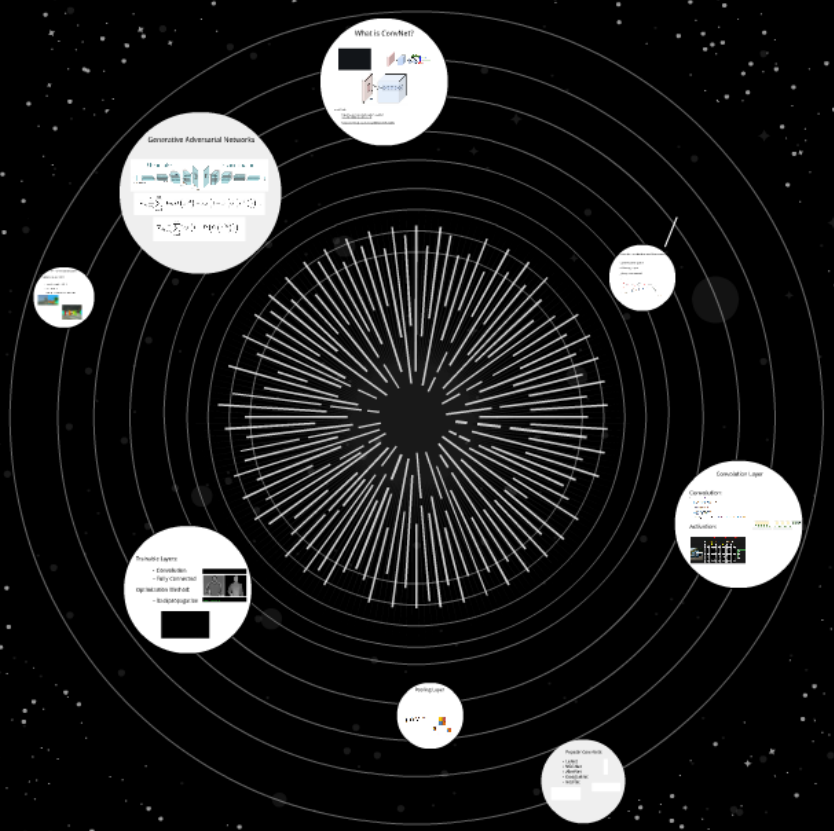
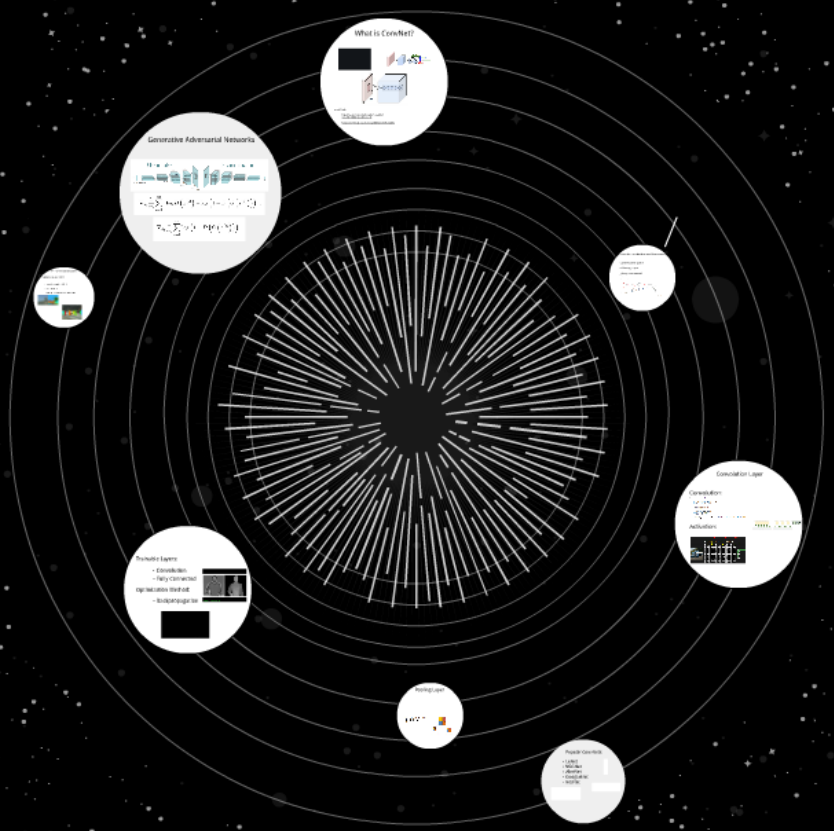


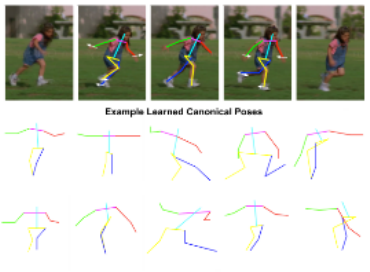
Abolfazl Meyarian
@mlghost



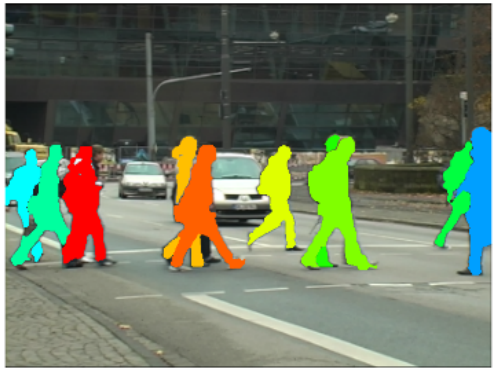
Abolfazl Meyarian
@mlghost

Convolutional Neural Network

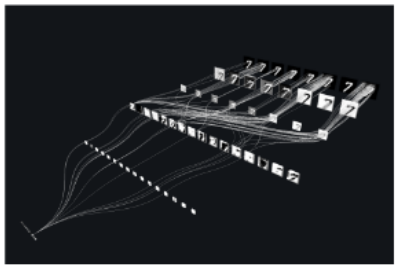
when we can use CNNs ?

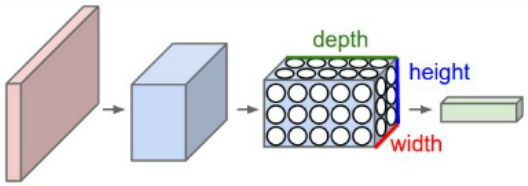


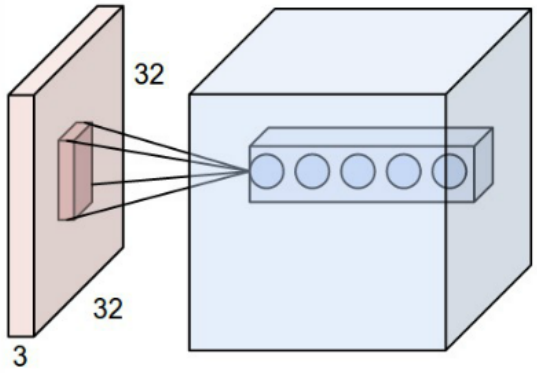
- high dimmensional Data
- Grid like Data
- widely used for image processing



What is ConvNet?







useful links:

http://cs.stanford.edu/people/karpathy/convnetjs/demo/cifar10.html

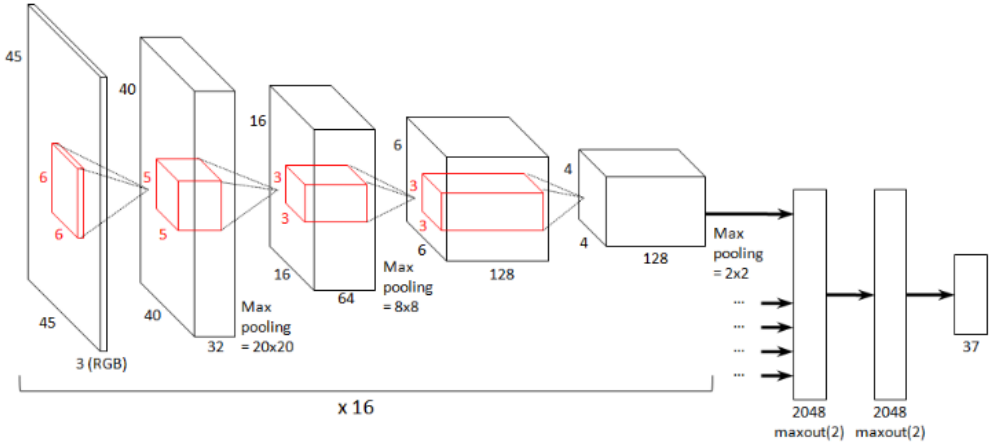
http://cs231n.github.io/convolutional-networks/

ConvNet Architecture and Operations

1.Convolution Layer

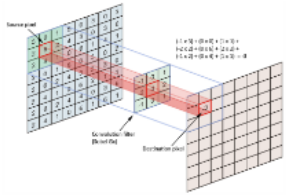
2.Pooling Layer

3.Fully Connected



Convolution Layer

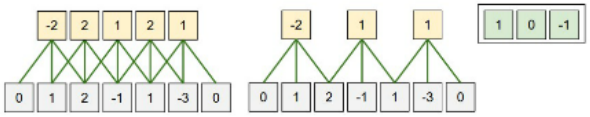
Convolution:

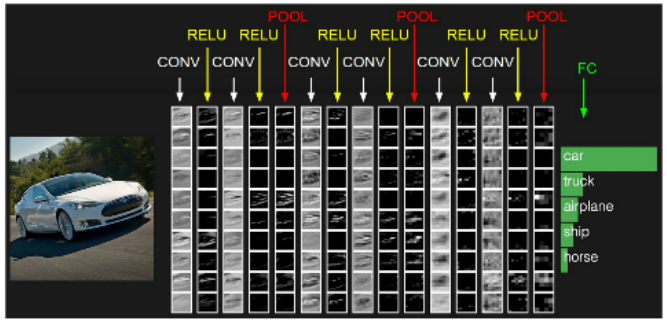


Summary. To summarize, the Conv Layer:

- Accepts a volume of size $W_1 \times H_1 \times D_1$
- Requires four hyperparameters:
  - Number of filters K,
  - their spatial extent F,
  - the stride S,
  - the amount of zero padding P.
- Produces a volume of size $W_2 \times H_2 \times D_2$ where:
  - $W_2 = (W_1 - F + 2P)/S + 1$
  - $H_2 = (H_1 - F + 2P)/S + 1$ (i.e. width and height are computed equally by symmetry)
  - $D_2 = K$

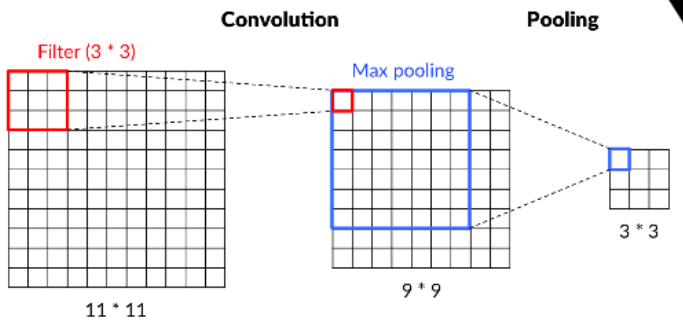
Activation:

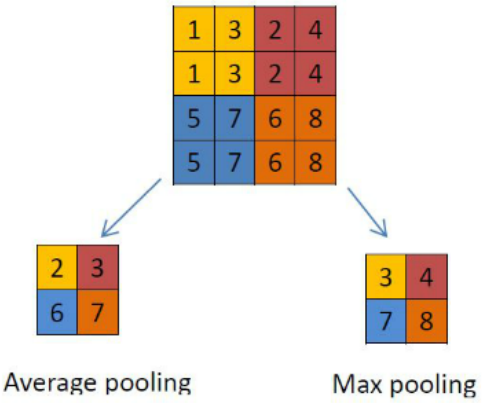




Pooling Layer

- Accepts a volume of size $W_1 \times H_1 \times D_1$
- Requires two hyperparameters:
  - their spatial extent F,
  - the stride S,
- Produces a volume of size $W_2 \times H_2 \times D_2$ where:
  - $W_2 = (W_1 - F)/S + 1$
  - $H_2 = (H_1 - F)/S + 1$
  - $D_2 = D_1$





Trainable Layers:

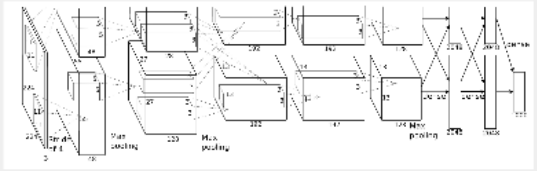
- Convolution
- Fully Connected

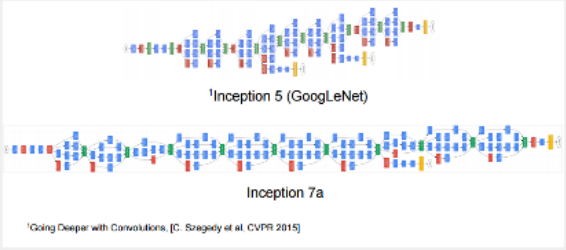
Optimization Method:

- Backpropagation

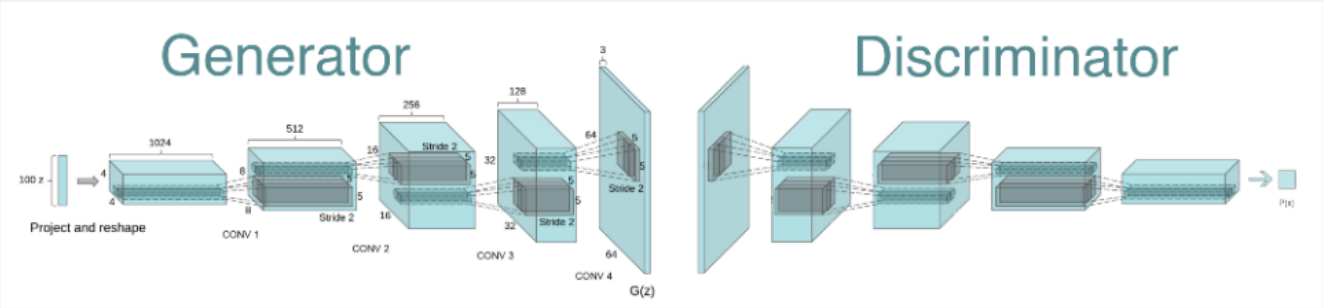
Popular ConvNets

- LeNet
- VGGNet
- AlexNet
- GoogLeNet
- ResNet



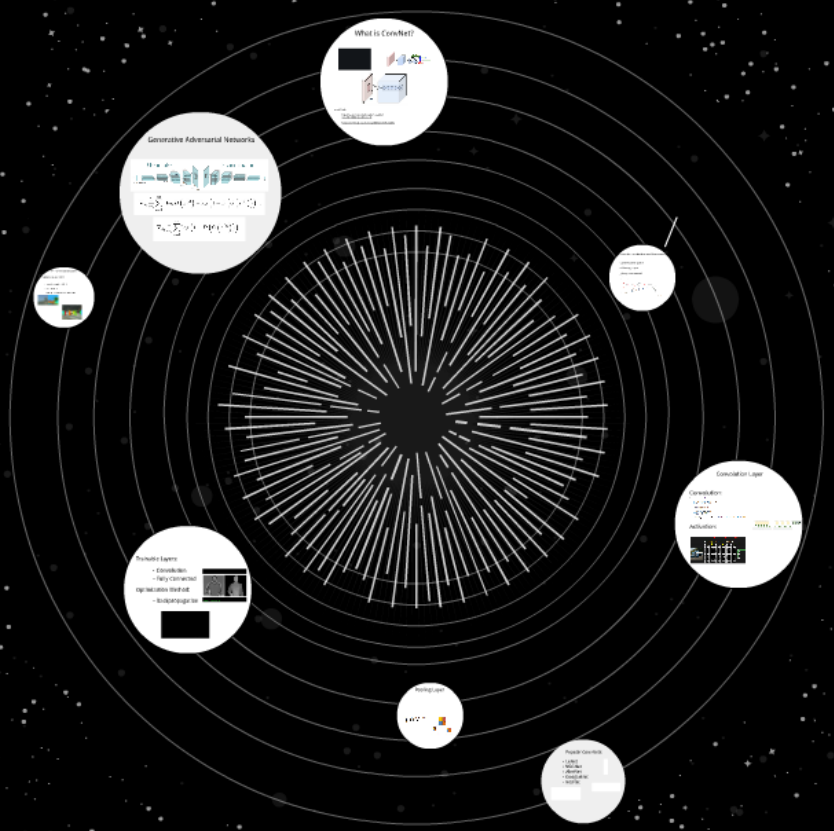


Generative Adversarial Networks



$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$



Abolfazl Meyarian
@mlghost