AlexNet: ImageNet Classification with Deep Convolutional Neural Networks

Robert McCraith

Maynooth University

29^{th} November 2016

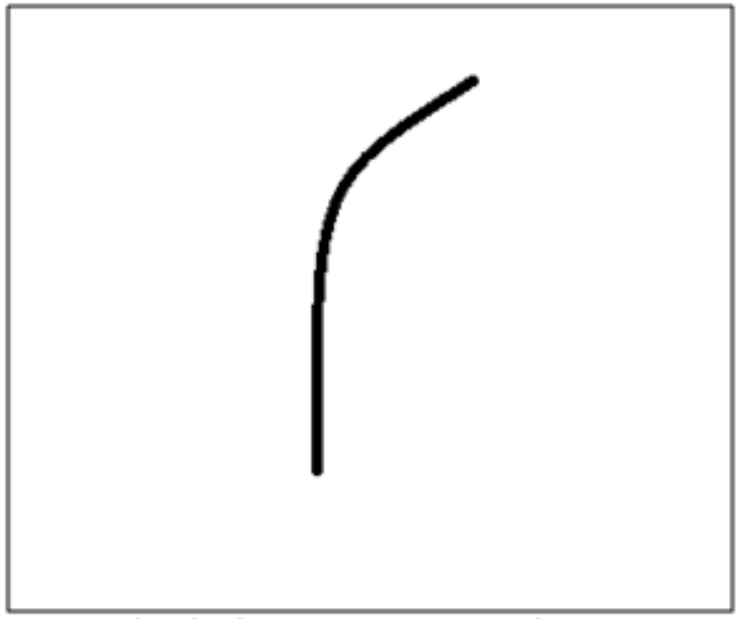
Convolutions

- Come from signal processing/computer vision (originally maths)
- Essentially can be thought of as a dot product of an underlying image and a filter
- When receptive field and filter match we get high values, when they differ we get low values
- Stanford cs231 Convolutions

Examples

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

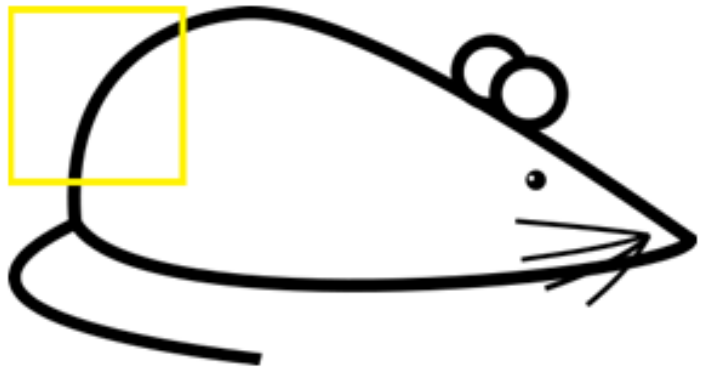


Visualization of a curve detector filter

[1] Source

Examples

Original image



Visualization of the filter on the image

Examples

Visualization of the receptive field

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

Pixel representation of the receptive field

*

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation = (50*30)+(50*30)+(50*30)+(20*30)+(50*30) = 6600 (A large number!)

Examples

Visualization of the filter on the image

0	0	0	0	0	0	0
0	40	0	0	0	0	0
40	0	40	0	0	0	0
40	20	0	0	0	0	0
0	50	0	0	0	0	0
0	0	50	0	0	0	0
25	25	0	50	0	0	0

Pixel representation of receptive field

*

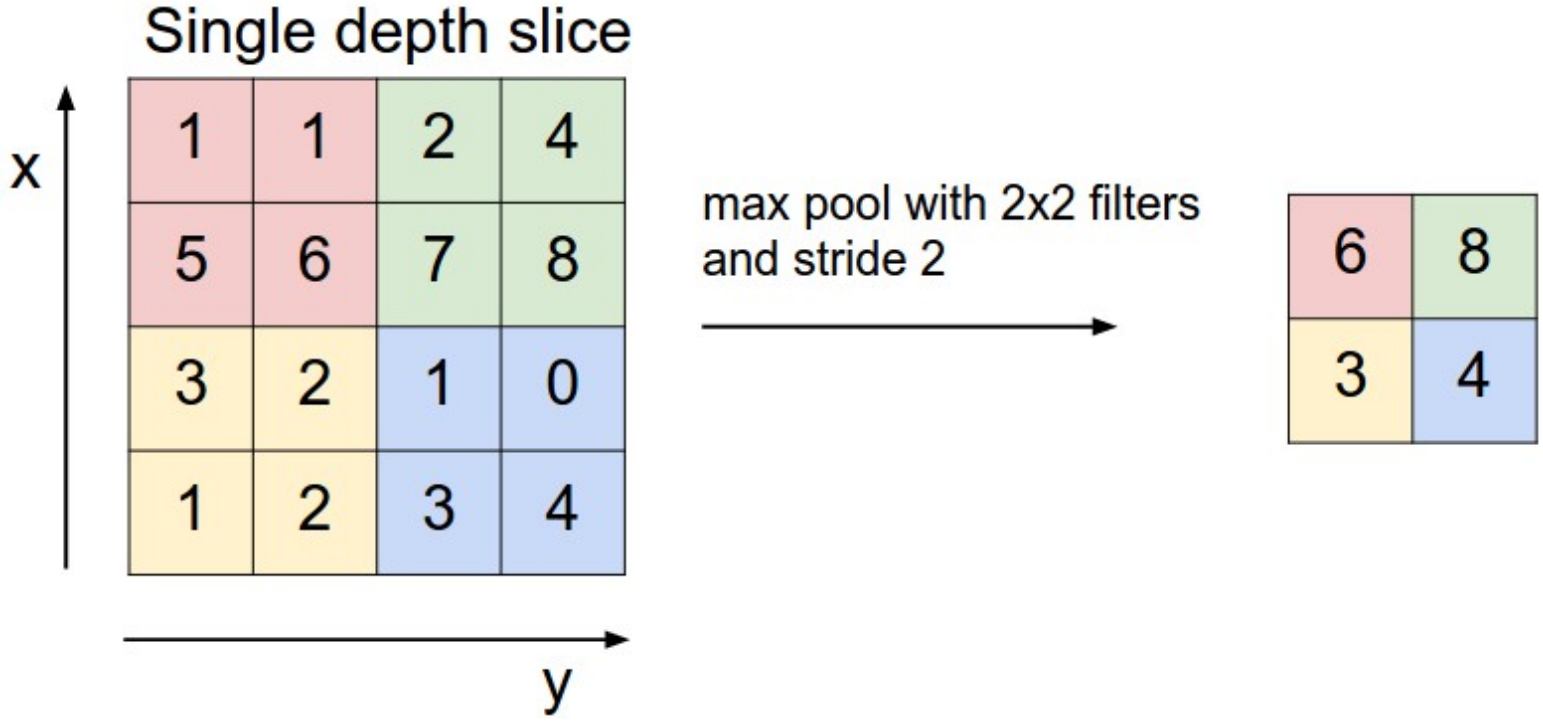
0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation = 0

Max Pooling

Down samples the volume spatially, independently in each depth slice of the input volume.



Response Normalisation

ReLU (Rectified Linear Unit)

$$f(x) = \max(0, x)$$

- Great speedup in training
- Simpler to compute than tanh, sigmoid
- Can "die" if large gradient flows through then weight might update to prevent this neuron activating causing it to remain 0. We can prevent this by setting a proper learning rate or using Noisy ReLUs, Leaky ReLUL or

Context

ImageNet

- 15 million labels images
- 22 thousand categories labeled by Amazon Mechanical Turk

ILSVRC

- Annual competition to classify images
- 1.2 million images and 1 thousand categories
- Top result and top 5 are considered

AlexNet

By. Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton

- Trained one of the largest (at the time) CNN's on the subset of ImageNet used for ILSVRC (Large Scale Recognition Challenge) and destroyed everyone else
- Wrote and published highly optimised GPU implementations of 2D convolution and other necessary operations
- New and unusual features that improve performance and reduce training time
- New techniques to prevent overfitting
- 5 convolution and 3 fully connected layers
- Size still limited by memory on GPUs at the time (they used 2x GTX 580's with 3GB VRAM each)

Architecture

ReLU Nonlinearity

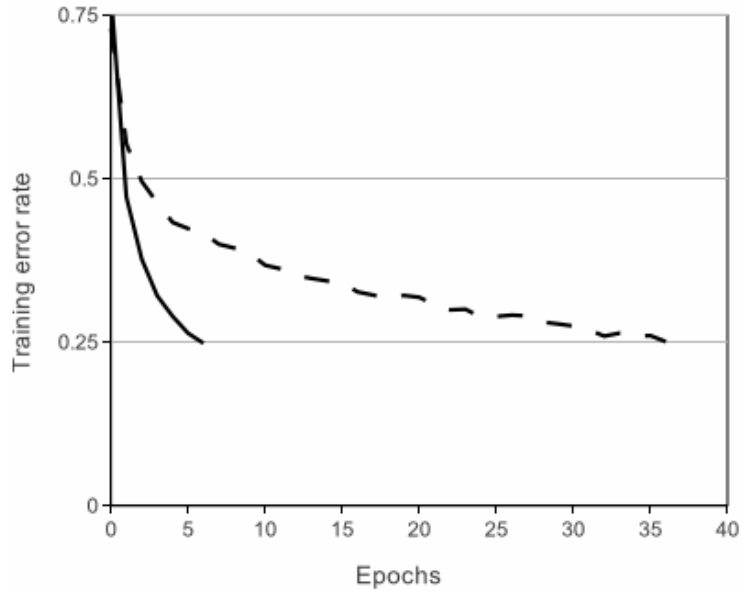
- Most people model neuron's output as

$$f(x) = tanh(x)$$

or

$$f(x) = (1 + e^{-x})^{-1}$$

- When training with gradient descent these saturating nonlinearities are much slower than non-saturating nonlinearity
- Deep CNNs with ReLU train several times faster than *tanh* networks



Mult GPU Training

- The GTX580 that they trained on only had 3GB of memory, turns out 1.2 million images $>$ 3GB
- Spread the network across two GPUs as current GPUs are well suited to cross GPU parallelisation as they can read/write to each others memory without going through host
- Put half the kernels on each GPU, GPUs only communicate in certain layers

---

(how to get the University to pay for your sweet gaming rig)

Local Response Normalisation

- ReLUs do not require input normalisation to prevent saturation, if at least one training examples gives positive input to ReLU learning happens
- Applying local normalisation aids generalisation
- ReLU neurons have unbounded activations, we need to normalise this. We want to detect high frequency features with large response
- Normalising around the neighbourhood of excited neuron makes it more sensitive compared to neighbours
- Normalisation also dampens responses that are uniformly large in a given neighbourhood

Overlapping Pooling

- Pooling Layers summarise outputs of neighbouring pixels in the same kernel map
- Traditionally the neighbourhoods didn't overlap
- Overlapping pooling results in greater accuracy and slightly more difficulty overfitting

Overall Architecture

- 8 layers, first 5 are convolutional, last 3 fully connected
- Output of last fully connected layer is fed into 1000-way softmax which produced distribution over the 1000 class labels
- Networks maximises multinomial logistic regression objective, equivalent to maximising average across training cases of log-probability

Layers

Layer 1

- Input: $224 \times 224 \times 3$
- Kernels: 96 of size $11 \times 11 \times 3$
- Stride 4 pixels (distance between centre of receptive fields

Layer 2

- Input: output of layer 1 after ReLU and max pooling
- Kernels: 256 of size $5 \times 5 \times 48$

Layer 3

- Input: output of layer 2 after ReLU and max pooling
- Kernel: 384 of size $3 \times 3 \times 256$

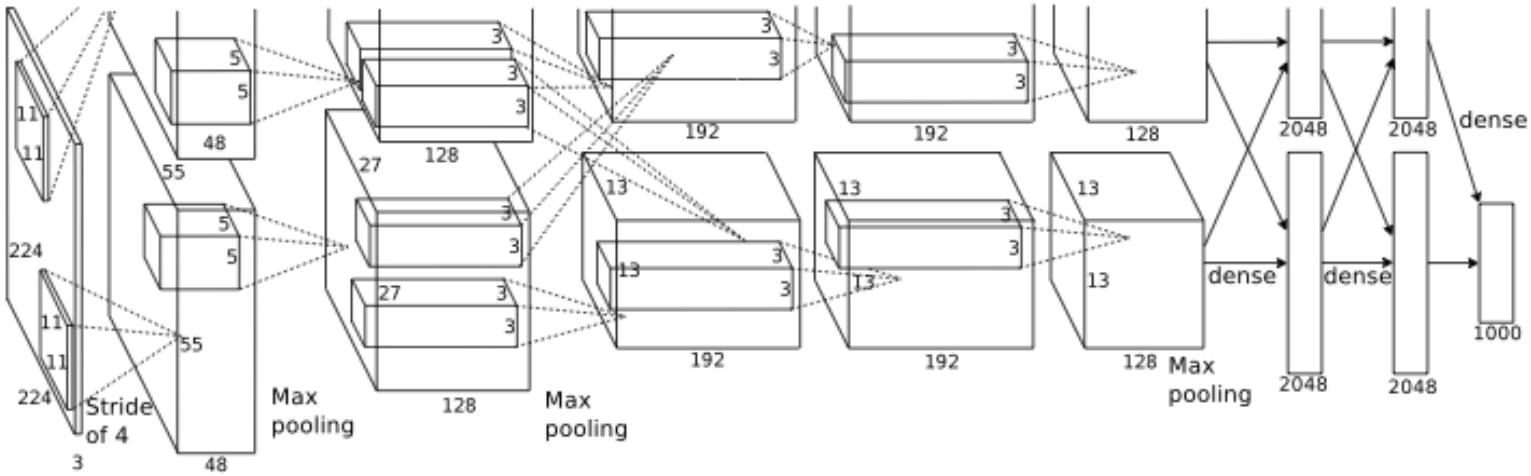
Layer 4

- Input: output of layer 3
- Kernel: 384 of size $3 \times 3 \times 192$ (note $192 = \frac{384}{2}$ so the data didn't transfer between GPUs on this level)

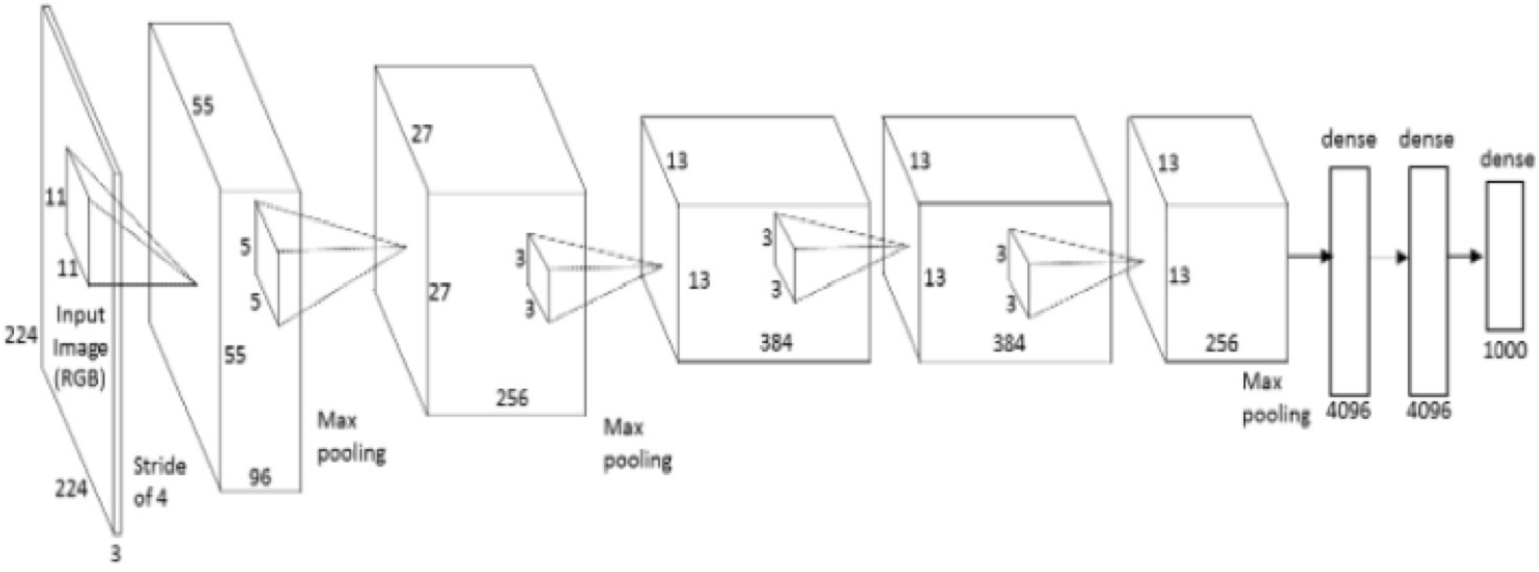
Layer 5(last convolutional)

- Input: output of layer 3
- Kernels: 256 of size $3 \times 3 \times 192$ (note again no inter GPU transfer)

Altogether


224
11
11
224
3
Stride
of 4
55
5
5
48
48
Max
pooling
27
128
3
3
3
128
Max
pooling
13
192
3
3
3
192
3
3
3
192
3
3
3
128
Max
pooling
dense
2048
2048
dense
dense
2048
2048
1000

Single GPU version



Reducing Overfitting

- Network has 60 million parameters, even with all the Imagenet data we can still easily overfit

Data Augmentation

- Artificially enlarge dataset using label preserving transforms
- Extract image and horizontal reflections. Image net images are 256×256, AlexNet input is 224×224
- this increases the input size by a factor of 2048 (although results are highly interdependent)
- Without this the network would substantially overfit thus requiring a smaller network
- At test time 5 224×224 patches are extracted (corners and centre) and their horizontal reflections
- We can also alternate the RGB values in the inputs

Dropout

- Combining predictions of many models is a successful way to reduce test errors but is very expensive in big networks
- Dropout on the other hand only adds a factor of 2 to training time
- Dropout involves setting neurons to 0 during training with probability 0.5
- Dropped neurons don't contribute in forward passes or backpropagation
- This results in a network that is different each time we pass in input data
- This reduces co-adaption of neurons (neurons doing the same thing) forcing learning of more robust features

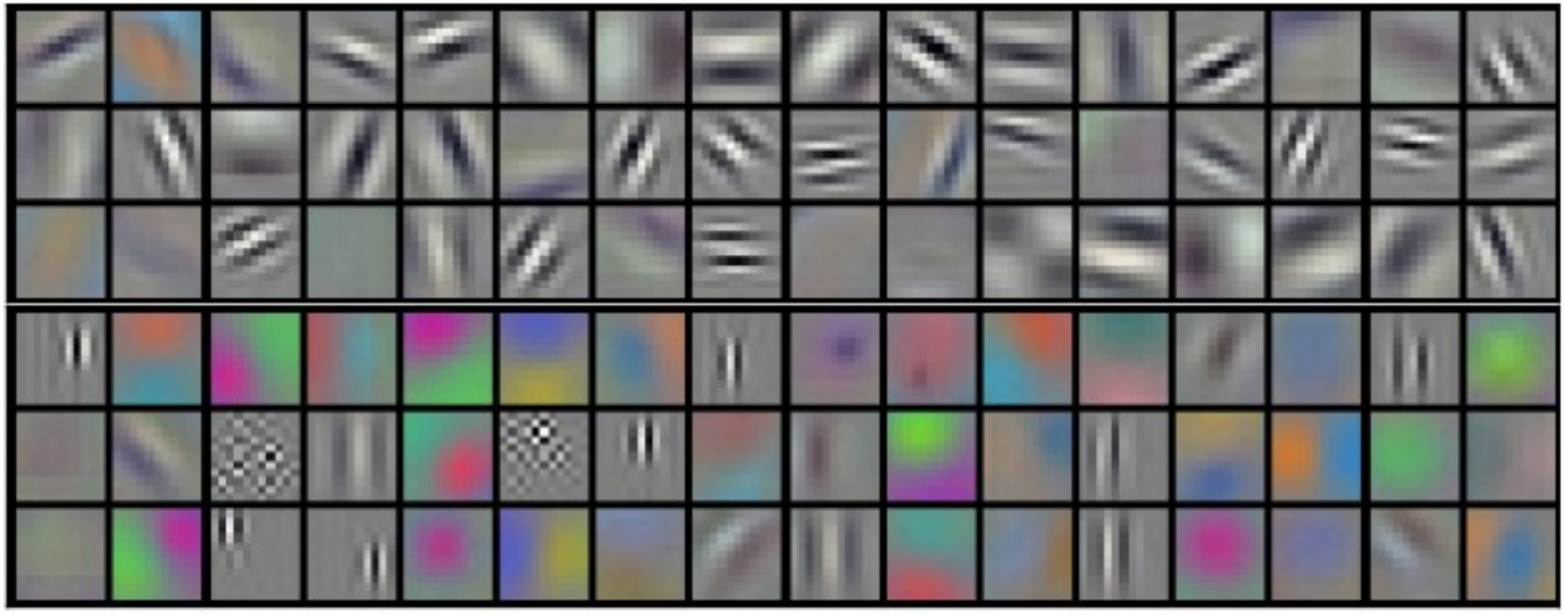
Training Details

- Trained using Stochastic Gradient Descent with batch size of 128, momentum 0.9, weight decay 0.0005
- Weight initialisation in each layer: zero mean Gaussian with $\sigma = 0.01$
- Neuron bias in layers 2, 3, 4, 5 and fully connected initialised as 1
- This initialisation accelerates early learning as ReLU start with positive inputs
- Learning rate of 0.01, reduced by factor of 10 when validation rate stopped improving
- 90 cycles through 1.2 million images taking 5/6 days on 2 GTX580's

Results

- top-1 error rate: 37.5
- top-5 error rate: 17.0
- Best performance on ILSVRC-2010 beating 47.1% and 28.2% respectively

Evaluating Convolutions



Here's the 96 convolutional kernels of size $11 \times 11 \times 3$ from the first layer

This shows the network has learned a variety of frequency and orientation selective kernels as well as colours blobs

Evaluating Fully Connected Layers

We can evaluate the fully connected layers by looking at images that excite the same collection of neurons
Here we see rows of images that caused the same neurons to excite, showing the network segments images similarly internally

Related further work

- Dropout (Srivastava, Hinton, Krizhevsky, Sutkever, Slakhutdinov)
- VGGNet (Oxford Vision Group)
- Inception (Google)
- ResNet (Facebook)