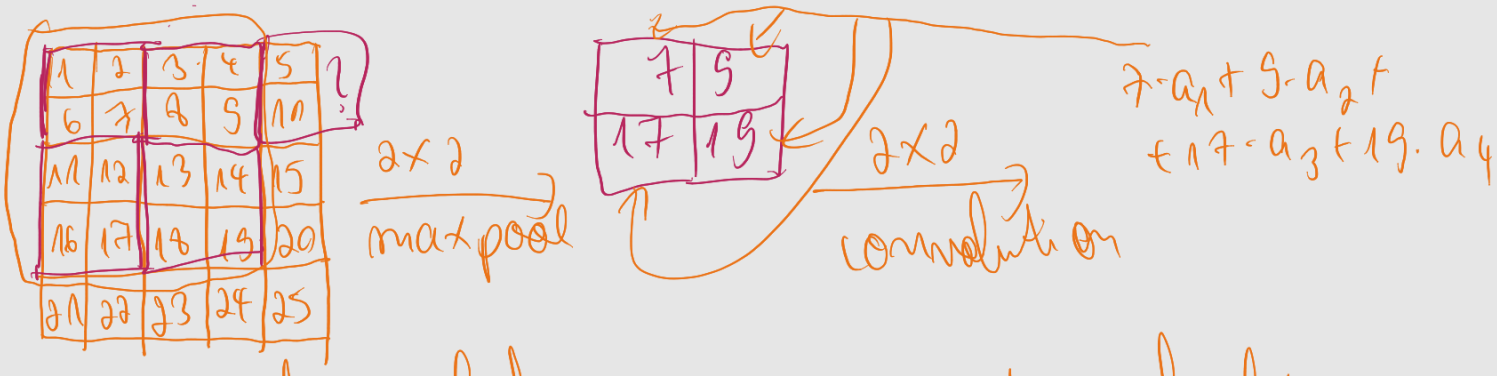

max pooling helps increase receptive field

LeNet

Input: 32×32 grayscale image => 1×32×32 shape

Layer 1: convolution: 1 input channel, 6 output channels,
kernel_size = 5, padding = 2 => 6×32×32

Layer 2: activation function: sigmoid

Layer 3: average pooling: kernel_size 2×2, stride 2×2
=> 6×16×16

Layer 4: convolutie cu 6 canale de intrare si 16
canale de iesire, kernel_size = 5,
no padding => resolution decreases => 16-4=12

Layer 5: another sigmoid

Layer 6: average pooling: kernel size 2×2, stride 2×2
=> half resolution => 16×5×5

At some point, transform de la convolutii la linear
layerrs:

Layer 7: flatten => vector de dim 16·5·5 = 16·25 =
= 400

Recipe for classification CNN:

- Bunch of convolutions and pooling layers
- flatten layer => from "image" to vector
- MLP that outputs as many values as there are classes

First linear regression: $y = a_1 \cdot x_1 + a_2 x_2 + b$

Second linear regression: $z = c \cdot y + d$

$z = c(a_1 \cdot x_1 + a_2 \cdot x_2 + b) + d$

$= \underbrace{a_1 c}_{m} \cdot x_1 + \underbrace{a_2 c}_{n} \cdot x_2 + \underbrace{bc + d}_{p}$

same thing

$z = m x_1 + n x_2 + p$

Between layers of linear „stuff": non-linear activation function!

$$y = \frac{1}{1 + e^{-x}}$$