Convolutional Neural Networks

- Computer Vision the a field of artificial intelligence that enables computers to interpret and make decisions based on visual information from the world, such as images and videos.

Applications of Computer Vision

- **Image and Video Recognition**: Identifying objects, and activities in images and videos.
- **Facial Recognition**: Used in security and user authentication.
- **Medical Image Analysis**: Helping in the diagnosis of diseases and in the planning of treatment (e.g., detecting tumors in MRI scans).
- **Autonomous Vehicles**: Enabling cars and drones to navigate and make decisions based on visual inputs.
- **Augmented Reality**: Overlaying virtual objects on the real world for gaming, design, or educational purposes.
- **Retail**: Analyzing in-store cameras for inventory management, customer behavior analysis, and theft prevention.
- **Agriculture**: Monitoring crop health and predicting yield using aerial imagery.

Complexity of Image Data

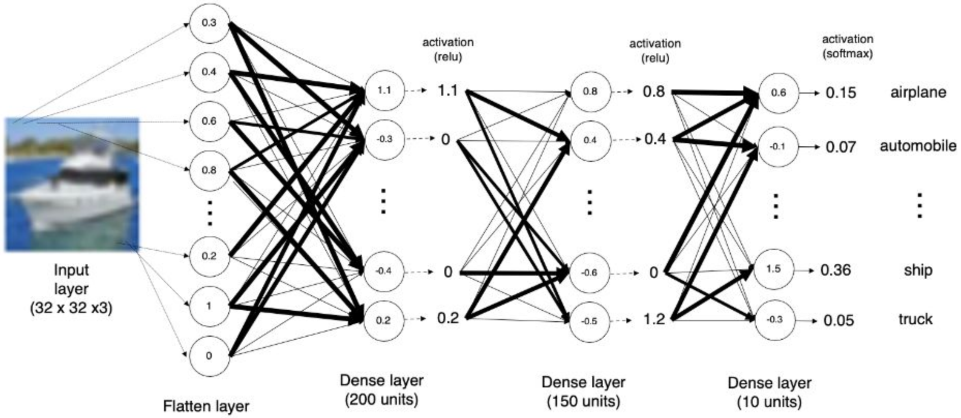
- **High Dimensionality**: Even a small image of size 256x256 pixels has 65,536 pixels, and if it's a color image with 3 color channels (RGB), that's 196,608 features for a single image.
- **Spatial Relationships**: The meaning of a pixel depends on its neighbors, requiring models to understand these spatial relationships.
- **Variability in Lighting, Pose and scale**: Images of the same object can look very different under different lighting conditions, angles or even sizes.
- **Deformations and Occlusions**: Objects can be deformed in various ways and can be partially or fully occluded.
- **Background Clutter**: Important objects can be surrounded by cluttered environments, making them hard to identify.
- **Intra-class Variation**: Objects of the same class can look quite different from each other, while objects of different classes can look similar.

Challenges in Image Classification

- **The Curse of Dimensionality**: As the number of features increase, the amount of data we need to generalize accurately increases exponentially
- **Need for Invariance**: Objects in images can be translated, rotated, or scaled. A model needs to recognize them regardless of their position, orientation, or size.
- **Local and Global Patterns**: Images have local patterns (edges, textures) and global patterns (shapes, objects). A model should capture both.

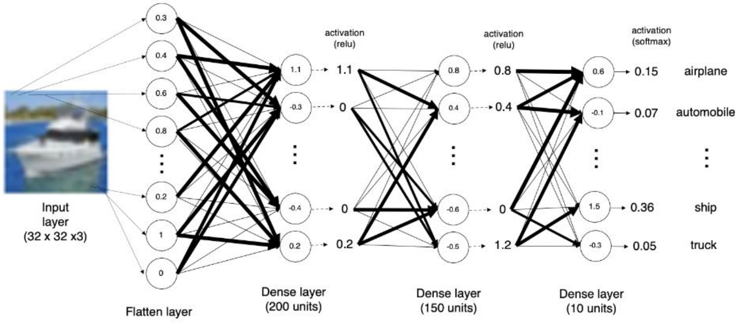
Limitations of MLP for image data

- **Parameter Explosion**: A fully connected layer in an MLP connects every input feature to every neuron in the layer. For image data, this results in a huge number of parameters.
- **Lack of Spatial Hierarchy**: MLPs do not take into account the spatial hierarchy and local patterns of images. They treat input features independently of each other.

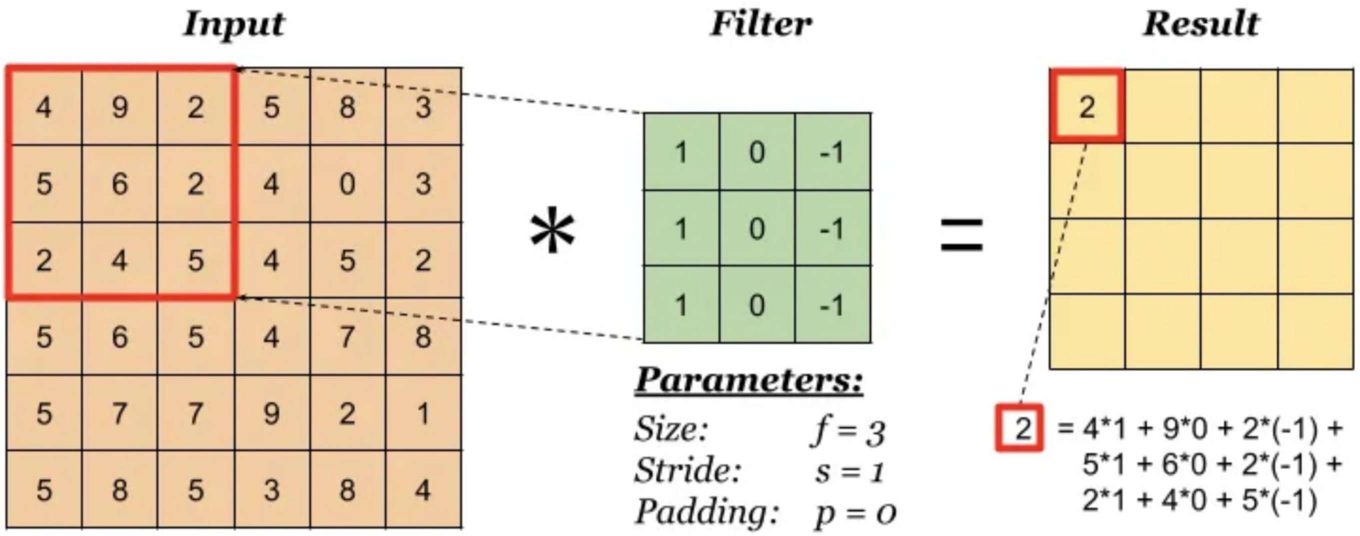


Limitations of MLP for image data

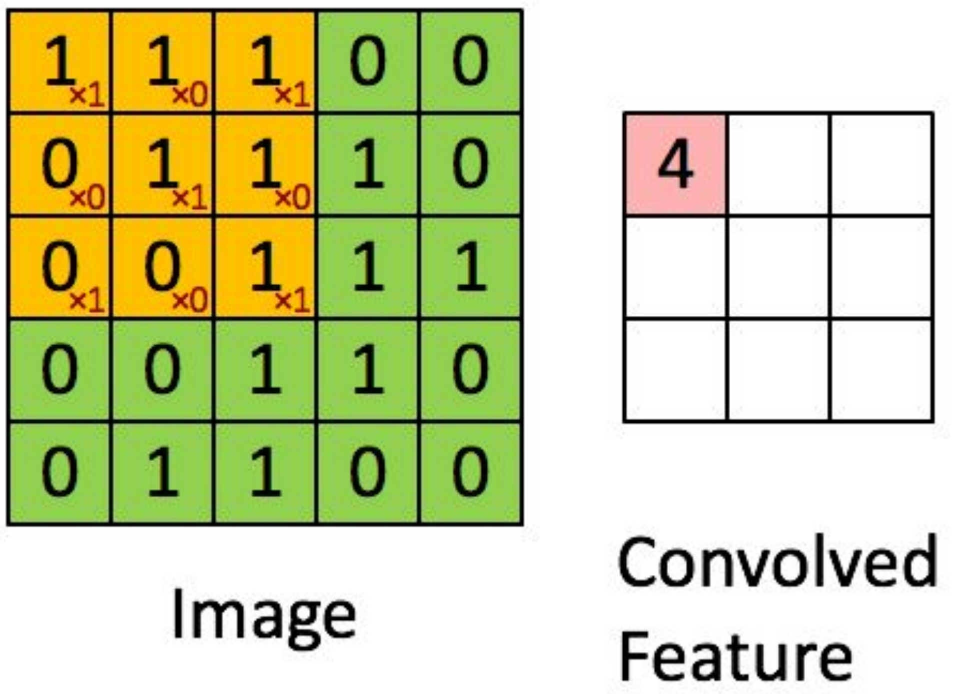
- **Parameter Explosion**: A fully connected layer in an MLP connects every input feature to every neuron in the layer. For image data, this results in a huge number of parameters.
- **Lack of Spatial Hierarchy**: MLPs do not take into account the spatial hierarchy and local patterns of images. They flatten the image and treat input features independently of each other.
- **Equal Treatment of All Connections**: Each neuron has its own weights, which are independently learned from the weights of the other neurons. The network can’t learn to treat spatially close neurons differently from spatially far neurons.



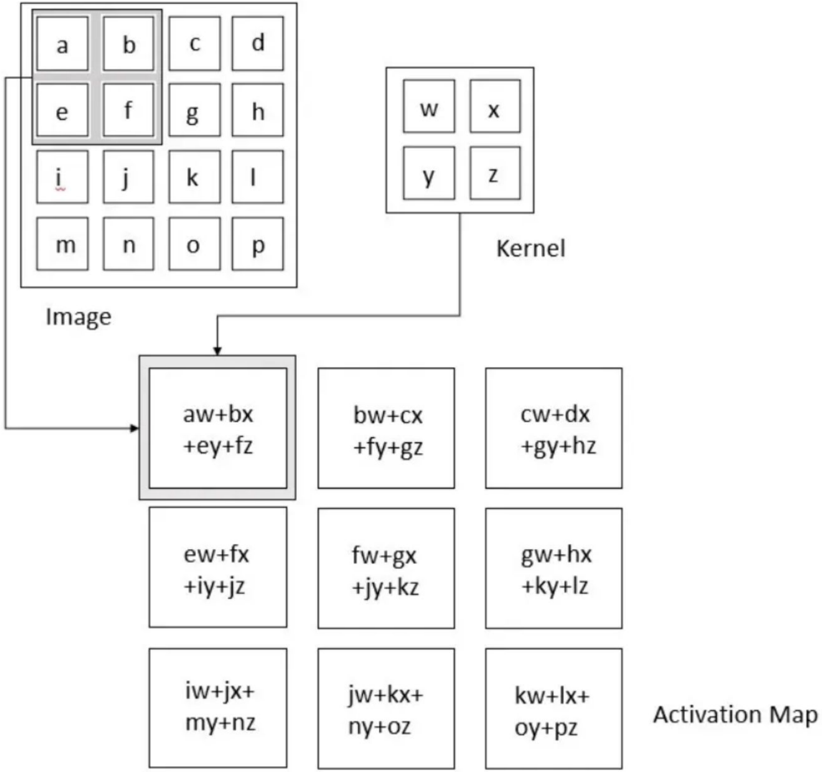
Convolutional Neural Networks



Convolutional Neural Networks

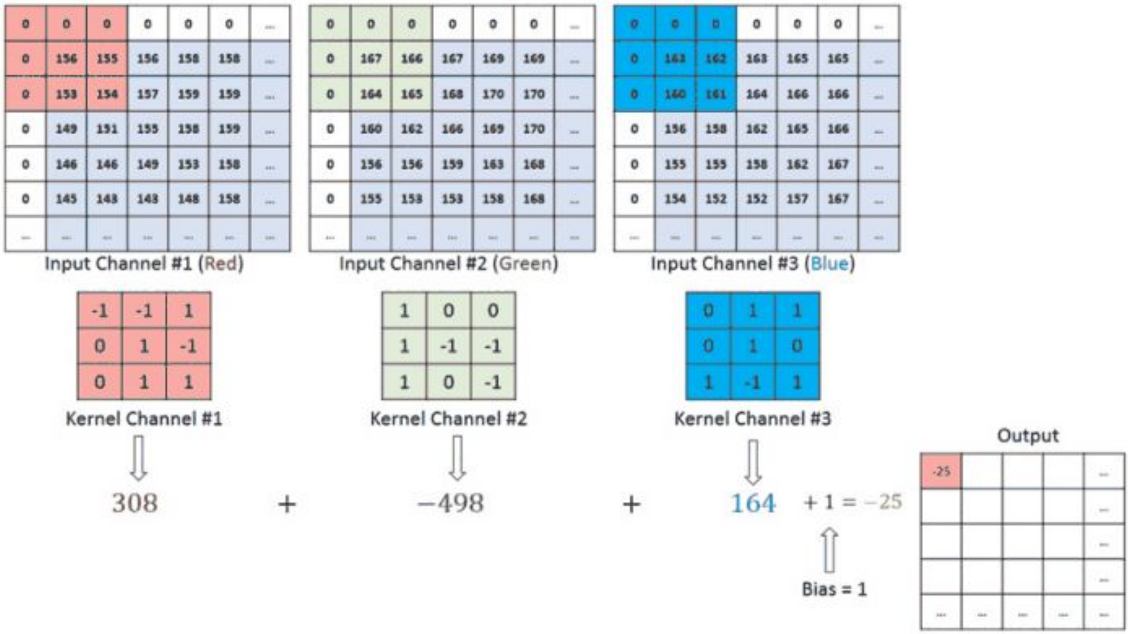


Convolutional Neural Networks



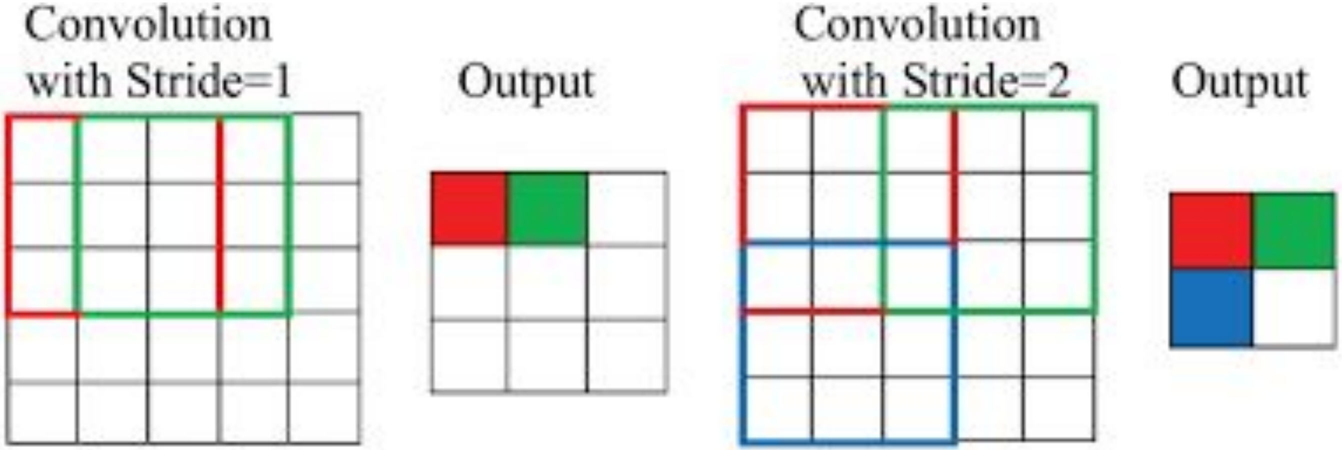
Convolutional Neural Networks

What about for 3-d input?



Convolutional Neural Networks

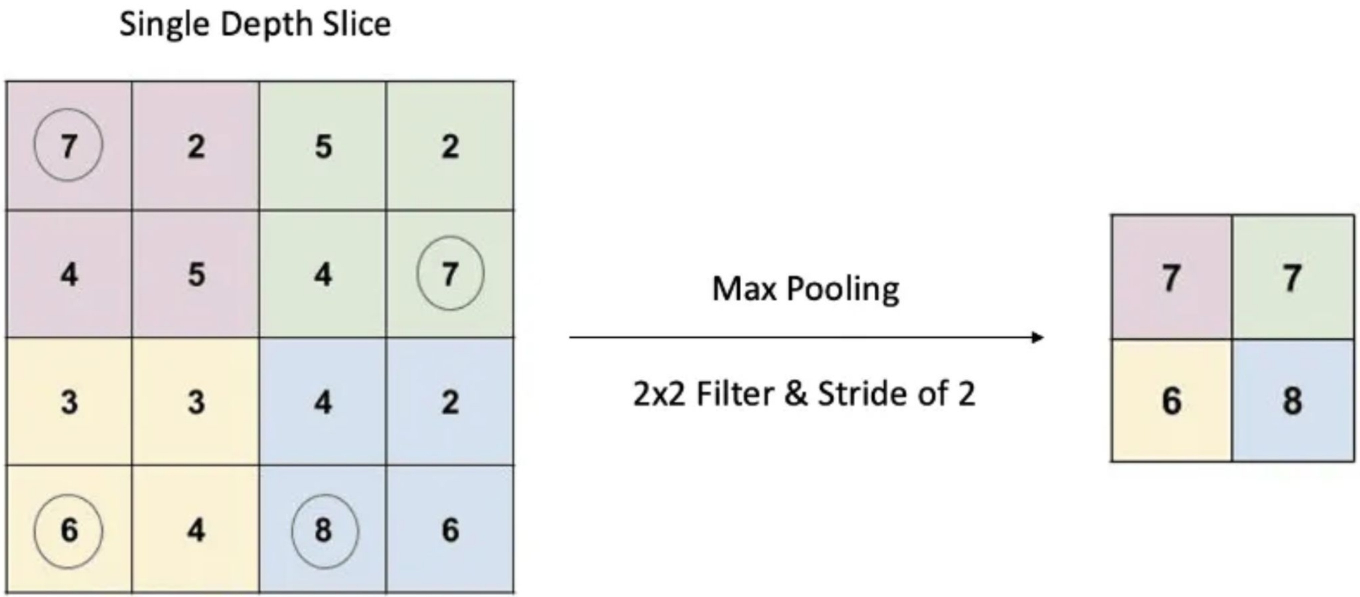
Stride (to reduce # params)



Convolutional Neural Networks

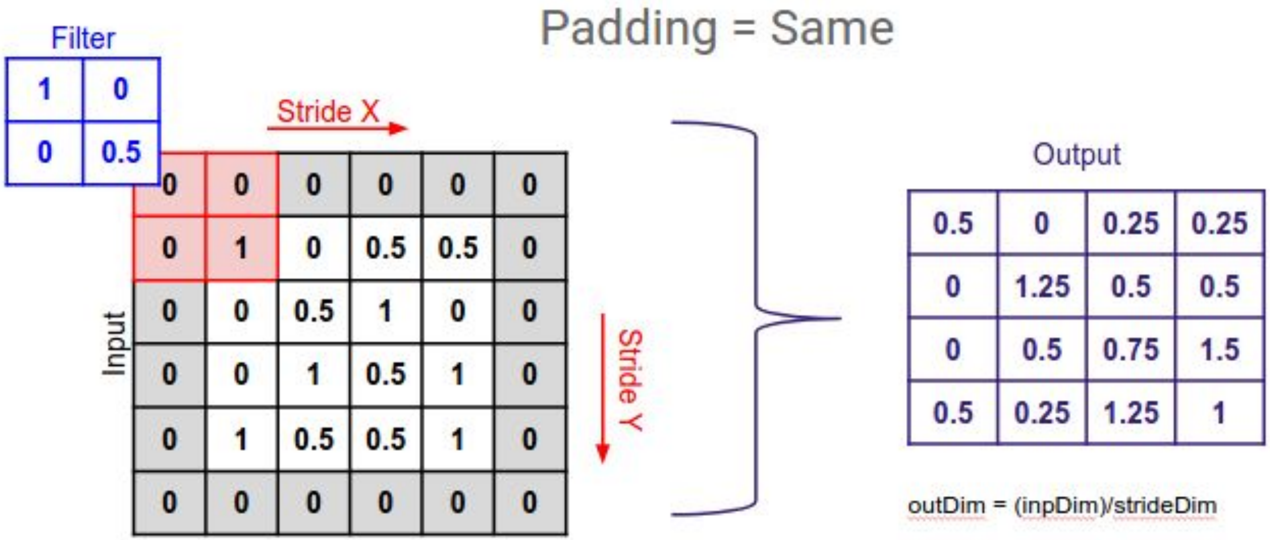
Pooling Layer (to reduce # params)

- Sometimes it may be enough to know a pattern is there



Convolutional Neural Networks

Padding



Convolutional Neural Networks

Output shape

$$n_{out} = \left\lfloor \frac{n_{in} + 2p - k}{s} \right\rfloor + 1$$

n_{in}: number of input features

n_{out}: number of output features

k: convolution kernel size

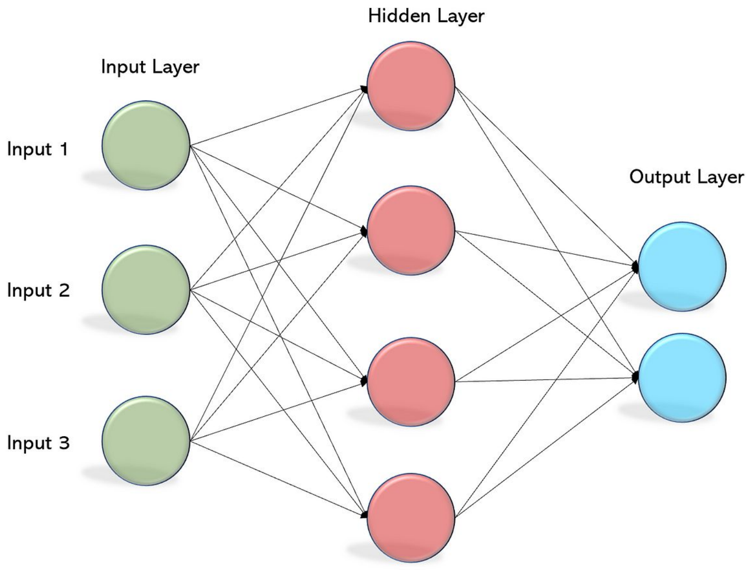
p: convolution padding size

s: convolution stride size

Convolutional Neural Networks

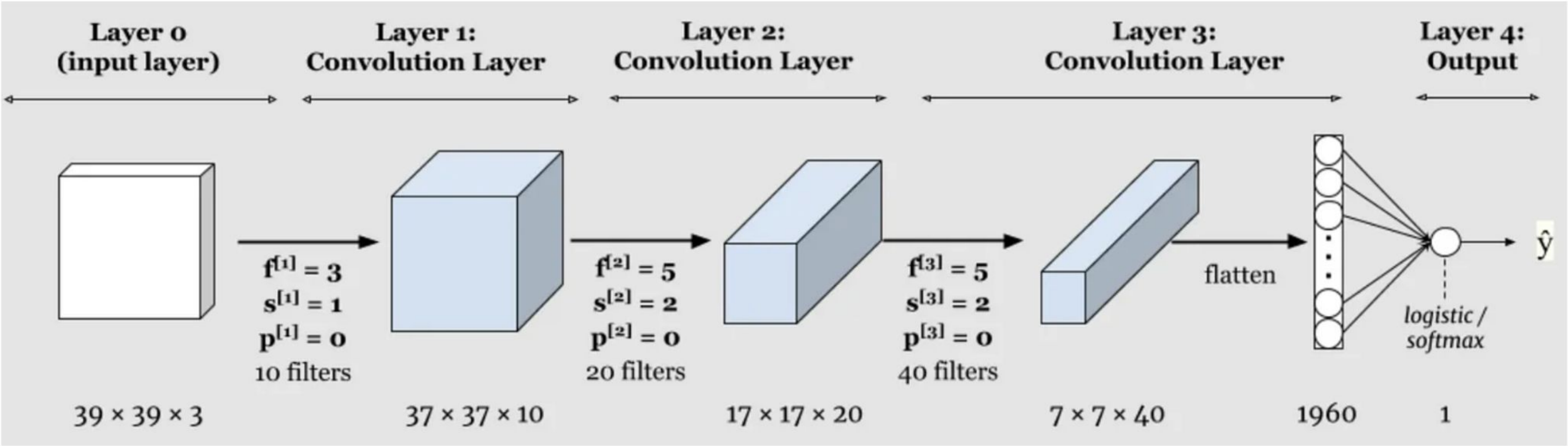
Dense Layer

- At the end of the convolutions we need a way to combine all patterns seen in an image



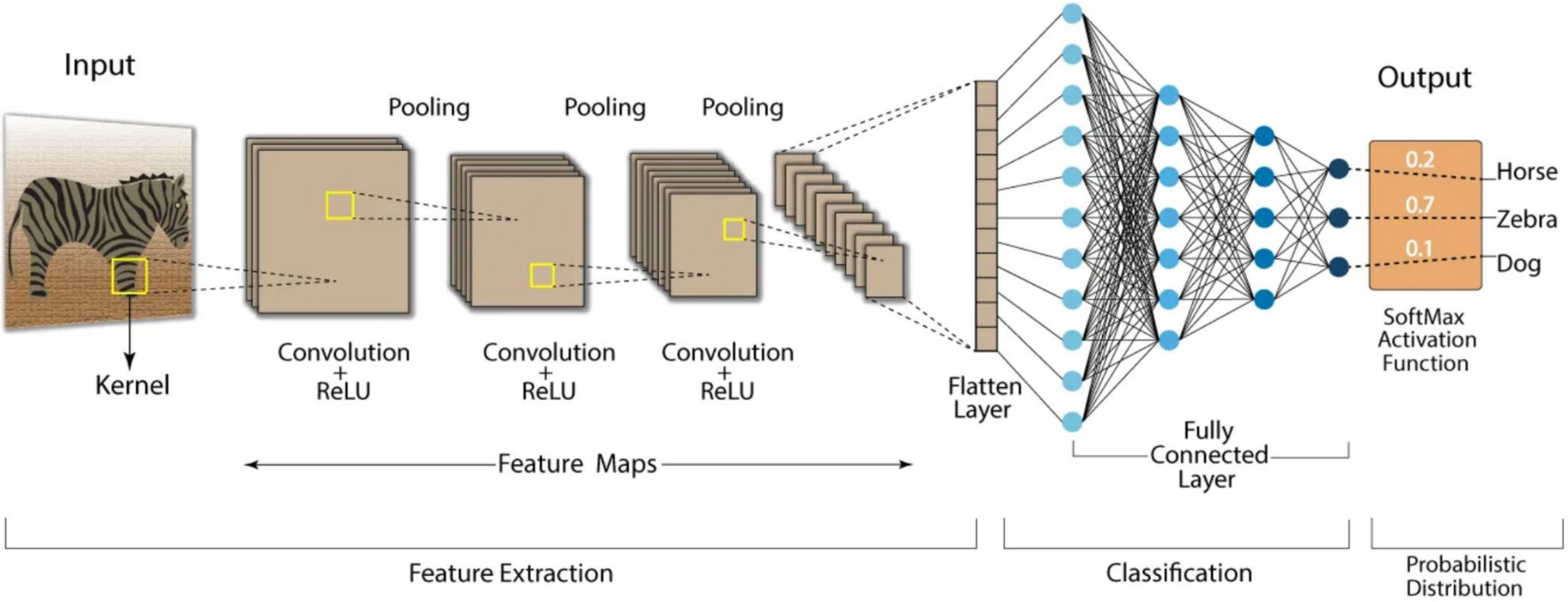
Convolutional Neural Networks

Overview



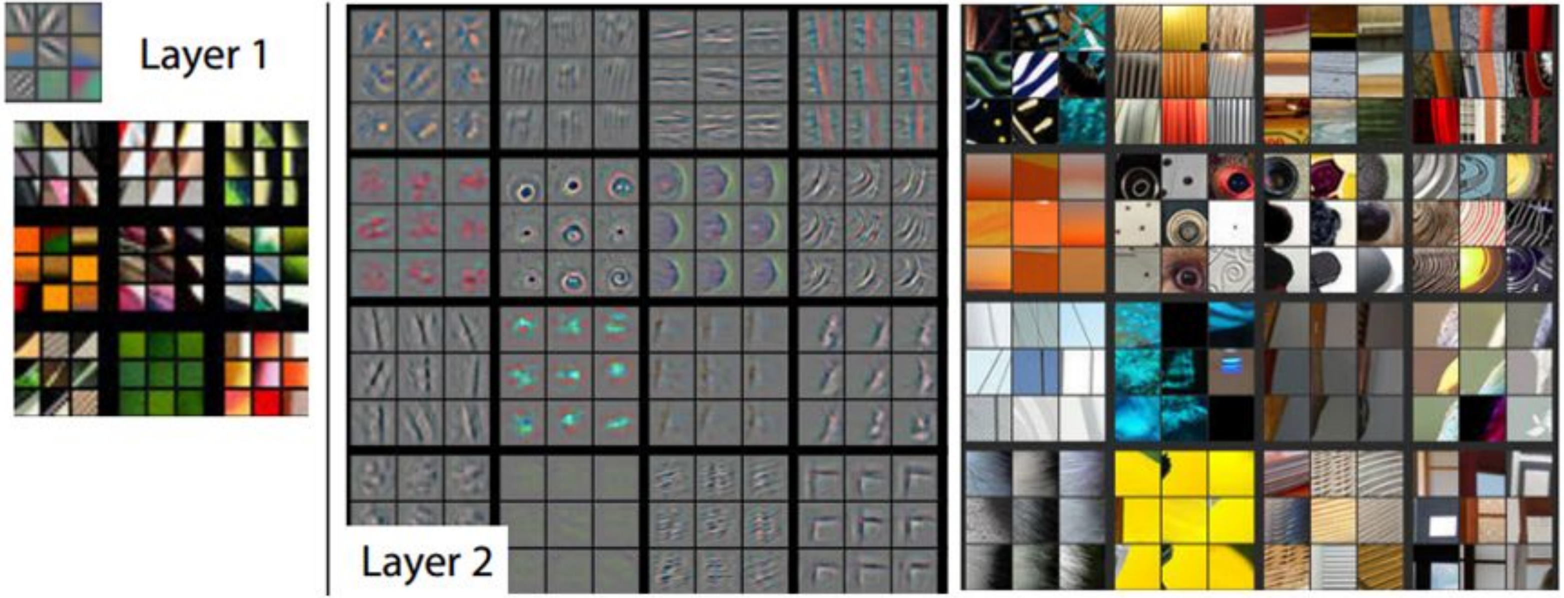
Convolutional Neural Networks

Overview



Convolutional Neural Networks

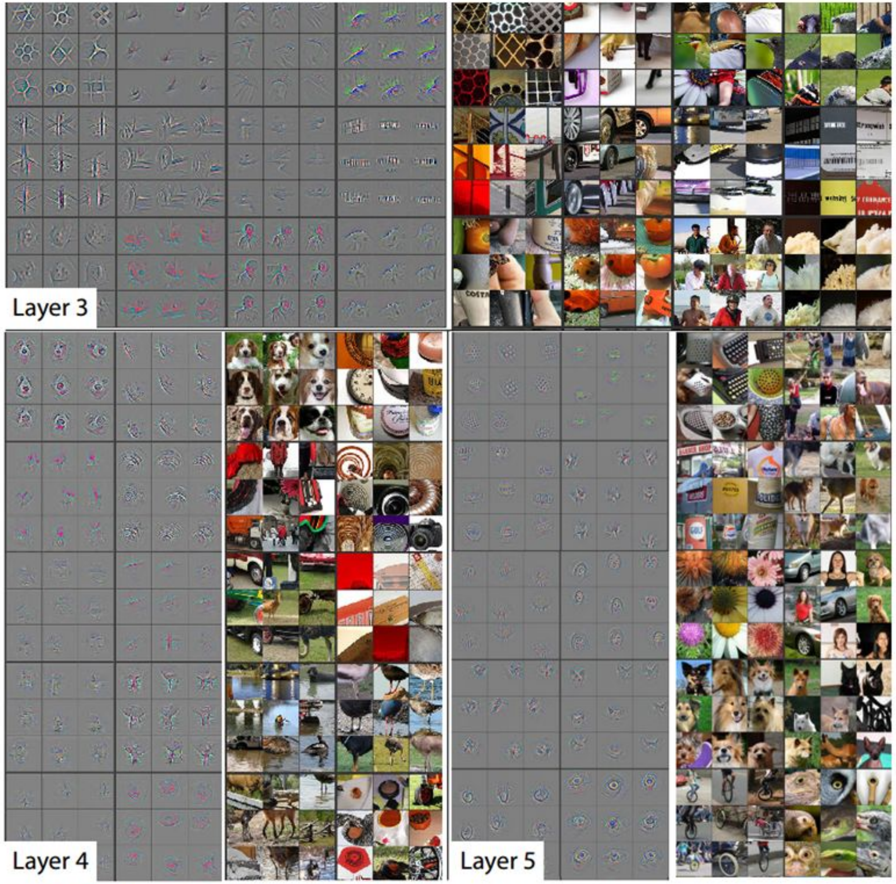
From low level to high level patterns



Visualizations of Layer 1 and 2. Each layer illustrates 2 pictures, one which shows the filters themselves and one that shows what part of the image are most strongly activated by the given filter. For example, in the space labled Layer 2, we have representations of the 16 different filters (on the left)

Convolutional Neural Networks

From low level to high level patterns



Visualizations of Layers 3, 4, and 5