

# Konvoluční neuronové sítě

Klasifikace číslic pomocí nich

# Neuronové sítě

- ▶ Jsou známé od 60. let
- ▶ Až s příchodem výpočetního výkonu a dostupností dat na trénování je možné dosáhnout „human level“ výkonu (například při klasifikaci). Někdy i lépe.
- ▶ Zásadní součástí NS pro práci s obrazem jsou konvoluční vrstvy
  - ▶ Neuronová síť se učí jaký filtr (jádro) je nejlepší aby co nejlépe rozlišil kočku od psa
    - ▶ Filtry mohou zvýraznit hrany, barvy, objekty, cokoliv. Uvnitř sítě je to již nepojmenovatelné.
  - ▶ Konvoluce lépe extrahuje příznaky (feature extractor), zbytek neuronky je pak klasifikátor.
- ▶ Dnes nejprogresivnější odvětví ve zpracování obrazu
  - ▶ CNN (konvoluční neuronové sítě) v podstatě převzaly všechny úlohy na které se vyvíjeli speciální algoritmy (proto jim věnujeme polovinu semestru).
    - ▶ Klasifikace - co je na obraze - liška, chodec, banán, číslovka 7..
    - ▶ Segmentace - tento pixel je auto, tento je chodec. Tady má člověk oči, tady ústa..
    - ▶ Generování obrazů - například fejkové generování obličejů: <https://thispersondoesnotexist.com/>
    - ▶ Atd..

# Rozpoznávání čísel

- ▶ V předchozím cvičení jsme rozpoznávali nulu od 7. V tomto případě není příliš složité přijít na jednoduché pravidlo, které čísla rozliší.
- ▶ Složitost výrazně roste s počtem pravidel, které bychom museli do algoritmu zanést v případě dalších čísel (7 je podobná jedničce, 0 osmičce, atd.)
- ▶ Navíc čísla nejsou vždy správně napsána (nula nemusí být pokaždé spojena, čísla mají různý úhel, atd.)
- ▶ V tomto cvičení je úloha podobná - rozpoznat již známý dataset nul a sedmiček.
  - ▶ Tentokrát však s použitím neuronové sítě
  - ▶ Jako bonus nám vznikne neuronová síť, která dokáže rozpoznat jakékoliv číslo
  - ▶ Jednoduchá pravidla jsme vyměnili za komplexní, obecnou neuronovou síť.

# Úloha obecně

- ▶ Natrénovat konvoluční neuronovou síť pro rozpoznání číslic
  - ▶ Tvorba datasetu pro trénování
  - ▶ Definice sítě
  - ▶ Práce s imageDatastore - fce transform
- ▶ Správně rozpoznat 0 a 7

# Úloha - tvorba CNN pro rozpoznávání číslic

- ▶ Budeme zde používat mnist dataset (<http://yann.lecun.com/exdb/mnist/>) pro natrénování CNN
- ▶ Matlab obsahuje část MNIST datasetu. Vytvořte proměnnou typu imageDatastore.
- ▶ `digitDatasetPath = fullfile(matlabroot, 'toolbox', 'nnet', 'nndemos', 'nndatasets', 'DigitDataset');`
- ▶ `imds = imageDatastore(digitDatasetPath, 'IncludeSubfolders', true, 'LabelSource', 'foldernames');`
- ▶ `[imdsTrain, imdsValidation] = splitEachLabel (imds, 0.7, 'randomize');`

# Úloha - tvorba CNN pro rozpoznávání číslí

Vytvořte vrstvy budoucí sítě (viz odkaz výše). Přidejte další konvoluční vrstvu - s 20 filtry o velikosti 8. Přidejte i normalizační vrstvu a též vrstvu, jenž do systému vnese nelinearitu (leakyRelu ).

Specifikujte vlastnosti pro trénování (zase to můžete téměř okopírovat)

```
layers = [
```

- ▶ `imageInputLayer([28 28 1])` % input layer (grayscale image with size of 28x28 pixels)
- ▶ `convolution2dLayer(5,16,'Padding','same')` % 16 convolution filters with size of 5
- ▶ `batchNormalizationLayer` % normalization layer
- ▶ `reluLayer`
- ▶ `fullyConnectedLayer(10)` % 10 - number of neurons
- ▶ `softmaxLayer` % normalization
- ▶ `classificationLayer`
- ▶ `];`

# Testování sítě

- ▶ Vytvořte imageDatastore z testovacích nul a sedmiček (imds\_test).
- ▶ Otestujte klasifikaci, všech 10 obrázků bude na konci správně určených
- ▶ Takto se dá využít funkce classify: `classify(net, imds_test)` (ale to zatím nefunguje, viz↓)
- ▶ Neuronová síť je natrénovaná na obrázky o velikosti 28x28x1.
  - ▶ Testovací obrázky mají velikost různou.
  - ▶ Zde se dá využít funkce transform, které se předhodí 2 parametry
    - ▶ Dataset který chcete transformovat
    - ▶ Funkci kterou jí budete transformovat (šablonu máte ve skriptu)
- ▶ MNIST dataset prošel celkem výrazným předzpracováním, viz: <http://yann.lecun.com/exdb/mnist/>
  - ▶ Zkuste toto zohlednit v transformační funkci (upravitObr).
  - ▶ Kdykoliv si můžete zobrazit výsledek transformace pomocí `montage(preproc_imds_test.readall())`
  - ▶ Pár rad:
    - ▶ Výstup transformační funkce musí být obrázek o dané velikosti ve formátu uint8
    - ▶ Samotné číslo z obrázku se dá „vyškrábnout“ pomocí `regionprops` a vl. „Image“
    - ▶ Síť je natrénovaná na číslech, které všechny mají určitý okraj. Zde pomůže fce `padarray`
    - ▶ Naučená čísla jsou bílá na černém pozadí

0000777777

# Vhodné zdroje

- ▶ Deep learning na tři řádky -článek, kde vysvětlují co se děje uvnitř neuronové sítě a proč je úloha rozpoznání něčeho tak složitá  
<https://www.itnetwork.cz/programovani/matlab/matlab-zlehka-deep-learning-na-tri-radky/>
- ▶ Perfektní video o neuronových sítích (s příkladem na rozpoznávání čísel):  
<https://www.youtube.com/watch?v=aircAruvnKk> (není tam ta konvoluční část, ale to vůbec nevadí)