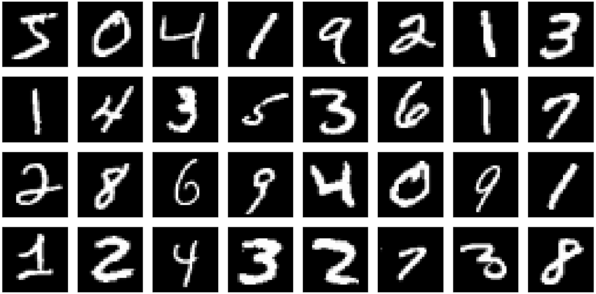
Convolutional Neural Networks (cont.)

Nan Ye

School of Mathematics and Physics
The University of Queensland

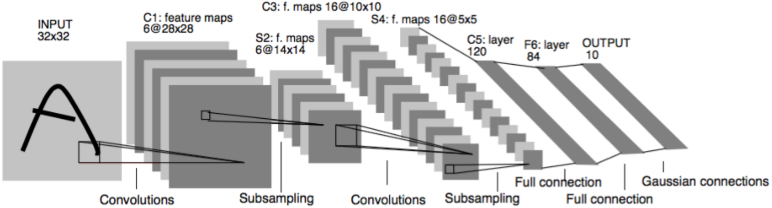
MNIST



http://yann.lecun.com/exdb/mnist/

- The MNIST dataset is an early large dataset used as a benchmark for evaluating handwritten digits recognition algorithms.
- There are 60,000 labeled training images, and 10,000 labeled test images.

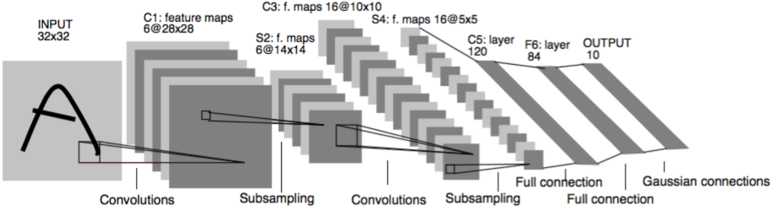
LeNet-5 (1998)



- 7 layers (excluding input layer)
- Layer 1,3,5 are convolution layers (C1, C3, C5)
- Layer 2,4 are sub-sampling layers (S2, S4)
- Layer 6 is fully-connected (F6)
- Layer 7 is the output layer

LeCun, Bottou, Bengio, and Haffner, Gradient-based learning applied to document recognition, 1998

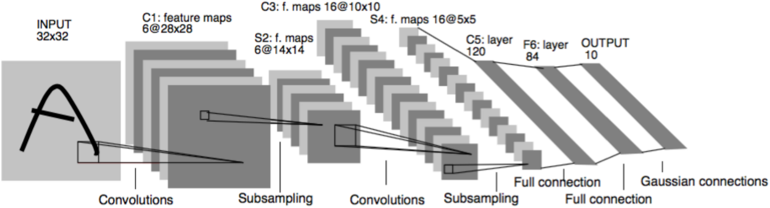
LeNet-5 (1998)



- Activation function is hyperbolic tangent up to F6.
- Output layer uses the Euclidean Radial Basis Function (RBF) units (each computes the squared distance between the input vector and the weight vector of the unit).

LeCun, Bottou, Bengio, and Haffner, Gradient-based learning applied to document recognition, 1998

LeNet-5 (1998)

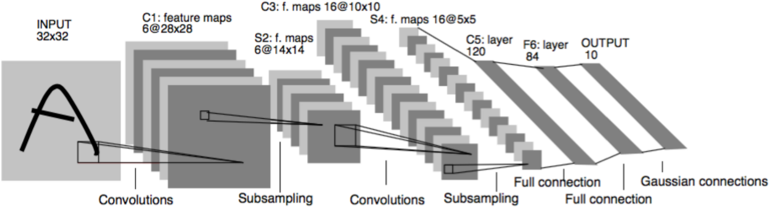


Convolutional layers

- Each convolutional layer has units organized as several 2D arrays.
- C1: 6 filters of size 5x5
- C2: 16 filters of size 5x5

LeCun, Bottou, Bengio, and Haffner, Gradient-based learning applied to document recognition, 1998

LeNet-5 (1998)

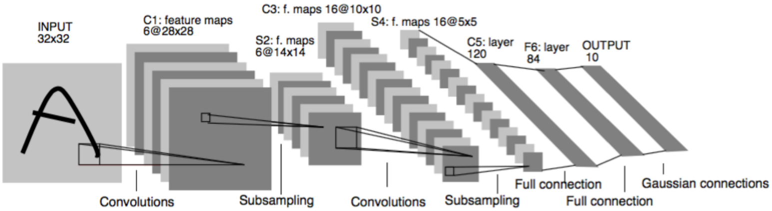


Sub-sampling/pooling layers

- Each sub-sampling layer has units organized as the same number of 2D arrays as previous convolutional layer.
- Reduces each 2D array in the previous convolutional layer to a lower resolution, by taking the sum of each non-overlapping 2x2 neighborhood and adding a bias to it.

LeCun, Bottou, Bengio, and Haffner, Gradient-based learning applied to document recognition, 1998

LeNet-5 (1998)



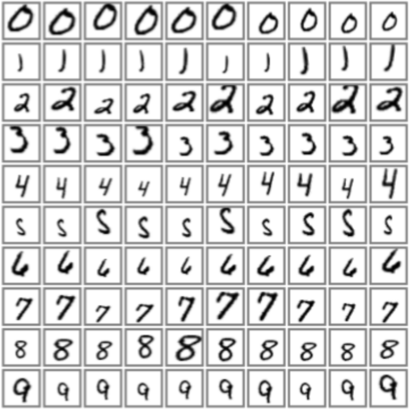
Trainable using backprop.

LeCun, Bottou, Bengio, and Haffner, Gradient-based learning applied to document recognition, 1998

Performance

- MNIST dataset: 60,000 training examples, 10,000 test examples, resized to 32x32.
- 0.95% error.

Adding distorted training data helps



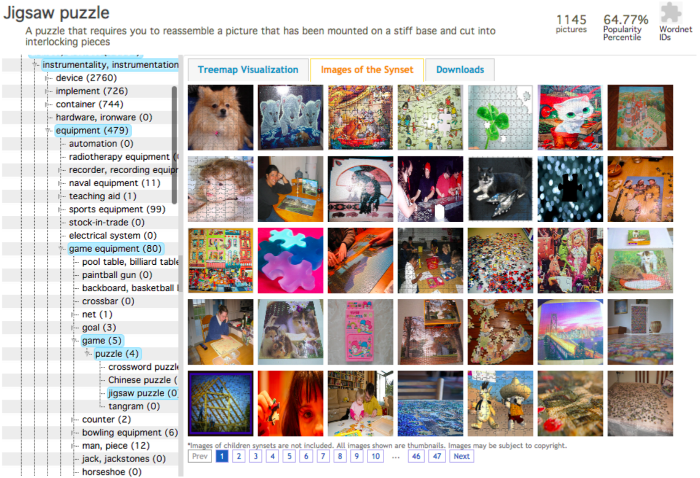
- Additional 540,000 distorted training examples.
- Error improved to 0.8%.

Errors made by LeNet5

Variants

- Max-pooling is found to work better than average-pooling.
- Overlapping pooling is sometimes used.
- Rectified linear unit (ReLU, $\max(0, x)$) is now often used instead of sigmoid units ($\tanh(x)$ or $\sigma(x)$).

ImageNet

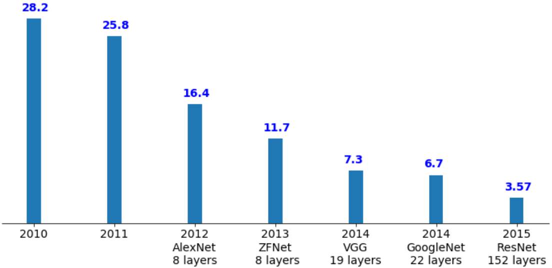


http://www.image-net.org/

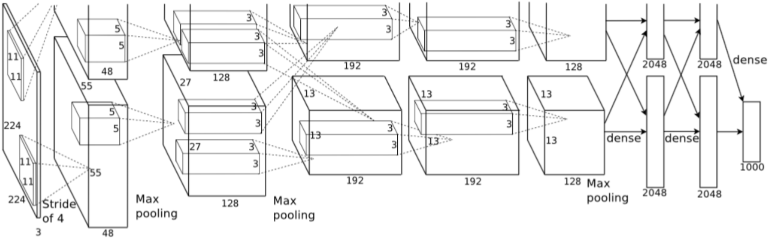
- ImageNet is a recent large image database.
- 1000 different object classes in 1.3 million high-resolution training images from the web.

ILSVRC

- ILSVRC (ImageNet Large Scale Visual Recognition Challenge) was a competition based on the ImageNet data.
- Top-5 classification error rates for the best systems



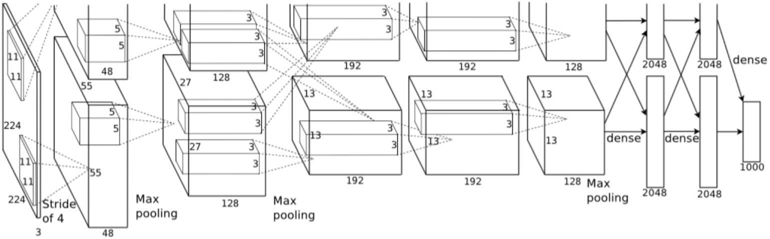
AlexNet (2012)



- Achieved one of the first strong results for deep neural networks.
- Reduced previous best top-5 error from 25.8% to 16.4%.

Krizhevsky, Sutskever, and Hinton, Imagenet classification with deep convolutional neural networks, 2012

AlexNet (2012)

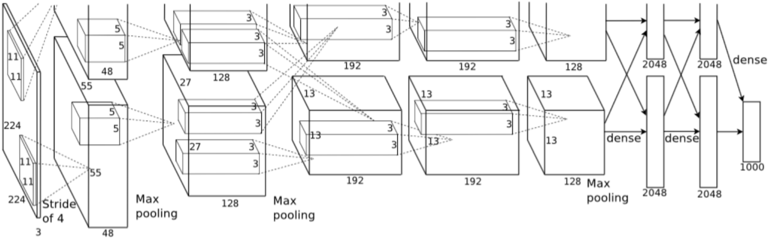


Layers

- Five convolutional layers
  - 1st and 2nd are followed by max-pooling and normalization layers (not common anymore)
- Three fully-connected layers
- 60 million parameters and 650,000 neurons

Krizhevsky, Sutskever, and Hinton, Imagenet classification with deep convolutional neural networks, 2012

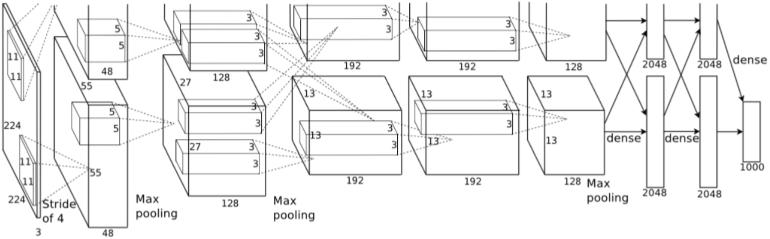
AlexNet (2012)



ReLU activation

- All hidden neurons use ReLU
  - About 6 times faster than sigmoid units
  - More expressive

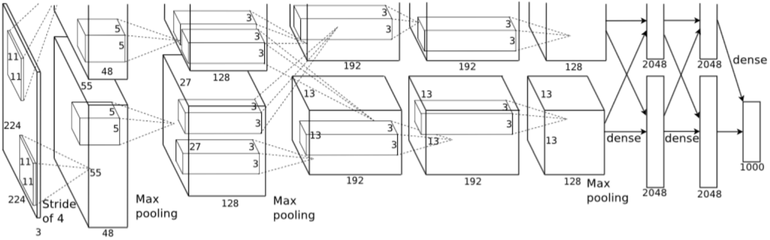
AlexNet (2012)



Training on multiple GPUs

- Used two Nvidia GTX 580 GPUs, essentially half of the neurons in each (which corresponds to the top and bottom parts of the architecture above)
- GPUs communicate only in certain layers.

AlexNet (2012)

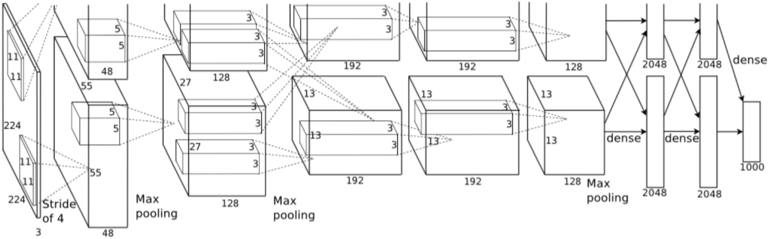


Regularization

- Train on random 224x224 patches from the 256x256 images to get more data.
- Use left-right reflections of the images.
- At test time, average class distributions for the corner and center 224x224 corner patches and their reflections (10 in total).
- Use a technique called dropout to regularize the weights in the fully connected layers (which contain most of the parameters).

Krizhevsky, Sutskever, and Hinton, Imagenet classification with deep convolutional neural networks, 2012

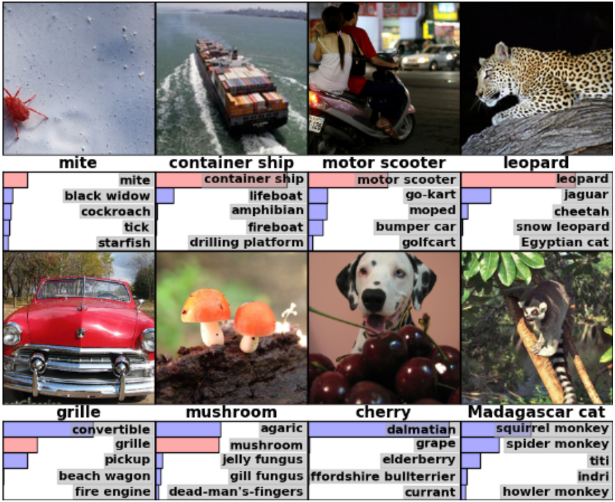
AlexNet (2012)



Training objective

- Convert outputs of last layer to a distribution using 1000-way softmax, and maximize likelihood
$$\text{softmax}(o_1, \ldots, o_m) = (e^{o_1}, \ldots, e^{o_m}) / \sum_i e^{o_i}.$$
- Equivalent to minimizing the cross entropy loss
$$L((o_1, \ldots, o_m), y) = -o_y + \ln \sum_i e^{o_i}.$$

input	3x224x224
conv1	96x3x11x11, stride 4
maxpool1	3x3 filters, stride 2
norm1	normalization
conv2	256x48x5x5
maxpool1	3x3 filters, stride 2
norm1	normalization
conv3	384x256x3x3
conv4	384x192x3x3
conv5	256x192x3x3
fc6	2048
dropout6	2048
fc7	2048
dropout7	2048
fc8	1000



AlexNet classification examples

ZFNet (2013)

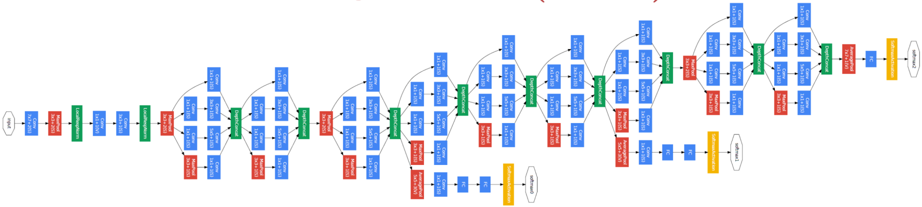
- ZFNet is the same as AlexNet except that
  - CONV1: change from (11x11, stride 4) to (7x7, stride 2)
  - CONV3, 4, 5: instead of 384, 384, 256 filters use 512, 1024, 512
- Reduced previous best top-5 error from 16.4% (AlexNet) to 11.7%.

VGGNet (2014)

- 16 or 19 layers (VGG16 and VGG19, VGG19 only slightly better but requires more memory)
- Small filters, deeper networks
  - Only CONV(3x3, stride 1, pad 1) and MAXPOOL(2x2, stride 2)
  - Stacking multiple small convolutional layers has the same effective receptive field as a larger convolutional layer, but deeper and more nonlinearity.
  - Fewer parameters
- Reduced previous best top-5 error from 11.7% (ZFNet) to 7.3%.

Simonyan and Zisserman, Very deep convolutional networks for large-scale image recognition, 2014

GoogLeNet (2014)

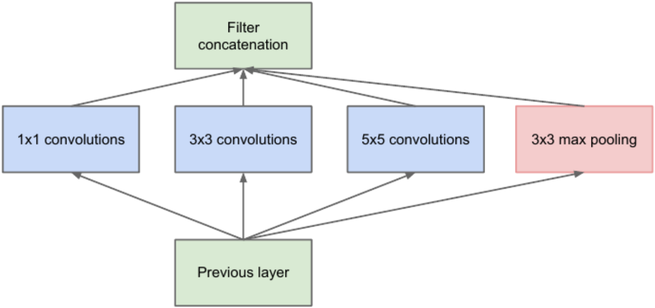


- Deeper networks (22 layers), with computational efficiency
- The key idea is to use sparse modules called *Inception* modules to replace computionally expensive fully connected layers (even inside the convolutions)
- Lower layers of the network are traditional convolutional layers, and then followed by a stack of Inception modules.
- Auxiliary classifiers are connected to intermediate layers to inject additional gradients.
- Reduced previous best top-5 error from 11.7% (ZFNet) to 6.7%.

Szegedy, Liu, Jia, Sermanet, Reed, Anguelov, Erhan, Vanhoucke, and Rabinovich, Going Deeper with Convolutions, 2014

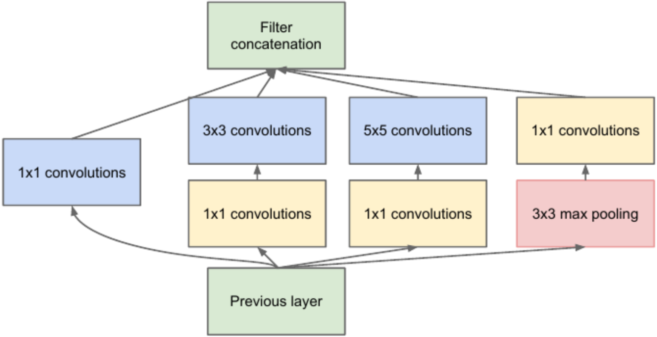
Inception module

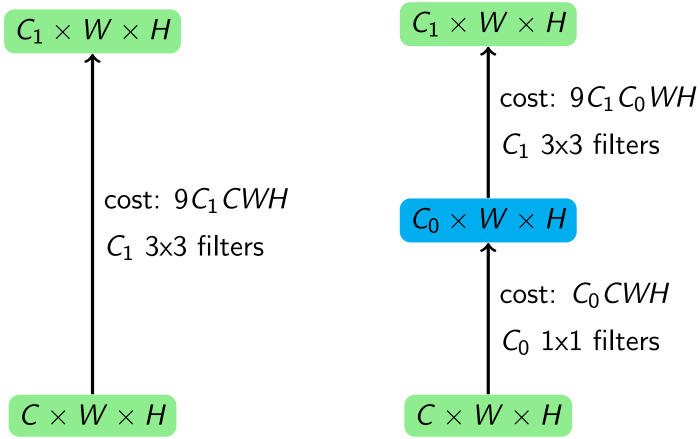
- A naive design is to concatenate several convolutional modules of different resolution together with a pooling module



- This is computationally very expensive.

- The actual Inception module first performs dimension reduction on a CxHxW layer to size C'xHxW by doing 1x1 convolution (with C' parameters)





If $C_0 = \frac{1}{3}C < C_1$, then

$$\frac{C_0CWH + 9C_1C_0WH}{9C_1CWH} \leq \frac{C_1CWH + 3C_1CWH}{9C_1CWH} = \frac{4}{9}.$$

ResNet (2015)

- ResNet uses a trick called skip connection to make it possible to train very deep neural networks.
- Reduced top-5 error from 6.7% to 3.57%.
- We will cover this in a few weeks.

What You Need to Know...

- The classic LeNet
- Key ideas in several modern CNNs
    - AlexNet: larger depth, ReLU activation, parallelization, regularization,
    - ZFNet: minor hyperparameter tuning
    - VGGNet: smaller filters, deeper networks
    - GoogLeNet: Inception module, auxiliary classifiers