Convolutional Neural Networks

Problems with traditional neural networks

- Previously, you have seen neural networks where every neuron in a layer is connected to every neuron in the next layer.
- These fully-connected layers can work well for layers with a small number of neurons, but as the size of the input and number of neurons increases, the number of weights becomes prohibitively expensive
- Fully connected neural networks would have too many weights if we were to try processing images with them
  - consider a 256 by 256 color image—there would be (256×256×3) = 196608 weights for *each* neuron in the first layer
  - Too many weights can also lead to overfitting training data

Convolutional Neural Networks (CNNs)

- Designed to process images—makes assumptions about image processing, and uses this to reduce weights
  - Features can be extracted locally: whether there is an edge at a location depends on the pixels around that location, not those on the other side of the image
  - The same weights used to extract a feature at some location are able to do so at any other location: there's

LeNet-5 (LeCun et al, 1998)
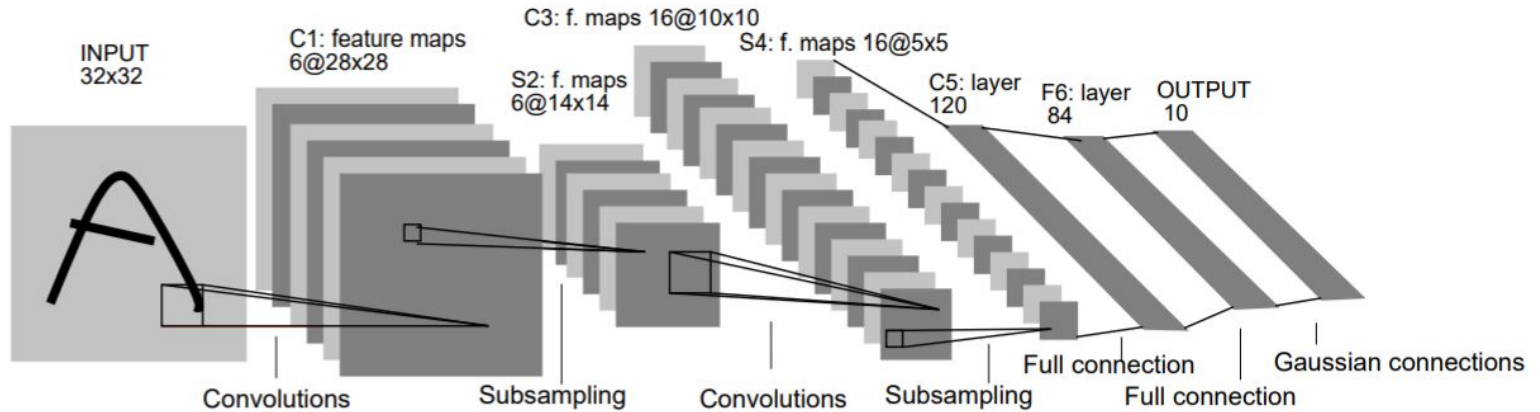
AlexNet (

ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky
University of Toronto
kriz@cs.utoronto.ca

Ilya Sutskever
University of Toronto
ilya@cs.utoronto.ca

Geoffrey E. Hinton
University of Toronto
hinton@cs.utoronto.ca

Abstract

We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers we employed a recently-developed regularization method called "dropout" that proved to be very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry.
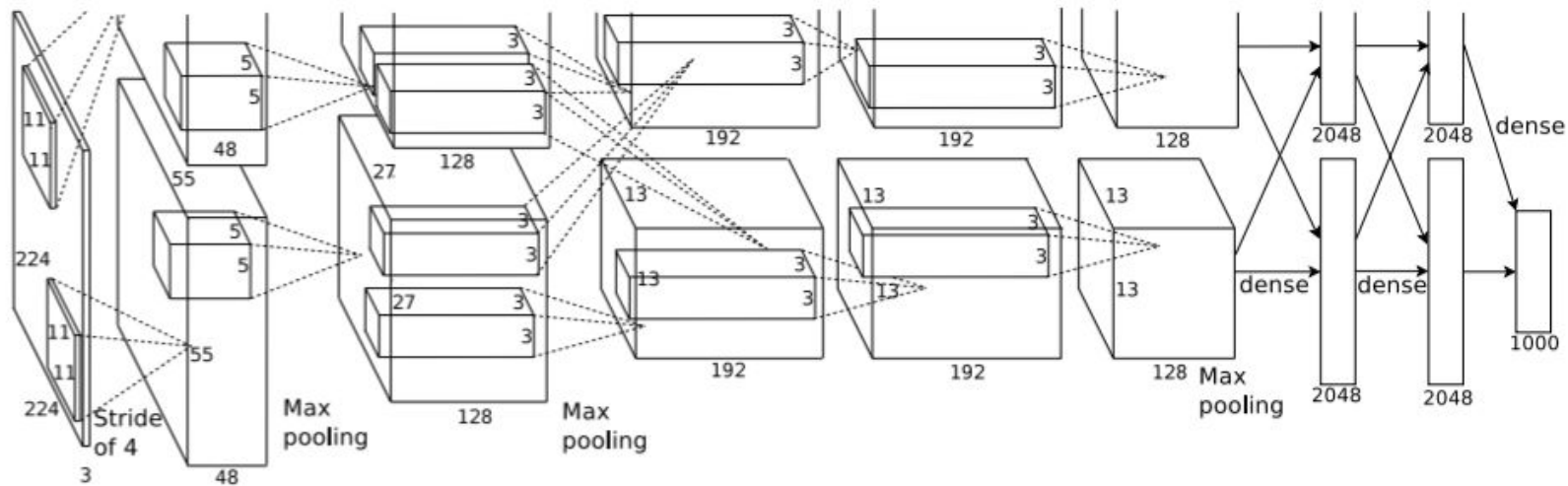
Key Points

- Usage of ReLu, a non-saturating function, as the activation function. (over tanh and sigmoid)
- Utilized two GPUs
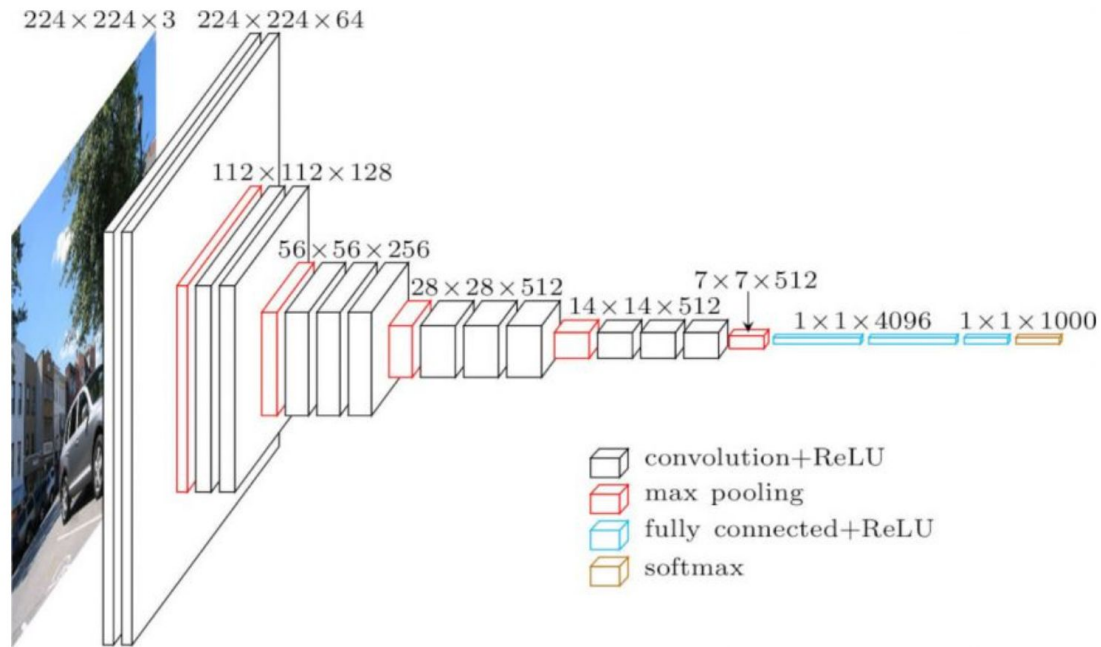- Data augmentation to reduce overfitting
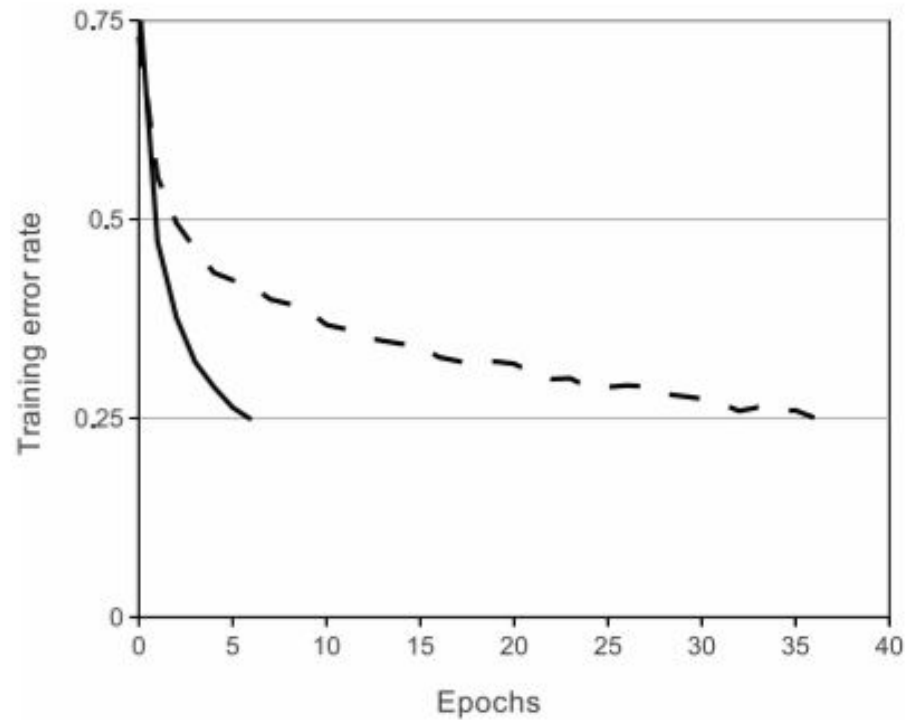
ice ReLu (Solid Line) reaches .25 training error rate approximately ~6x faster than Tanh(dotted line) does.

Data Augmentation

- Extracted 224x224 patches out of the 256x256 images, resulting in 2048 more training points. (Why our network has inputs of size 224x224)
- During testing, extracted the 4 corner 224x224 patches and the center patch, along with their horizontal reflections. (total 10 patches)

- Altered the intensity of RGB color channels, utilizing PCA to pertain the important

VGGN

$224 \times 224 \times 3$ $224 \times 224 \times 64$

$112 \times 112 \times 128$

$56 \times 56 \times 256$

$28 \times 28 \times 512$

$14 \times 14 \times 512$

$7 \times 7 \times 512$

$1 \times 1 \times 4096$ $1 \times 1 \times 1000$

convolution+ReLU

max pooling

fully connected+ReLU

softmax

VGG16 Architecture

Key Points

- Only uses 3x3 convolution filters

Why 3x3 Filters?

- It is the smallest size that still captures the notion of left, right, up, down, and middle.
- Instead of a single large 7x7 or 11x11 filter, they use multiple 3x3 filters.
- If we apply a 3x3 filter to a 7x7 image, we get a 5x5 matrix. Applying 2 more 3x3 layers results in a 1x1 matrix, which is equivalent to applying a single 7x7 filter.
- Using smaller filters results in more non-linear ReLu layers, which makes our network more discriminative.

5

5

2 layers of 3×3 filters already covered the 5×5 area

1st 3x3 conv. layer
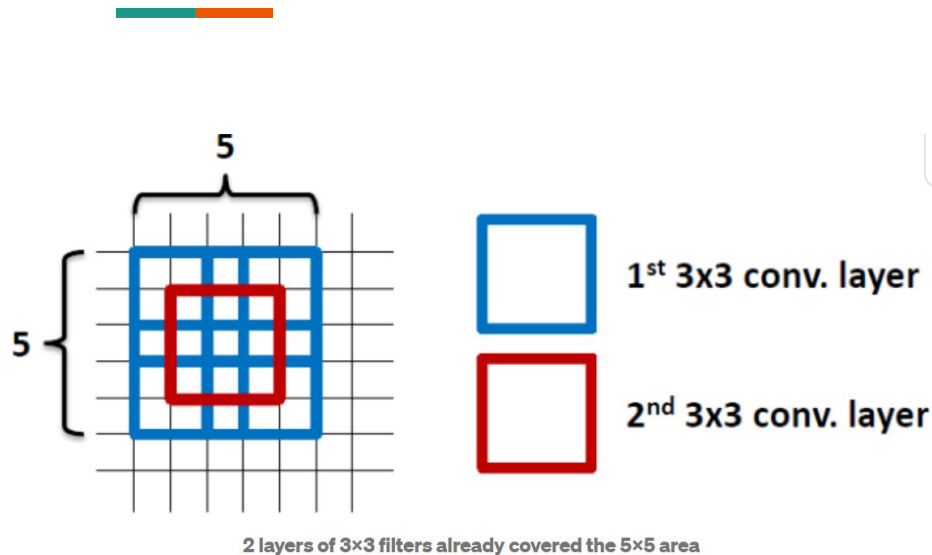
2nd 3x3 conv. layer

3 3x3 filters => 27 weights

1 7x7 filter => 49 weights

Using only 3x3 conv. layers requires 45% fewer weights.

2 3x3 filters => 18 weights

1 5x5 filter => 25 weights

Using only 3x3 layers requires 28% fewer weights

GoogleNet

Abstract

We propose a deep convolutional neural network architecture codenamed Inception that achieves the new state of the art for classification and detection in the ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14). The main hallmark of this architecture is the improved utilization of the computing resources inside the network. By a carefully crafted design, we increased the depth and width of the network while keeping the computational budget constant. To optimize quality, the architectural decisions were based on the Hebbian principle and the intuition of multi-scale processing. One particular incarnation used in our submission for ILSVRC14 is called GoogLeNet, a 22 layers deep network, the quality of which is assessed in the context of classification and detection.
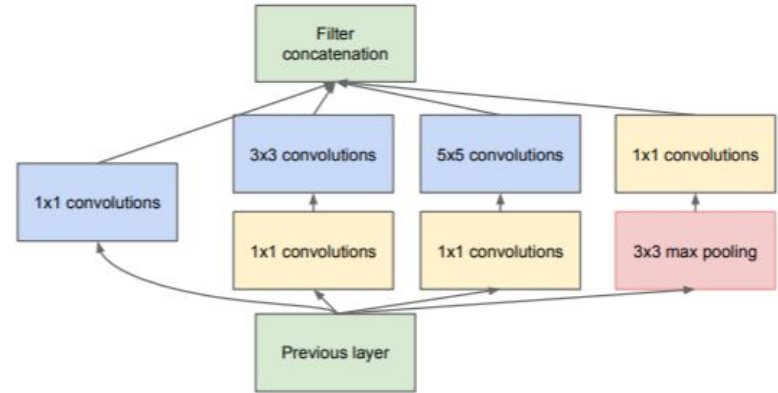
Key Points

- Inception Modules
- Average Pooling instead of FC layers, which eliminates a large amount of parameters.

Inception Modules
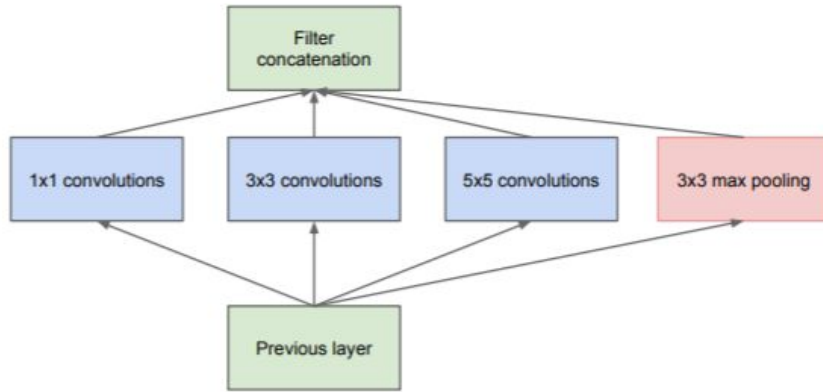
(a) Inception module, naïve version

(b) Inception module with dimensionality reduction

Figure 2: Inception module

ResNet (He et al, 2015)

Abstract

Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet dataset we evaluate residual nets with a depth of up to 152 layers—8× deeper than VGG nets [41] but still having lower complexity. An ensemble of these residual nets achieves 3.57% error on the ImageNet test set. This result won the 1st place on the ILSVRC 2015 classification task. We also present analysis on CIFAR-10 with 100 and 1000 layers.

The depth of representations is of central importance for many visual recognition tasks. Solely due to our extremely deep representations, we obtain a 28% relative improvement on the COCO object detection dataset. Deep residual nets are foundations of our submissions to ILSVRC & COCO 2015 competitions[1], where we also won the 1st places on the tasks of ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation.

Key Points

- "Shortcut" connections allow easier learning of the identity mapping
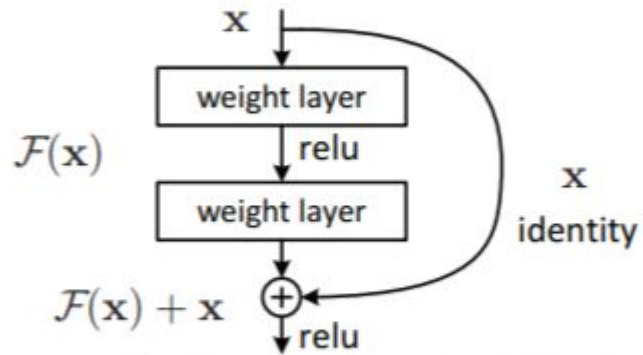- Speeds up training of deep networks

Figure 2. Residual learning: a building block.

If the desired underlying function we want to learn is H(x), the network learns the residual mapping, ie F(x):= H(x) - x. Thus, the output of this network is F(x) + x = H(x)
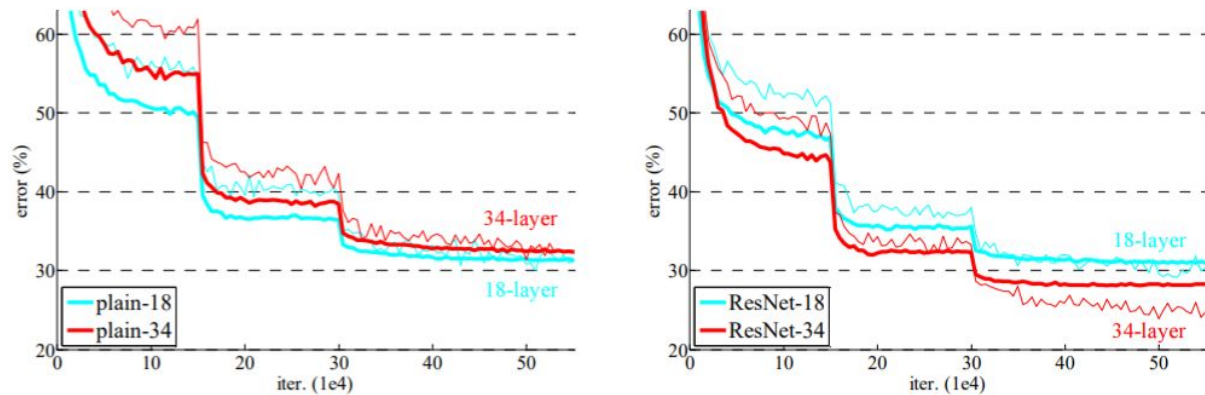
Figure 4. Training on **ImageNet**. Thin curves denote training error, and bold curves denote validation error of the center crops. Left: plain networks of 18 and 34 layers. Right: ResNets of 18 and 34 layers. In this plot, the residual networks have no extra parameter compared to their plain counterparts.

Allows us to build deeper networks that will still likely converge.

Adversarial Examples

- Szegedy et al. (2014) has found that most state of the art CNN's are susceptible to adversarial examples.