

## Lekce 4 [25]

Write down the equation of how convolution of a given image is computed. Assume the input is an image  $I$  of size  $H \times W$  with  $C$  channels, the kernel  $K$  has size  $N \times M$ , the stride is  $T \times S$ , the operation performed is in fact cross-correlation (as usual in convolutional neural networks) and that  $O$  output channels are computed. [5]

$$(K \star I)_{i,j,o} = \sum_{m,n,c} I_{i \cdot S + m, j \cdot T + n, c} K_{m,n,c,o} \quad (1)$$

V zásadě je pixel na pozici  $i, j, o$  vytvořen posouváním  $o$ -tého kernelu kolem pozice  $i \cdot S, j \cdot T$  v  $I$ .

Explain both **SAME** and **VALID** padding schemes and write down the output size of a convolutional operation with an  $N \times M$  kernel on image of size  $H \times W$  or both these padding schemes (stride is 1). [5]

**SAME** zachová původní rozměry, krajní hodnoty tím pádem nejsou kombinovány z plného kernelu, ale jen z jeho validní části. Je implementovaný tak, že se původní obrázek dopaduje nulami, a pak se spustí **VALID**.

**VALID** bere v úvahu jen ty pozice, u kterých byl celý kernel "uvnitř" vstupního obrázku. Výsledné rozměry jsou tedy kratší o šířku (výšku) kernelu.

Describe batch normalization and write down an algorithm how it is used during training and an algorithm how it is used during inference. Be sure to explicitly write over what is being normalized in case of fully connected layers, and in case of convolutional layers. [10]

Když se během SGD změní distribuce hodnot vrstev vlevo, musíme jen kvůli této změně měnit i vrstvy napravo. Proto chceme všechny vrstvy nějakým způsobem normalizovat, tak, aby tyto změny byly co nejmenší.

Během tréninku batchnorm pracuje následovně, vstupem jsou  $x_i$  z jednoho batche.

$$\begin{pmatrix} \mu \leftarrow \frac{1}{m} \sum_{i=1}^m \mathbf{x}^{(i)} \\ \sigma^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (\mathbf{x}^{(i)} - \mu)^2 \end{pmatrix} \quad (2)$$
$$\hat{\mathbf{x}}^{(i)} \leftarrow (\mathbf{x}^{(i)} - \mu) / \sqrt{\sigma^2 + \epsilon}$$
$$\mathbf{y}^{(i)} \leftarrow \gamma \hat{\mathbf{x}}^{(i)} + \beta$$

Tj. spočítáme střední hodnotu a rozptyl batche, ručně znormalizujeme vstupy, a pak dovolíme síti pomocí naučených parametrů  $\gamma$  a  $\beta$  přeskálovat a posunout standardní rozdělení na libovolnou střední hodnotu a rozptyl.

Během inference už nepře počítáme  $\mu$  a  $\sigma^2$ , ale používáme hodnoty získané při tréninku. Aktivační funkci používáme po batchnormu, biasy přeskakujeme.

$$f(BN(\mathbf{W}\mathbf{x})) \quad (3)$$

Když je batchnorm použitý na dense vrstvách, každý neuron je normalizován sám za sebe (v rámci batche). V CNN normalizování probíhá v rámci celého kanálu (tak, jak bychom čekali).

Describe overall architecture of VGG-19 (you do not need to remember the exact number of layers/filters, but you should describe which layers are used). [5]

- 
1. 2 x conv3 64
  2. maxpool
  3. 2 x conv3 128
  4. maxpool
  5. 4 x conv3 256
  6. maxpool
  7. 4 x conv3 512
  8. maxpool
  9. 4 x conv3 512
  10. maxpool
  11. 3 x fully connected
  12. softmax

Nepoužívají se 5x5 konvoluce, protože dvě 3x3 za sebou mají zhruba stejné receptive field, ale mají méně parametrů a jsou rychlejší. Vždy po max poolingů zdvojnásobíme počet kanálů (než nám dojde paměť v posledním bloku, that is).

Kvůli finálním FC vrstvám museli používat obrázky fixní velikosti; samotné konvoluce by si s různě velkými obrázky poradit zvládly.