

Colloquium

Prepare your presentation (10 - 15 min)

- Shortly present the problem/task.
- Present your design/workflow/abstraction.
- Present your code.

Q&A (5 min) after the presentation

Template for slides can be found [here](#).

***For everybody(!):** Deadline for submitting code solutions is 05.02.2021.*



TECHNISCHE
UNIVERSITÄT
DRESDEN

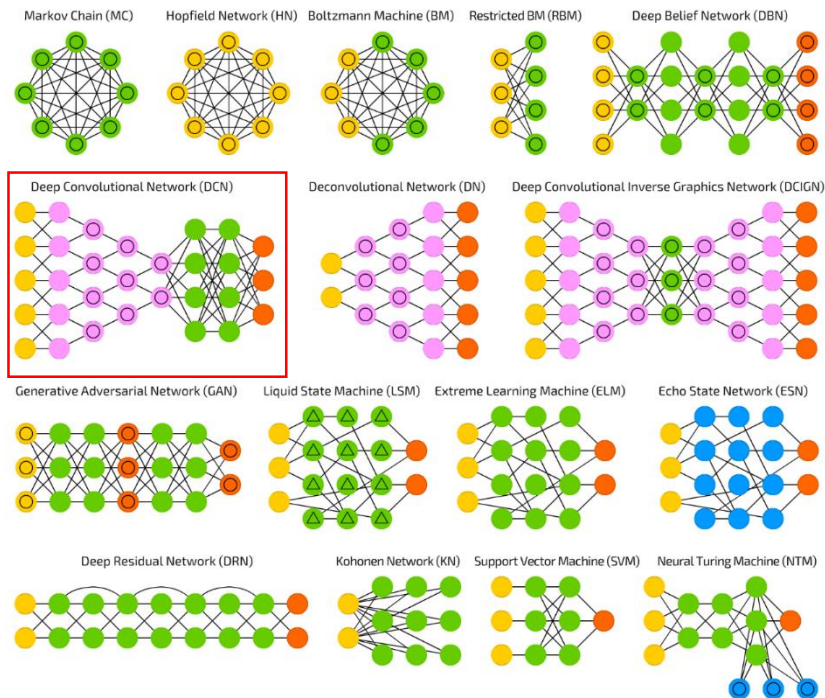
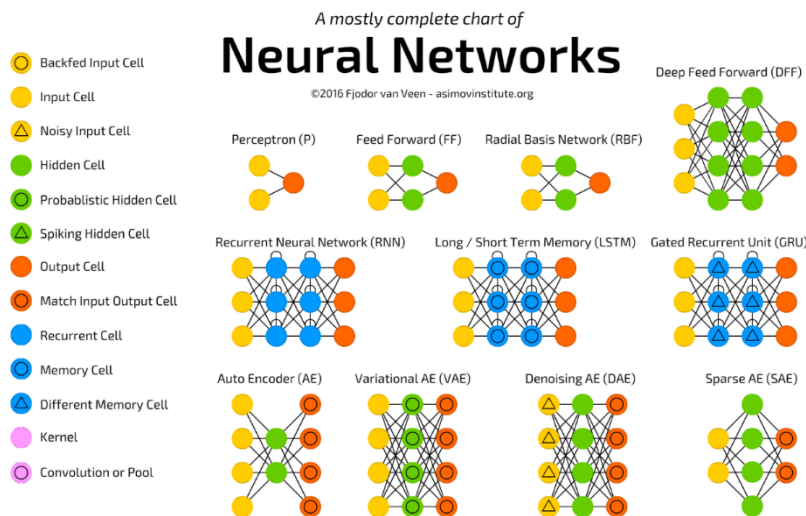


Dresden Database
Systems Group

Neural Networks

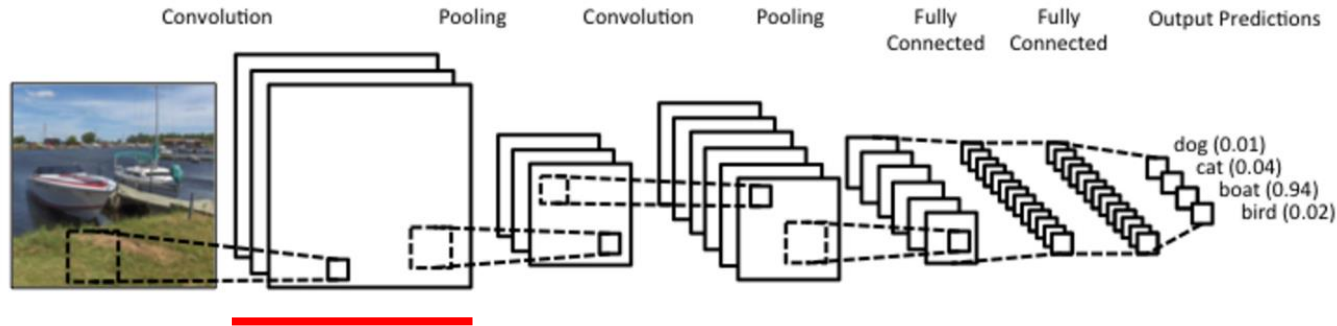
Programming for Data Science

Architectures



Convolutional Neural Networks (CNN)

- Processing of Image & Audio data
- Convolutional and pooling layers
- Biological paradigm: receptive field → all sensory receptors that transmit to one subsequent neuron



Convolutional Neural Networks

- Convolutional Layer: “small” convolution matrix / kernel is moved over the input
- Input to the neuron is scalar product of kernel \rightarrow neighboring neurons have overlapping input
- Most common activation function: ReLu

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

Input

1	0	1
0	1	0
1	0	1

Filter / Kernel

1x1	1x0	1x1	0	0
0x0	1x1	1x0	1	0
0x1	0x0	1x1	1	1
0	0	1	1	0
0	1	1	0	0

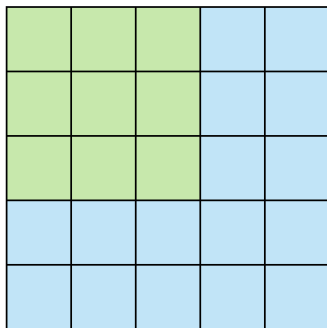
-

4		

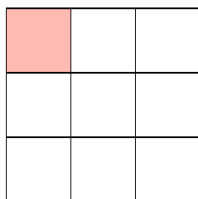
-

Convolutional Neural Networks

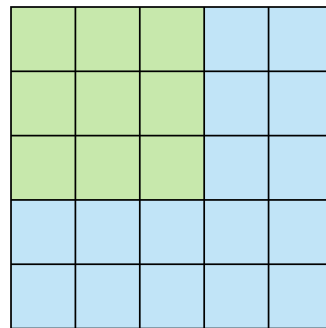
- Parameter: window size, stride, number
- Window size: typically 3x3 (5x5, 7x7, 1x1,...)
- Stride: step size for moving the kernel (default 1)
- Stride influences overlap and number of created convolutions
- Number of filters / kernels : typically power of two



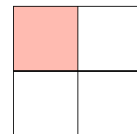
Stride 1



Feature Map



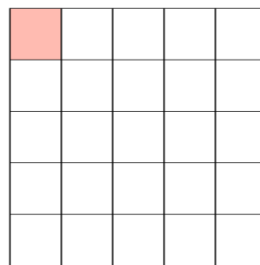
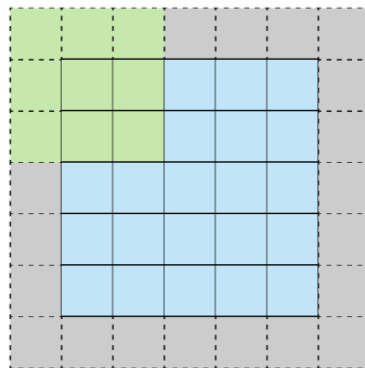
Stride 2



Feature Map

Convolutional Neural Networks

- Parameter: padding
- Kernel has to fit input \rightarrow reduces dimensionality of feature map
- Padding “frames” input \rightarrow feature map retains dimensionality of input
- Padding with 0 or edge values of input

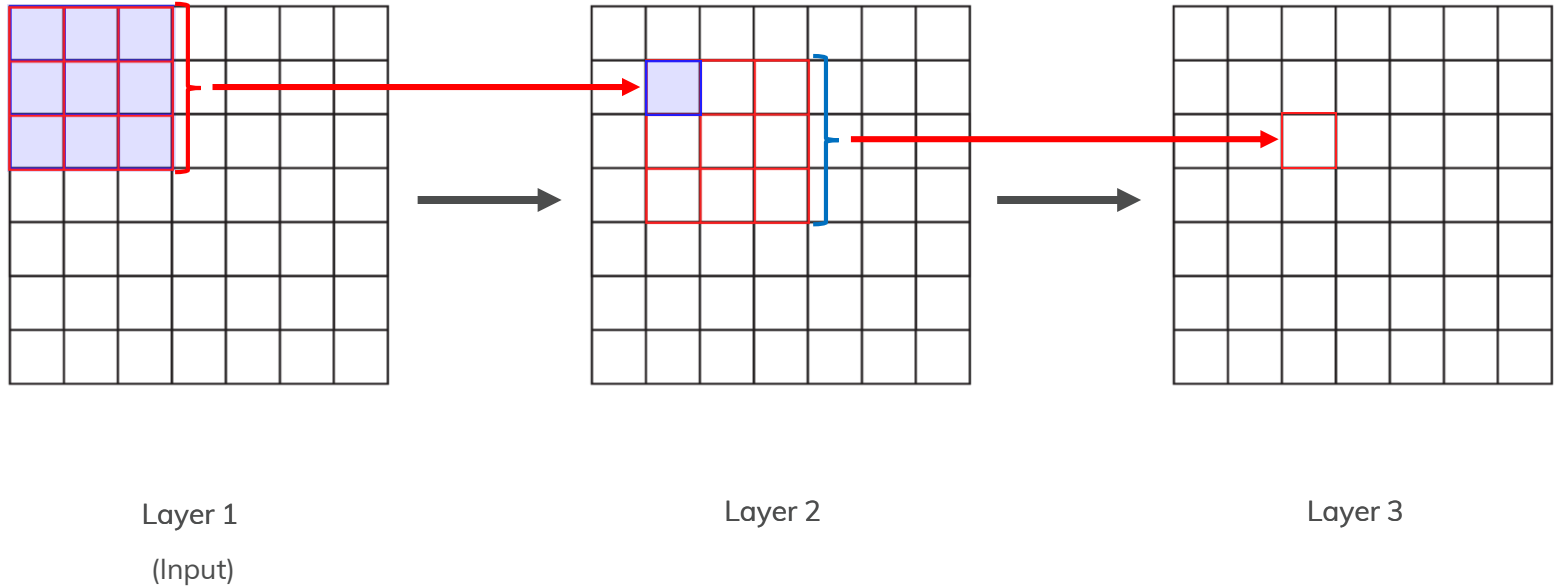


Stride 1 with Padding

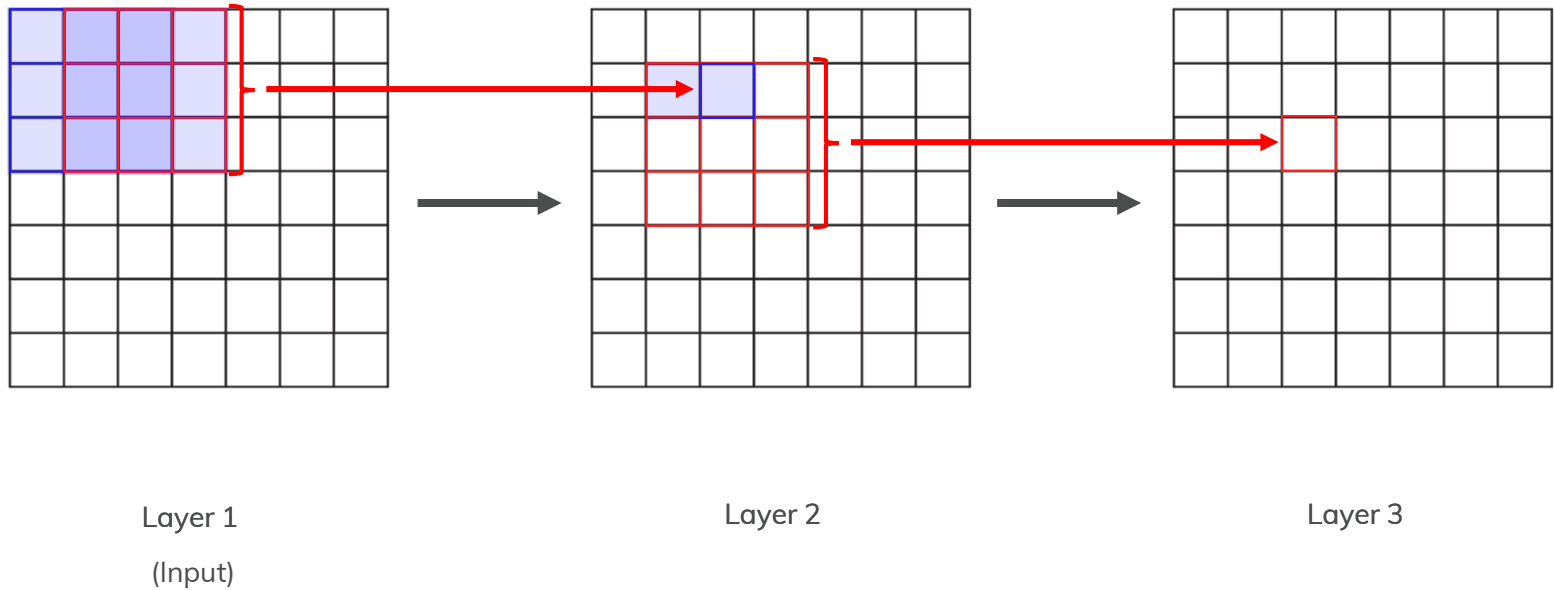
Feature Map

Neural Networks

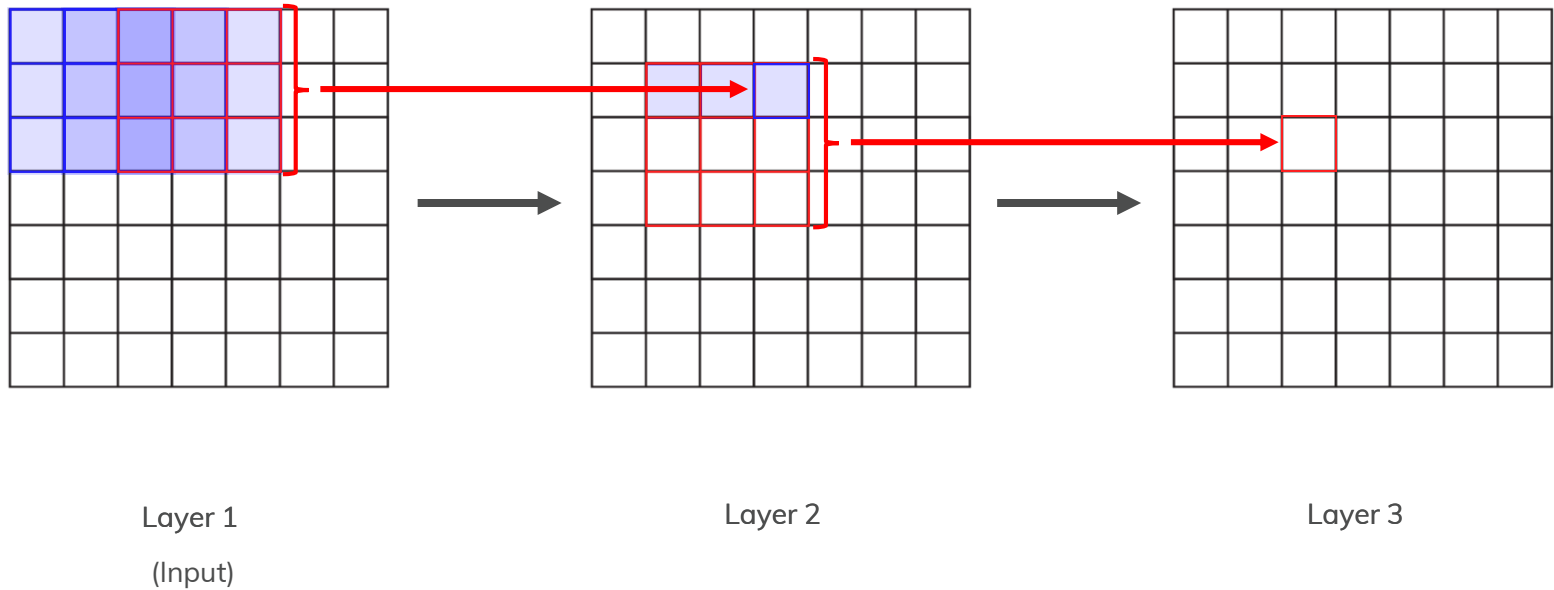
Convolutional Neural Networks



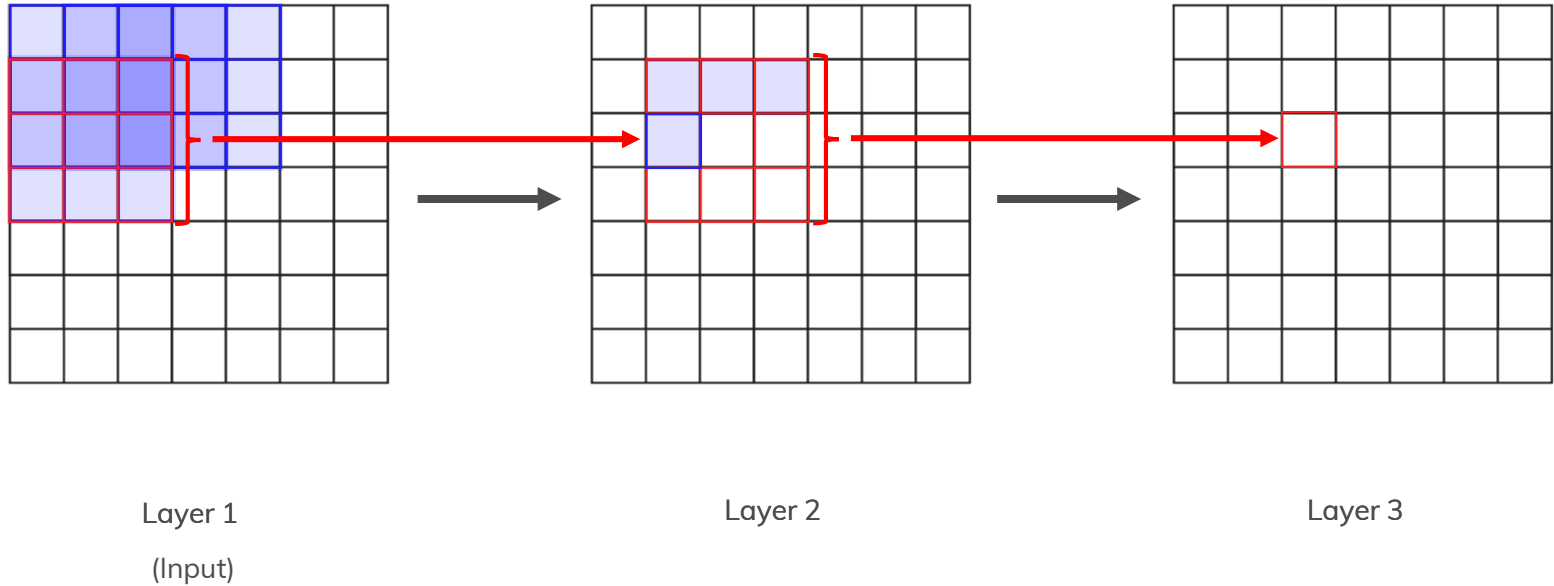
Convolutional Neural Networks



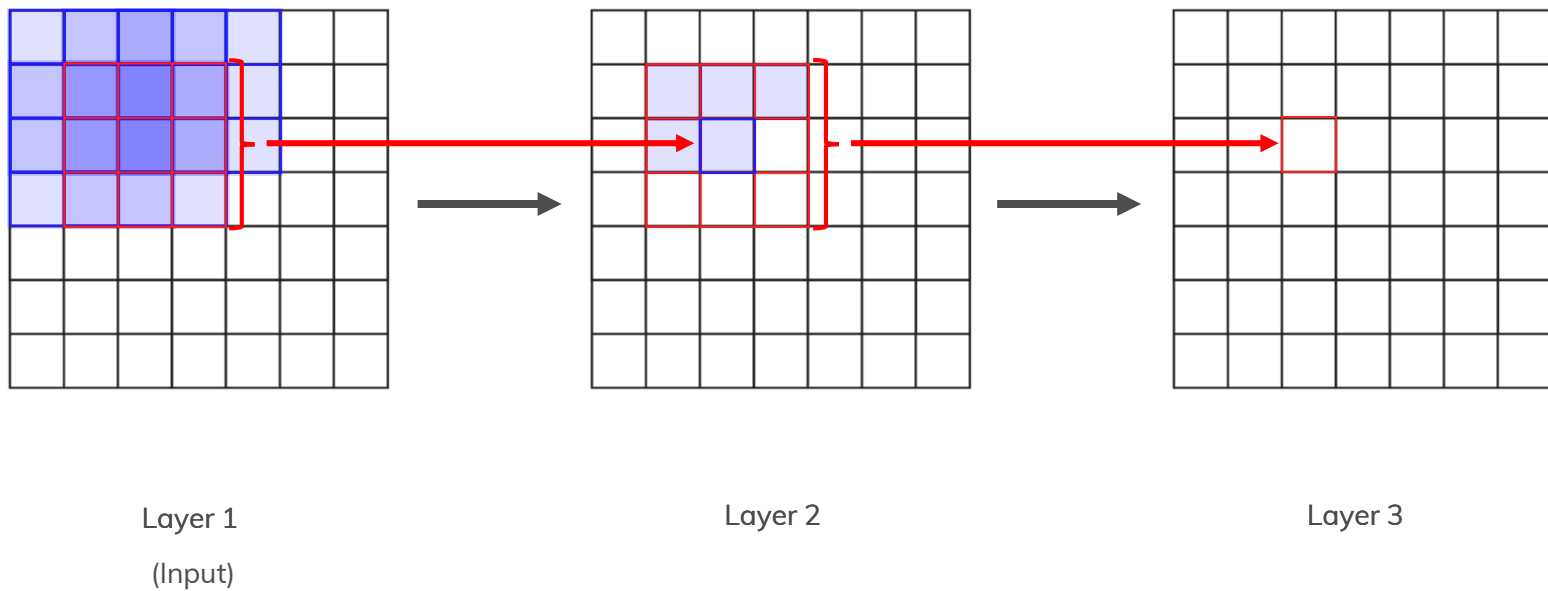
Convolutional Neural Networks



Convolutional Neural Networks

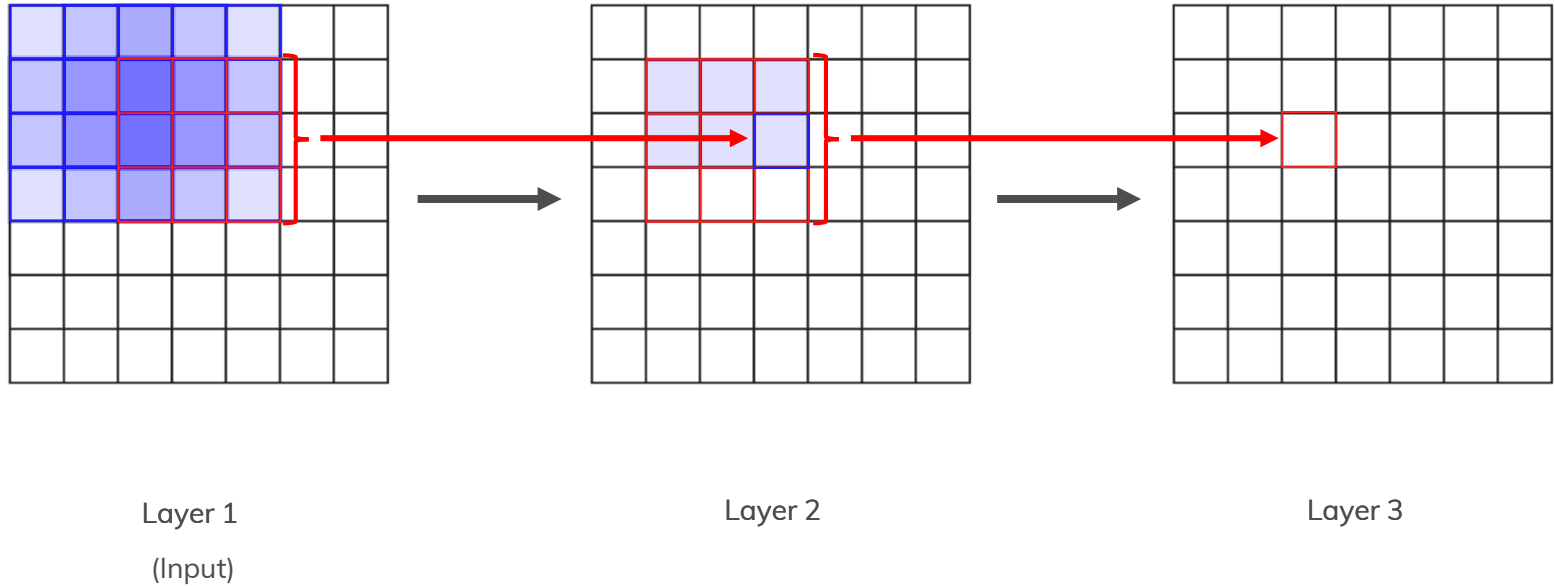


Convolutional Neural Networks

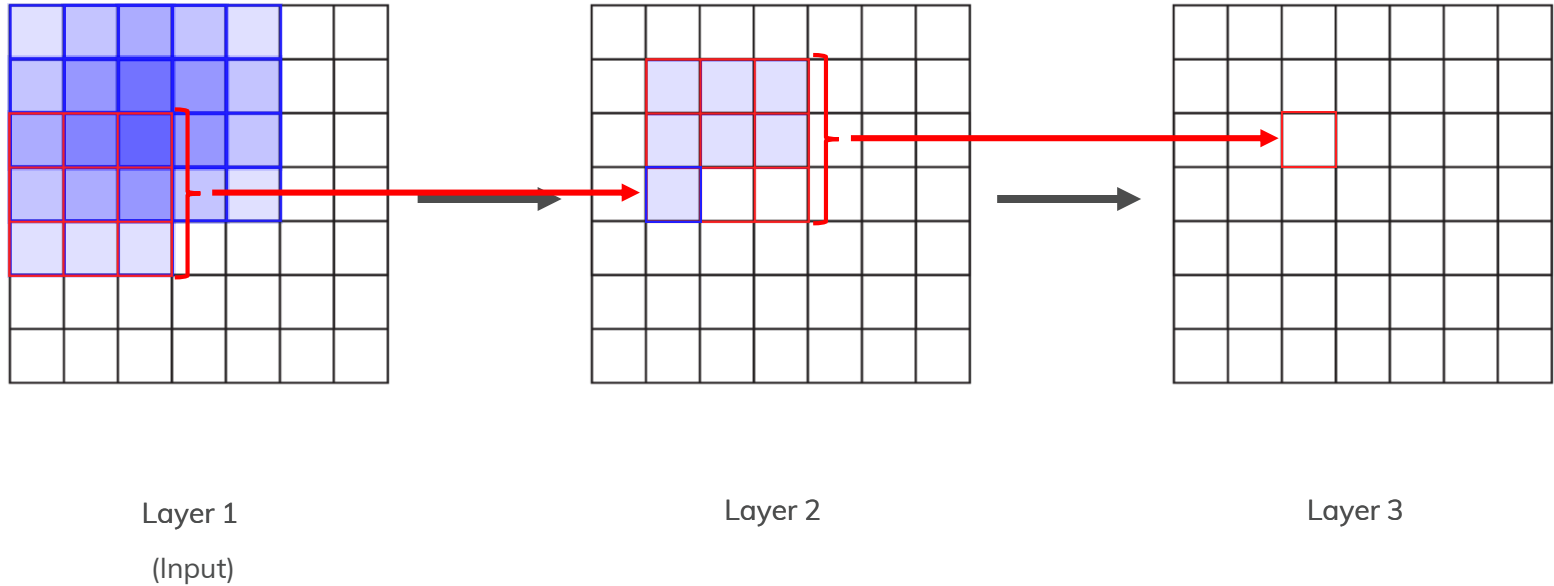


Neural Networks

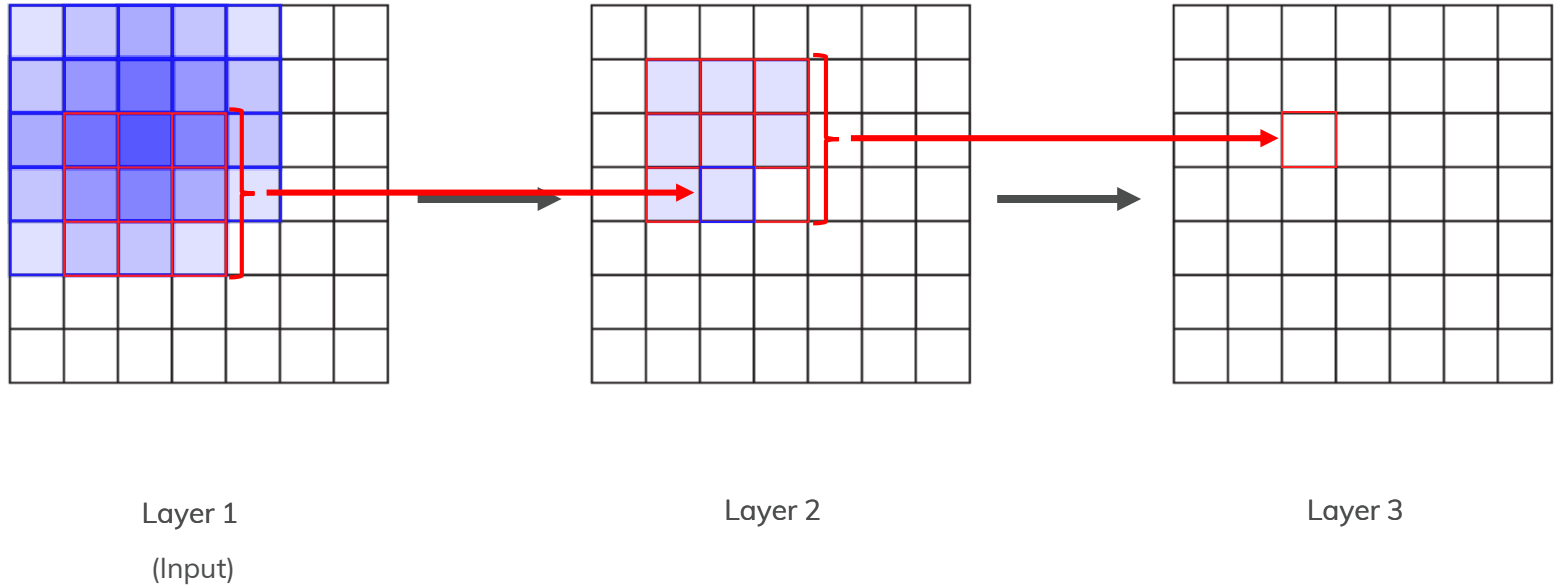
Convolutional Neural Networks



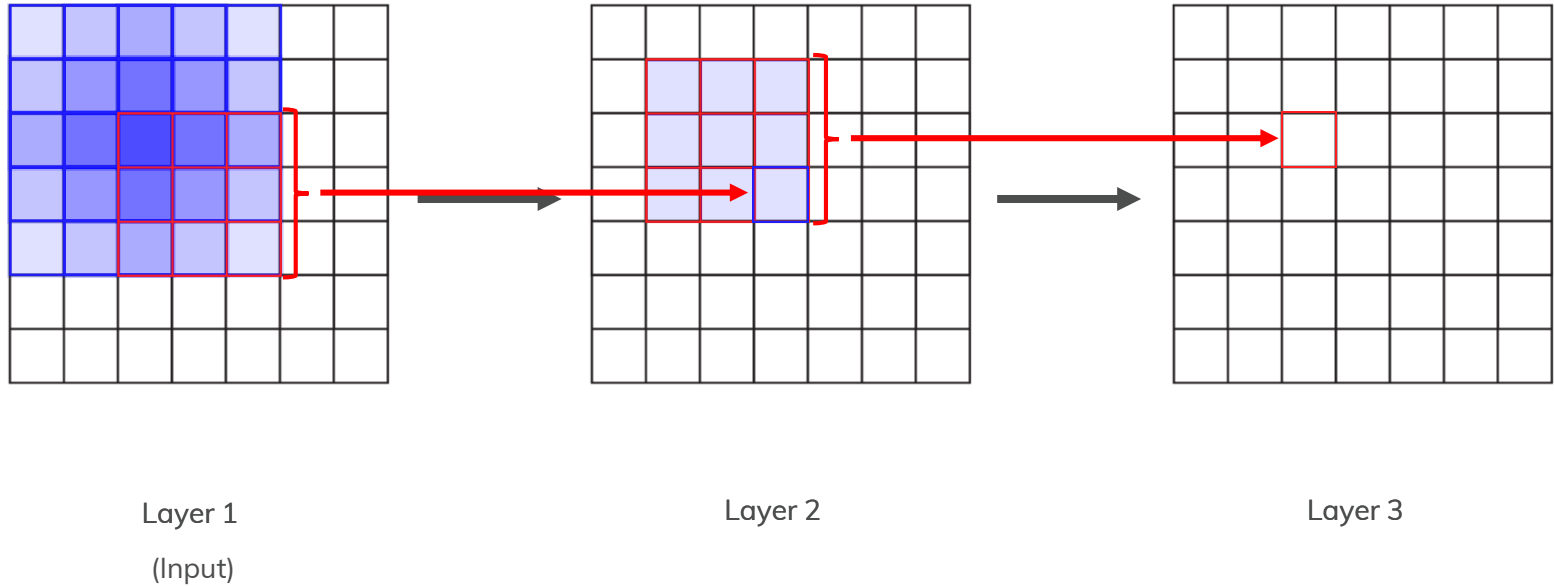
Convolutional Neural Networks



Convolutional Neural Networks

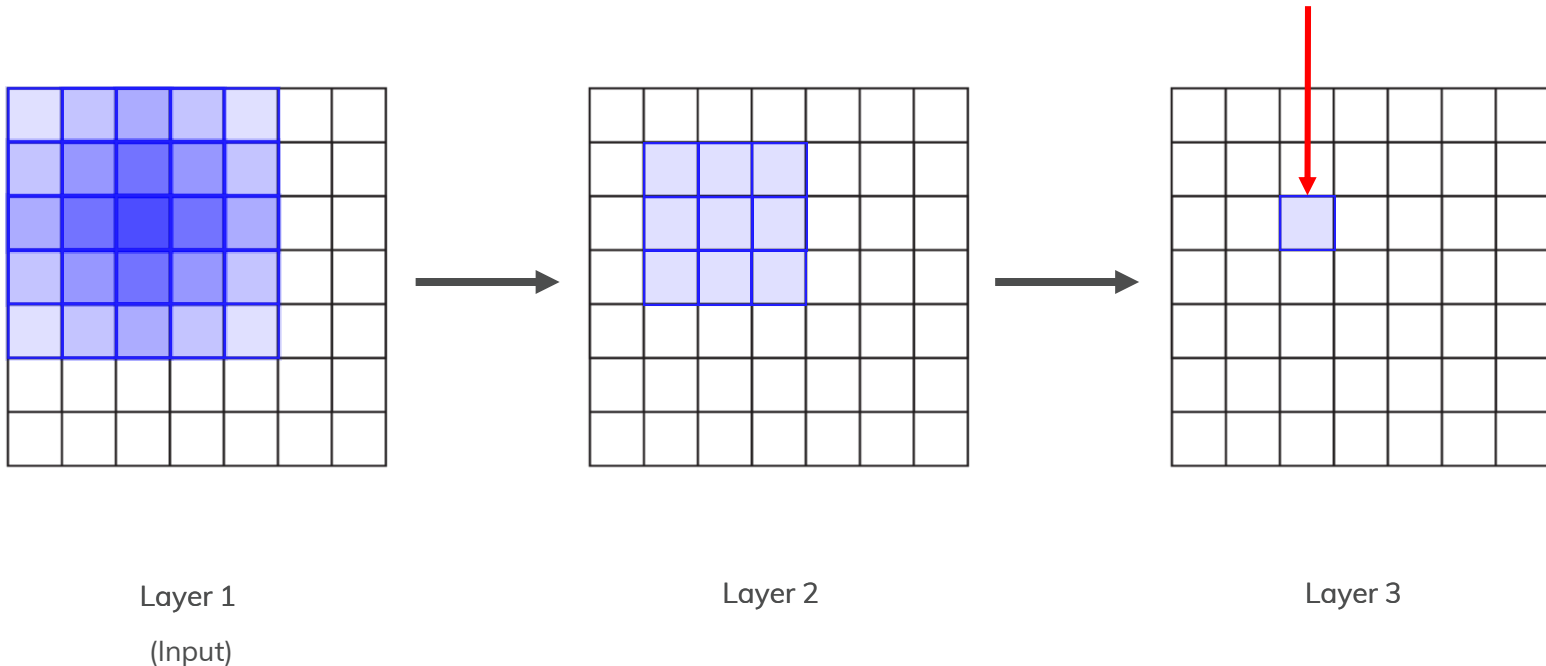


Convolutional Neural Networks



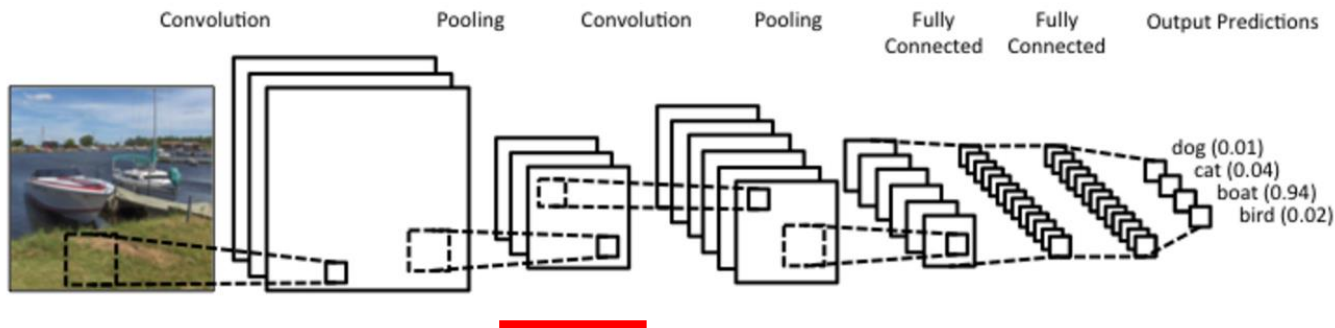
Convolutional Neural Networks

- Each neuron in layer 3 “sees” 5x5 neurons from layer 1



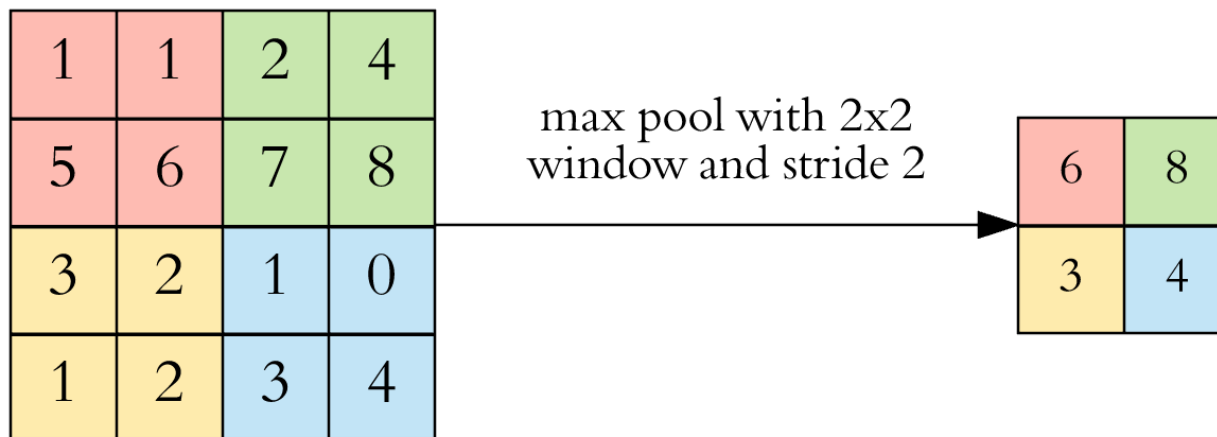
Convolutional Neural Networks

- Subsequent Pooling Layer combine outputs and remove redundant information
- Example: exact position of an edge < approximate position
- Reduces storage requirements, prevents overfitting
- Biological paradigm: lateral inhibition \rightarrow excited neuron inhibits surroundings \rightarrow spread of activation is suppressed \rightarrow creates contrast and better perception



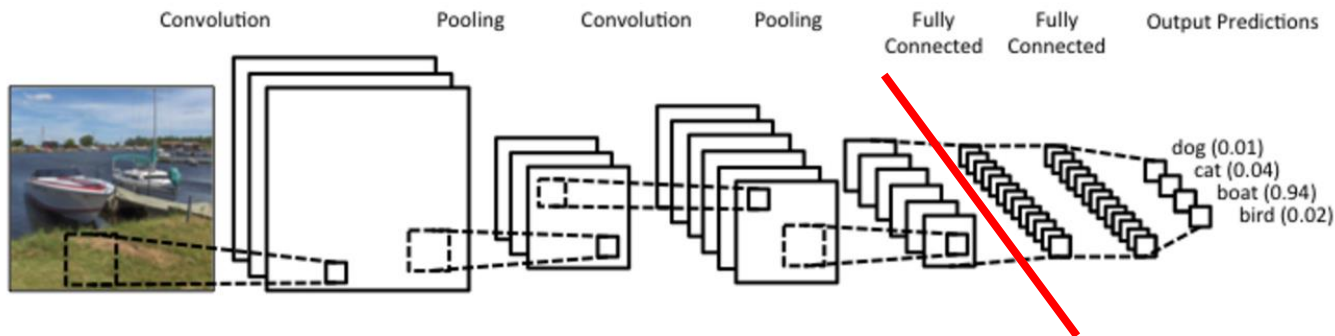
Pooling - Convolutional Neural Networks

- Most common type of pooling layer: max
- 2x2 window with stride 2
- Reduction by 75%



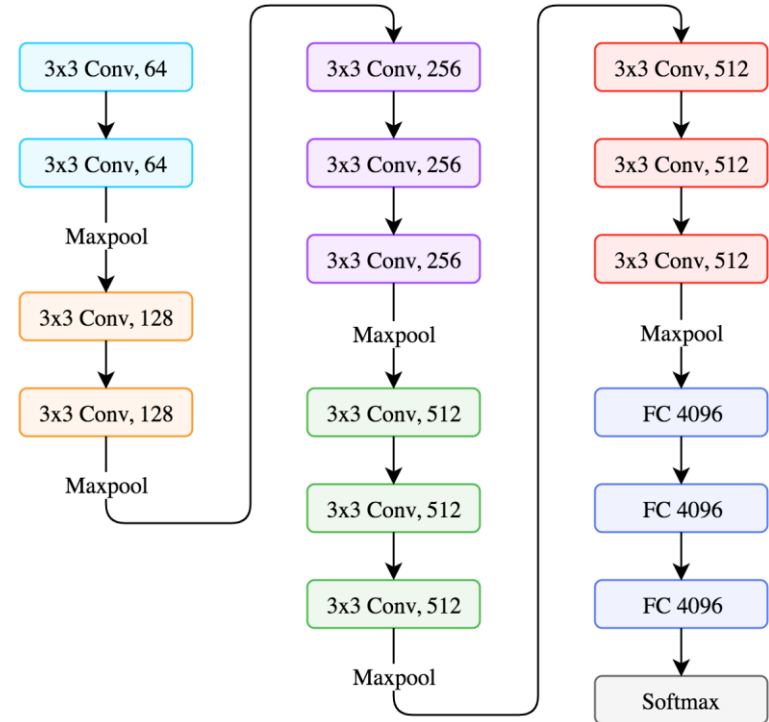
Convolutional Neural Networks

- After convolution and pooling layers, “standard” NN layers are added
- Convolution & pooling extract and filter features
- Later feedforward neuron layers use these features for classification
- Flattening: for NN layers, the 2d feature map must be reduced to 1d vector
- Transfer Learning: keep trained convolution and pooling layers, switch NN layers



VGG16 – State-of-the-art CNN

- Oxford's Visual Geometry Group, 2014
- ImageNet classification challenge (2nd place, 7.3% error rate)
- ImageNet complements nouns of WordNet hierarchy (118K) with images (ca. 500 images per noun)
- Stacked convolutional layers:
 - Collection: $2(3 \times 3) = 5 \times 5$, $3(3 \times 3) = 7 \times 7$
 - Multiple ReLU \rightarrow more non linearity
- Training time: 3 weeks on 4 GPUs
- 16 layer \rightarrow simple and compact!
- Winner 2015: Microsoft ResNet (3.6% error rate) \rightarrow 152 layer !!



Task

Step 0

- Get the MNIST dataset from keras.
- Use the predefined split into training and test data.
- Reshape and scale your data.

Step 1

- Use a CNN* with two convolutional layers (32 and 64 filters, kernel size 3x3, same padding), one max pooling layer (pool size 2x2) and at least one dense layer to classify all digits from their images.
- Other parameters: RELU activations, categorical cross entropy, accuracy metric, the Adam optimizer, 12 epochs and a batch size of 32.
- Evaluate the NN on the test data.

[Step 2]

- Find a better structure for your CNN.

*use your own implementation

Package suggestions

R

- (data.table)
- keras

python3

- numpy
- pandas
- keras

Exercise Appointment

We compare and discuss the results

- Tuesday, 21.01.2020,
- Consultation: 16.01.2020,
- Please prepare your solutions! Send us your code!

If you have questions, please mail us:

claudio.hartmann@tu-dresden.de Orga + Code + R

lucas.woltmann@tu-dresden.de Tasks + Python

