# AlexNet – 2012

Pablo Ruiz – Harvard University – October 2018

## What are the innovations?

### CNN to be the best model for image classification

Very daring from my point of view, authors affirm that:

> *Convolutional neural networks make strong and mostly correct assumptions about the nature of the images, namely stationarity of statistics and locality of pixel dependencies.*

Based on that assumption and the need of big models to learn from millions of images, AlexNet [1] reuse the convolutional neural network model introduced by LeCun in 1998 [2] to outperform state-of-art models on image classification by increasing the size.

### Rectified Linear Unit

They make use a new activation function introduced by Hinton and Nair in 2010 [3] called ReLU. The reason to use this **non-saturating nonlinear function** is to speed up the training. Figure 1 clearly shows this decrease in training time by replacing all the *tanh* activations by ReLU.

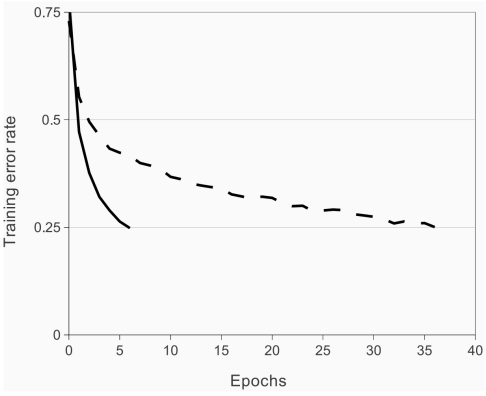


*Figure 1. Speed of training using tanh vs ReLU*

## Dropout

Dropout was introduced by Hinton and Srivastava also in 2012 [4] as a powerful regularization technique.

---

*Dropout randomly (with an assigned probability to each neuron) drops a neuron from the overall network structure, leading to a smaller network*

---

Figure 2 show this phenomenon for 1 single pass. This pass could be 1 single image, or more frequently a mini-batch of images, since neural networks are mostly trained using stochastic gradient descent and updating the weights after each batch of images.

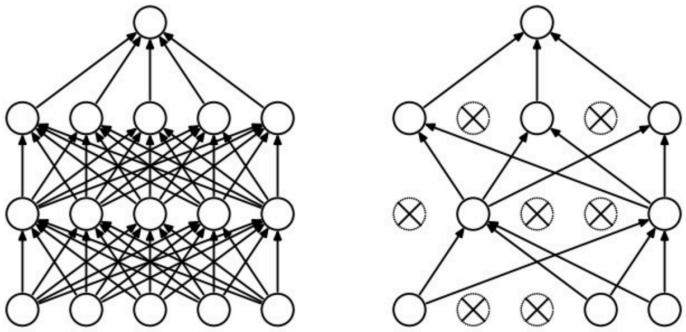

*Figure 2. Dropout scheme*

Dropout is considered an inherent way to build ensembles[1]. It is precisely an ensemble build using bagging as the data distribution technique and having each member of the ensemble a different architecture (with duplication) instead of ensembles where the exact same model is trained on different subset of data.

Figure 3 shows this behavior with a simplified network as the base network. All the 16 possible networks that could be constructed after the 4 neuron base networks gives perspective about the inherent ensemble property mentioned.

Note now how each mini-batch of images is going to go through a different network, and when their outputs are averaged to compute the gradient, it mimics when outputs of different models in an ensemble are averaged. In my opinion, I think dropout is in fact more ensemble-aware, since the group-decision are being considered in the backpropagation of the error, whereas classical ensembles are trained independently.

Another important fact is that **dropout reduces co-adaptation of neurons**, since now one neuron cannot rely on the presence on another, because in the next mini-batch could have been dropped out.

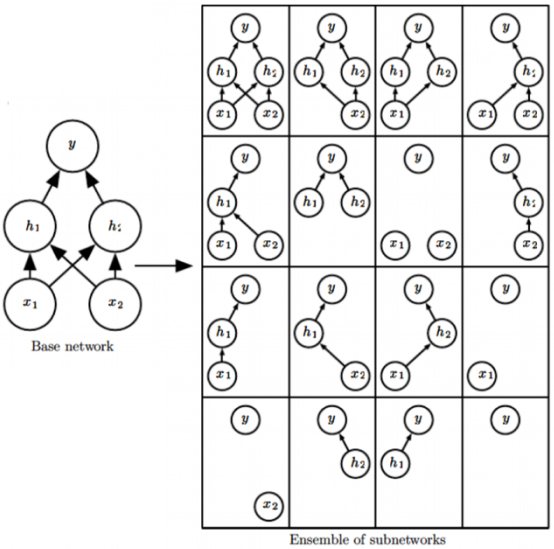


*Figure 3. Dropout as bagging*

[1] I am currently working on a post on ensemble learning: http://pabloruizruiz10.com/aiblog/el/el.html

## Multi-GPU Implementation

The size of a network is limited by the memory available on the GPU(s) and the training time aloud. The 1.2 images of ImageNet dataset used in this paper is enough data to train a model as big as not being able to fit in a single GPU – note that at that time hardware options were less. They used GTX 580 with 3GB of memory.

So, authors came up with a solution to this, spreading the model into 2 GPUs as a new level of parallelization, named cross-GPU parallelization.

---

*GPUs allows to read and write to one another's memory directly, without going through host machine memory.*

---

The result is 2-GPU net takes slightly less time to train that the 1-GPU net. However, this comparison is biased favoring the 1-GPU since it has more parameters than half the size of the 2-GPU. Furthermore, there is an increase on the accuracy with the 2-GPU (≈ 1.5 %).

# Structure

### Note

I need to make a confession at this point. I am not being able to follow the paper on how they construct the network. Basically, if I follow the indications it does not match with the already so famous figure shown in where the architecture is shown. I opened two issues on Stack Overflow here and here to see if the community or you guys can help me again ! : )

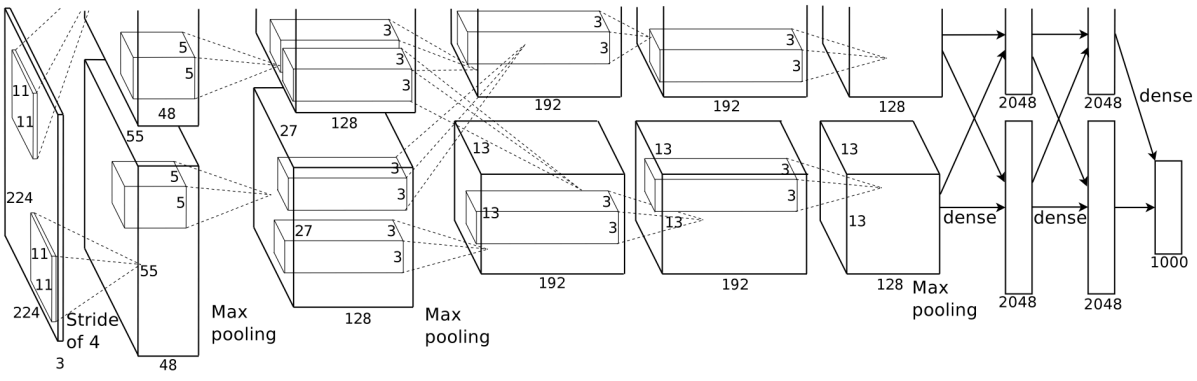


*Figure 4. Architecture of the AlexNet efficiently split into 2 GPU (up and down)s.*

However, there is a next publication by the same author Alex Krizhevsky [5] where uses a completely new structure that matches the PyTorch official implementation. Therefore, we will follow this optimized version here.

### Optimized Structure

Ongoing…

# Results

First let's introduce how the performance of the model is measured.

ILSVRC-2012 competition was where these models were submitted. That competition requires the models to train on ImageNet dataset (1.2 million images and 1000 classes) and calculate the validation and test classification accuracies at 2 levels.

- Top1

Top 1 means that an example only counts as well classified if the network after the softmax layer assigns the biggest probability to the correct label. This is the classic understanding of a well-classified example.

- Top5

Top 5 means that an example is well-classified if the correct labels is between the 5 biggest probabilities output by the network after the softmax. This is an attempt to better say whether the network did a good although not enough work classifying each particular example, or it was completely misclassified.

## Last best results

The best results of the ILSVRC-2010 where achieved by:

- Averaging predictions of six space-coding models trained on different features
- Averaging the predictions of 2 classifiers trained on Fisher Vectors (FV).

For that edition of 2010, these networks would have classified first giving the next score:

*Table 1. Results for ILSVRC-2010 competition (TEST SET)*

| Model | Top-1 | Top-5 |
|---|---|---|
| Sparse coding | 47.1 % | 28.2 % |
| SIFT + FV | 45.7 % | 25.7 % |
| CNN | 37.5 % | 17.0 % |

For the edition of 2012, the results where the following:

*Table 2. Results for ILSVRC-2012 competition (VALIDATION SET)*

| Model | Top-1 | Top-5 |
|---|---|---|
| 1 CNN | 40.7 % | 18.2 % |
| 5 CNNs | 38.1 % | 16.4 % |
| 7 CNNs* | 37.7 % | 15.4 % |

Where 5 CNNs simply means averaging the output of 5 same models architectures but after all the randomness of the training (initialization, batches, dropout, stochastic gradient descent…) and where the * means that 2 of those CNNs were pre-trained on the entire Fall 2011 release with the previous 5 CNNs.

# Bibliography


[1] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," 2012.

[2] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," 1998.

[3] N. Vinod and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," 2010.

[4] G. E. Hinton, N. Srivastava and A. Krizhevsky, "Improving neural networks by preventing co-adaptation of feature detectors," 2012.

[5] A. Krizhevsky, One weird trick for parallelizing convolutional neural networks, 2014.

[6] K. He, X. Zhang, S. Ren and J. Sun, "Deep Resifual Learning for Image Recognition," in *CVPR*, 2016.