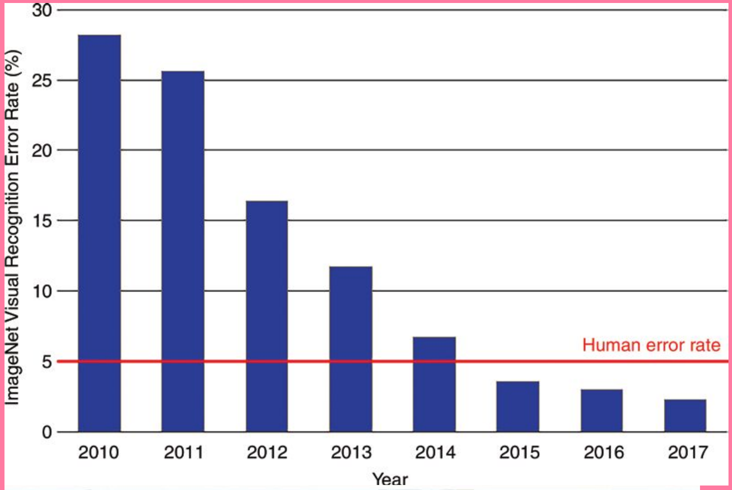
ImageNet Classification with Deep Convolutional Neural Networks

Authors: Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton

Lit Review by:
Ella Yan,
Nora Shao

Context/Overview

- ImageNet Large Scale Visual Recognition Challenge 2012
- Perceptron developed 1957, backpropagation developed 1980
- Previous winners worked with feature detection and landmarks (SIFT, FVs)
- AlexNet showed that machine learning could be used on large scale problems
- Revolutionized machine learning



Model Architecture

Non-linear ReLU Activation

To speed up training and
avoid saturation

Multiple GPUs

Trained on two separate GTX
580 GPUs (3GB memory)

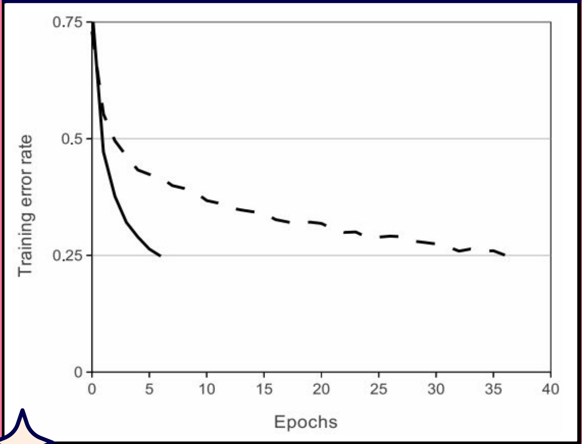
Local Response Normalization

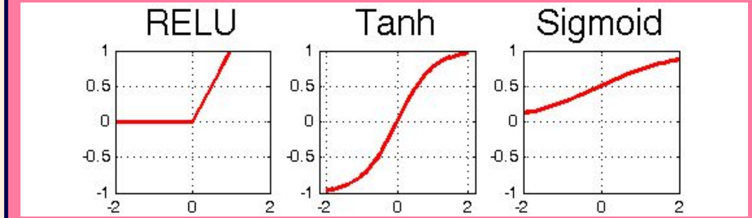
Encourage competition
between neurons

Overlapping Pool

To help with overfitting
and reduce computations

Using ReLU Activation





Nonlinearity

- ReLU was 25% faster than tanh (CIFAR-10)
- Computationally simple max(0,x)
- Avoids issue with saturation
- Could lead to dead neurons… (leaky ReLU would solve this)

Local Response Normalization

Local brightness normalization scheme aids generalization

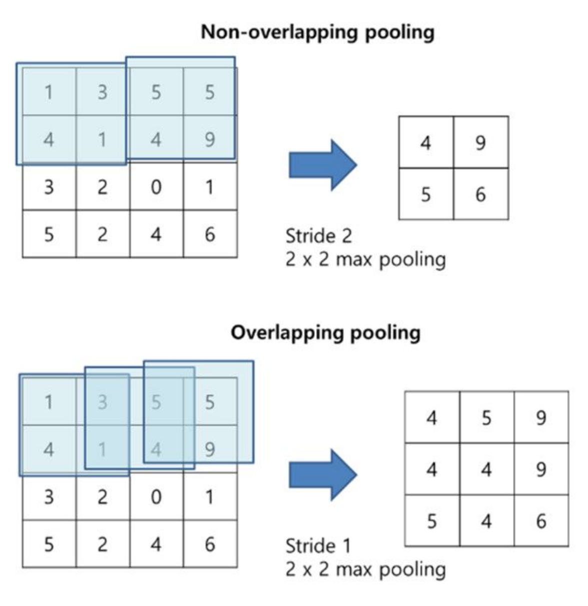
- Highlights low-activation neurons
- Increases sparsity
- Reduces overfitting
- Largely replaced by batch norm (etc) in the modern era

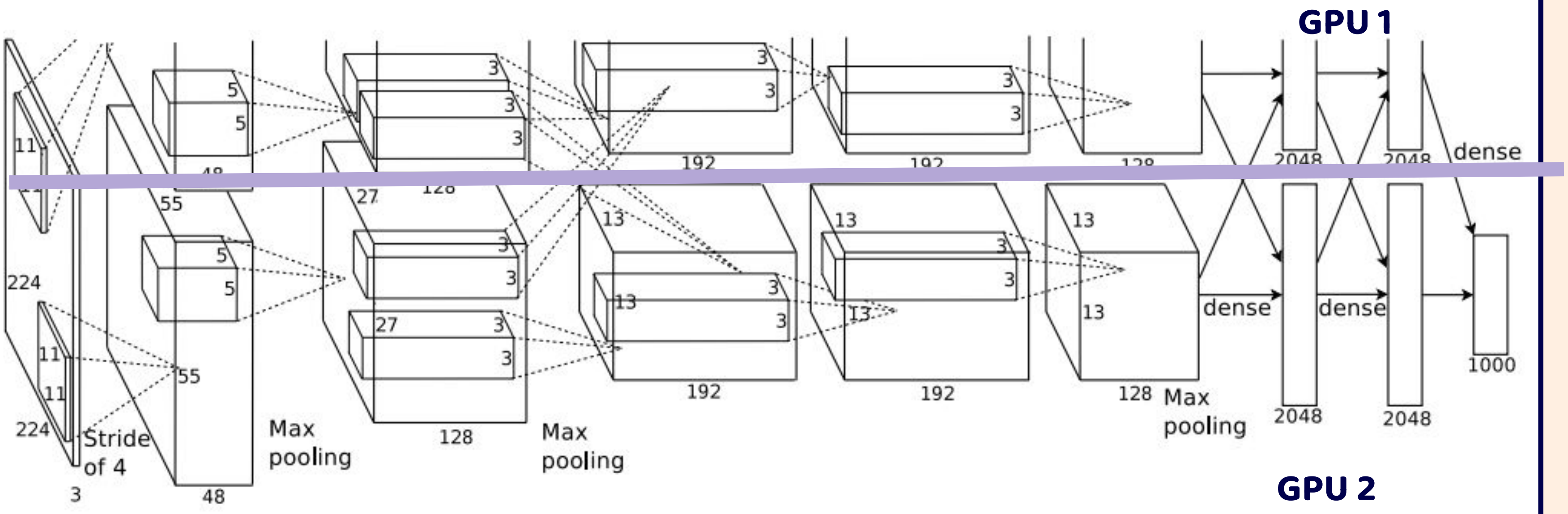
$$b^i_{x,y} = a^i_{x,y} / \left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a^j_{x,y})^2 \right)^\beta$$

Local Response Normalization

Overlapping Pooling

- Pooling reduces data after each convolution layers
- Overlapping pooling adds redundancy
- Observed to avoid overfitting by reducing dependency on pixel location
- Not very popular anymore (we love strided convolutions)





Overall Architecture

The separate GPUs only communicate on some layers to maintain speed
Model was extremely sensitive to depth
Question: what effects would this have on your model?

Learning

Stochastic gradient descent

Weight decay of 0.0005 important for learning

Reducing Overfitting

- Data Augmentation: Adding variety to dataset with cropped images, mirrored images and colour adjustments

- Dropout: Zeroing output of random neurons to make them less dependent on their neighbours

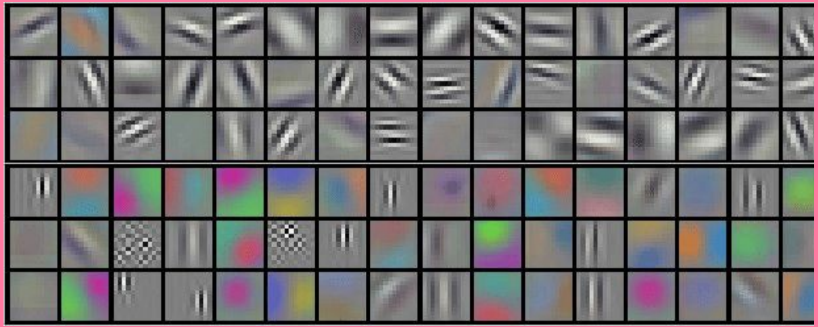
Results

Kernels

- Various edge detectors in the top half
- Checker pattern detectors on the bottom left
- Colour blob detection as well

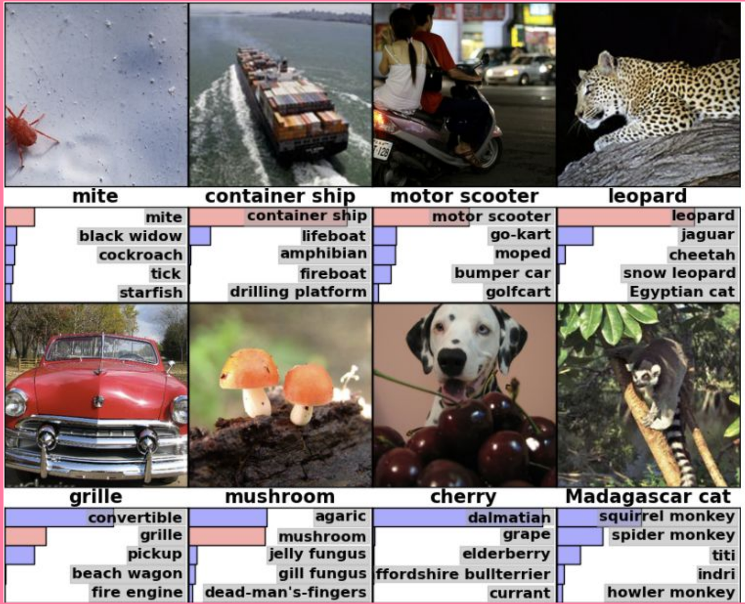
Model	Top-1 (val)	Top-5 (val)	Top-5 (test)
SIFT + FVs [7]	—	—	*26.2%*
1 CNN	40.7%	18.2%	—
5 CNNs	38.1%	16.4%	**16.4%**
1 CNN*	39.0%	16.6%	—
7 CNNs*	36.7%	15.4%	**15.3%**

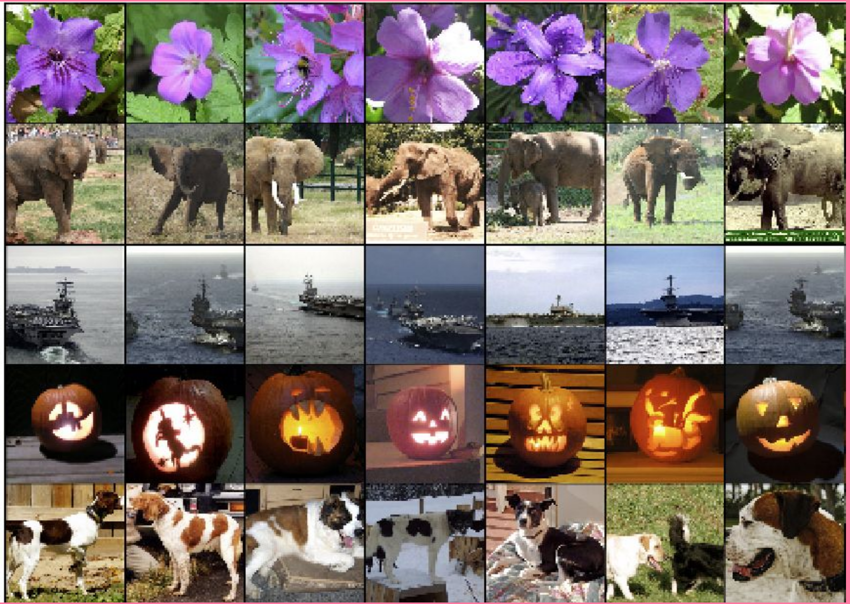
Table 2: Comparison of error rates on ILSVRC-2012 validation and test sets. In *italics* are best results achieved by others. Models with an asterisk* were "pre-trained" to classify the entire ImageNet 2011 Fall release. See Section 6 for details.



Kernels learned in the first convolutional layer. Top are GPU1, bottom are GPU 2

Results





Left: Top 5 Guesses

Right: Nearest Neighbours in Feature Space

Final Thoughts

"All of our experiments suggest that our results can be improved simply by waiting for faster GPUs and bigger datasets to become available"

1) What kind of effects does having your GPUs only communicate on certain layers impose?
2) Why did GPU1 learn a lot about shapes and edges while GPU2 learned more about colours?
3) At what point does adding more layers/depth cause the model to perform worse?

slidesgo