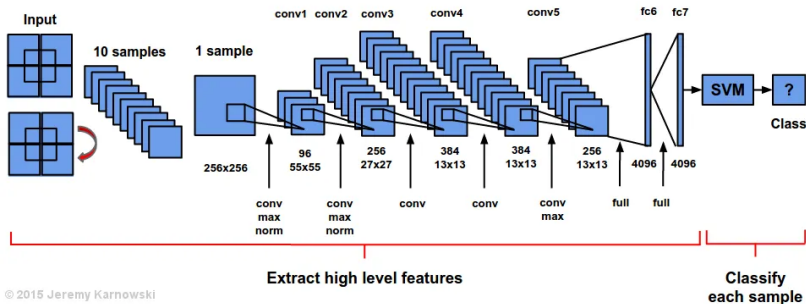


Zdroj obrázku:

https://cs.stanford.edu/people/karpathy/cnnembed/cnn_embed_full_1k.jpg

AlexNet

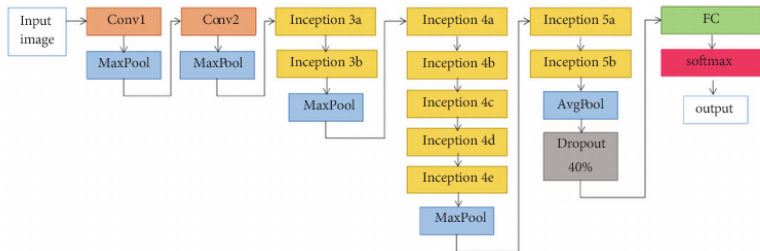
- Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton (vítěz soutěže ILSVRC 2012 - úspěšnost 84,7 % v top-5), již výrazně složitější



Zdroj obrázku: <https://medium.com/@jkarnows/alexnet-visualization-35577e5dcd1a>

GoogLeNet (Inception v1), 2014

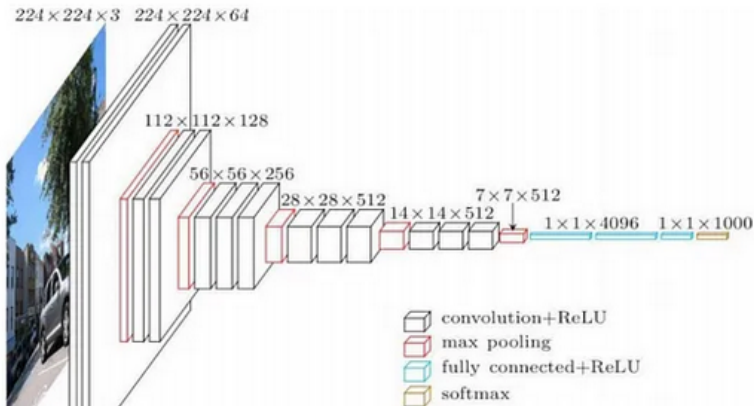
- vítěz soutěže ILSVRC 2014 - úspěšnost 93,33 % v top-5



Zdroj obrázku: Zhang, Keke et. al.: Can Deep Learning Identify Tomato Leaf Disease?. Advances in Multimedia. 2018.

VGGNet

Karen Simonyan a Andrew Zisserman, 2014, rodina modelů (např. VGG16, VGG19)



Další populární architektury

- ResNet (Residual Network, 2015) - zavádí skip connections (řeší problém tiché pošty), vítěz ILSVRC 2015
- Inception v2, v3 (2015, 2016)
- DenseNet: (Gao Huang, 2016)
- MobileNet (Google, 2017), EfficientNet (2019)- úspora výpočetních prostředků
- NASNet (Neural Architecture Search Network, 2017) - model se naučil vhodnou architekturu sám, genetické algoritmy, reinforcement learning
- EfficientNet, 2019
- SqueezeNet
- ...

Další populární benchmarkové datové sady

CIFAR-10, CIFAR-100

<https://www.cs.toronto.edu/~kriz/cifar.html>

- CIFAR-10: 60000 barevných obrázků, 10 tříd (auta, psi, lodě,...)
- CIFAR-100: 100 tříd
- malé obrázky, dataseta populární pro rychlé testování a porovnání modelů

COCO (Common Objects in Context)

- detekce objektů, segmentace a popis obrázků

COCO 2020 Panoptic Segmentation Task



Zajímavé odkazy

- vizualizace konvoluční neuronové sítě
<https://poloclub.github.io/cnnexplainer/>
- Matlab onramp kurzy k hlubokému učení
<https://matlabacademy.mathworks.com/#ai>
<https://matlabacademy.mathworks.com/details/deep-learning-with-matlab/mldl>