Convolutional Neural Networks

Max Pichler

16.01.2019

Classification vs Regression

- Regression:

  Approximiating a mapping function (f) from input variables to a **continous** output variable

  „How large will the beetles be?“

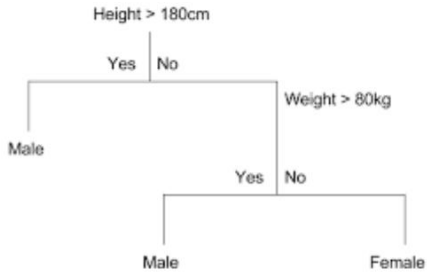
- Classification:

  Approximiating a mapping function (f) from input variables to **discrete** output variables

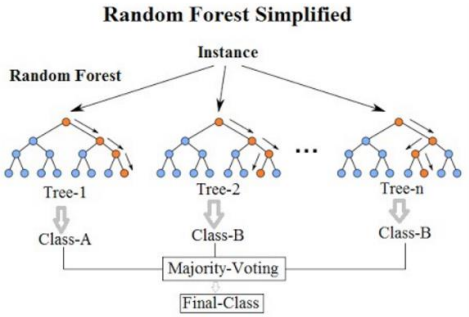
  „Will the beetles survive or die?“

Classification algorithms that were already covered here

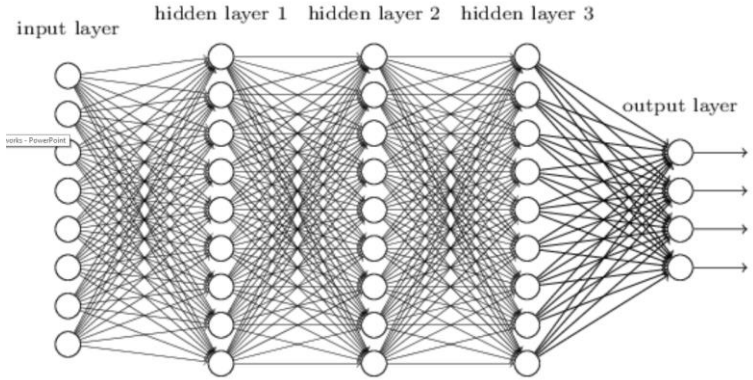
Classification Trees



Random Forest

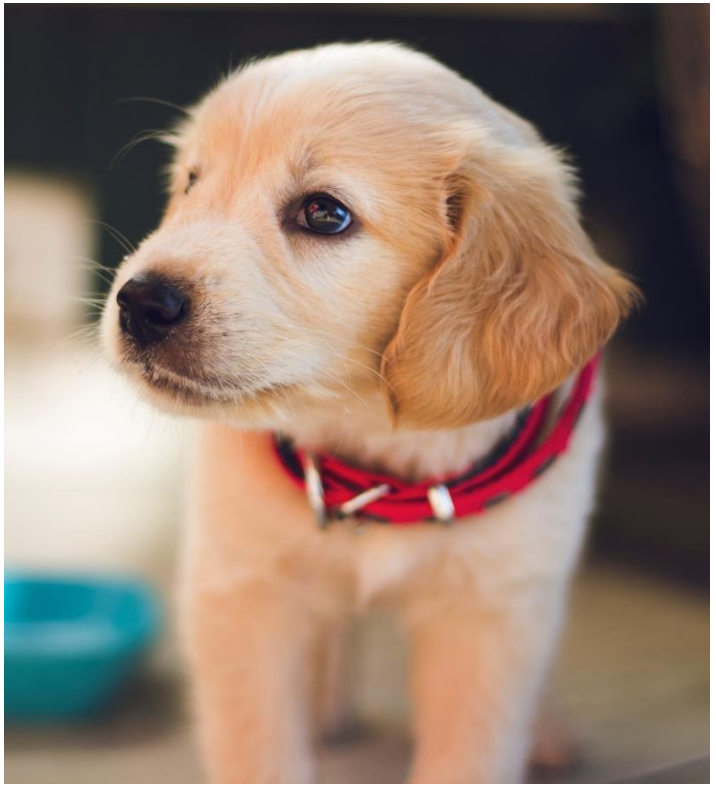


Deep Neural Networks



They work well with low dimensional data but images with millions of pixel (each pixel is a input variable)?

A new Task

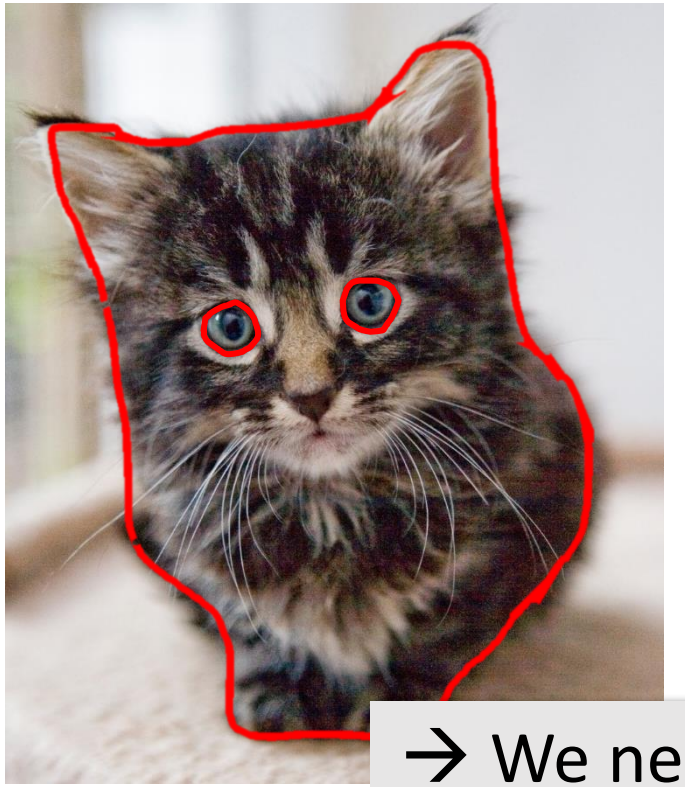


A new Task

- High dimensional data / feature space (many pixels)
- Interested in patterns in the feature space

  → Features of interest are spatially correlated

A new Task



Problems with fully connected ANN for image classification:

- Spatial information is not explicity used(Matrix → Vector)
- Translational invariance? (cat should be identified independent of the position)
- Large amount of weights. (1000 x 1000 Pixel Image and one layer with 1*10^8

→ We need a model capable of spatially looking at the image

Motivation

CNNs are deep feed-forward artifical neural networks that are applied for analyzing grid like data

- 1-D Sequences (time series, text)
- 2-D Images
- 3-D Images
- …

Motivation

CNNs are deep feed-forward artifical neural networks that are applied for analyzing grid like data

Recent ecological applications:

- Tabak et al., 2018: Machine learning to classify animal species in camera trap images: Applications in ecology
- Lasseck et al., 2018: Audio-based bird species identification with deep convolutional neural networks
- Guirado et al., 2018: Automatic whale counting in satellite images with deep learning

Motivation

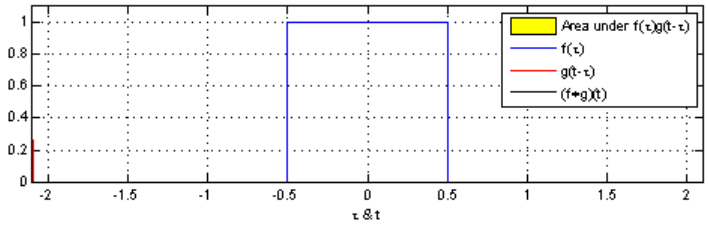
CNNs are deep feed-forward artifical neural networks that are applied for analyzing grid like data

- Nothing new! LeCun et al., 1998
- Rebirth at 2012 ImageNet classification competition: Krizhevsky et al., 2012

Convolutional Neural Networks

Convolution:

$$(f * g)(x) = \int_{-\infty}^{\infty} f(\tau) * g(\tau - t)dt$$

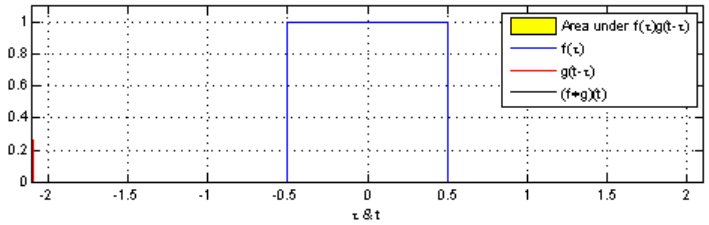


From wikipedia

Convolutional Neural Networks

Convolution:

$$(f * g)(x) = \int_{-\infty}^{\infty} f(\tau) * g(\tau - t)dt$$

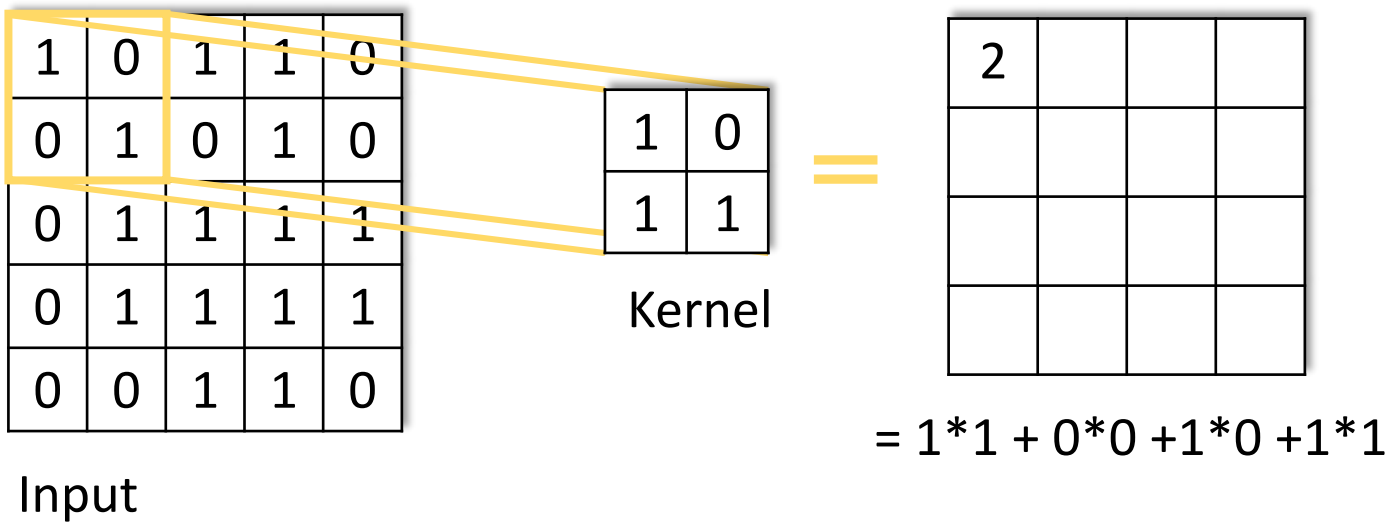


$f(\tau) = \text{Input (e.g. Image)}$
$g(\tau - t)$ = Kernel (Weights)

From wikipedia

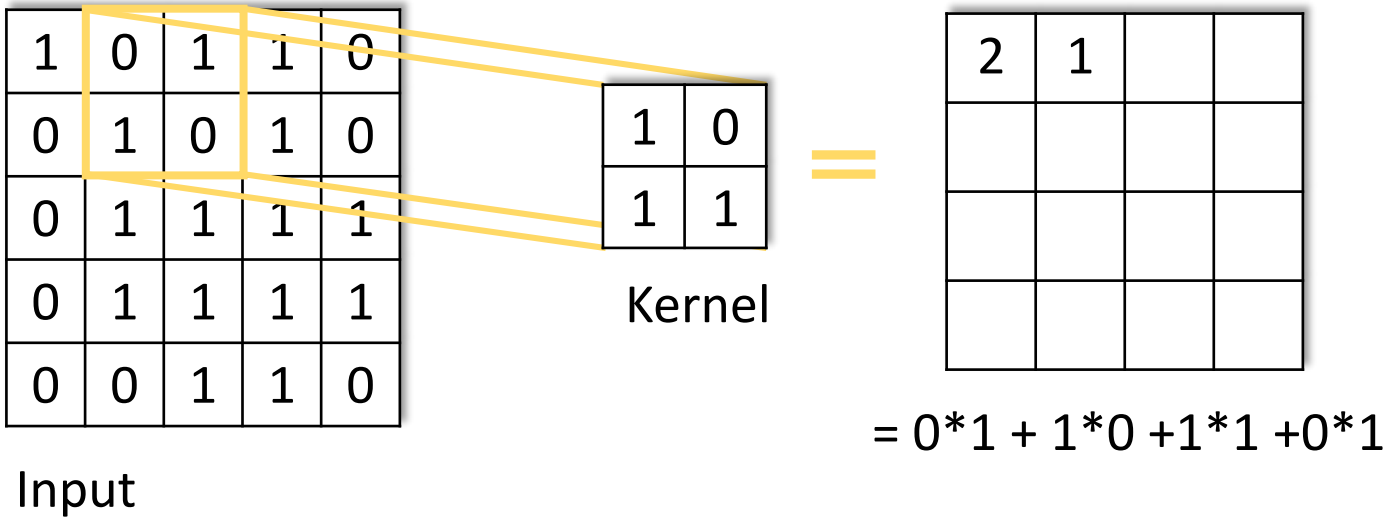
Convolutional Neural Networks

Stride: 1*1
Kernel: 2*2



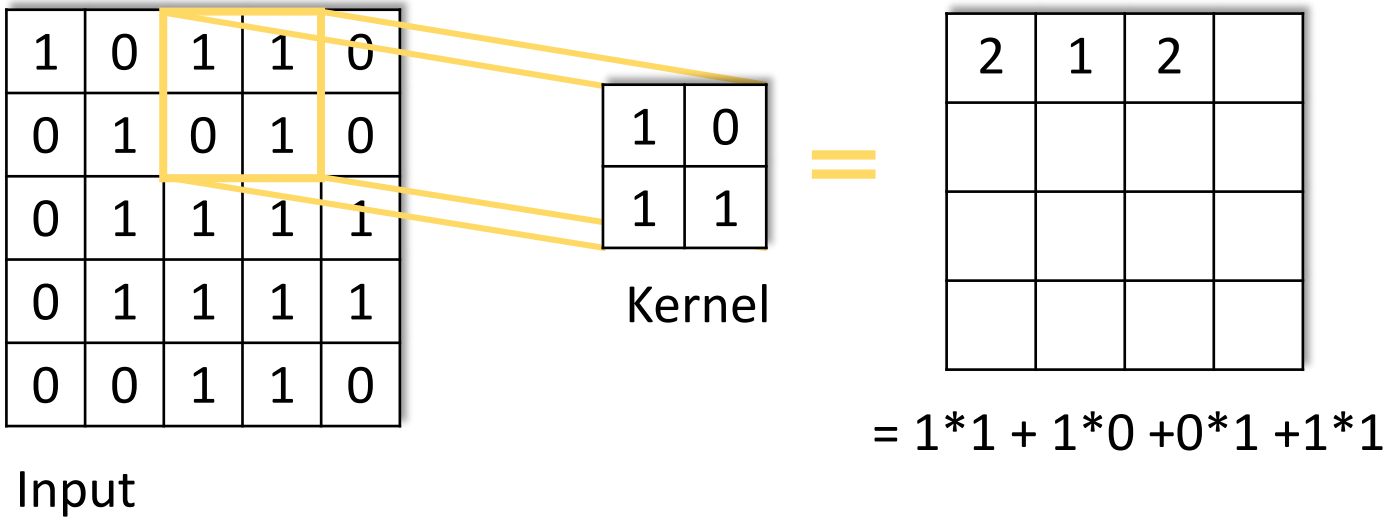
Convolutional Neural Networks

Stride: 1*1
Kernel: 2*2



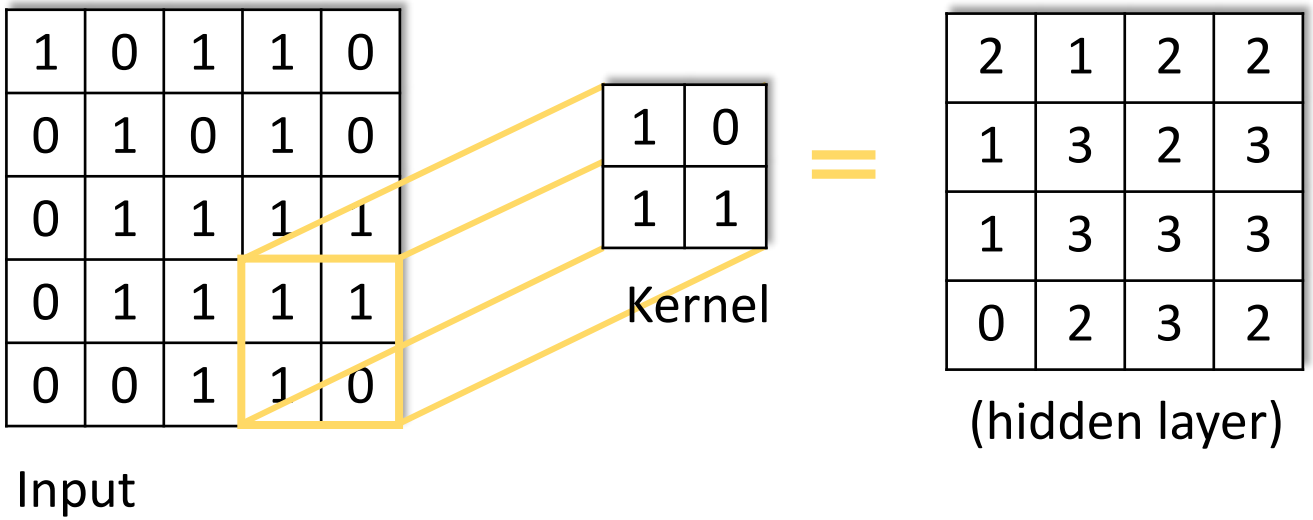
Convolutional Neural Networks

Stride: 1*1
Kernel: 2*2



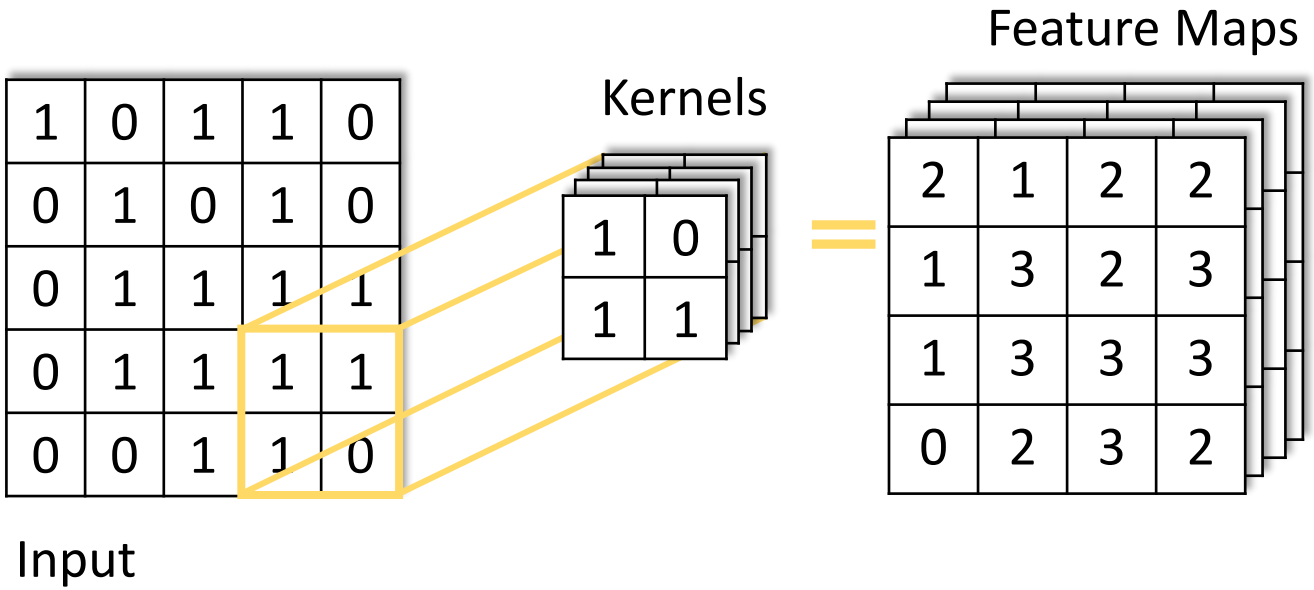
Convolutional Neural Networks

Stride: 1*1
Kernel: 2*2



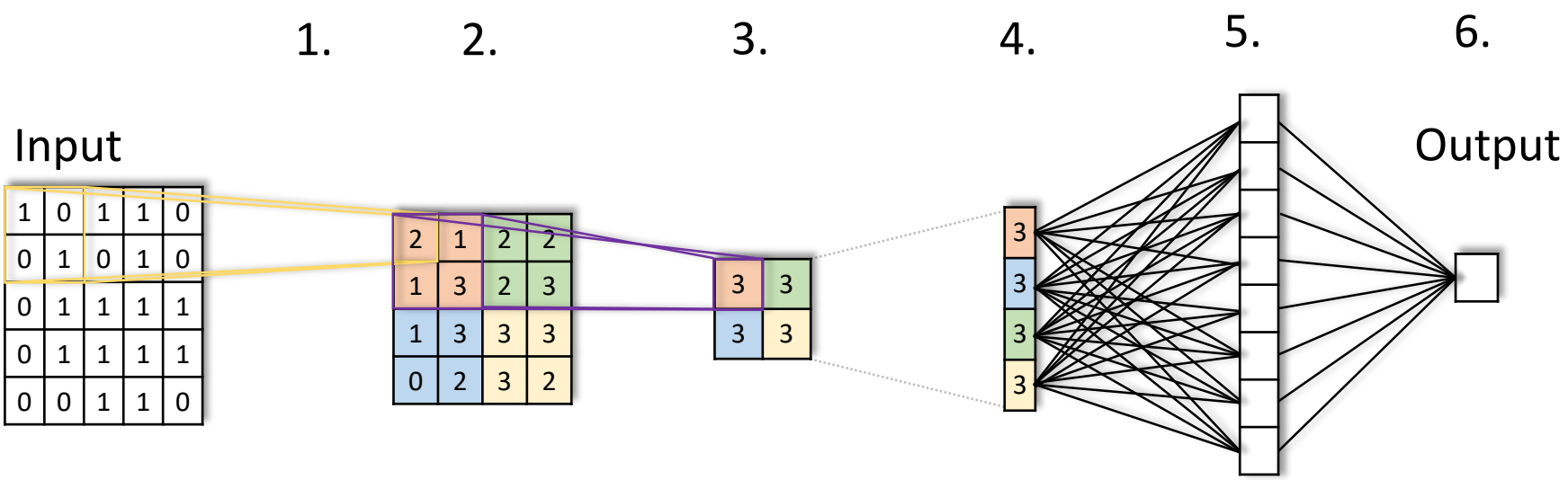
Convolutional Neural Networks

Stride: 1*1
Kernel: 2*2



→Several Kernels! (e.g.32,64…)

Basic Architecture



1. Convolution
2. Activation function
3. Pooling (max or average = aggressive downsampling)
4. Flatten matrix
5. Fully connected layer
6. Output neuron

CNN MNIST

Key Ideas

- Sparse connectivity (one output in a feature map is connected to a few features in the previous layer)
- Parameter sharing (each weight in a kernel is used at every position of the input)
  → translational invariance

Key Ideas

- Sparse connectivity (one output in a feature map is connected to a few features in the previous layer)
- Parameter sharing (each weight in a kernel is used at every position of the input)
  → translational invariance

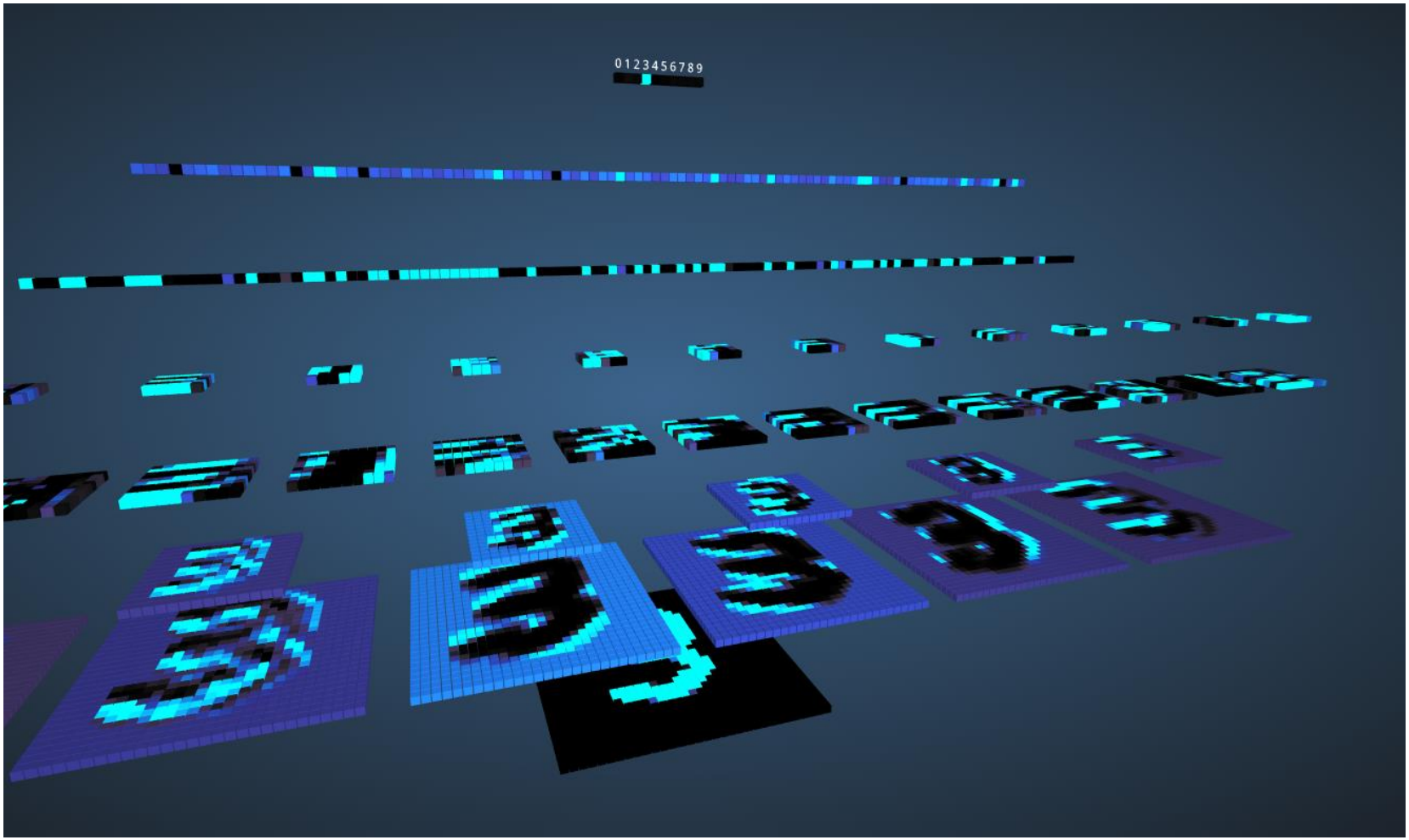
…However there's no consensus about the 'right' architecture!

Major Architectures

… over time several ‘state-of-the-art’ architectures were designed:

- 2012: AlexNet: 60m
- 2014: VGG-16: 134m
- 2014: GoogleLeNet Inception: 5m/23m
- 2015: ResNet: 25m
- 2017: DenseNet: 0.8-40m
- 2018: NasNet: RNN learns to predict CNN Architecture

A practical example



CNN

Application in R

https://github.com/MaximilianPi/CNN

Limitations

- Architecture design → Simple designs, do not try to tune
- Regularization → Dropout, L1, L2,…
- Overfitting/Evaluation → CV
- Computational resources → Take pre-trained models: ‘Transfer Learning’
- Huge amounts of data → Augmentation, Generative Adversarial Networks

Questions?...

Convolutional Neural Networks

Convolution:

$$(f * g)(x) = \int_{-\infty}^{\infty} f(\tau) * g(\tau - t)dt$$

$$S(i,i) = (K * I)(i,j) = \sum_m \sum_n I(i+m, j+n)K(m,n)$$

I = Input (e.g. Image)
K = Kernel (Weights)

Major Architectures

Inception: