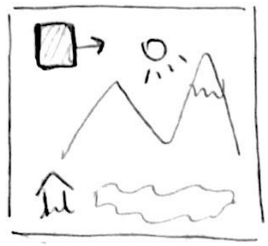
NETWORKS ARCHITECTURES

▷ DEEP CONVOLUTIONAL NN

LET'S CONSIDER AN IMAGE.

Image

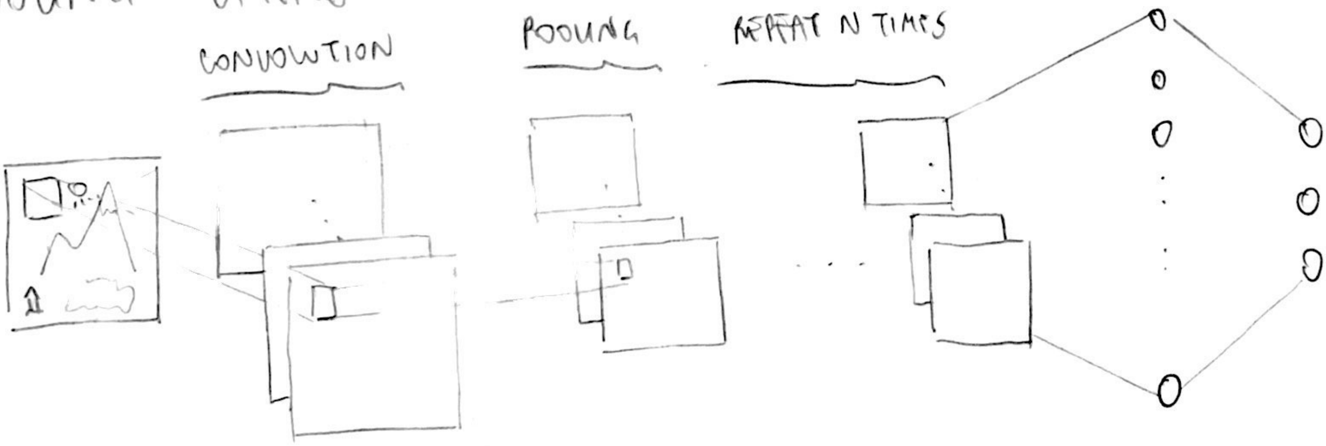


THE CONVOLUTIONAL FILTER CAPTURES THE IMPORTANT FEATURES OF THE IMAGE

IMPORTANT IMAGE TRANSFORMATIONS:

- TRANSLATIONS
- ROTATIONS
- DIALATION

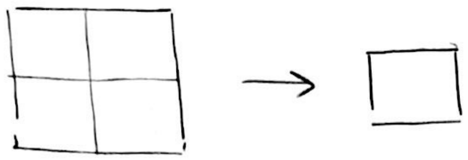
TRANSLATION AND DIALATION ARE ADDRESSED BY MOVING FILTERS OF DIFFERENT SIZE. THE WEIGHTS OF THE FILTER CAN BE INITIALIZED AT RANDOM AND SINCE THERE ARE MANY LOCAL MINIMA, AFTER, TH LEARNING, THEY WILL PRODUCE DIFFERENT FEATURE SPACE. AFTER CONVOLUTION WE REDUCE THE SIZE OF THE FEATURE SPACE WITH POOLING LAYERS.



WE REPEAT DIFFERENT TIMES CONVOLUTION AND POOLING LAYERS AND AFTER THAT WE FLATTEN THE PIXELS AND USE A STANDARD HIDDEN LAYER.

EVERY CONVOLUTIONAL LAYER IS PARAMETRIZED BY THE FILTER SIZE AND THE STRIDE

POOLING



WE CAN TAKE THE AVERAGE OR THE MAXIMUM.

EVERY FEATURE SPACE CAN BE INDEPENDENT OR CAN COMMUNICATE WITH THE OTHERS.

DUE TO THE HIGH NUMBER OF PARAMETERS ANN CAN EASILY OVERFIT UNLESS A BIG AMOUNT OF DATA IS USED. WE CAN REDUCE OVERFITTING WITH DROPOUT. IN DROPOUT WE RANDOMLY TURN-OFF NEURONS

D NN FOR DYNAMICS

LET'S CONSIDER A DYNAMICAL SYSTEM AND $x \in \mathbb{R}^n$

$$\frac{d}{dt} x = f(x) \quad (\cdot)$$

WE CAN SOLVE IT WITH EULER FORMULA:

$$\frac{x(t+\Delta t) - x(t)}{\Delta t} = f(x(t+\Delta t)) \rightarrow x_{k+1} = x_n + \Delta t \, f_{k+1}$$

THIS IS THE EASIEST WAY TO INTEGRATE $(\cdot)$. A MORE SOPHISTICATE TOOL IS RUNGE-KUTTA SCHEME:

$$x(t+\Delta t) = x(t) + \Delta t \, \phi$$

$$\phi = f_1 + f_2 + f_3 + f_4$$

WE CAN SEE RUNGE-KUTTA METHOD AS A KIND OFF NN: