A Closer Look at AlexNet

Tutorial 6 – CNNs

February 12, 2018

CSC321

The ImageNet Recognition Challenge

The ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) had ∼1200 images for each of 1000 categories in 2012.

IMAGENET

AlexNet on ILSVRC-2012

AlexNet revolutionized the state-of-the-art in object recognition at the time.

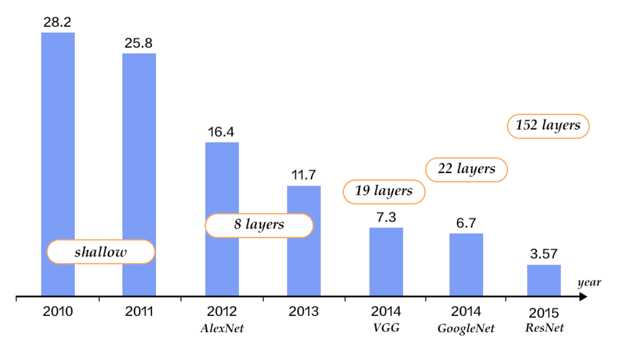
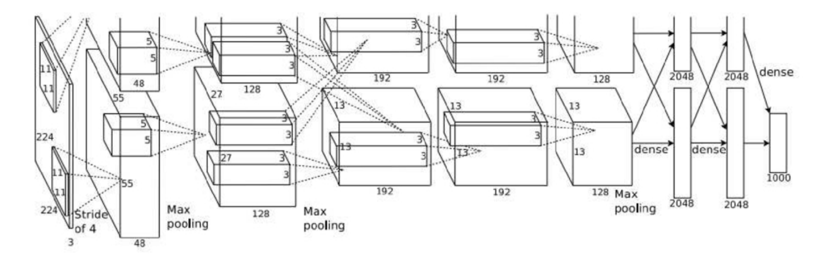


Figure 1: Top-5 error rates on ILSVRC image classification over Time

Source: http://paddlepaddle.org/docs/develop/book/03.image_classification/index.html

The Overall Architecture

Consists of 8 layers: 5 convolutional + 3 fully connected



The split (i.e. two pathways) in the image above are the split between two GPUs.

Inputs: RGB images with $224 \times 224 \times 3 = 150528$ values

Recall: Measuring Network Size (I)

Consider an input of size $W \times H \times C$ going to a convolutional layer L with square kernel size K and M output maps (channels). Then L has:

- There are WHM neurons (output units) in L, one for each of the WH “pixels” in the input and across the channels in the output M.
- There are K^2CM weights. K^2C per filter (the size of each “piece” of the input run through the filter), and M filters in total (one per output channel).
- There are WHK^2CM connections. A single kernel processes K^2C values in the input; this occurs for each of the WHM output units.

Recall: Measuring Network Size (II)

In contrast, a fully connected layer mapping *WHC* inputs to *WHM* outputs has far more weights and connections for small filter size *K*.

	Fully Connected	Convolutional
Number of Neurons	WHM	WHM
Number of Weights	W^2H^2CM	K^2CM
Number of Connections	W^2H^2CM	WHK^2CM

Notice that the *weight sharing* (i.e. using the same kernel per channel across the output units) has decreased the number of weights considerably.

We will omit the bias weights for simplicity.

The Convolutional Layers (I)

Let $N_i = WHM$, $P_i = K^2CM$, and $U_i = WHK^2CM$ be the number of output units, parameters (weights), and connections of layer L_i. Note that stride and max pooling each reduce the input W and H by the given factor.

- Layer L_1

  96 kernels (output channels) each of size $11 \times 11 \times 3$.

  Stride 4: the input W and H shrink by a factor of 4.
  Thus: $W = H = 55$, $C = 3$, $M = 96$, $K = 11$.

  - $N_1 = 55^2 \times 96 = 290{,}400$
  - $P_1 = 96 \times 11^2 \times 3 = 34{,}848$
  - $U_1 = 55^2 \times 11^2 \times 3 \times 96 = 105{,}415{,}200$

The Convolutional Layers (II)

- Layer L_2: 256 kernels each of size $5 \times 5 \times 48$. (Max pooling: $55/2 = 27$.) So $N_2 = 27^2 \times 256 = 186{,}624$; $P_2 = 2(5^2 \times 48 \times 128) = 307{,}200$; $U_2 = 223{,}948{,}800/2 = 111{,}974{,}400$.
- Layer L_3: 384 kernels each of size $3 \times 3 \times 256$. (Max pooling: $27/2 = 13$.) So $N_3 = 13^2 \times 384 = 64{,}896$; $P_3 = 3^2 \times 256 \times 384 = 884{,}736$; $U_3 = 13^2 \times 3^2 \times 256 \times 384 = 149{,}520{,}384$.

Note: layers 2, 4, & 5 are not connected to the preceding layer between GPUs; thus, one computes them separately and multiplies by 2. Recall: $N_i = WHM$, $P_i = K^2CM$, $U_i = WHK^2CM$

The Convolutional Layers (III)

- Layer L_4: 384 kernels each of size $3 \times 3 \times 192$. So $N_4 = 13^2 \times 384 = 64{,}869$; $P_4 = 2(3^2 \times 192^2) = 663{,}552$; $U_4 = 13^2 \times 3^2 \times 384^2/2 = 112{,}140{,}288$.
- Layer L_5: 256 kernels each of size $3 \times 3 \times 192$. So $N_5 = 13^2 \times 256 = 43{,}264$; $P_5 = 2(3^2 \times 192 \times 128) = 442{,}368$; $U_5 = 13^2 \times 3^2 \times 384 \times 256/2 = 74{,}760{,}192$.

Note: layers 2, 4, & 5 are not connected to the preceding layer between GPUs; thus, one computes them separately and multiplies by 2. Recall: $N_i = WHM$, $P_i = K^2CM$, $U_i = WHK^2CM$

The Fully Connected Layers

For the fully connected layers, again let N_i, P_i, and U_i be the number of output units, parameters (weights), and connections of layer L_i.

- Layer L_6: 4096 units. (Max pooling: input $13/2 = 6$.)
  $N_6 = 4096$; $P_6 = U_6 = 6 \times 6 \times 256 \times 4096 = 37{,}748{,}736$
- Layer L_7: $N_7 = 4096$ units. $P_7 = U_7 = 4096 \times 4096 = 16{,}777{,}216$
- Layer L_8: $N_8 =$1000 units. $P_8 = U_8 = 4096 \times 1000 = 4{,}096{,}000$

Notice that the number of parameters is much larger for the dense layers than the convolutional ones.

Summary of Results

Layer	Units	Weights	Connections
L_1 (Conv)	290,400	34,848	105,415,200
L_2 (Conv)	186,624	307,200	111,974,400
L_3 (Conv)	64,896	884,736	149,520,384
L_4 (Conv)	64,869	663,552	112,140,288
L_5 (Conv)	43,264	442,368	74,760,192
L_6 (Dense)	4096	37,748,736	37,748,736
L_7 (Dense)	4096	16,777,216	16,777,216
L_8 (Dense)	1000	4,096,000	4,096,000
Conv Subtotal	650,080	2,332,704	553,810,464
Dense Subtotal	9192	58,621,952	58,621,952
Total	659,272	60,954,656	612,432,416

Conclusion

Overall, AlexNet has about 660K units, 61M parameters, and over 600M connections.

Notice: the convolutional layers comprise most of the units and connections, but the fully connected layers are responsible for most of the weights.

More modern networks can do better with fewer parameters (e.g. GoogLeNet).

Further Reading

The original paper:

- Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. “Imagenet classification with deep convolutional neural networks.” *Advances in neural information processing systems.* 2012.

It’s possible to reduce the number of parameters of AlexNet by 9x without losing accuracy:

- Han, Song, et al. “Learning both weights and connections for efficient neural network.” *Advances in neural information processing systems.* 2015.