Data Mining & Machine Learning

CS57300
Purdue University

March 29, 2018
Some Deep Learning Applications (Supervised Learning)

- Classification

- Denoising (regression)
Role of Structure

• Feedforward networks use image as input as a vector

• From image to vector… hard to account for spatial correlations in the vector representation (which pixels are next to each other?)

• Even harder to account for location and colors
Convolutional Neural Network (CNN)

Transforms large image patch into one output

Non-linear activation

Basic unit: single neuron of filter $G$

Image patch

Vectorize image $x$ vector representing image

Weights used by this neuron $W^{(G)}_{1,1}$

Output $\sigma(W_{i,j}^G x + b)$

$9x9$ filter $G$
Equation View

• Equation view of a 2x2 filter applied over a 3x4 image

Deep Learning, Ian Goodfellow and Yoshua Bengio and Aaron Courville, MIT Press
How to represent color images?
Color Image Convolutional Neural Network (CNN)

- A filter can be defined with one neuron over multiple channels
- For instance, one filter for colors Green and Red

Weights used by red: $W^{(R)}_{1,2}$
Weights used by green: $W^{(G)}_{1,1}$

+ bias
Forward Pass of an RGB filter

- Forward pass of a convolution
- Input: RGB image, 3 matrices
- Filter Wi: 3x3 pixels
  1 neuron
  3 weight vectors, one for each color
  1 bias
- Apply the neuron with stride 1 and padding 1

http://cs231n.github.io/convolutional-networks/
Forward Pass of an RGB filter

- Forward pass of a convolution
- Input: RGB image, 3 matrices
- Filter Wi: 3x3 pixels
  1 neuron
  3 weight vectors, one for each color
  1 bias
- Apply the neuron with stride 1 and padding 1

The final goal of the CNN is to learn weights (81) and biases (2) that minimize the loss

http://cs231n.github.io/convolutional-networks/
Also Possible to Have one Filter Per Color

- If the input has 3 channels (R,G,B), 3 separate k by k filter can be applied to each channel
- Output of convolving 1 feature is called a feature map
Pooling

- Max pooling is a way to get a single output out of a filter

Weights used by this neuron

$W^{(G)}_{1,1}$

$W^{(G)}_{1,1}$

output

Can also be average, min or any permutation-invariant operation
Convolutional Network (ConvNet)

- Non-Linearity: sigmoid, rectified linear units (ReLU)
- Pooling: max, average,…
- Training: Image labels

Original: Y. LeCun, Facebook, modified by Ribeiro
Learning parameters of the neuron in the filter

• Our goal is to learn the parameters of the neuron in the filter

• Algorithm to learn the CNN weights:
  1. Do a forward pass as described above
  2. Perform a backward pass to get the gradient of the weights
  3. Update all parameters
  4. GoTo 1
AlexNet [Krizhevsky, Sutskever, Hinton 2012]

- Method: large convolutional net
  - 650K neurons, 832M synapses, 60M parameters
  - Trained with backprop on NVIDIA GPU
  - Trained “with all the tricks Yann came up with in the last 20 years, plus dropout” (Hinton, NIPS 2012)
  - Rectification, contrast normalization,...

- Error rate: 15% (whenever correct class isn't in top 5) Previous state of the art: 25% error

- A revolution in computer vision

- Acquired by Google in Jan 2013
- Deployed in Google+ Photo Tagging in May 2013

Credit: Y. LeCun, Facebook
AlexNet Architecture

-Won the 2012 ImageNet LSVRC. 60 Million parameters, 832M Matrix multiplication + accumulation operations per forward pass

96 filters: each with $3 \times (11 \times 11) + 1$ parameters = 34,849

Credit: Y. LeCun, Facebook
Filters: Layer 1 (7x7) and Layer 2 (7x7)

– Layer 1: 3x96 filters, RGB->96 feature maps, 7x7 Filters, stride 2

– Layer 2: 96x256 filters, 7x7

Credit: Y. LeCun, Facebook
AlexNet

• Detailed View:

Full (simplified) AlexNet architecture:

[227x227x3] **INPUT**

[55x55x96] **CONV1**: 96 11x11 filters at stride 4, pad 0
[27x27x96] **MAX POOL1**: 3x3 filters at stride 2
[27x27x96] **NORM1**: Normalization layer

[27x27x256] **CONV2**: 256 5x5 filters at stride 1, pad 2
[13x13x256] **MAX POOL2**: 3x3 filters at stride 2
[13x13x256] **NORM2**: Normalization layer

[13x13x384] **CONV3**: 384 3x3 filters at stride 1, pad 1
[13x13x384] **CONV4**: 384 3x3 filters at stride 1, pad 1
[13x13x256] **CONV5**: 256 3x3 filters at stride 1, pad 1
[6x6x256] **MAX POOL3**: 3x3 filters at stride 2

[4096] **FC6**: 4096 neurons
[4096] **FC7**: 4096 neurons
[1000] **FC8**: 1000 neurons (class probability output)
Deep learning = learning hierarchical representations

It's **deep** if it has **more than one stage** of non-linear feature transformation

Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Credit: Y. LeCun, Facebook