Introduction to Deep Learning

Donghui Yan

Department of Mathematics and
Program in Data Science
University of Massachusetts Dartmouth

June 26, 2017

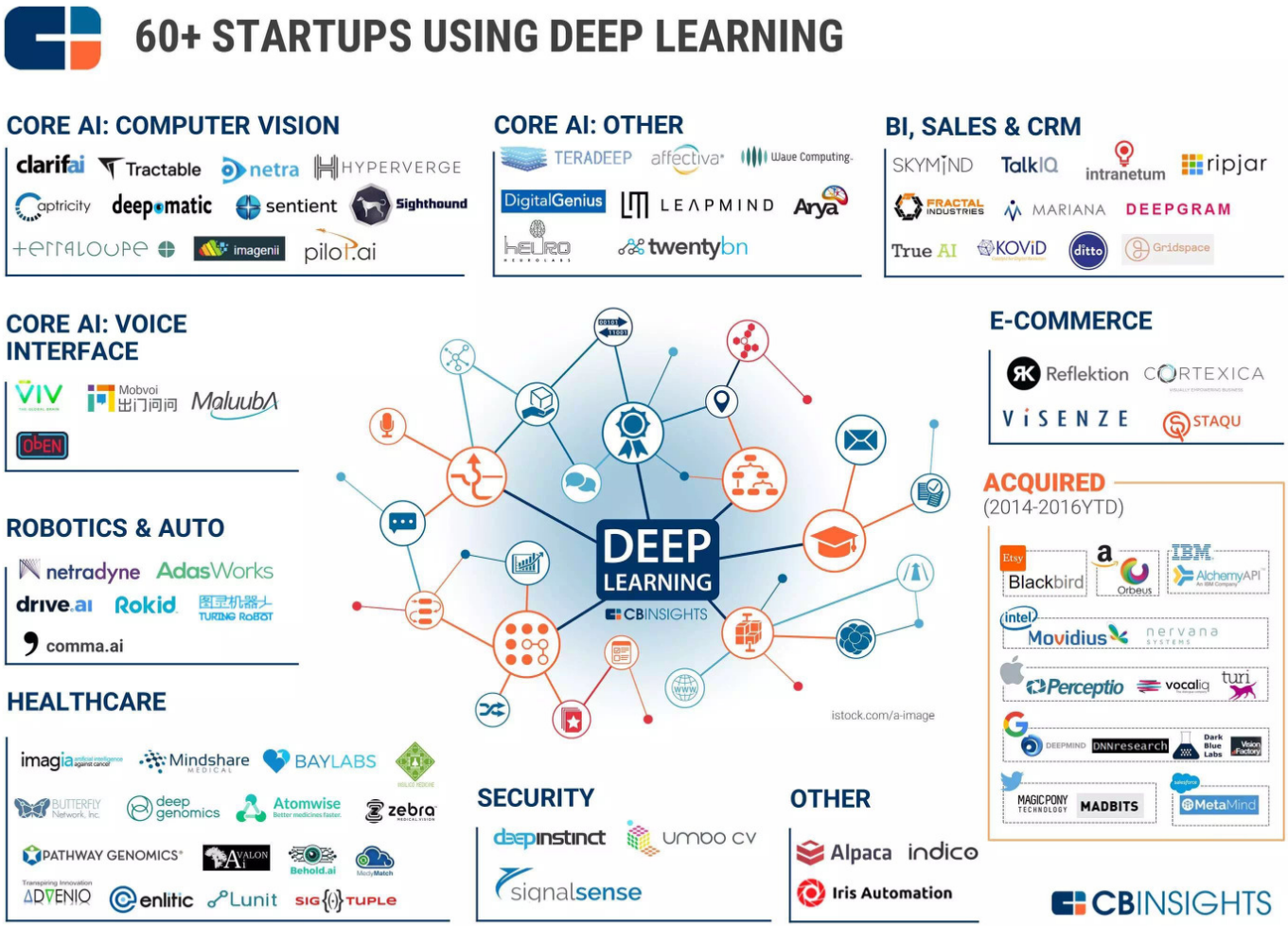
Outline

- Introduction
- Deep learning
- Practical issues
- Convolutional NN

The exponential popularity of deep learning

- Emerges in recent years (since 2006)
  - Significant advance in various domains
    - e.g., image, speech, natural language processing etc
  - Central role behind several technology breakthroughs
    - e.g., *AlphaGo*, self-driving, conversational agent etc
- Deep roots in neural network
  - Vast majority of work actually deep neural network
    - e.g., winners of *ImageNet* typically have 100-200 layers, and millions of parameters
- Warmly embraced by many major corporations
  - *Google, Facebook, Uber, Baidu, Didi, Goldman Sachs, Citadel* etc.

Ventures related to deep learning



The story of deep learning

- Hinton asked Salakhutdinov to work on a deep NN (2006)
  - About the computational issue
  - *Accidently*, some intermediate results turned out to be
    - 'Good' *representation* of the data
  - Remarkable performance when using such representation
    - This marks the "birth" of deep learning
    - *However, deep learning should not be limited to NN.*
- ♠ Why did it take so long for deep learning to evolve?
  - Fundamental building block available long ago.

Challenges in deep neural network

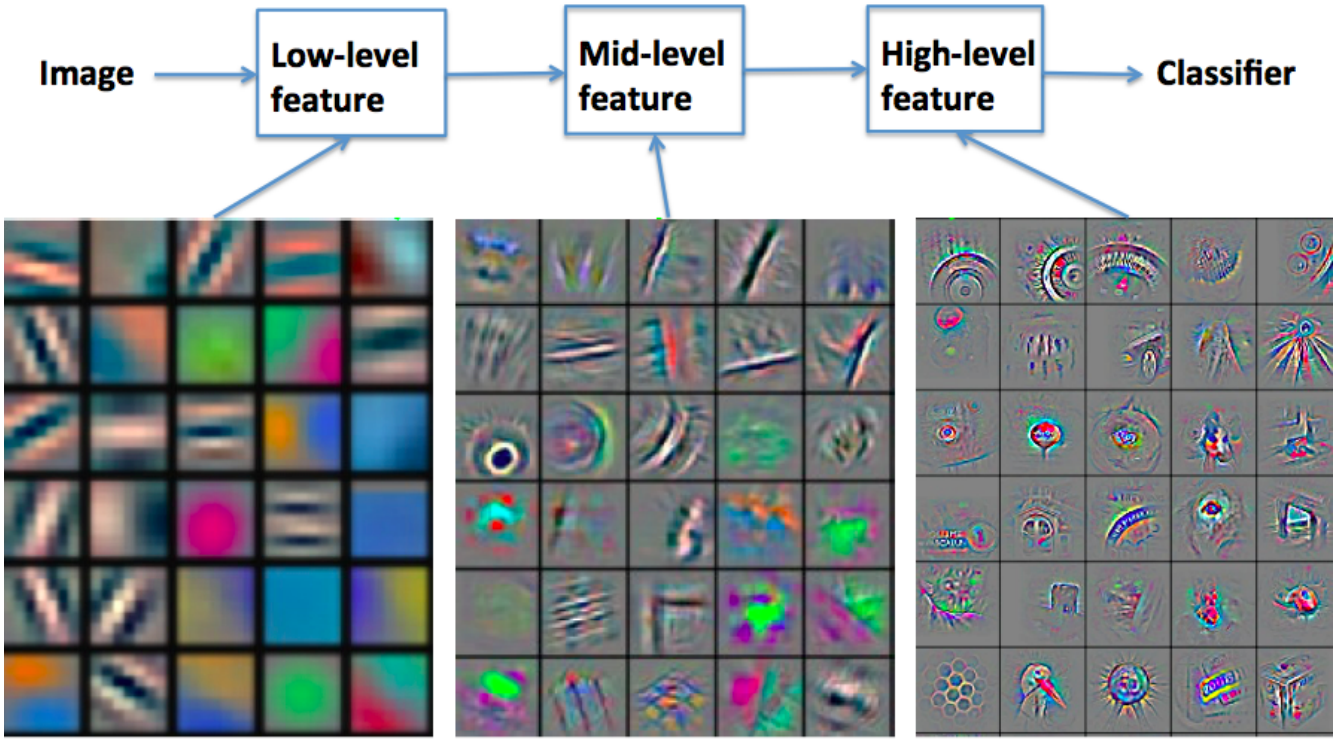
- Neural network only converges to local optima
  - A good starting value is crucial
- Prune to overfitting
  - Too many parameters
  - Require large dataset to prevent overfitting
- High computational complexity.

Deep learning

Deep Learning

What exactly is deep learning?

- A popular view is *representation* learning [Hinton et al 2006]
  - A hierarchy of representations
    - Low-level transformed into a higher-level representation
  - Current work mostly deep NN.



Example deep representation hierarchy

- Image recognition
  - $Pixel \rightarrow edge \rightarrow texton \rightarrow motif \rightarrow parts \rightarrow object$
- Natural language processing
  - $Character \rightarrow word \rightarrow word\ grp \rightarrow motif \rightarrow clause \rightarrow sentence \rightarrow \ldots$
- Speech recognition
  - $Sample \rightarrow spectral\ band \rightarrow sound \rightarrow phoneme \rightarrow word \rightarrow phrase \rightarrow sentence \rightarrow \ldots$

Deep Vs shallow learning

- Shallow: features obtained by knowledge/feature engineering
  - Typically requires substantial human expertise and efforts
  - May take several months or years' time
    - e.g., SIFT took about 10 years (Lowe 2004)
- Deep: features learned by algorithm
  - Typically *multiple levels of (hierarchical) representation*
    - Some def of deep learning require only a representation
  - Often called end-to-end learning (without human involved)
  - Such features typically beyond usual feature engineering.

Are these models deep?

- 2-layer models
- Neural network with 1 hidden layer
- SVM and kernel methods
- Classification trees

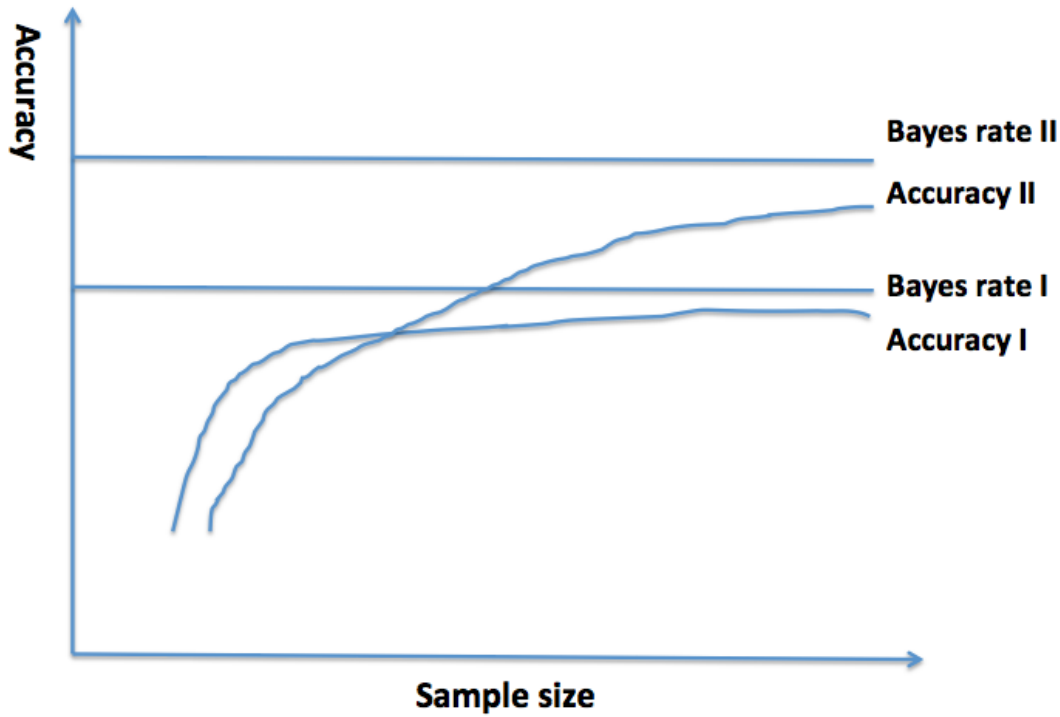
♠ *No* as there is no feature hierarchy.

Why may it work?

- A crude view by Bayes rate
- Intuitions from layered feature extraction
- Some constructs in a deep net make algorithm
  - Less sensitive to noise, contrast, and small transformations
    - Details in discussion about CNN.

Why may it work? A crude view

- 'I' indicates original data, 'II' with deep features
  - 'II' leads to a larger model space thus higher Bayes rate
    - Large sample size quickly leads to better accuracy.



What is the message here?

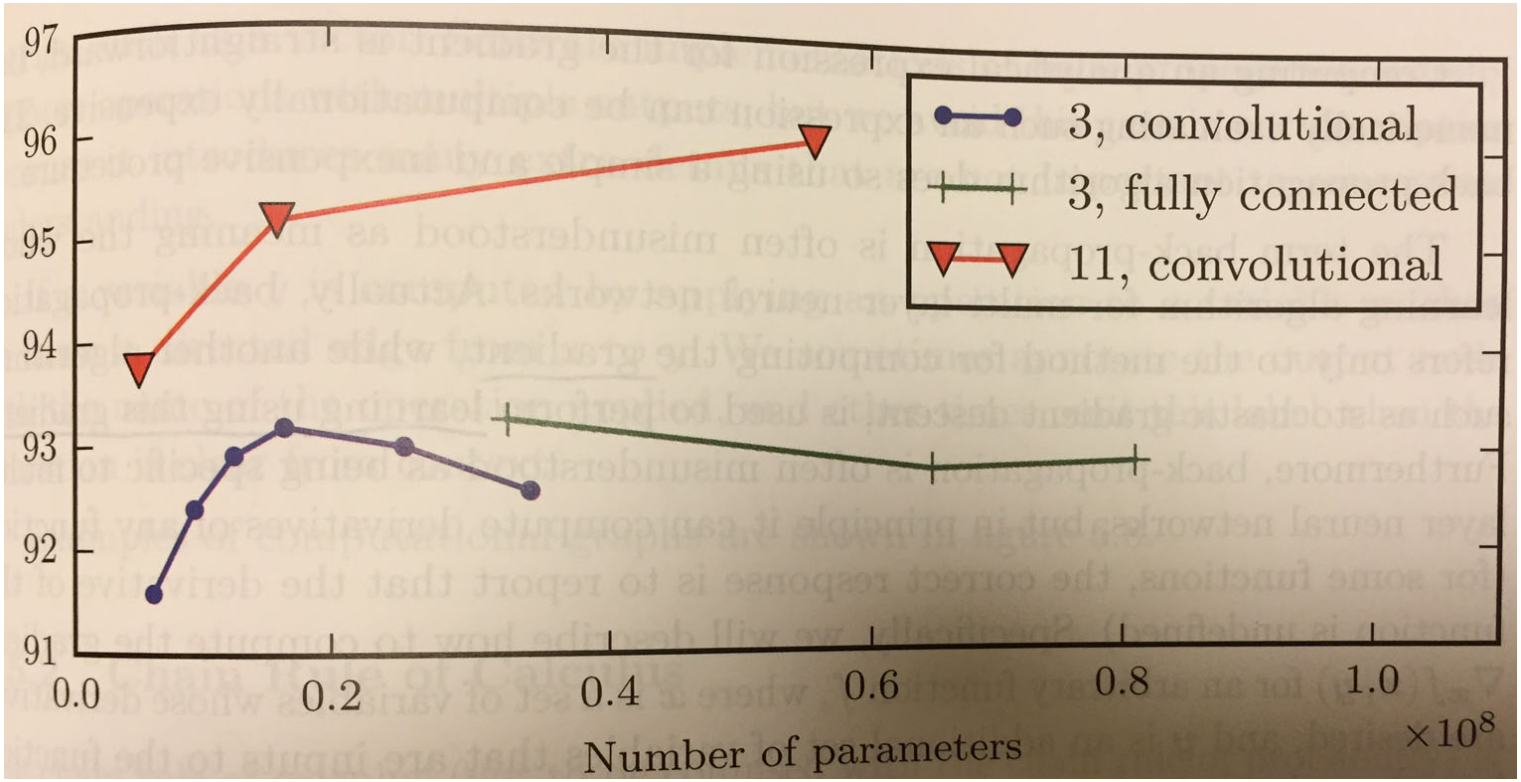


Figure: Image courtesy Goodfellow et al 2015.

- Deeper is beneficial
  - Exponential advantage of a deep network
    - Montufar et al show (2004) show # linear regions separated by deep NN with d inputs, depth l, and n units per hidden layer is

$$O\left(\binom{n}{d}^{d(l-1)} n^d\right).$$

- Structure also important.

Deep features may be informative

- 1st layer of a deep NN gives edge-like feature
  - 2nd and higher layer texture or shape etc information.

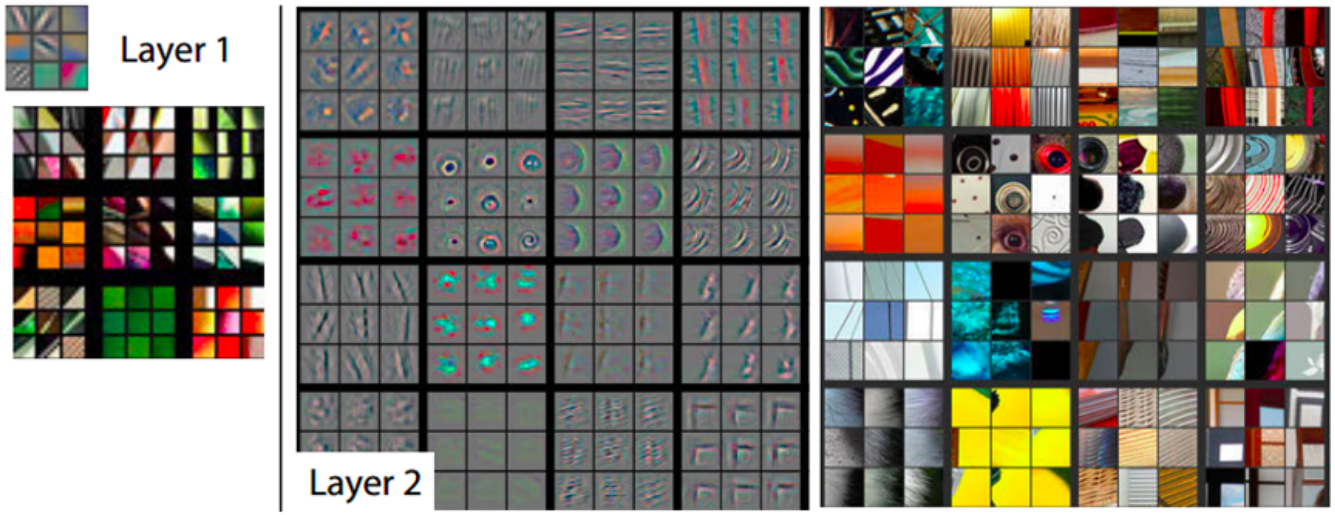


Figure: *Matthew D. Zeiler and Rob Fergus (2014).*

Deep features may be informative (continued)

- 3rd layer of a deep NN gives object parts

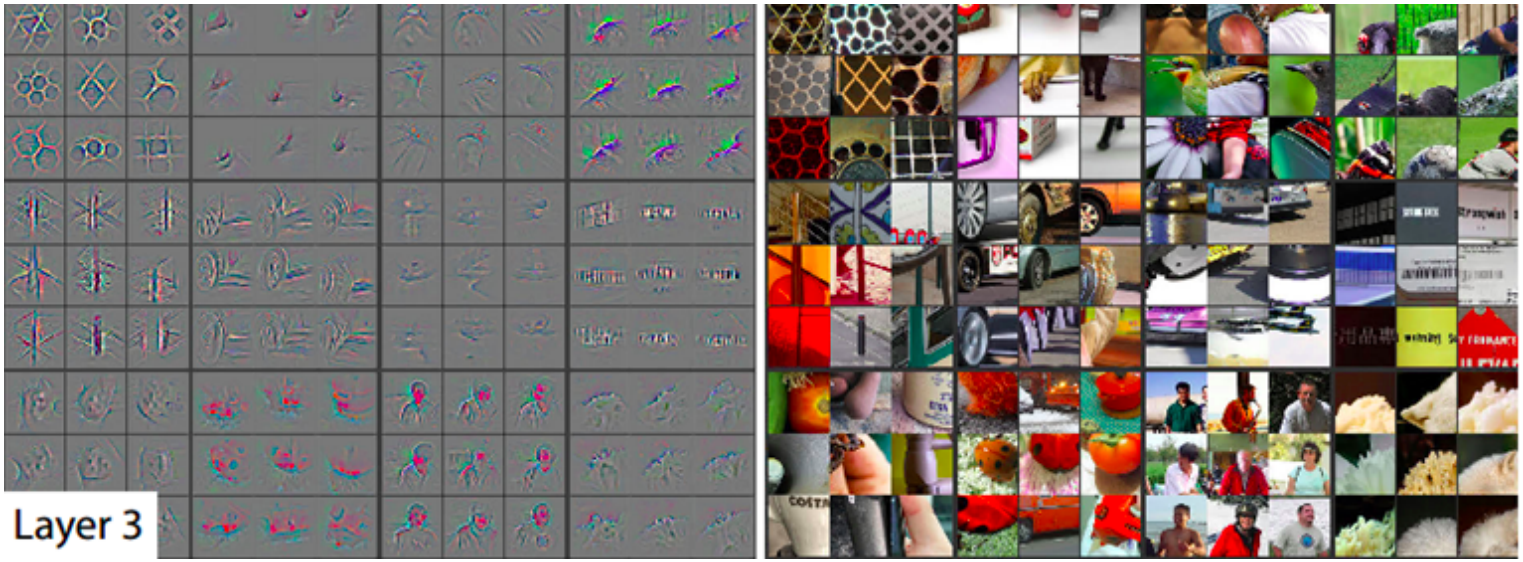


Figure: *Matthew D. Zeiler and Rob Fergus (2014).*

Deep features may be informative (continued)

- 4th and above layers gives simple objects

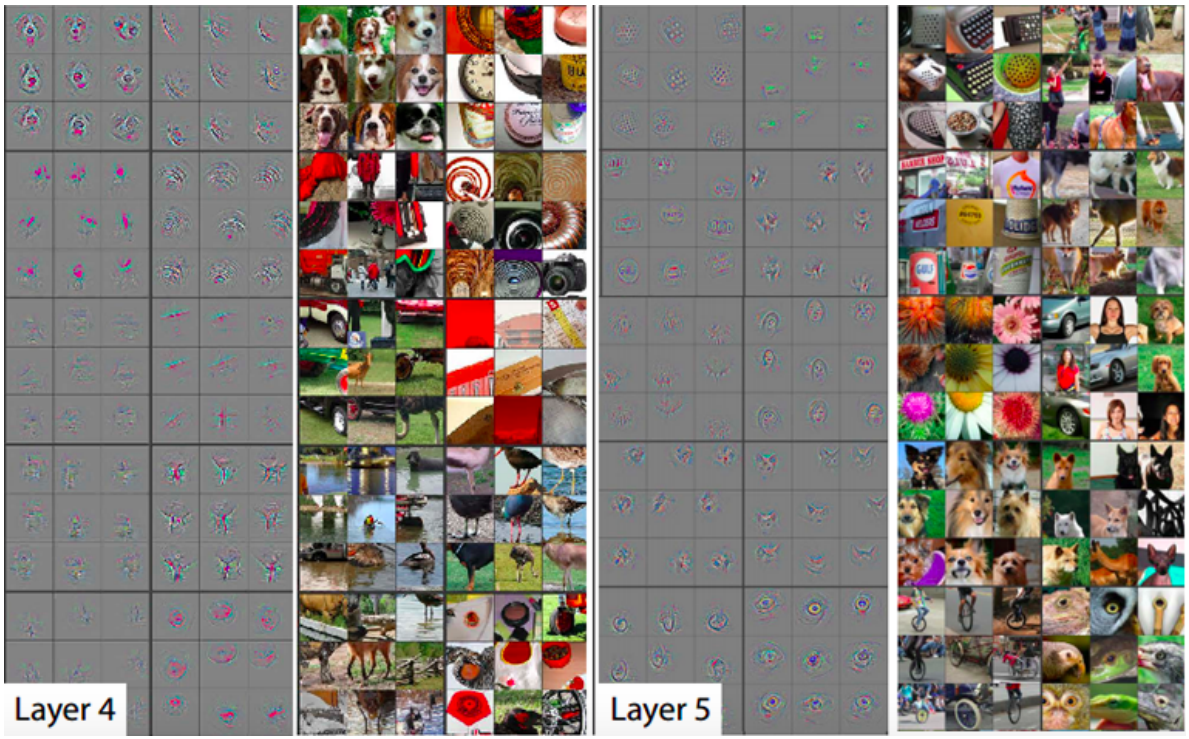


Figure: *Matthew D. Zeiler and Rob Fergus (2014).*

MNIST digits recognition

- A database of handwritten digits
  - Image size 28 x 28
  - 60000 training, 10000 test images
  - State of the art error rate: 0.27%
    - 6-layer CNN 784-50-100-500-1000-10-10

Experiment with 4 hidden layers

```
> library(deepnet);
> dnn <- dbn.dnn.train(train_x_n, mnist$train$yy,
        hidden_dropout=0.6,
        hidden = c(300, 100, 100,100),
        numepochs = 5, cd = 3);
> err.dnn <- nn.test(dnn, test_x_n, mnist$test$yy);
> dnn_predict <- nn.predict(dnn, test_x_n);
> print(err.dnn)
[1] 0.0445
> print(dnn_predict[1,])
 [1] 3.876123e-03 3.326393e-04 2.584891e-03 1.029923e-02
 [5] 5.424675e-04 1.807185e-04 2.153286e-05 9.921616e-01
 [9] 1.166920e-04 3.344022e-02
> print(mnist$test$y[1])
[1] 7
```

Deep learning packages

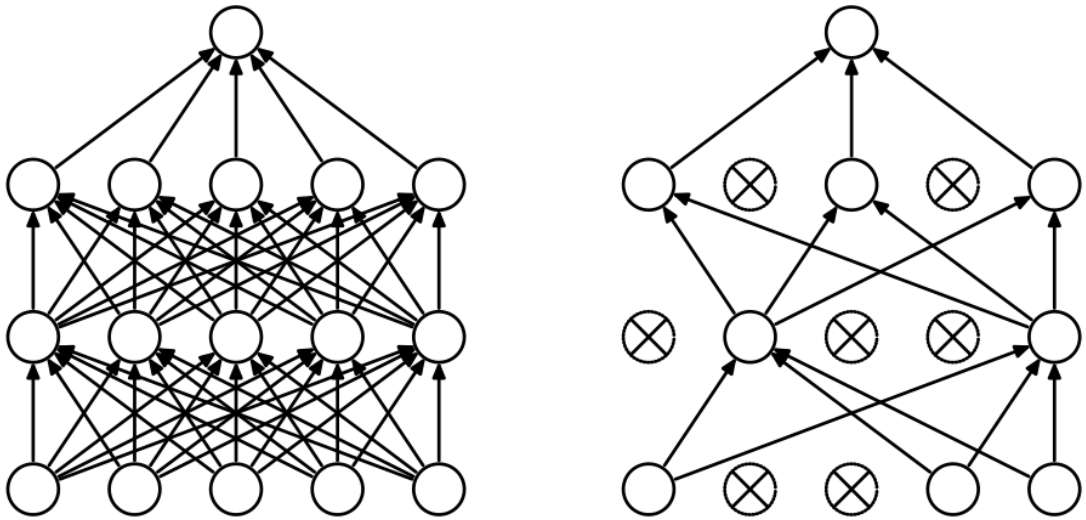
- TensorFlow (Google)
- Caffe (Jia, Berkeley)
- Theano (LISA lab)
- Cuda-convnet (Alex Krizhevsky, Google)
- Overfeat (NYU)
- H2O
- R packages
  - *deepnet*
  - *darch*
  - *MXNet*
  - *h2o*

Feasibility of deep neural networks

- Large training set
- Sufficient computational resources
- ♠ *Not always feasible*
  - One idea is 'dropout'
  - Other ideas include *stochastic gradient descent* and hashing etc
  - GPU to speed up computation.

Dropout in deep learning

- Random dropout in deep learning (Hinton et al 2012)
  - Substantially improved performance
    - On MNIST, error rate drops from 1.6% (Simard et al 2003)
    - To 0.79% by DBM + dropout.

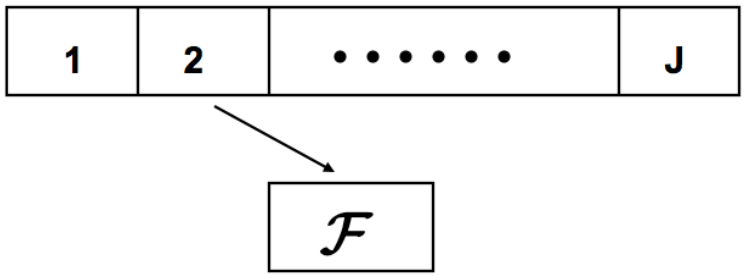


(a) Standard Neural Net

(b) After applying dropout.

Why does the idea of dropout work?

- Insights from a result on 'thinning' (Yan et al 2012)



- The loss on Bayes rate is negligible
  - When dropping a substantial fraction of features for classification task on high dimensional data.

Convolutional NN

CNN

Approaches to challenges in deep NN

- Possible approaches
  - Incorporate structures
  - Parameter sharing
  - The use of memory/forget
- Resulting network architecture
  - Convolutional NN (CNN, LeCun 1989)
  - Recurrent NN
    - Rumelhart, Hinton and Williams 1986
  - Long short-term memory (LSTM)
    - Hochreiter and Schmidhuber 1997.

CNN

- NN with specialized connectivity structure
  - Target at data with a grid topology
  - e.g., time-series data, images
- Name due to the *convolution* operation
  - Slightly different from usual definition
- Tremendous success in practice
  - e.g., LeNet 1-5 for USPS digits recognition (LeCun 1989-1998)
  - Winner of *imageNet challenge* in several years.

CNN

- Stack multiple layers of feature extractors
- Deeper layers extract more global/invariant features
- Features extracted from the final layer input to a classifier.

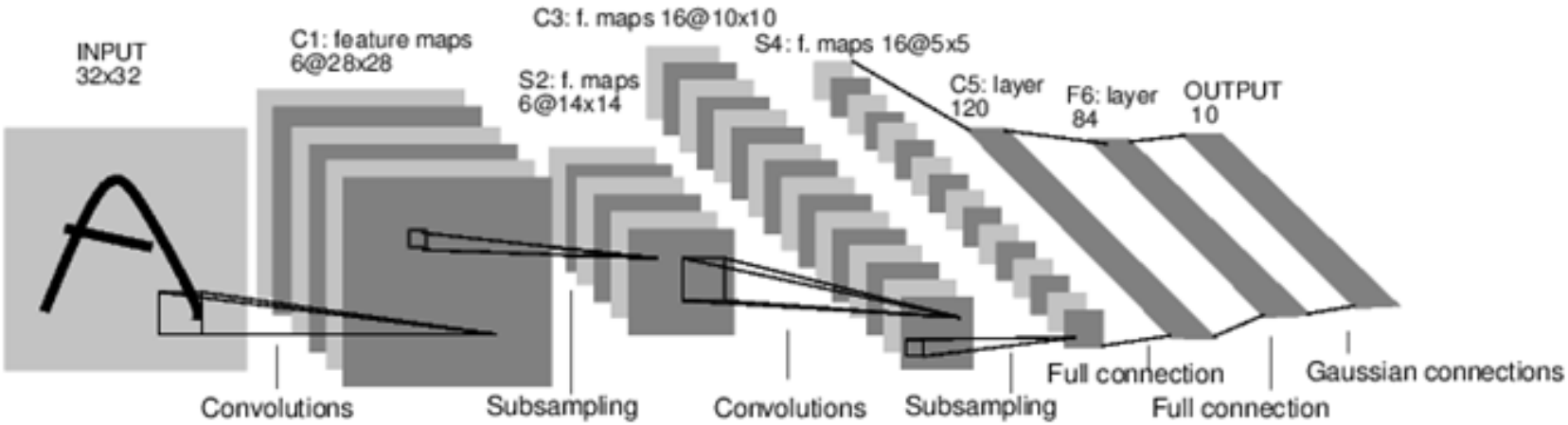
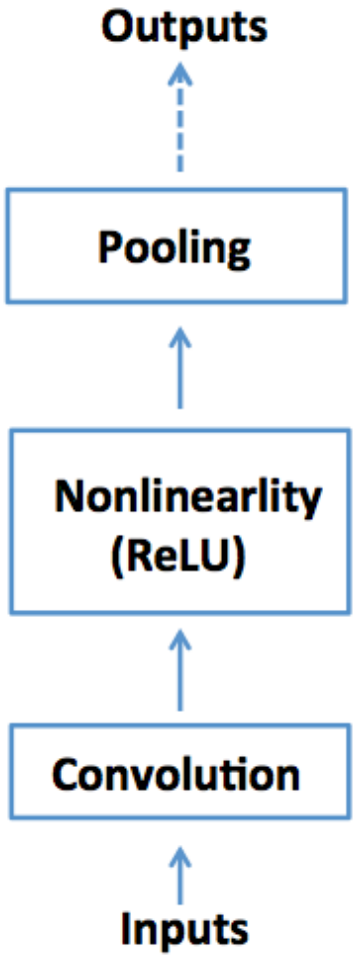


Figure: *LeCun, Bottou, Bengio, and Haffner, 1998*

Operations in CNN

- Convolution
- Nonlinearity
  - Activation function (aka detector stage)
- (Spatial) pooling
- Normalization
- As usual, training of CNN by back-propagation of classification error.



Convolution

- To encode local dependency via a filter
  - Can be viewed as *weighted average* of neighbors
- Mathematically, convolution is defined by

$$s(t) = \int x(a)w(t-a)da \triangleq (x \star w)(t)$$

- Image $I(.,.)$ convolve with kernel K

$$\begin{aligned} S(i,j) &= (I \star K)(i,j) = \sum_m \sum_n I(m,n)K(i-m, j-n) \\ &= \sum_m \sum_n I(i-m, j-n)K(m,n). \end{aligned}$$

- Desirable property
  - Fewer parameters (filter weights).

Illustration of convolution

- $(3 \times 4 \text{ input}) \star (2 \times 2 \text{ kernel}) \Rightarrow 2 \times 3 \text{ output}.$

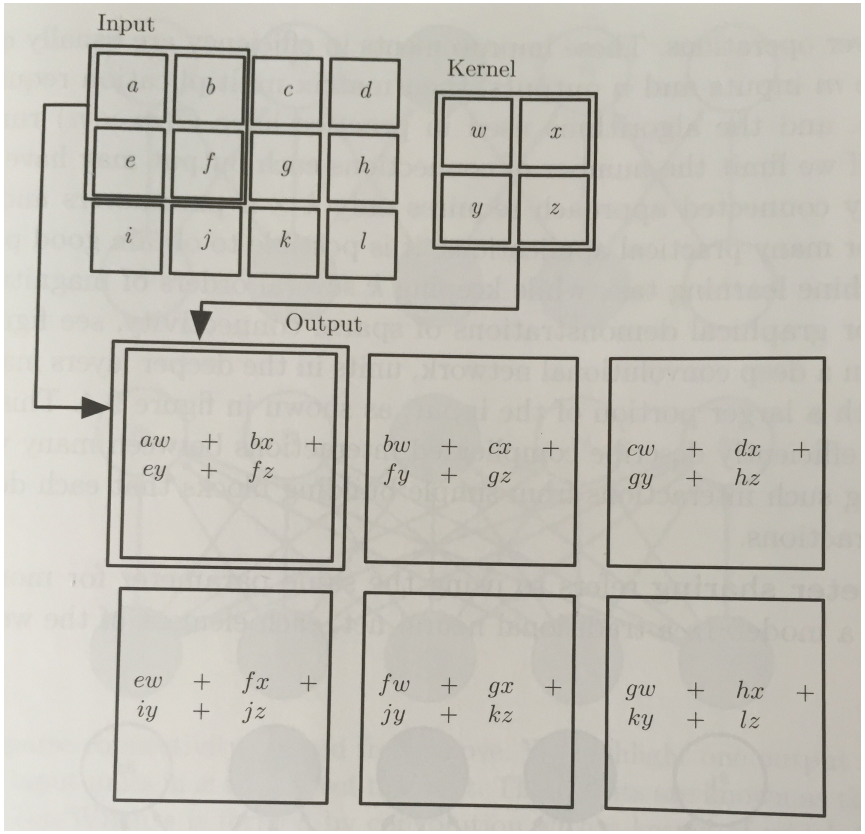


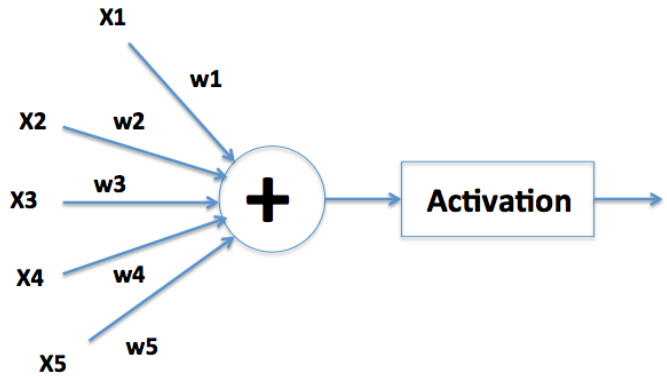
Figure: *Goodfellow et al 2016*

Effect of convolution

Figure: *Image courtesy LeCun 2014.*

Nonlinearity

- Achieved by activation function
- Options
  - $Tanh(x)$
  - Sigmoid function $1/(1+exp(-x))$
  - Rectified linear unit (ReLU).



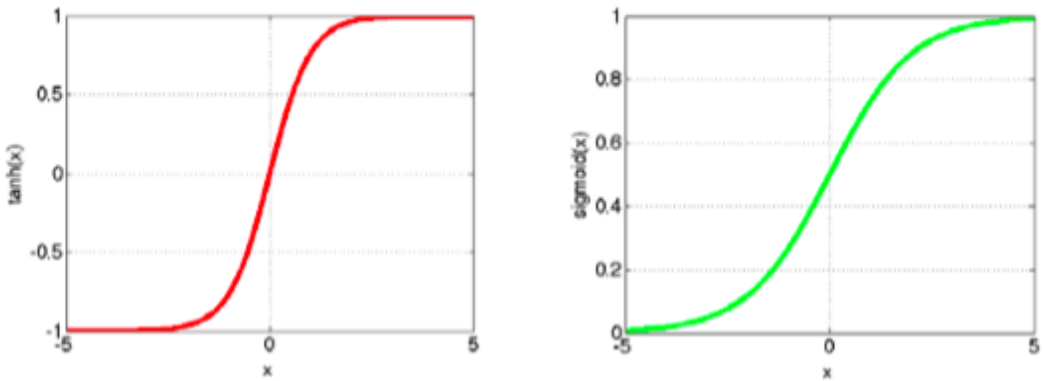


Figure: *Image courtesy LeCun 2014.*

ReLU

- Preferred option for nonlinearity
- Desirable properties
  - Easy to optimize with gradient-based methods
    - Derivative 0 when inactive and constant when active
  - Good generalization ability as linear models
- "single most important factor in progress" (Jarrett 2009).

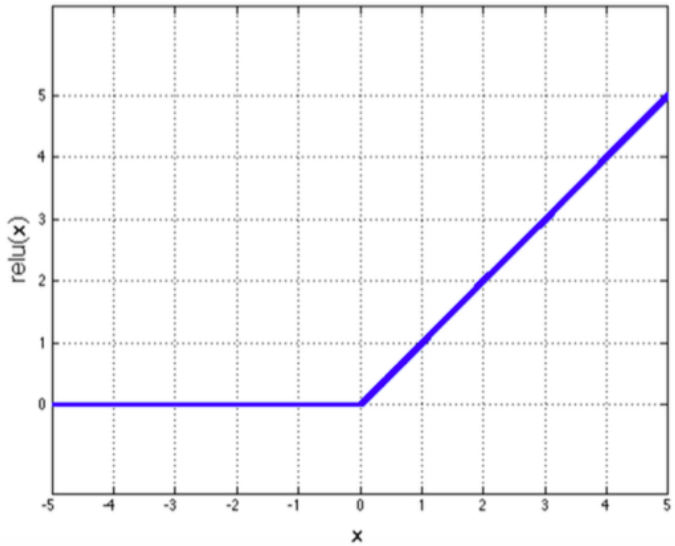


Figure: *Image courtesy LeCun 2014.*

Pooling

- Replace the output of certain nodes
  - With summary statistics of neighbors
- Pooling functions
  - Max pooling (Zhou and Chellappa 1988)
    - Maximum of outputs within a neighborhood
  - (Weighted) average, L^2-norm etc
- Property
  - Invariance to small transformations in inputs
  - Larger receptive fields (see larger scope in inputs)
  - Possible for downsampling
    - # pooling units $\leq$ # detector units
    - Improves computational efficiency
  - Essential in handling images of varying sizes.

Illustration of max pooling

- Sensitive to a neighborhood but not individual neighbors
  - Eg 1: pooling region *width* 3 and a *stride* of one pixel
  - Eg 2: same width but with downsampling.

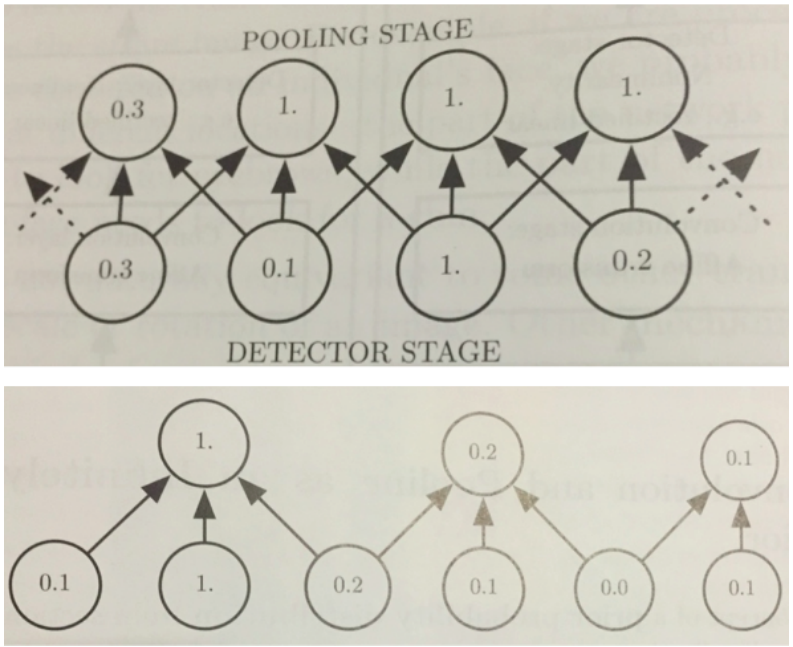


Figure: *Image courtesy Goodfellow et al 2016.*

CNN pooling example

- Invariance to transformations
  - Depending on how the detector units are designed.

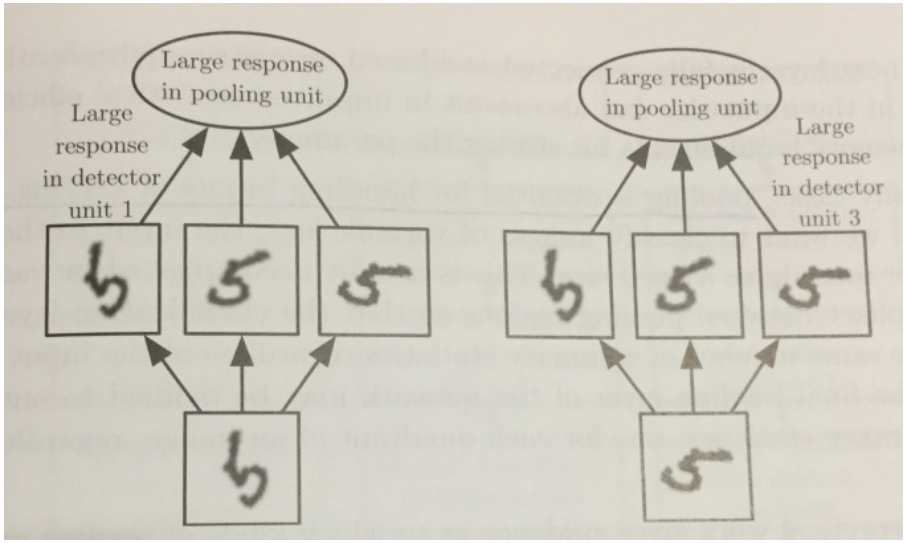


Figure: *Image courtesy Goodfellow et al 2016.*

CNN normalization

- Contrast normalization being the most common
  - To reduce variations due to contrast
- Usually refers to standardization or its extension
  - Global: over entire image
  - Local: small image window

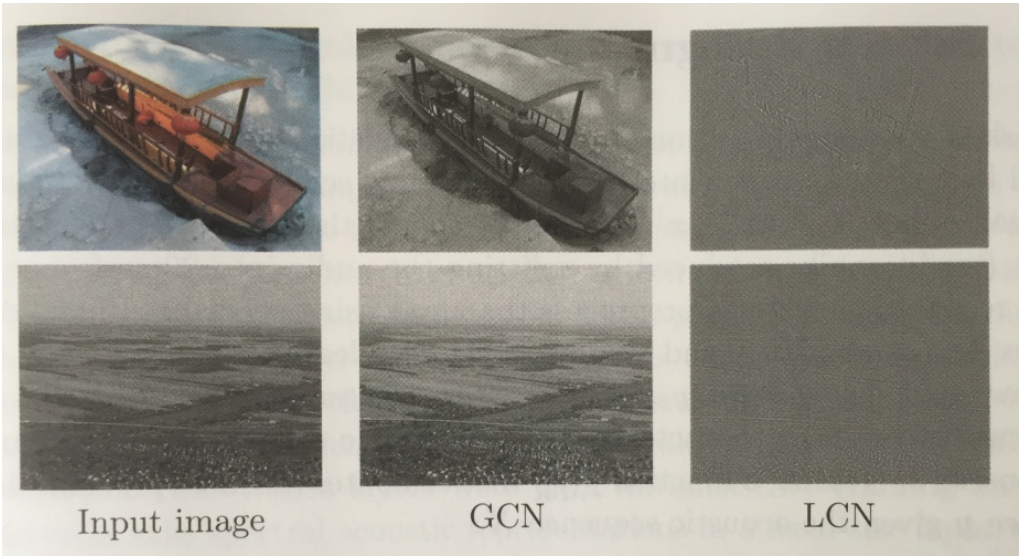


Figure: *Image courtesy Goodfellow et al 2016.*

Zero paddings in CNN

- Representation shrinks quickly without zero padding
  - # layers of representations limited
  - Padding makes it possible to build deep networks.

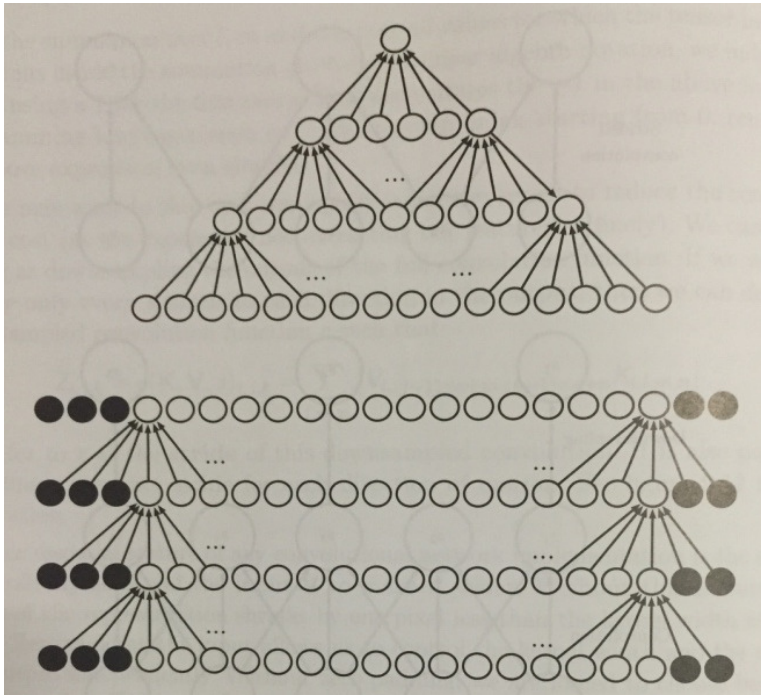


Figure: *Image courtesy Goodfellow et al 2016.*

CNN Vs locally connected NN

- Both substantially fewer connections than a full net
  - Parameter sharing in CNN
  - No parameter sharing in *locally connected* NN
    - So a lot more parameters than CNN.

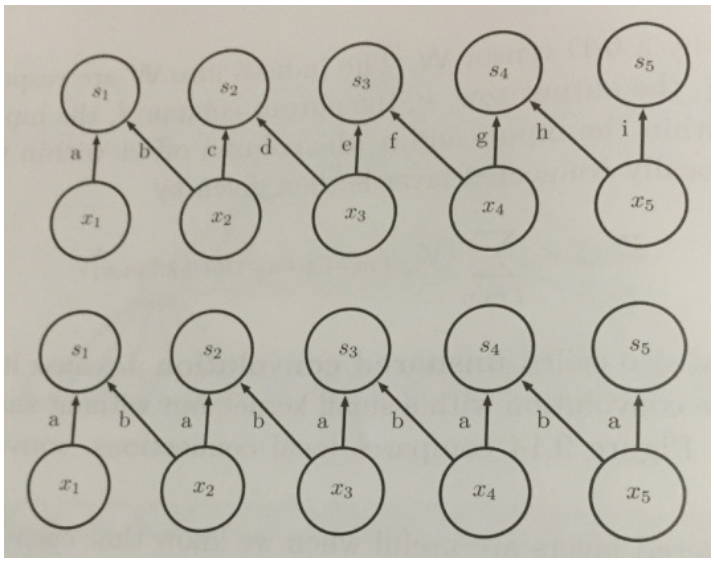


Figure: *Image courtesy Goodfellow et al 2016.*

Benchmark results

- MNIST handwritten digits
  - 60000 training + 10000 test images
  - 0.17% error rate (Ciresan et al 2011)
- ImageNet
  - 14m labeled images, 20k classes
  - Human labels via Amazon Mechanical Turk
  - Challenge: 1.2m training images, 1000 classes.

[Deng et al. CVPR 2009]

ConvNet (Krizhevsky, Sutskever and Hinton 2012)

- Similar to LeNet-5 (LeCun 1998)
  - 7 hidden layers, 650000 units, 60m parameters
  - GPU implementation (50x speedup over CPU)
  - Regularization with DropOut
  - Top-5 error rate 16.4%.

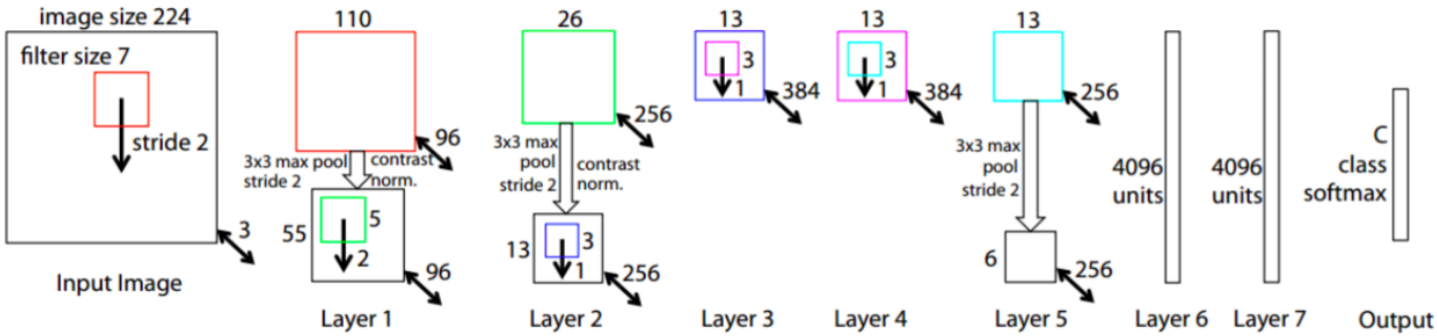


Figure 3. Architecture of our 8 layer convnet model. A 224 by 224 crop of an image (with 3 color planes) is presented as the input. This is convolved with 96 different 1st layer filters (red), each of size 7 by 7, using a stride of 2 in both x and y. The resulting feature maps are then: (i) passed through a rectified linear function (not shown), (ii) pooled (max within 3x3 regions, using stride 2) and (iii) contrast normalized across feature maps to give 96 different 55 by 55 element feature maps. Similar operations are repeated in layers 2,3,4,5. The last two layers are fully connected, taking features from the top convolutional layer as input in vector form ($6 \cdot 6 \cdot 256 = 9216$ dimensions). The final layer is a C-way softmax function, C being the number of classes. All filters and feature maps are square in shape.

Figure: *Image courtesy Zeiler and Fergus (2013).*

How important is depth to *ConvNet*?

	Removing layers	# parameters
18.1%	None	60m
19.2%	7	44m
23.8%	6,7	10m
21.1%	3,4	59m
51.6%	3,4,6,7	—

Table: *Top-5 error rate on ImageNet.*

- For each data point
  - CNN first predicts class probability for each class
  - Then check if the 'true' label is one of the top 5 predictions
    - i.e., those with top-5 highest probabilities
- *Top-5 error rate* = Percentage 'true' label NOT among top-5.

The end

Thank you!