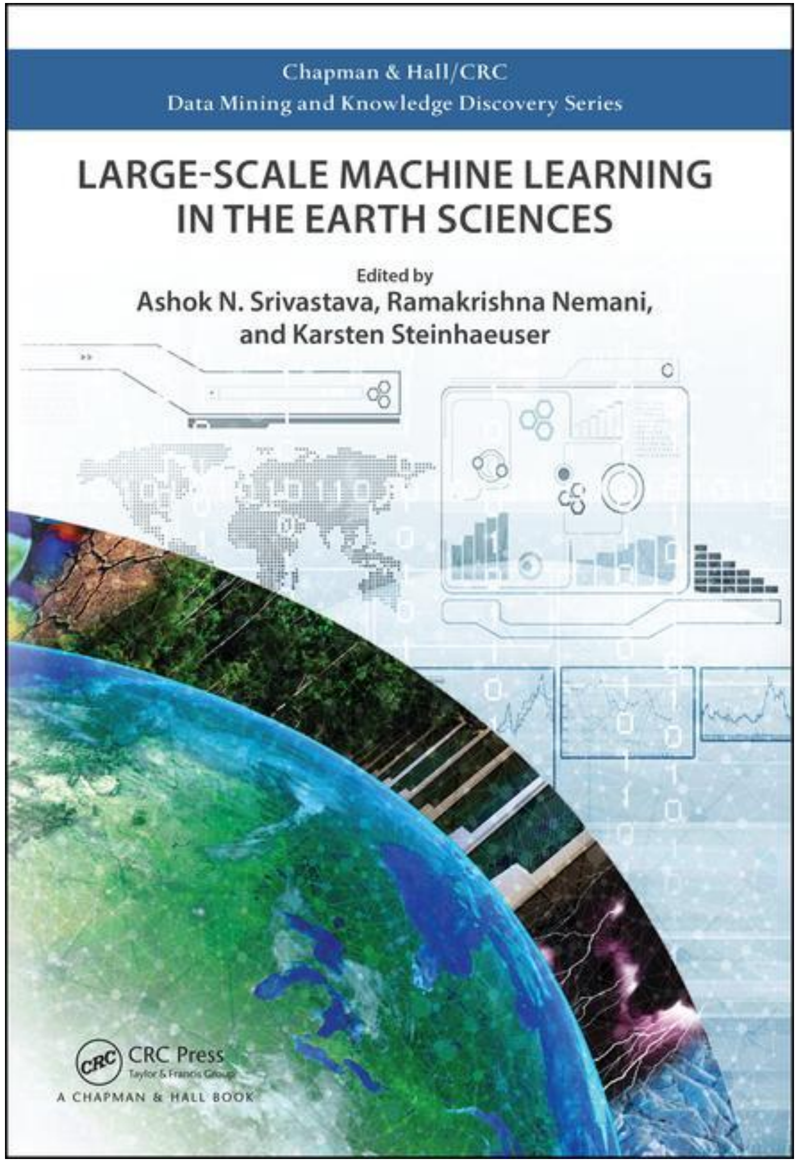
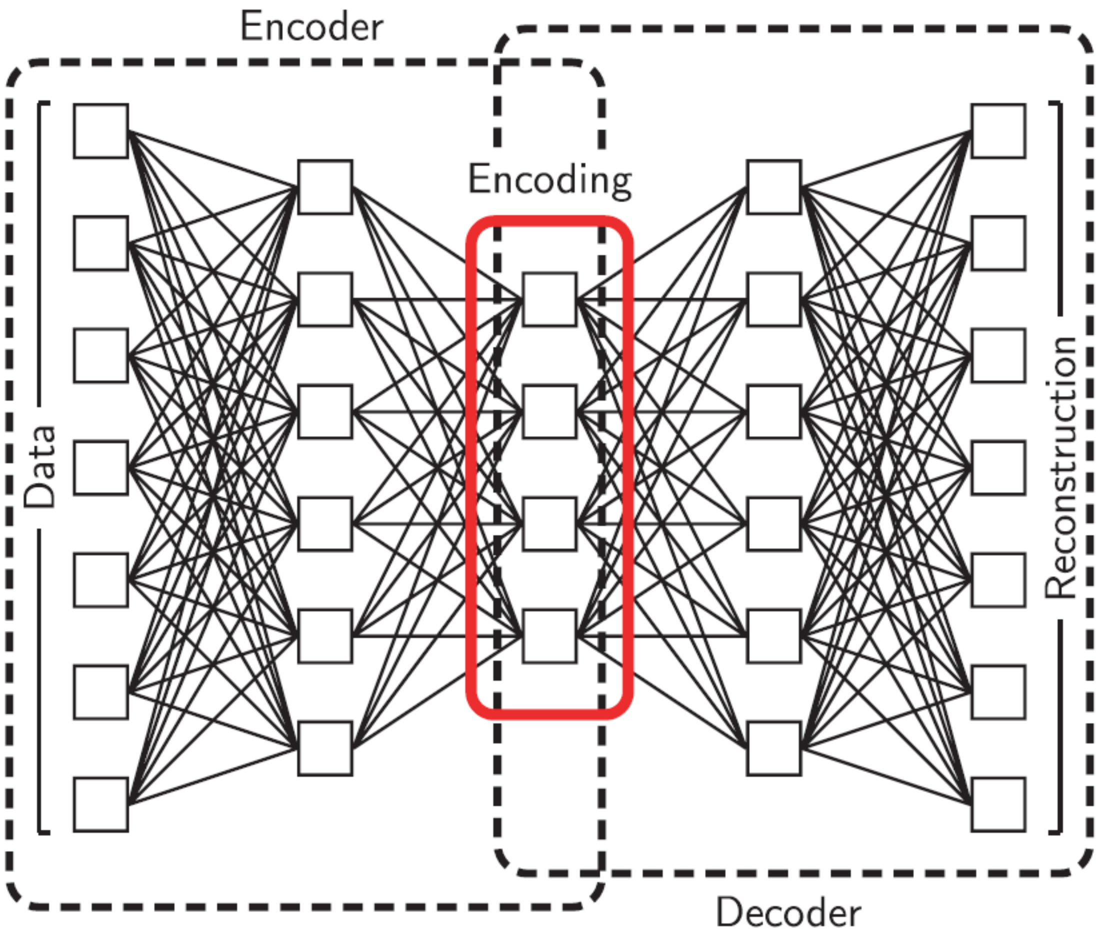
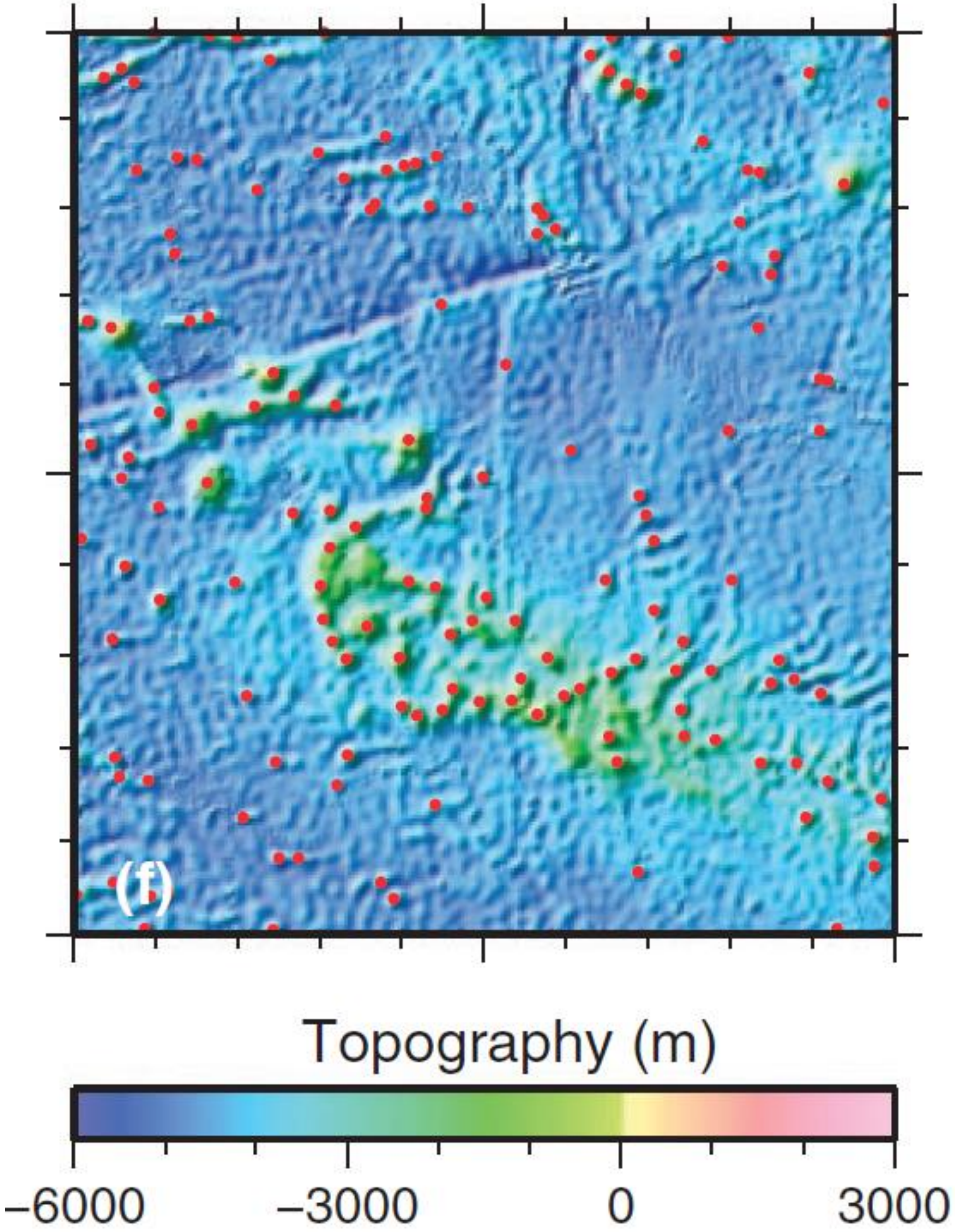
CRC Press



Part 2:
ML in Geosciences

Examples in Geo

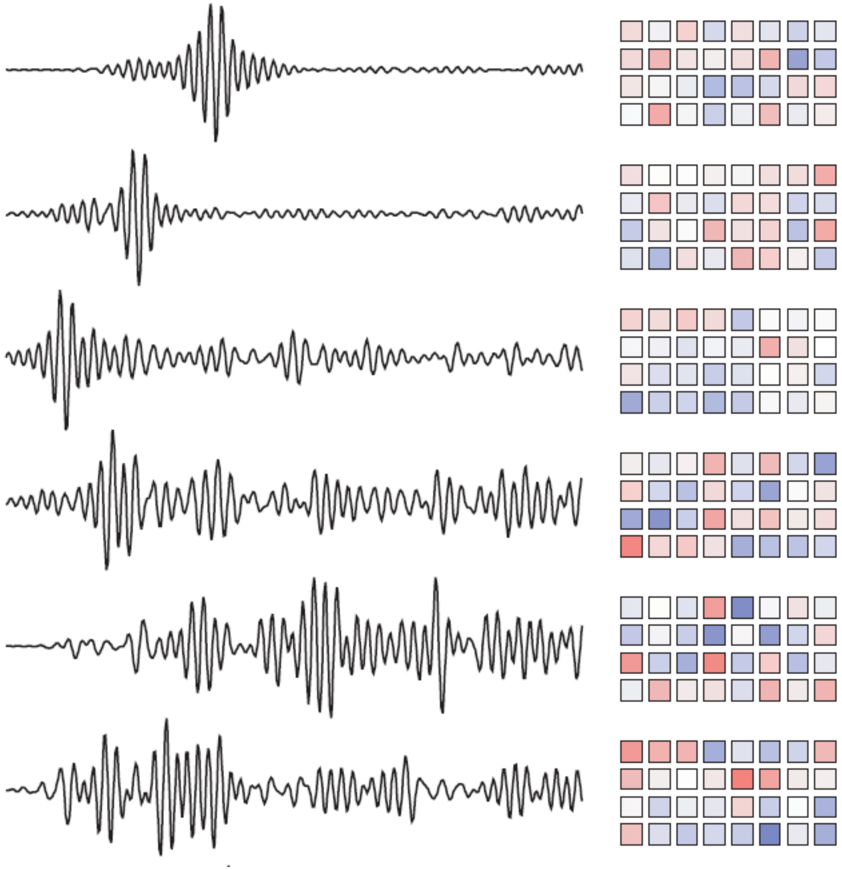




Valentine et al. (2012, 2013)

Examples in Geo

Valentine & Trampert (2012)



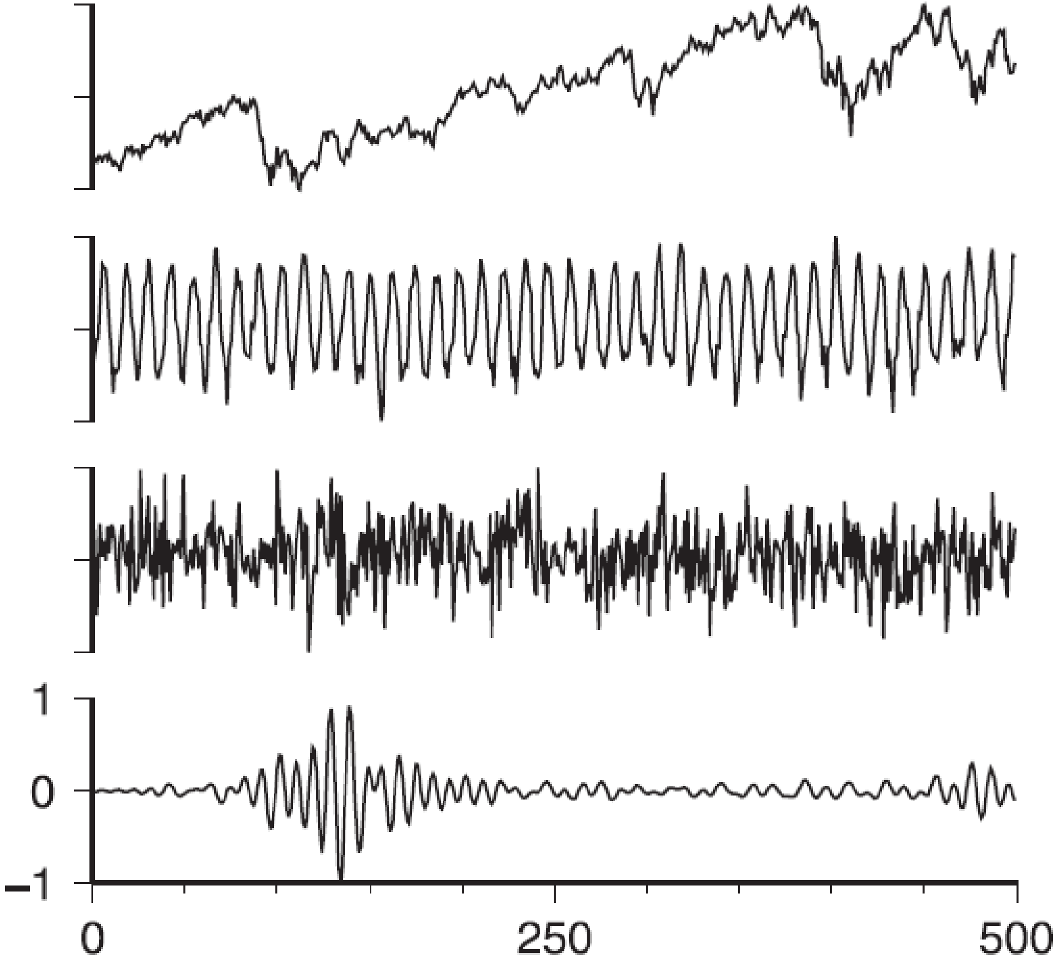
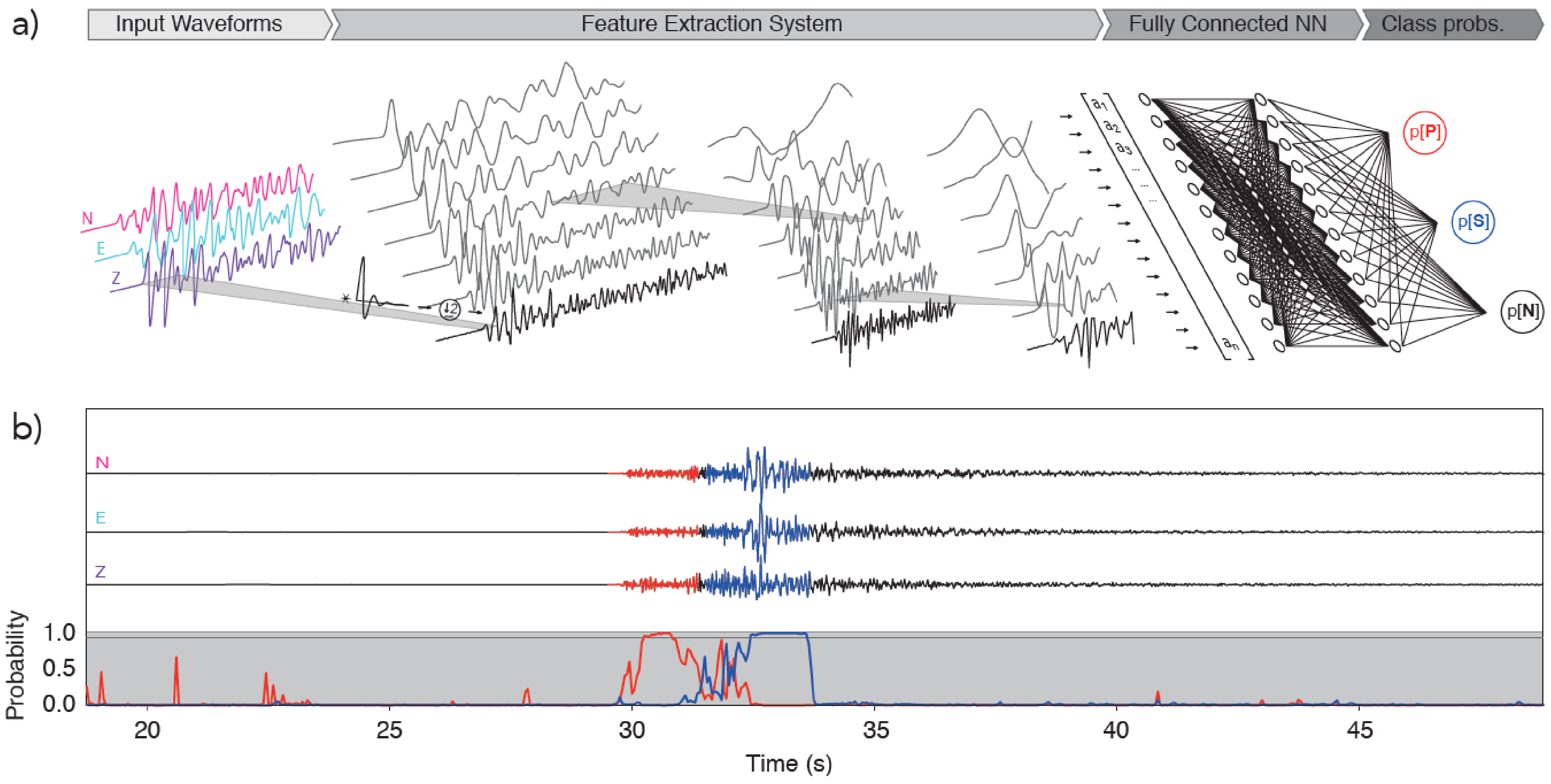


Figure 1. 'Spot the seismogram'. Four 500-point time-series, normalized to take amplitudes in the range [−1, 1]—but the seismogram has sufficient characteristic features to make it instantly recognizable. From top: FTSE 100 closing prices, 2009 June–2011 May; monthly mean temperature for central England, 1950–1991 (Parker *et al.* 1992); Gaussian random noise; long-period surface wave seismogram.

Examples in Geo

Ross et al. (2018)



Examples in Geo

Matteo et al. (in prep) – FAULTS R GEMS

Examples in Geo

www.kaggle.com

Research Prediction Competition

LANL Earthquake Prediction

Can you predict upcoming laboratory earthquakes?

$50,000
Prize Money

Los Alamos National Laboratory · 4,540 teams · 3 months ago

Overview | Data | Notebooks | Discussion | Leaderboard | Rules | **Join Competition**

Overview

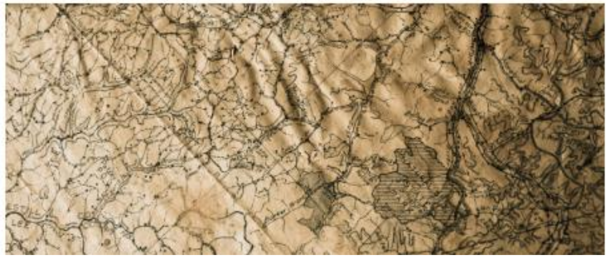
Description

Evaluation

Timeline

Prizes

Forecasting earthquakes is one of the most important problems in Earth science because of their devastating consequences. Current scientific studies related to earthquake forecasting focus on three key points: **when** the event will occur, **where** it will occur, and **how large** it will be.



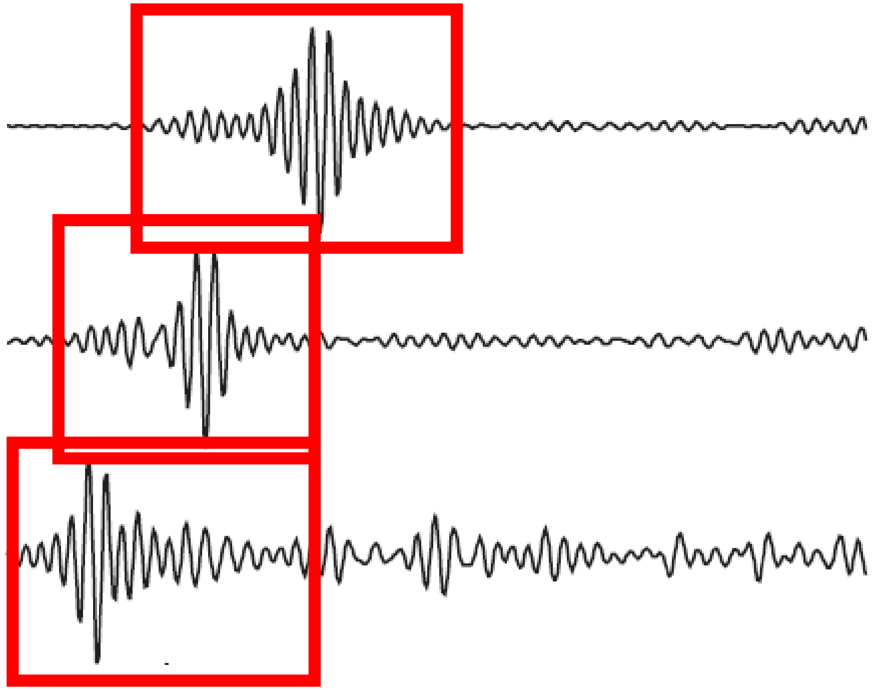
The Workhorse: Convolution

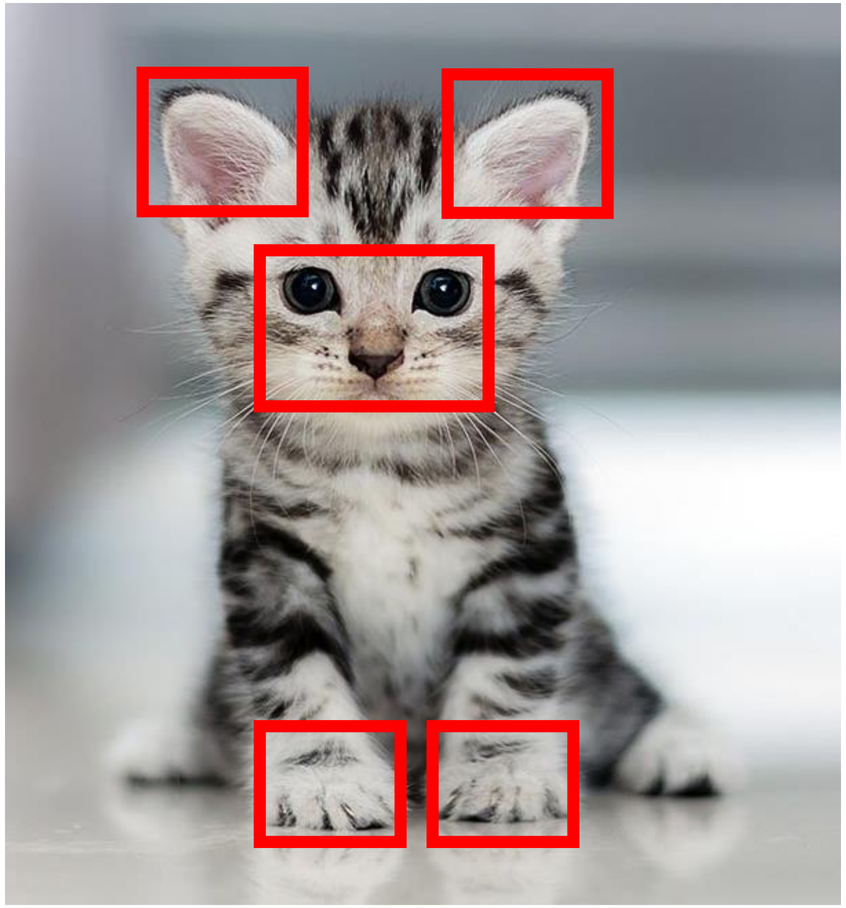
- Most Geoscientific problems involve analysis of time-series, images, or volumetric data
- Fully-connected Neural Networks do not optimally leverage spatial correlations in the data
- Convolutional Neural Networks (CNNs) do a better job at this

Is this the same cat?

Convolutional Neural Networks (CNN)

Rationale: signal correlations are mostly local





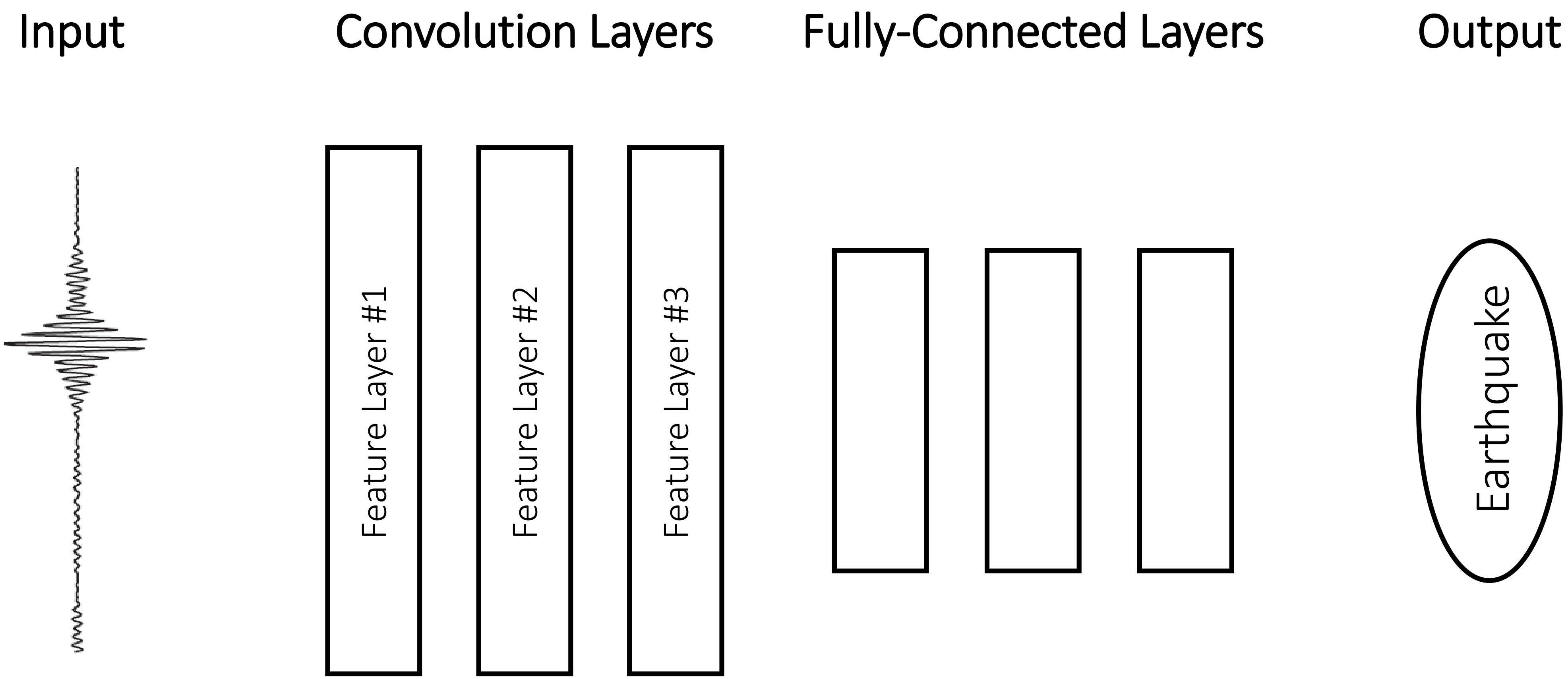
CNN Properties: Shift-Invariance

CNN Properties: Scale-"Invariance"

CNN Properties

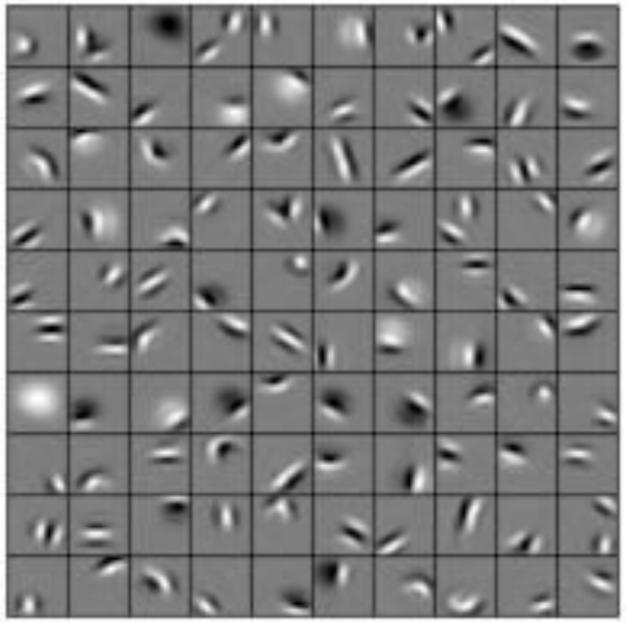
- Practically speaking, most CNNs are robust to:
  - ✓ Translation
  - ✓ Rotation
  - ✓ Scaling
- Intuition: CNNs look for local patterns (“features”) in the data

CNN Architecture

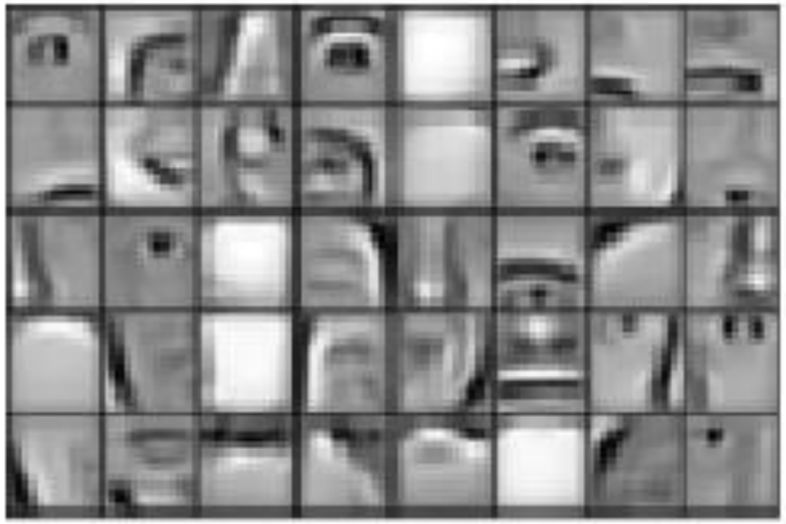


CNN Features

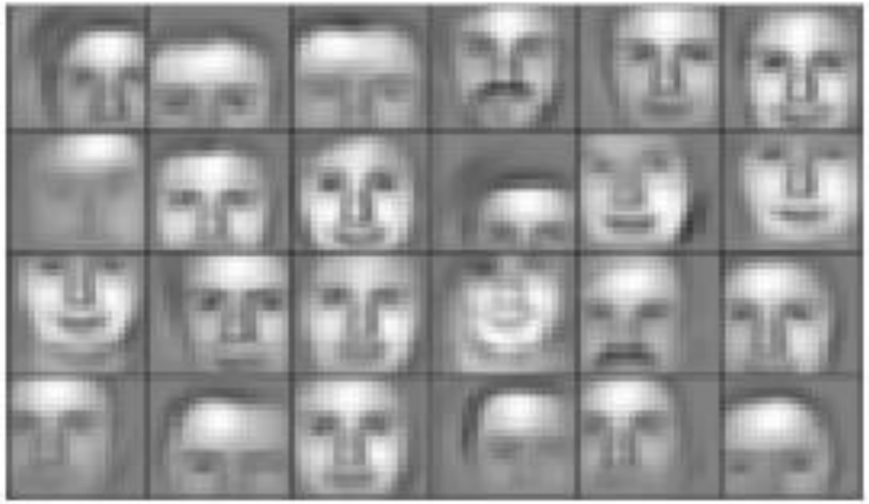
Layer 1



Layer 10

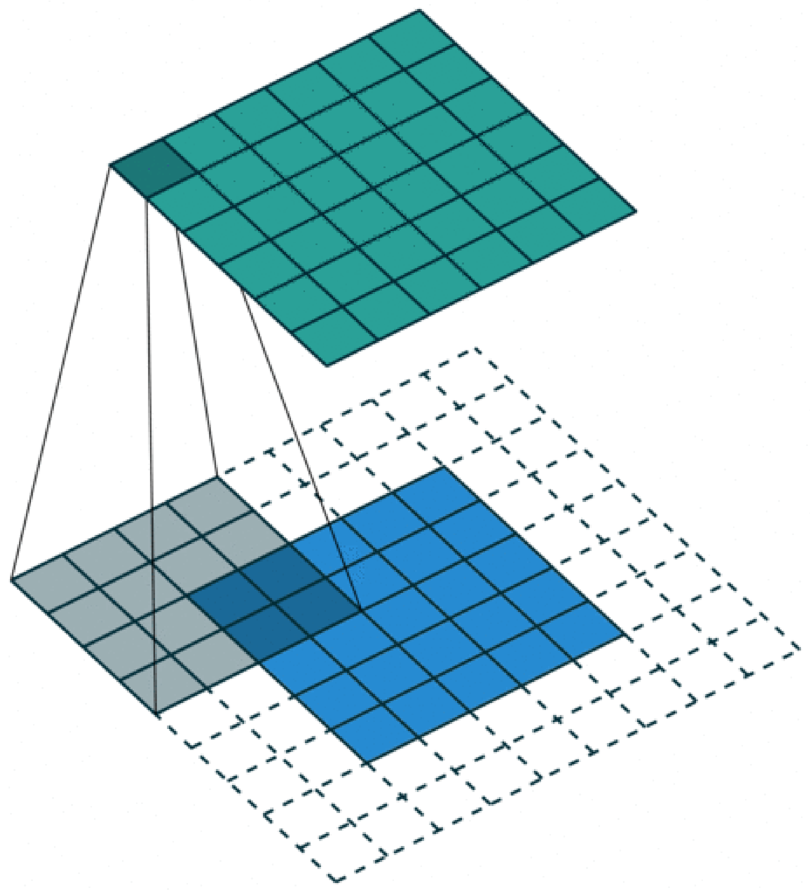


Layer 20



https://devblogs.nvidia.com

Basic CNN Mechanics

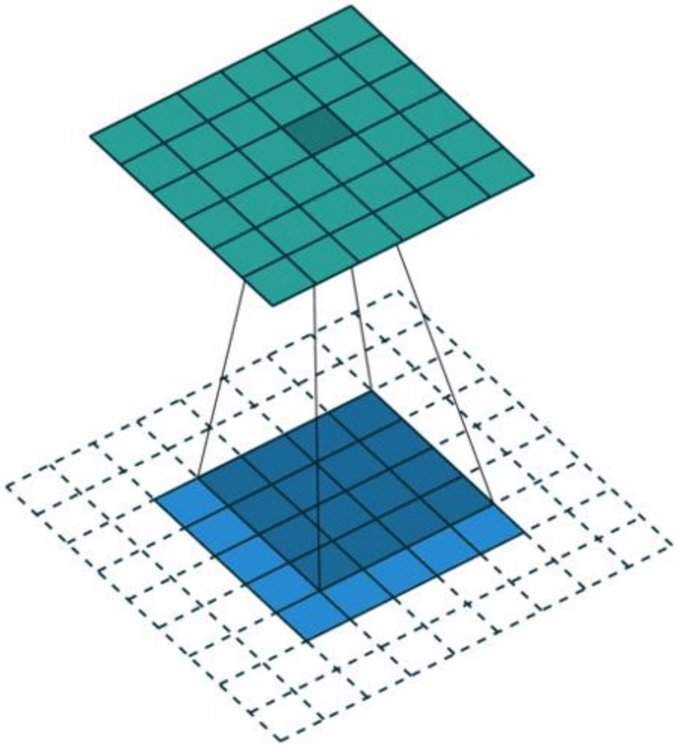


See https://github.com/vdumoulin/conv_arithmetic for more

Basic CNN Mechanics

$$x_j^{(n)} = \sum_{i=1}^{16} k_i x_{i+\mathrm{d}j}^{(n-1)}$$

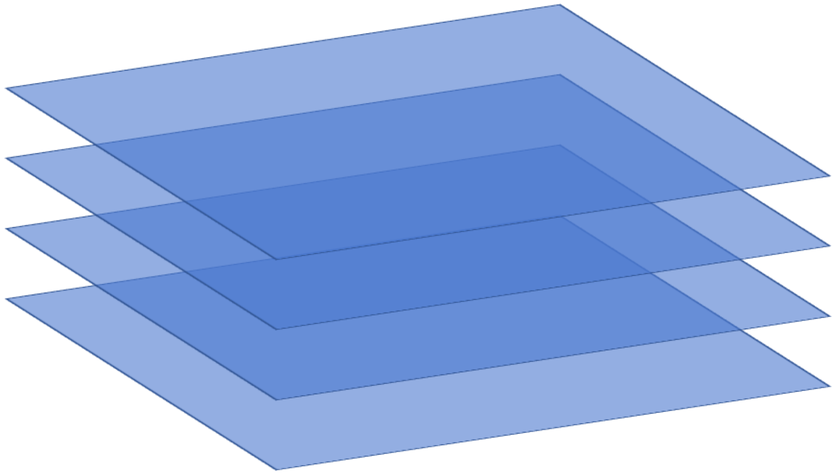
$$x^{(n)}[i,j] = \left(K * x^{(n-1)}\right)[i,j]$$



Basic CNN Mechanics

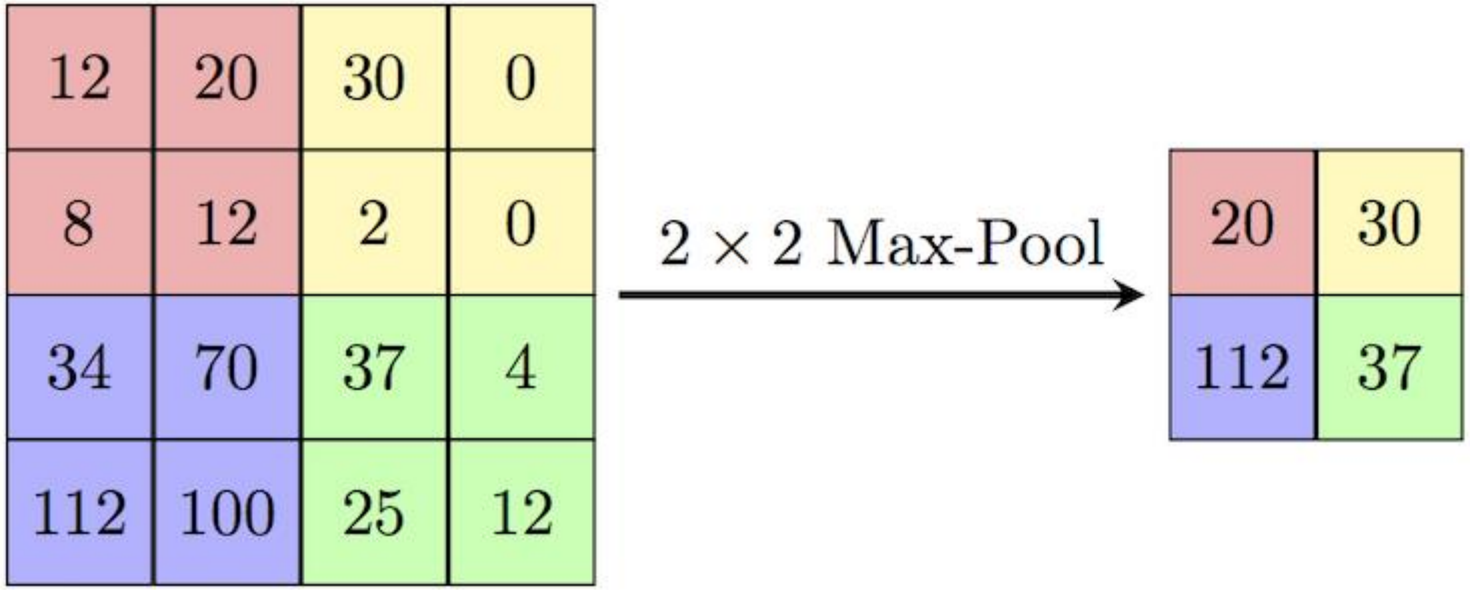
<u>1 convolution layer:</u>

- $k_x \times k_y$ kernel size
- f filters (in this example: 4)
- Input: $N_x \times N_y \times N_f$
- Output: $N_x \times N_y \times f$

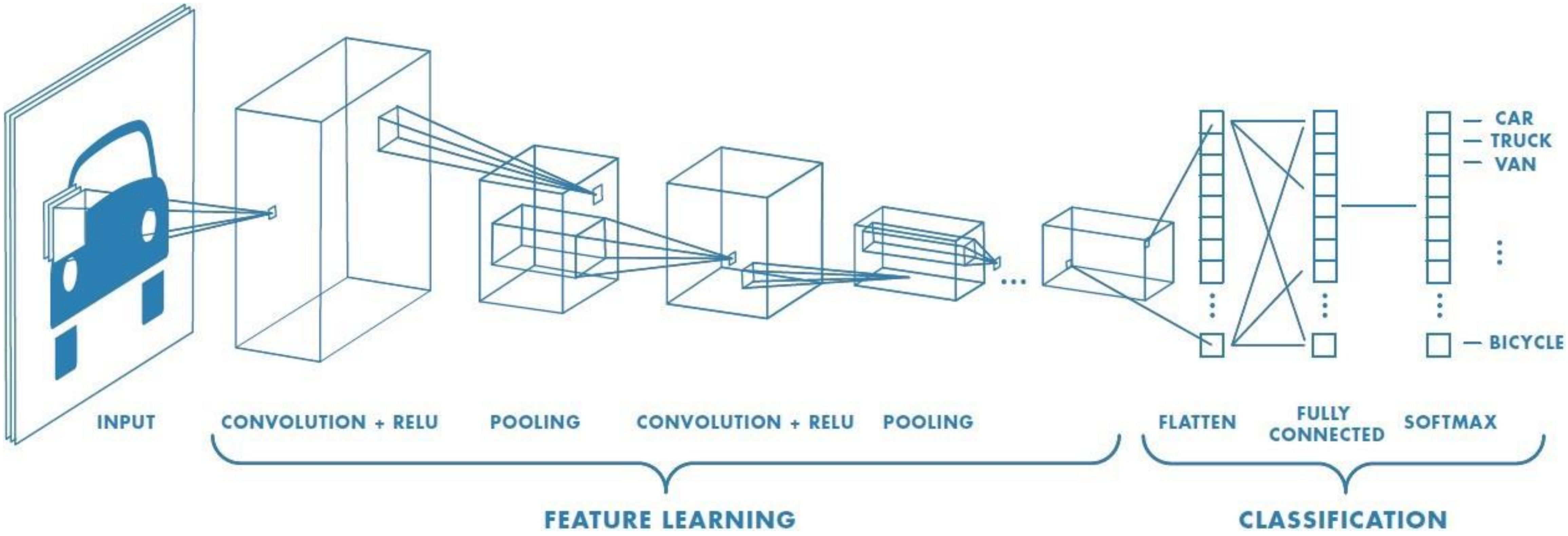


Downsampling / pooling

- Input data often contains redundant information
- Incremental downsampling of the data: pooling



CNN Architecture



Small Network – Big Reach

- In fully-connected networks, the size of a layer is proportional to the size of the input: $O(n)$
- Number of weights scales as $O(n^2)$
- Input of 1000 elements => 1M weights
- Larger networks require more data and more time to train

Small Network – Big Reach

- CNN: size of a layer is $O(1)$ (constant, depending on kernel size)
- Kernels in CNN are usually small (3x3, 7x7, etc.)
- Fewer parameters = faster training, less data
- **Or**: make network much bigger (= deeper)

Time for dirty hands again…