CNN algorithms

Liao Yiming
Department of Computer Science
Shanghai Jiao Tong University

Baseline

- 1 basic architecture
- 2 history and evolution
- 3 our algorithms

1.1 overview of architecture

- why CNN?

  Regular Neural Nets: too much parameters

  Cifar10(32*32*3)

- Layers used to build ConvNets

  Convolutional Layer

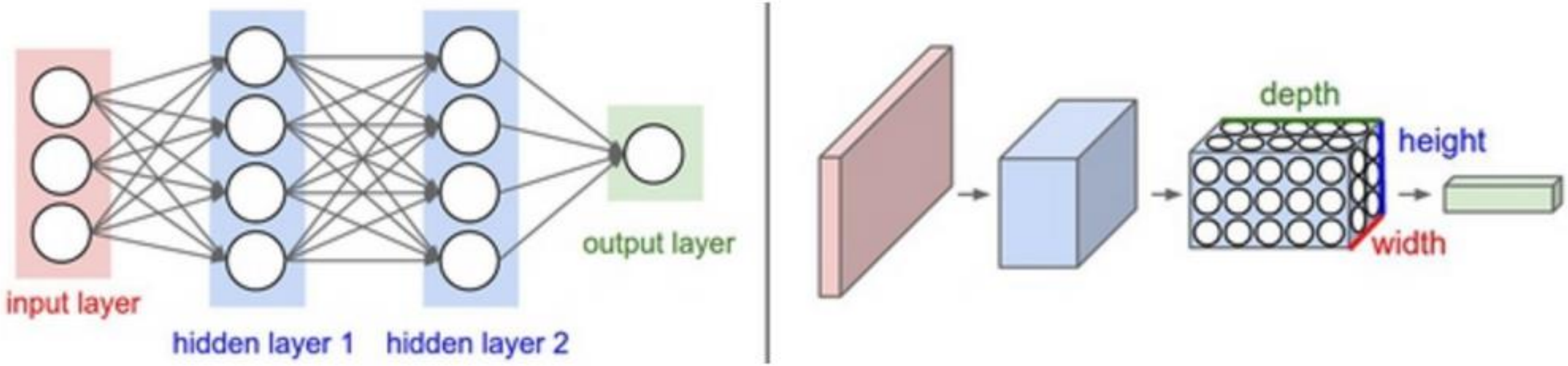
  Pooling Layer

  Fully-Connected Layer

1.2 Convolutional Layer

- Local Connectivity
- Spatial arrangement
- Parameter Sharing

Local Connectivity



Spatial arrangement

- Depth
- Stride
- Zero-padding

Parameter Sharing

- Control the number of parameter
- Catch feature

1.3 Pooling Layer

- Pooling layer between successive Convolution layers
- reduce the spatial size of the representation
  to reduce the amount of parameters
- control overfitting

- MAX operation

  filters of size 2x2 applied with a stride of 2 every depth slice

  Every MAX operation would in this case be taking a max over 4 numbers

- general pooling

- No pooling layer

1.4 Fully-connected layer

- Neurons in a fully connected layer have full connections to all activations in the previous layer, as seen in regular Neural Networks

- INPUT -> [CONV -> RELU -> POOL]*2 -> FC -> RELU -> FC.

  Here we see that there is a single CONV layer between every POOL layer.

2.1 LeNet

- The first successful applications of Convolutional Networks were developed by Yann LeCun in 1990’s

- Convolutional Layer

  Pooling Layer

  Fully-connected layer

2.2 Alexnet

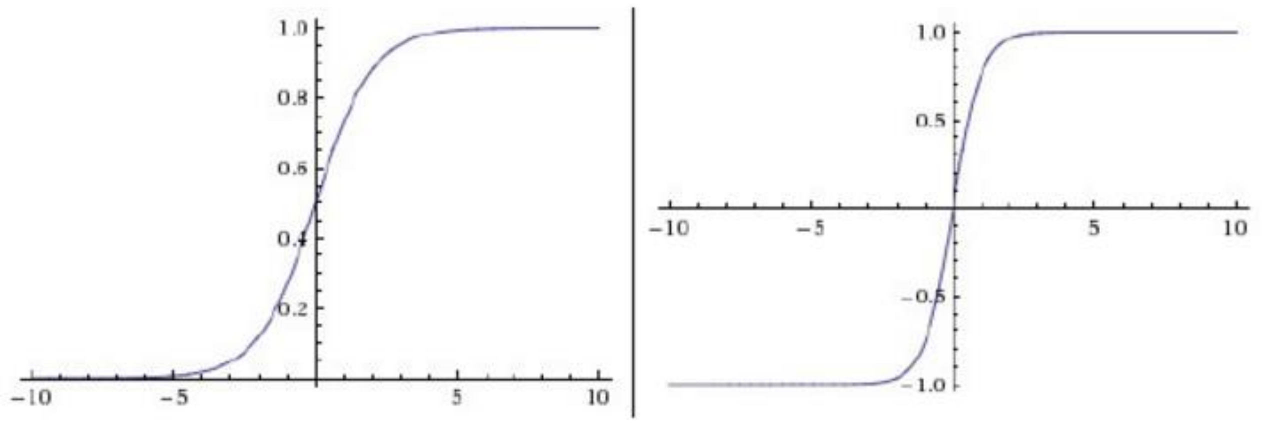
- The AlexNet was submitted to the ImageNet ILSVRC challenge in 2012 and significantly outperformed the second runner-up (top 5 error of 16% compared to runner-up with 26% error).

- deeper, bigger, and featured Convolutional Layers

2.3 LeNet to AlexNet

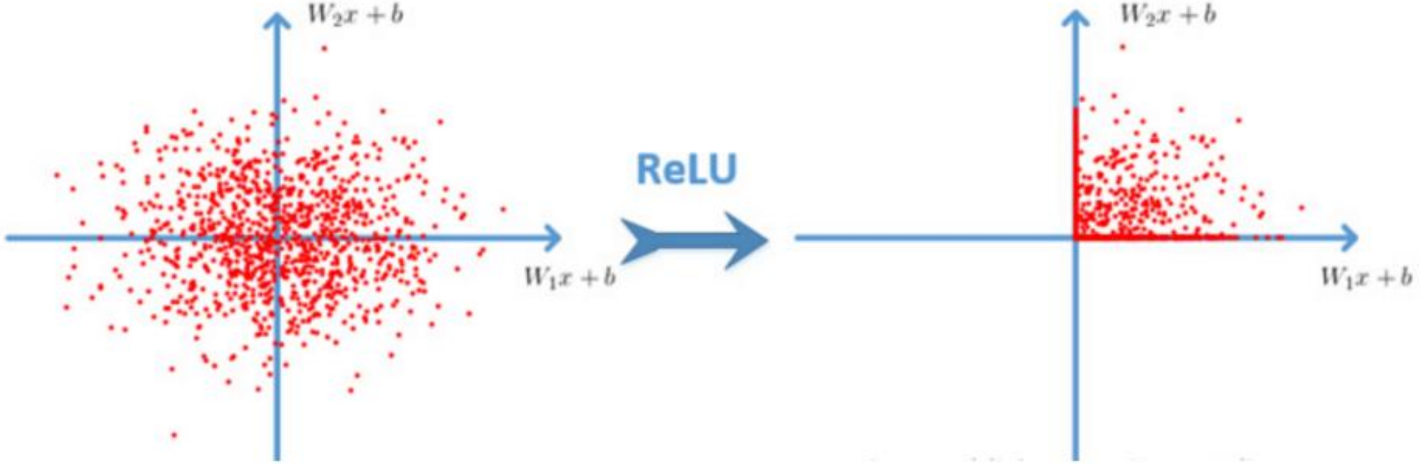
- Big data
- GPU
- data augmentation
- Relu and dropout

Sigmoid



RELU

f(X)=MAX(X,0)



SGD

- SGD (stochastic gradient descent)
- learning models are often trained stochastically, i.e. using a method where the objective function changes at each iteration.
  - dataset is divided into sub-datasets (mini-batches)
  - in each epoch, using different sub-datasets at different iteration. different datasets, different objective functions
- motivation to use SGD:
  - training data may be too large to fit in memory
  - objective functions are typically non-convex (e.g. neural networks), SGD helps prevent the model fom settling into a local minimum. (local minimum of one objective function is not that of another)