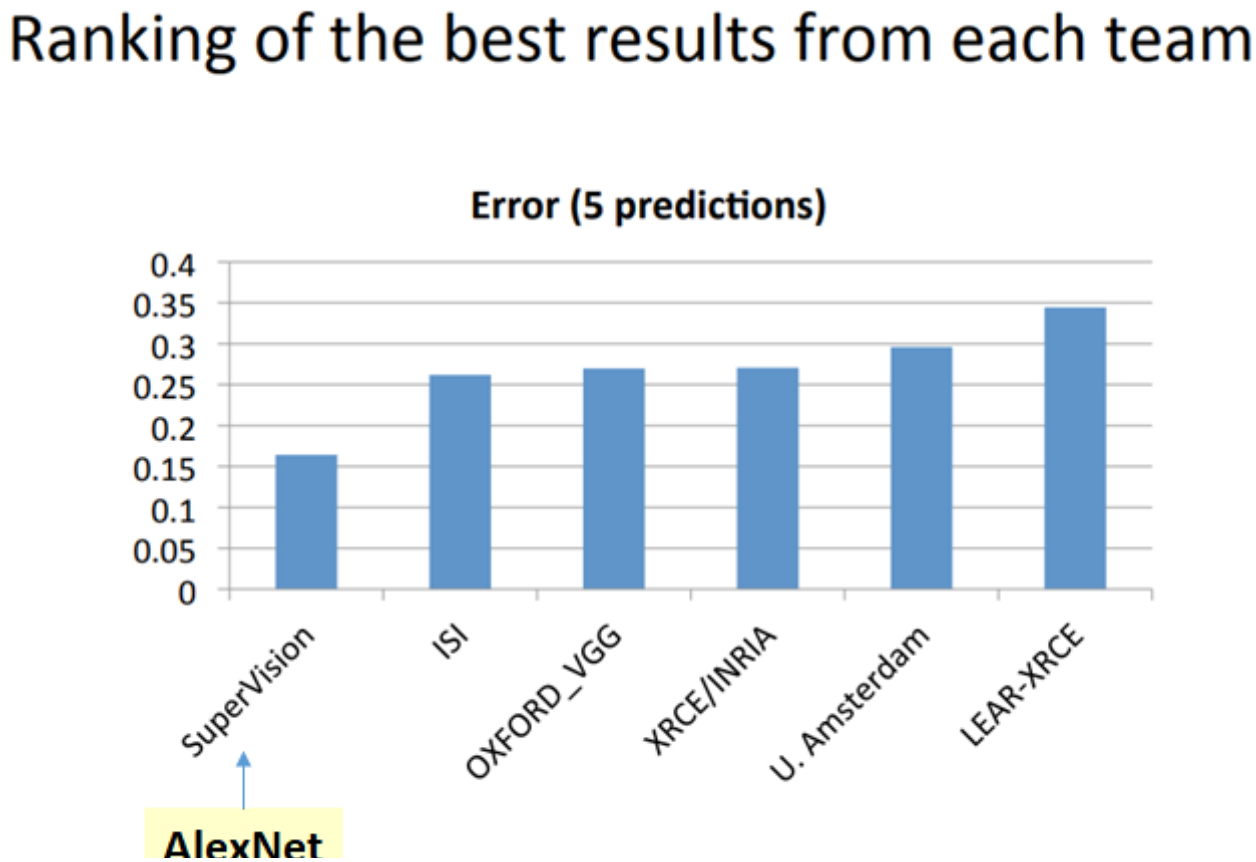
Review: AlexNet, CaffeNet — Winner of ILSVRC 2012 (Image Classification)

Sik-Ho Tsang
Aug 9, 2018 · 6 min read

In this story, **AlexNet** and **CaffeNet** are reviewed. AlexNet is the **winner of the ILSVRC (ImageNet Large Scale Visual Recognition Competition) 2012**, which is an image classification competition.

This is a **2012 NIPS** paper from Prof. Hinton's Group with **about 28000 citations when I was writing this story.** It has an **essential breakthrough in deep learning which substantially reduce the error rate in ILSVRC 2012** as the figure shown below. Thus, this is a must read paper!! (Sik-Ho Tsang @ Medium)

ImageNet, is a dataset of over 15 millions labeled high-resolution images with around 22,000 categories. ILSVRC uses a subset of ImageNet of around 1000 images in each of 1000 categories. In all, there are roughly 1.2 million training images, 50,000 validation images and 150,000 testing images.



AlexNet, the winner in ILSVRC 2012 image classification with remarkable lower error rate

A. For **AlexNet**, we will cover:

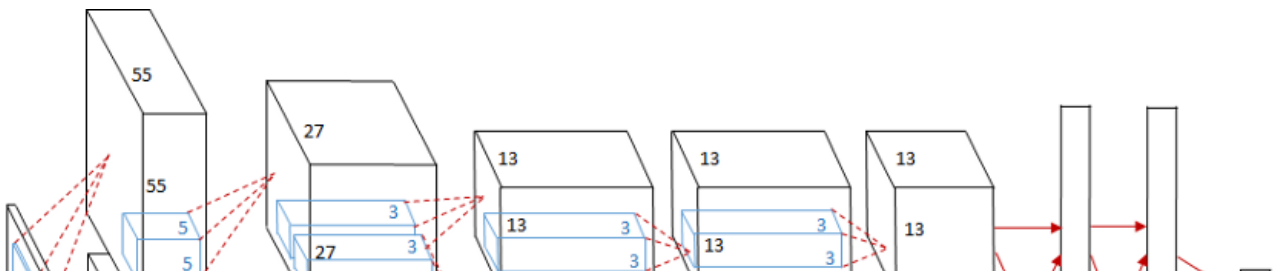
1. **Architecture**
2. **ReLU (Rectified Linear Unit)**
3. **Multiple GPUs**
4. **Local Response Normalization**
5. **Overlapping Pooling**
6. **Data Augmentation**
7. **Dropout**
8. **Other Details of Learning Parameters**
9. **Results**

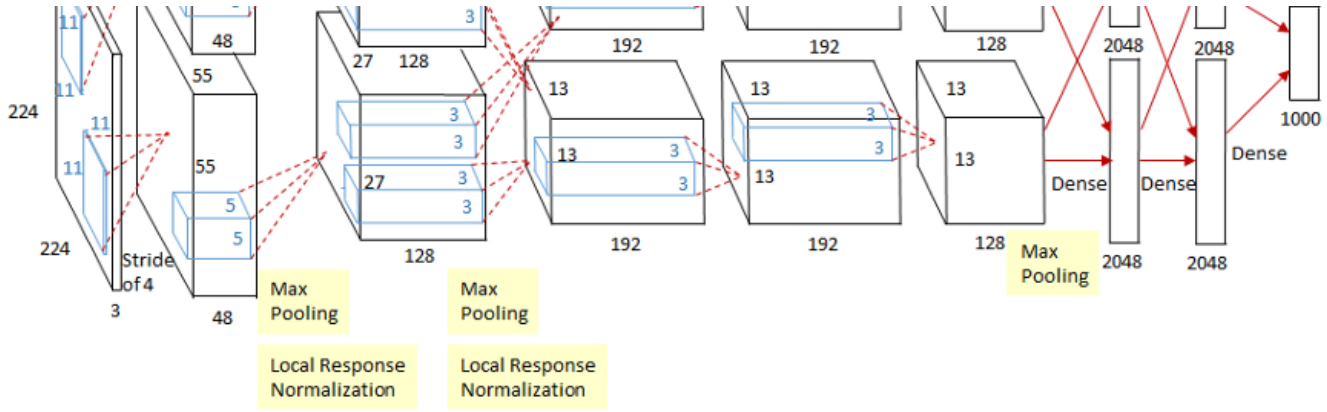
B. For **CaffeNet**, it is just a **single-GPU version of AlexNet**. Since normally, people would only have one GPU, CaffeNet is a single-GPU network to simulate AlexNet. We will cover this as well at the end of this story.

By going through each component, we can know the importance of each component. Some of them are not so useful by now. But they do inspire for invention of other networks.

A. AlexNet

1. Architecture





AlexNet

AlexNet contains **eight layers**:

Input: 224×224×3 input images

1th: Convolutional Layer: 96 kernels of size 11×11×3 (stride: 4, pad: 0)
55×55×96 feature maps
Then **3×3 Overlapping Max Pooling (stride: 2)**
27×27×96 feature maps
Then **Local Response Normalization**
27×27×96 feature maps

2nd: Convolutional Layer: 256 kernels of size 5×5×48 (stride: 1, pad: 2)
27×27×256 feature maps
Then **3×3 Overlapping Max Pooling (stride: 2)**
13×13×256 feature maps
Then **Local Response Normalization**
13×13×256 feature maps

3rd: Convolutional Layer: 384 kernels of size 3×3×256 (stride: 1, pad: 1)
13×13×384 feature maps

4th: Convolutional Layer: 384 kernels of size 3×3×192 (stride: 1, pad: 1)
13×13×384 feature maps

5th: Convolutional Layer: 256 kernels of size 3×3×192
(stride: 1, pad: 1)
13×13×256 feature maps
Then **3×3 Overlapping Max Pooling (stride: 2)**
6×6×256 feature maps

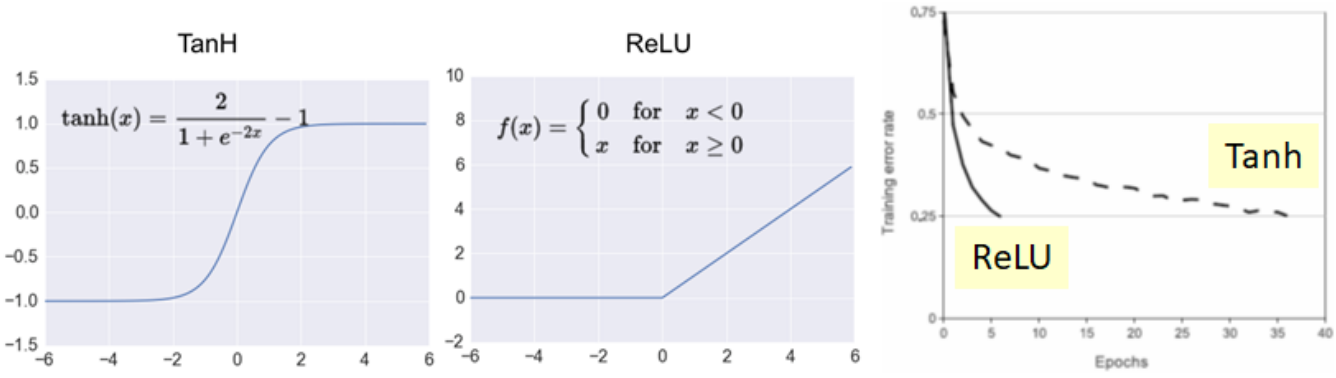
6th: Fully Connected (Dense) Layer of
4096 neurons

7th: Fully Connected (Dense) Layer of
4096 neurons

8th: Fully Connected (Dense) Layer of
Output: 1000 neurons (since there are 1000 classes)
Softmax is used for calculating the loss.

In total, there are 60 million parameters need to be trained !!!

2. ReLU



Activation function

Before Alexnet, Tanh was used. **ReLU is introduced in AlexNet.**
And **ReLU is six times faster than Tanh** to reach 25% training error rate.

3. Multiple GPUs

At that moment, NVIDIA GTX 580 GPU is used which only got 3GB of memory. Thus, we can see in the architecture that they split into two paths and use 2 GPUs for convolutions. Inter-communications are only occurred at one specific convolutional layer.

Thus, using 2 GPUs, is due to memory problem, NOT for speeding up the training process.

With the whole network **compared with a net with only half of kernels** (only one path), **Top-1 and top-5 error rates are reduced by 1.7% and 1.2% respectively.**

4. Local Response Normalization

$$b^i_{x,y} = a^i_{x,y} / \left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a^j_{x,y})^2 \right)^{\beta}$$

a: activity of a neuron
i: i-th kernal
N: total number of kernel
k, n, α, β: hyper-parameters

Local Response Normalization

Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β
Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma \widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.

Batch Normalization

Normalization

In AlexNet, local response normalization is used. It is different from the batch normalization as we can see in the equations. Normalization helps to speed up the convergence. Nowadays, batch normalization is used instead of using local response normalization.

With local response normalization, Top-1 and top-5 error rates are reduced by 1.4% and 1.2% respectively.

5. Overlapping Pooling

Overlapping Pooling is the pooling with stride smaller than the kernel size while Non-Overlapping Pooling is the pooling with stride equal to or larger than the kernel size.

With overlapping pooling, Top-1 and top-5 error rates are reduced by 0.4% and 0.3% respectively.

6. Data Augmentation

Two forms of data augmentation.

First: Image translation and horizontal reflection (mirroring)

A random 224×224 is extracted from one 256×256 image plus horizontal reflection. The size of training set is increased by a factor of 2048. This can be calculated as follows:

By image translation: $(256-224)^2=32^2=1024$

By horizontal reflection: $1024 \times 2 = 2048$

At the test time, four corner patches plus the centre patch as well as their corresponding horizontal reflections (10 patches in total), are used for prediction, and get the average of all results to obtain the final classification result.

Second: Altering the intensity

PCA is perform on the training set. For each training image, add the quantity:

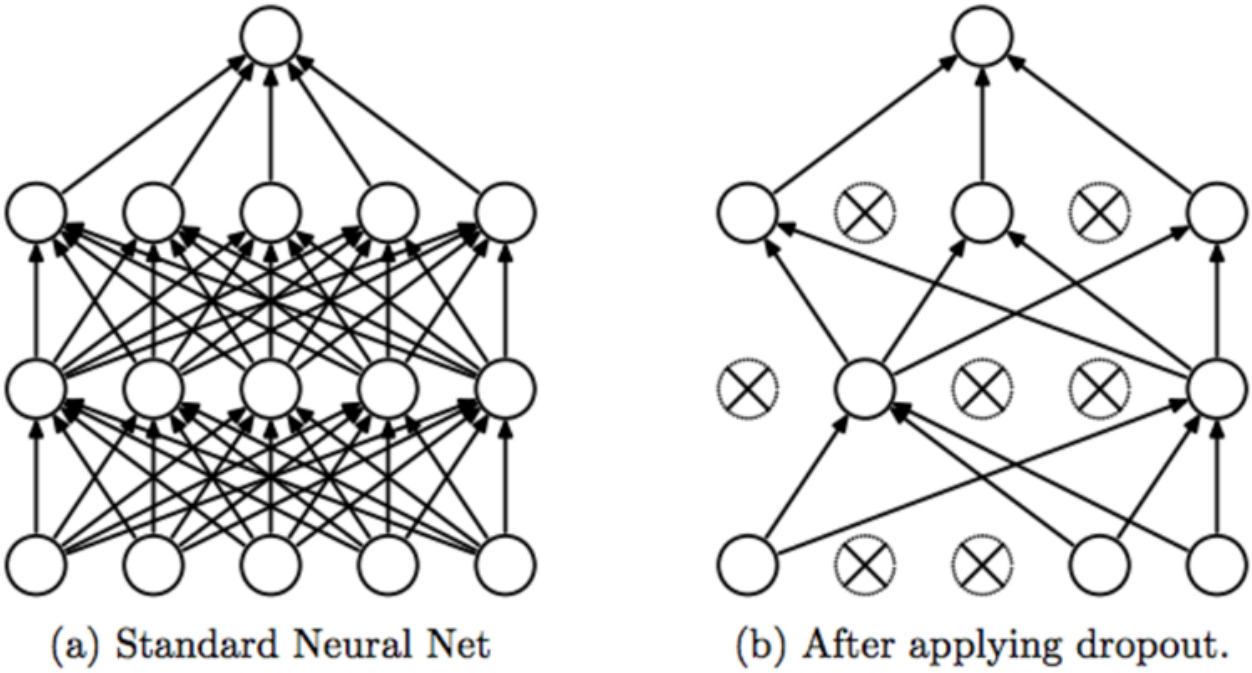
$$[\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3][\alpha_1\lambda_1, \alpha_2\lambda_2, \alpha_3\lambda_3]^T$$

Quantity of intensity altered

where pi and λi are ith eigenvector and eigenvalue of the 3×3 covariance matrix of RGB pixel values, respectively, and αi is the random variable with mean 0 and standard variation 0.1.

By increasing the size of training set with data augmentation, Top-1 error rate is reduced by over 1%.

7. Dropout



Dropout

With the layer that using dropout, during training, each neuron has a probability not to contribute to feed forward pass and participate in backpropagation. Thus, each neuron can have a larger chance to be trained, and not to depend so much for some very “strong” neuron.

During test time, there will be no dropout.

In AlexNet, probability of 0.5 is used at the first two fully-connected layers. Dropout is a kind of regularization technique to reduce the overfitting.

8. Other Details of Learning Parameters

Batch size: 128

Momentum v: 0.9

Weight Decay: 0.0005

Learning rate ϵ: 0.01, reduced by 10 manually when validation error rate stopped improving, and reduced by 3 times.

$$v_{i+1} := 0.9 \cdot v_i - 0.0005 \cdot \epsilon \cdot w_i - \epsilon \cdot \left\langle \frac{\partial L}{\partial w}\Big|_{w_i} \right\rangle_{D_i}$$

$$w_{i+1} := w_i + v_{i+1}$$

The update of momentum v and weight w

Training set of 1.2 million images.

Network is trained for roughly 90 cycles.

Five to six days on two NVIDIA GTX 580 3GB GPUs.

9. Results

Model	Top-1	Top-5
Sparse coding [2]	*47.1%*	*28.2%*
SIFT + FVs [24]	*45.7%*	*25.7%*
CNN	**37.5%**	**17.0%**

ILSVRC 2010

Error Rate in ILSVRC 2010

For ILSVRC 2010, AlexNet got the Top-1 and top-5 error rates of 37.5% and 17.0% respectively, which outperforms other approaches.

Without averaging 10 predictions over ten patches by data augmentation, **AlexNet only got the Top-1 and top-5 error rates of 39.0% and 18.3% respectively.**

Model	Top-1 (val)	Top-5 (val)	Top-5 (test)
SIFT + FVs [7]	—	—	*26.2%*

1 CNN	40.7%	18.2%	—
5 CNNs	38.1%	16.4%	**16.4%**
1 CNN*	39.0%	16.6%	—
7 CNNs*	36.7%	15.4%	**15.3%**

ILSVRC 2012

Error Rate in ILSVRC 2012

By **1 AlexNet (1 CNN)**, the validation error rate is **18.2%**.

By **Averaging the prediction from 5 AlexNet (5 CNNs)**, the error rate is reduced to **16.4%**. This is **a kind of boosting technique** already used in LeNet for digit classification.

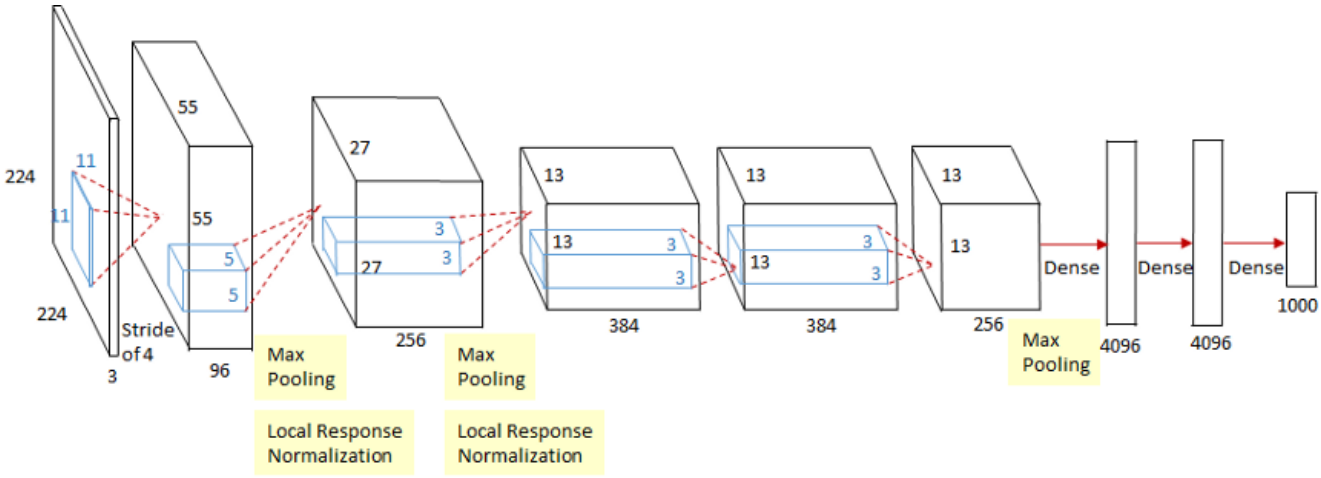
By **adding one more convolutional layer to AlexNet (1 CNN*)**, the validation error rate is reduced to **16.6%**.

By **Averaging the prediction from 2 modfiied AlexNet and 5 original AlexNet (7 CNNs*)**, the validation error rate is reduced to **15.4%**.

Some Top-5 results by AlexNet

B. CaffeNet

CaffeNet is a 1-GPU version of AlexNet. The architecture is:



CaffeNet

We can see that the 2 paths in AlexNet are combined to become one path.

It is noted that for early version of CaffeNet, the order of pooling and normalization layers is reversed, this is by accident. But in the current version of CaffeNet provided by Caffe, it has already provided the Caffenet with the correct order of pooling and normalization layers.

By investigating each component one by one, we can know the effectiveness of each component. :)

If interested, there is also a tutorial about CaffeNet quick setup using Nvidia-Docker and Caffe [3].

References

1. [2012 NIPS] [AlexNet]
ImageNet Classification with Deep Convolutional Neural Networks

2. [2014 ACM MM] [CaffeNet]
   Caffe: Convolutional Architecture for Fast Feature Embedding

3. VERY QUICK SETUP of CaffeNet (AlexNet) for Image Classification Using Nvidia-Docker 2.0 + CUDA + CuDNN + Jupyter Notebook + Caffe

4. ILSVRC
   ImageNet Large Scale Visual Recognition Competition