THE UNIVERSITY of ADELAIDE

Neural networks and Deep learning

Guosheng Lin

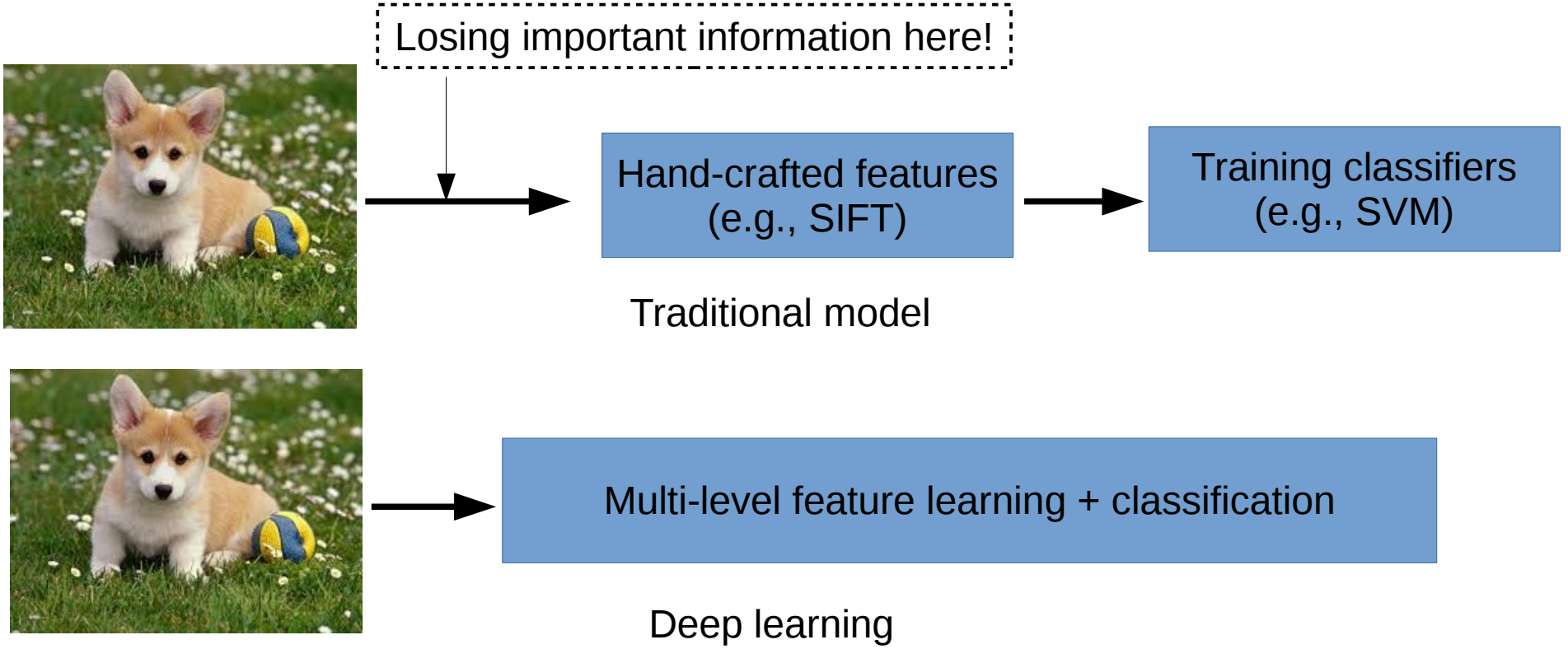
Australian Centre for Visual Technologies (ACVT)
School of Computer Science
Aug. 2015

What's deep learning

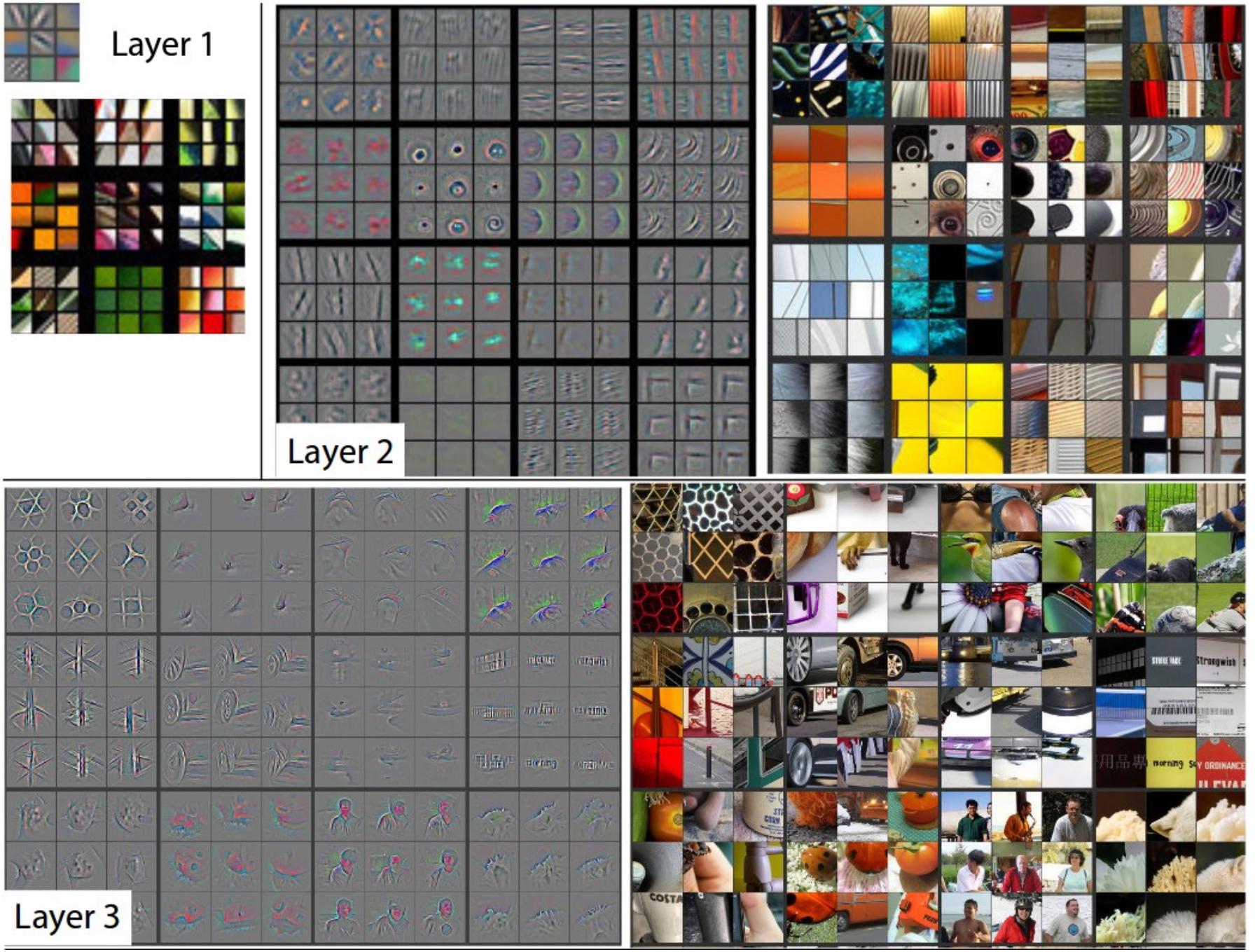
- https://en.wikipedia.org/wiki/Deep_learning
- Deep Learning is:
  - Deep neural networks (DNN)
    - Neural networks with many layers
  - Multi-level representation learning
    - Automatic feature learning, rather than using hand-crafted features
    - Hierarchical feature learning
  - Learning techniques for deep models
    - Efficient and effective deep model training.
      - RELU activation, Drop-out, training in GPUs
- Different types of network architectures
  - Convolutional neural networks, Stacked auto-encoders, Recurrent neural networks, Deep belief networks, Deep Boltzmann Machines …

Deep learning is representation learning

- Example: image classification
  - Traditional model for recognition
    - First extract hand-crafted features from images (e.g., SIFT, HOG, LBP)
    - Then train classifiers (e.g., SVM, boosting, …)
  - Deep learning for recognition
    - Automatic feature learning, Learning multi-level representation of images
    - end-to-end learning: joint learning of features and classifiers

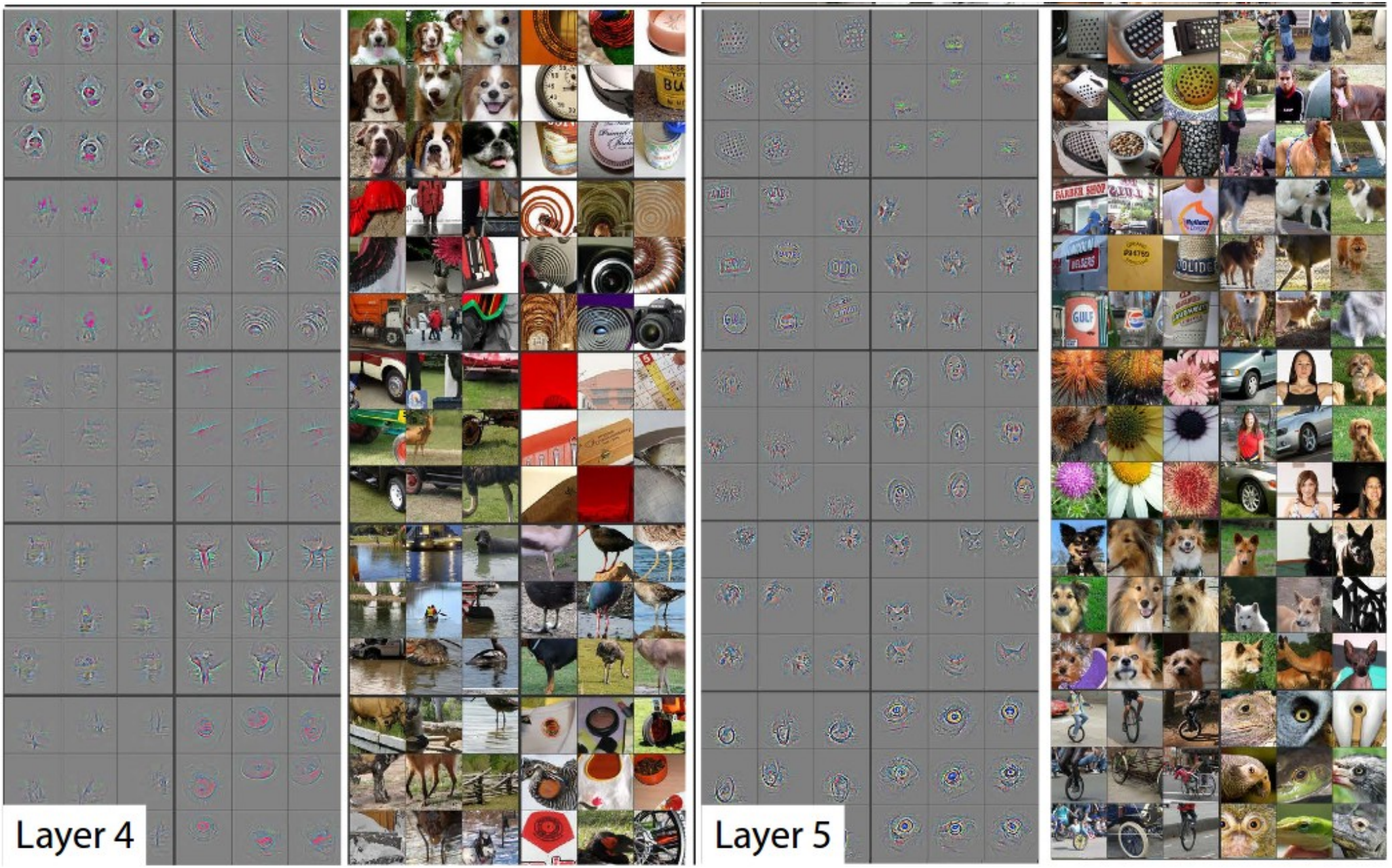


Multi-level features/representation



(Matthew D. Zeiler, Rob Fergus, “Visualizing and Understanding Convolutional Networks”, ECCV 2014)

Multi-level features/representation



(Matthew D. Zeiler, Rob Fergus, “Visualizing and Understanding Convolutional Networks”, ECCV 2014)

Large-scale image classification

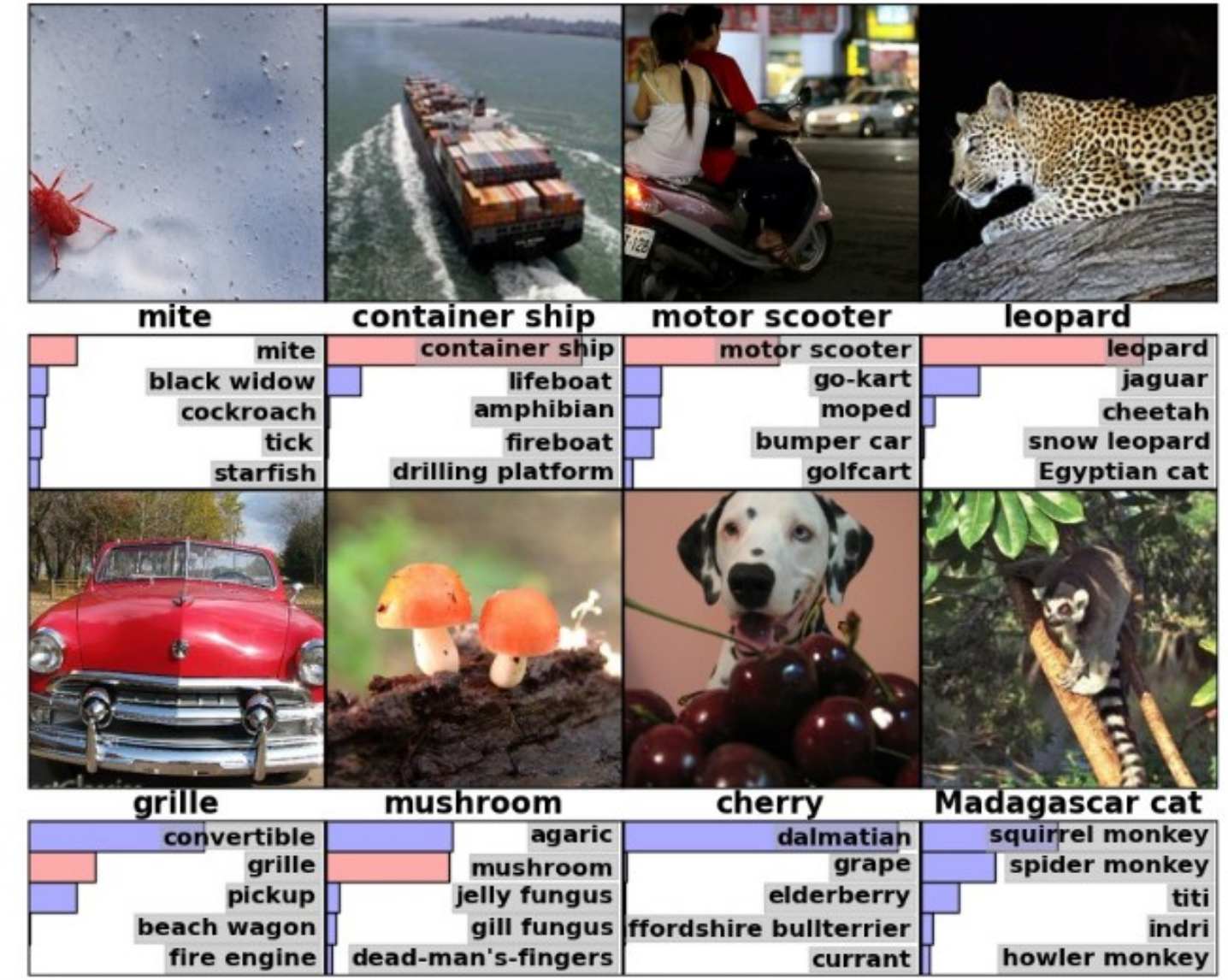
- Image-Net image classification competition:
  - 1000 categories, more than 1 million images
- Breakthrough in computer vision
  - Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton; “ImageNet Classification with Deep Convolutional Neural Networks”, NIPS 2012
  - Convolutional neural network (CNN)
    - LeNet (Yann LeCun etc. 1998)

Rank	Name	Error rate	Description
1	**U. Toronto**	0.15315	Deep learning
2	U. Tokyo	0.26172	Hand-crafted features and learning models. Bottleneck.
3	U. Oxford	0.26979	
4	Xerox/INRIA	0.27058	

Object recognition over 1,000,000 images and 1,000 categories (2 GPU)

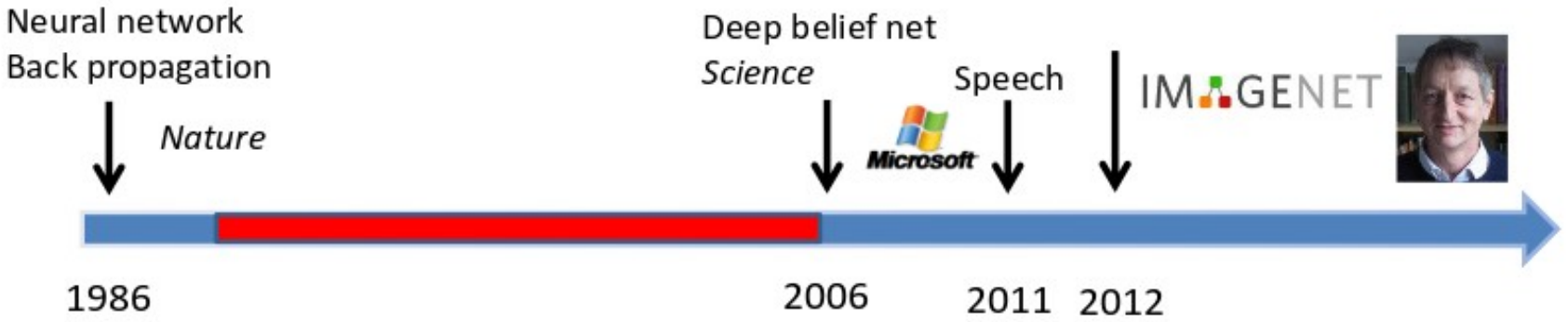
(Image-Net classification challenge ILSVRC 2012 , Table from Xiaogang Wang)

Image-Net classification prediction examples



(Figure from Alex Krizhevsky)

The Deep Learning Revolution



(Figure from Xiaogang Wang, researcher in the figure: Prof. Geoffrey Hinton)

- A breakthrough in machine learning
  - Currently the most popular machine learning technique
  - Set new performance records in
    - Speech recognition
    - Natural language processing
    - Computer vision
      - Image classification, Object detection, Semantic segmentation, Recognition in videos, etc.
- Actively explored in industry
  - Google (machine translation, image classification)

    Facebook (face recognition), Baidu (image retrieval)
  - Many start-up companies on deep learning applications

Resources

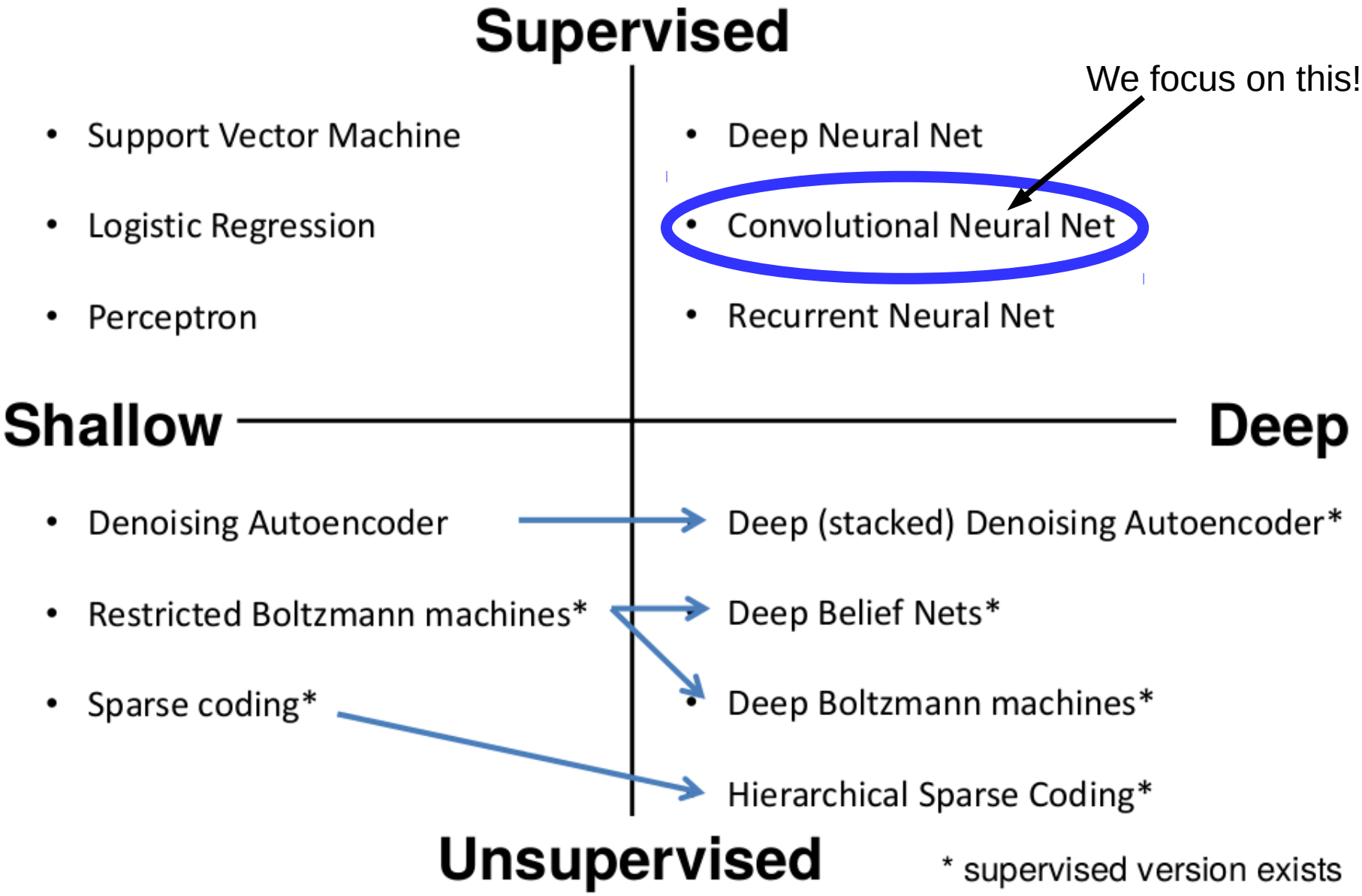
- http://deeplearning.net
- Research groups:
  - Geoff Hinton, Yoshua Bengio, Yann Lecun
  - Stanford, Oxford, Google, Microsoft, etc.
- Basics of neural networks:
  - http://ufldl.stanford.edu/tutorial/
- A comprehensive course in deep learning:
  - https://ift6266h15.wordpress.com/
    - Prof. Aaron Courville; Université de Montréal
- Book: DEEP LEARNING
  - Yoshua Bengio, Ian Goodfellow and Aaron Courville. MIT Press (available online)
- Recent advance:
  - Deep Learning Summer School 2015

    https://sites.google.com/site/deeplearningsummerschool/schedule

Resources

- Toolbox
  - MatConvNet: a MATLAB based toolbox
    - http://www.vlfeat.org/matconvnet/
  - Caffe (Python, C++)
    - http://caffe.berkeleyvision.org/
  - Theano/Pylearn2 (Python)
    - http://www.deeplearning.net/software/theano/
  - Torch7 (Lua)
    - http://torch.ch/
  - Chainer (Python)
    - http://chainer.org/
  - Others...

Taxonomy of feature learning methods



(slide from Honglak Lee)

Outlines

- Basics of neural networks
  - Reference:
    - http://ufldl.stanford.edu/tutorial/supervised/MultiLayerNeuralNetworks/
- Convolutional neural networks (CNNs)
  - translation invariance
- CNN for image classification:
  - AlexNet
    - Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton; “ImageNet Classification with Deep Convolutional Neural Networks”, NIPS 2012

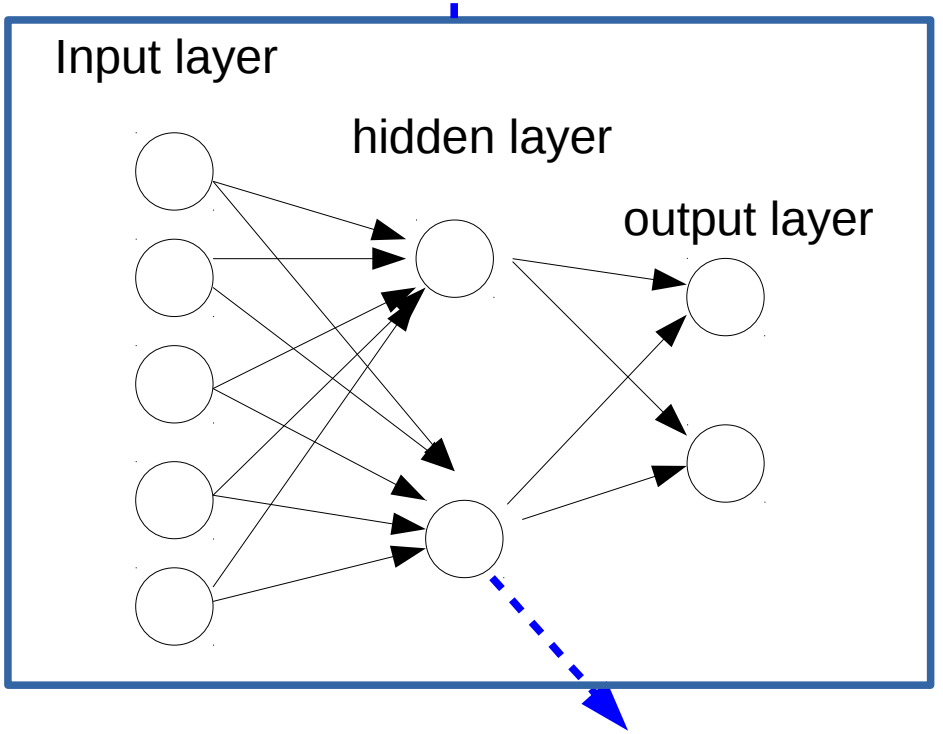
Multi-Layer Neural Network

Example: classification of 2 classes.
A network is a complex non-linear mapping function:

$$y = f(\boldsymbol{x})$$

Output (2-dimension)
Confidence values of 2 classes

Input (vector)



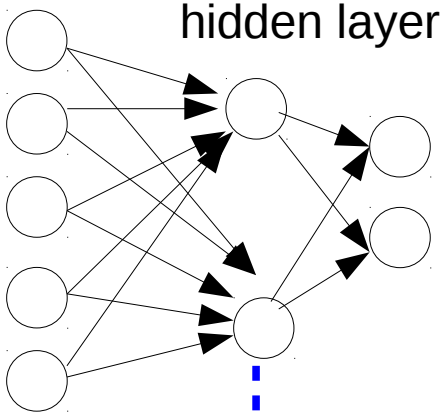
A 3-layer fully-connected network

A unit / neuron
(represents a non-linear operation)

Multi-Layer Neural Network

Input layer

hidden layer



A unit / neuron

3 key elements in a unit:

3. Activation function: RELU
or other functions (e.g., sigmoid, tanh)

$$\sigma(\boldsymbol{w}^\top \boldsymbol{x}) = \max(\boldsymbol{w}^\top \boldsymbol{x}, 0)$$

$$y = g(\boldsymbol{x}; \boldsymbol{w}) = \sigma(w_1 x_1 + w_2 x_2 + \dots)$$

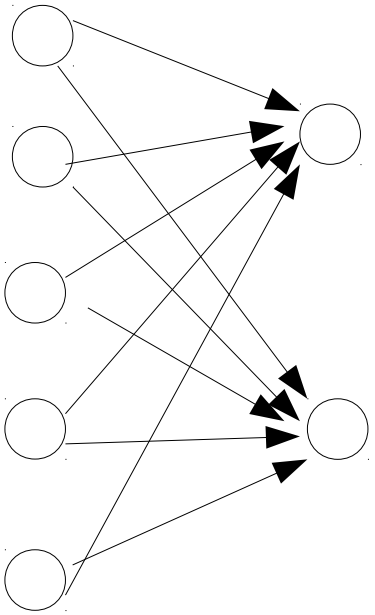
1. connections:
identify the input nodes
from previous layer

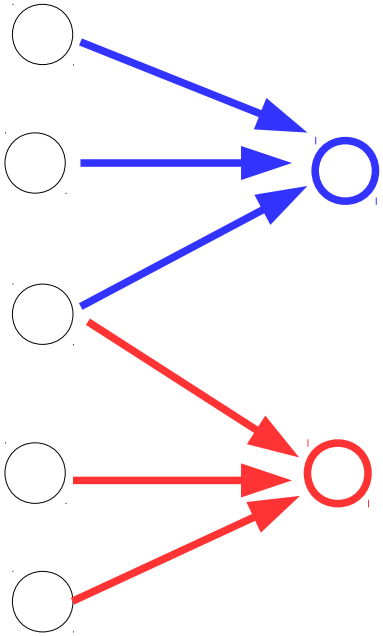
2. Model parameter, need to learn

Convolutional Neural Network (1D)

Note: one edge corresponds to one weight.

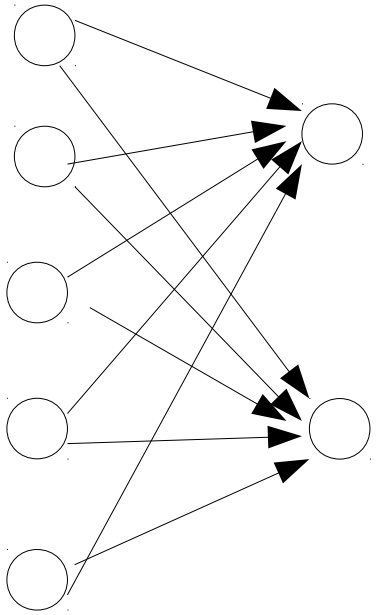
Fully connected net ⟵ VS ⟶ Convolutional neural net
(1-Dimensional convolution)



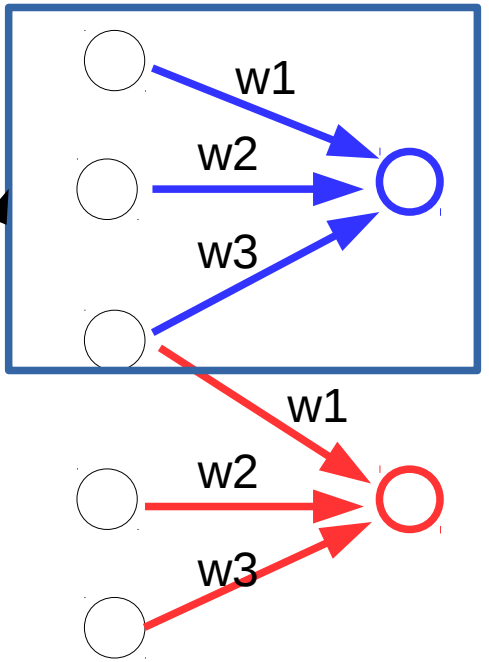


Convolutional Neural Network (1D)

Fully connected net ⟵ VS ⟶ Convolutional neural net
(1-Dimensional convolution)



Number of weights: 10
(model parameter)

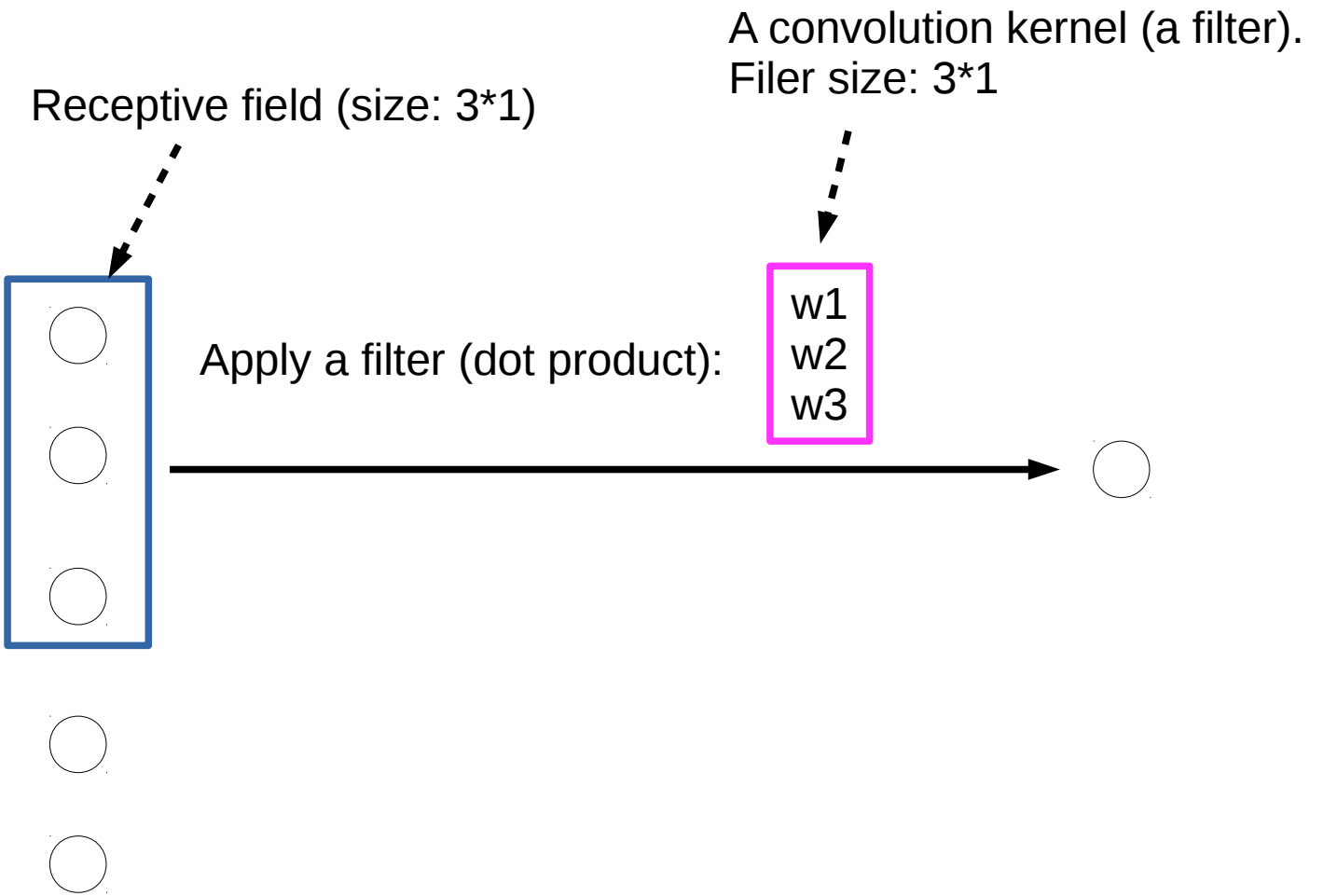


Number of weights: 3

CNN layer properties:
1. local connectivity (spatially-local connection)
2. sharing weights
   (red and blue indicate two groups of edges
   which use the same set of weights)

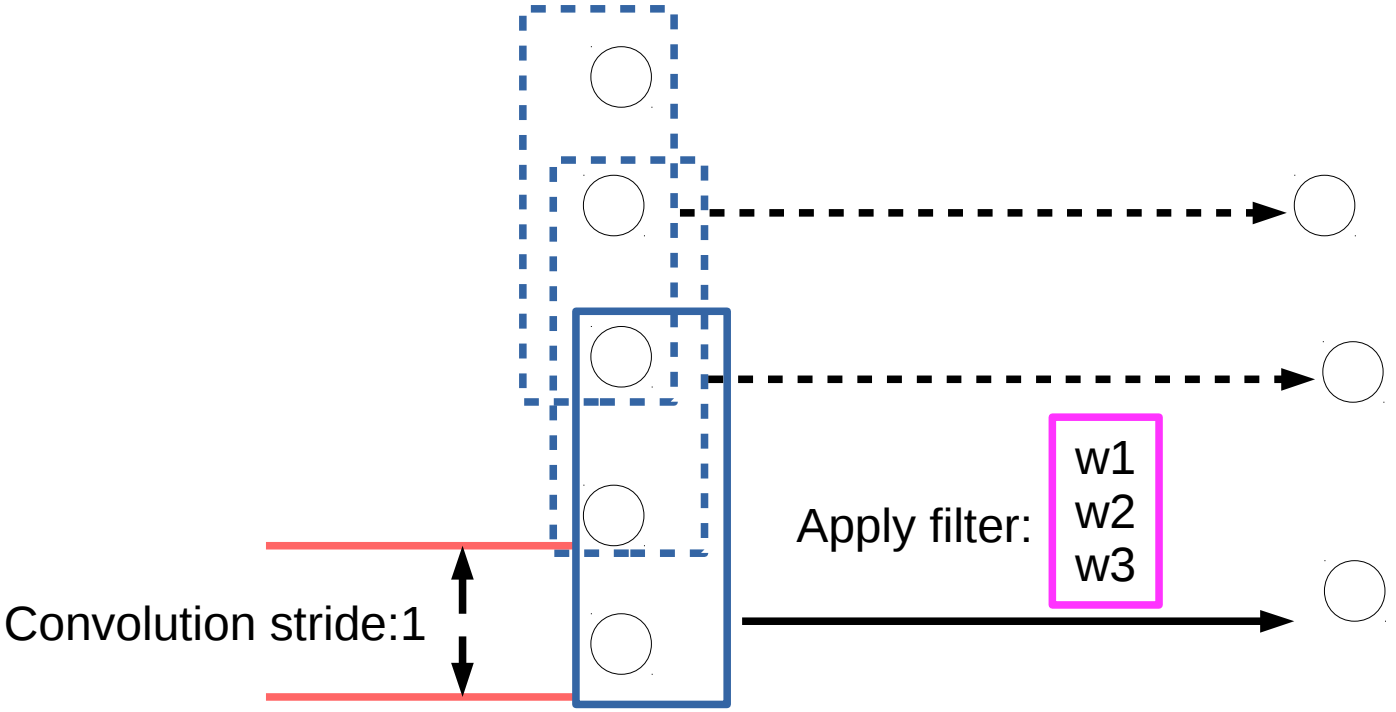
Convolutional Neural Network (1D)

CNN layer in another aspect: filtering/convolutional operation.
Apply a filer in all spatial locations to generate outputs:



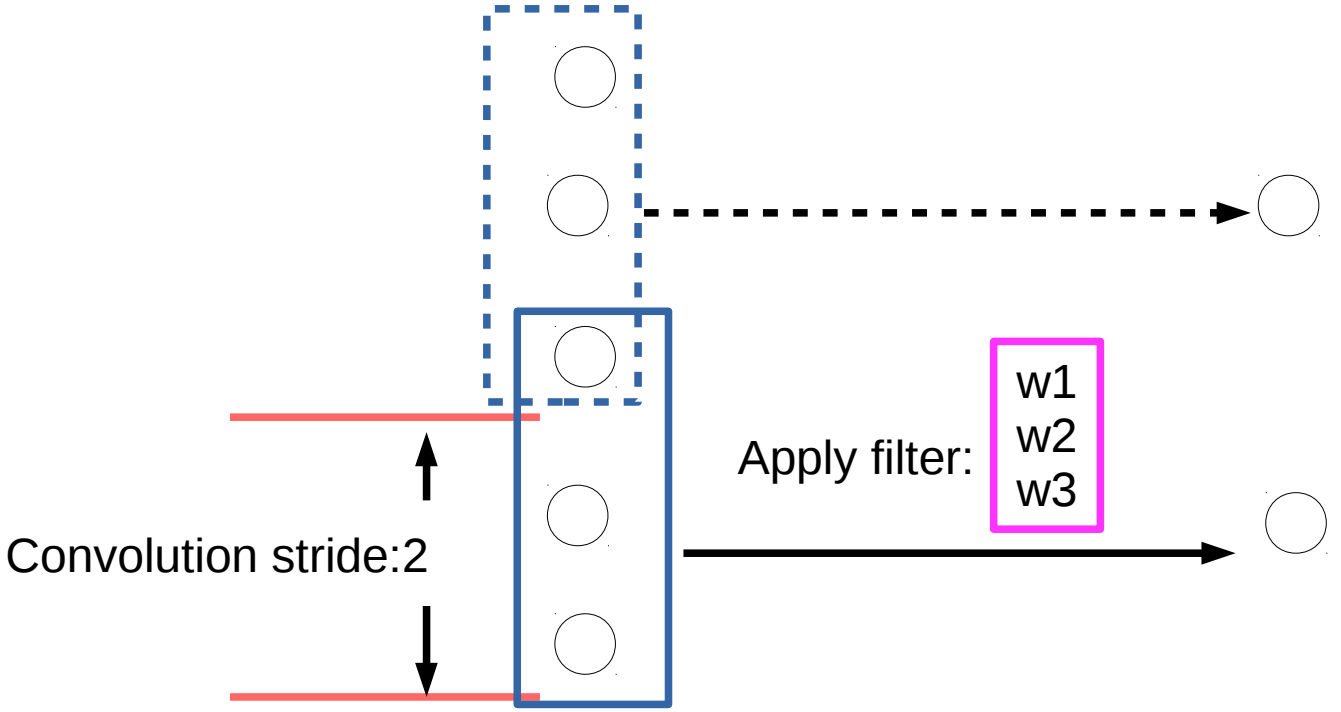
Convolutional Neural Network

Convolution (stride=1)



Convolutional Neural Network

Convolution (stride=2)

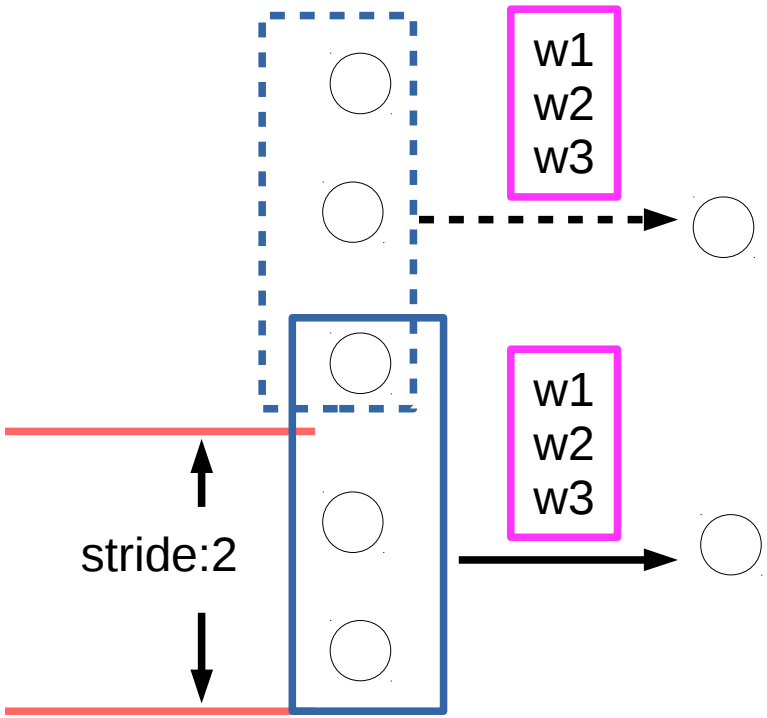


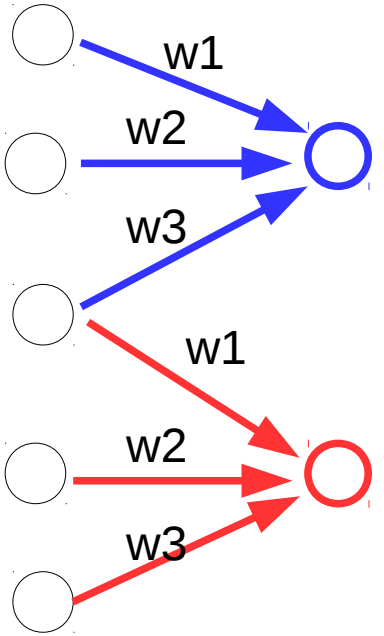
Convolutional Neural Network

Different aspects of
understanding CNN layers

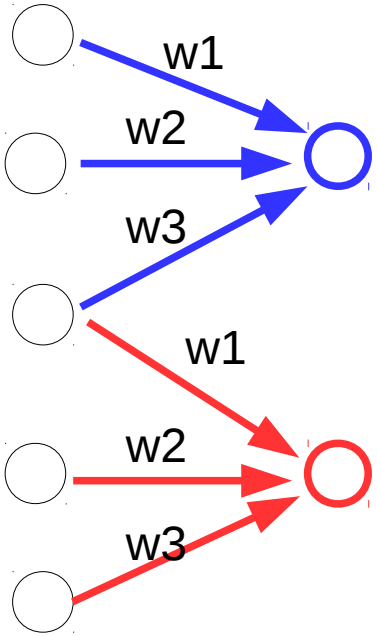
Apply filers.
Convolution (stride=2)

Local connection,
Share weights:

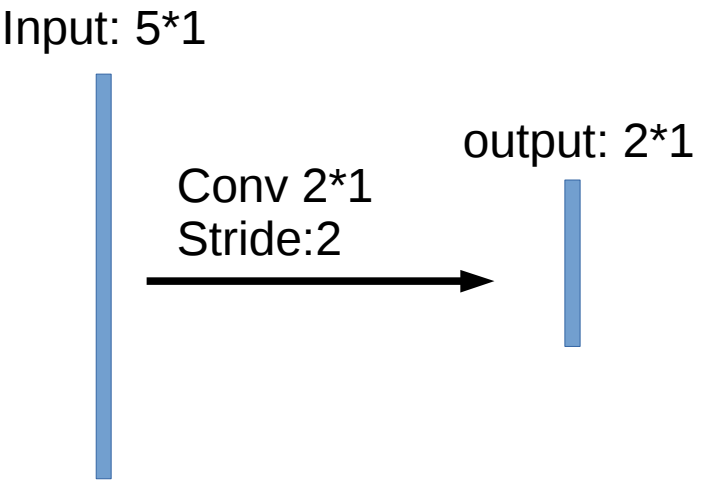




Convolutional Neural Network



A compact way to describe CNN layers



CNN 2D Convolution

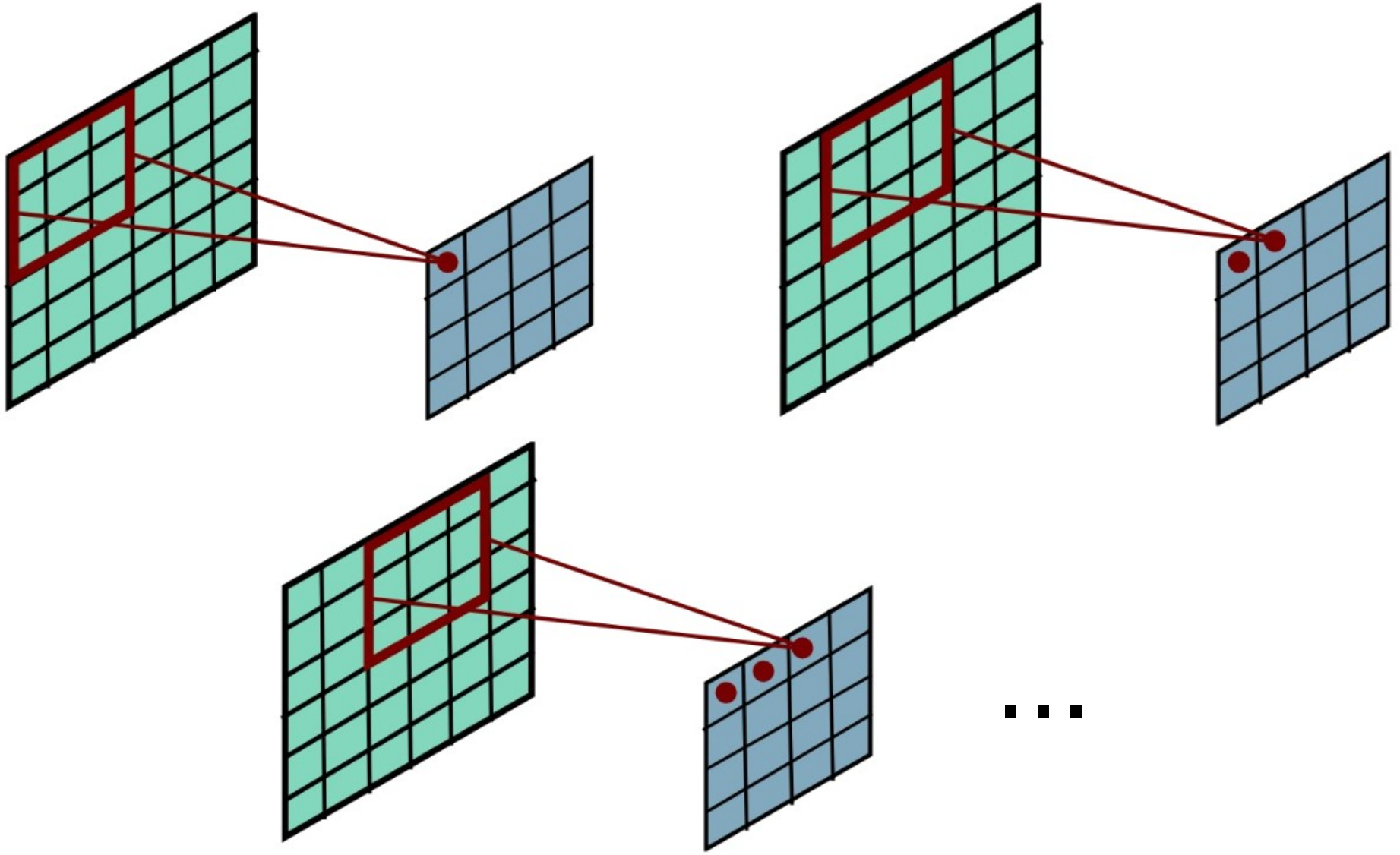
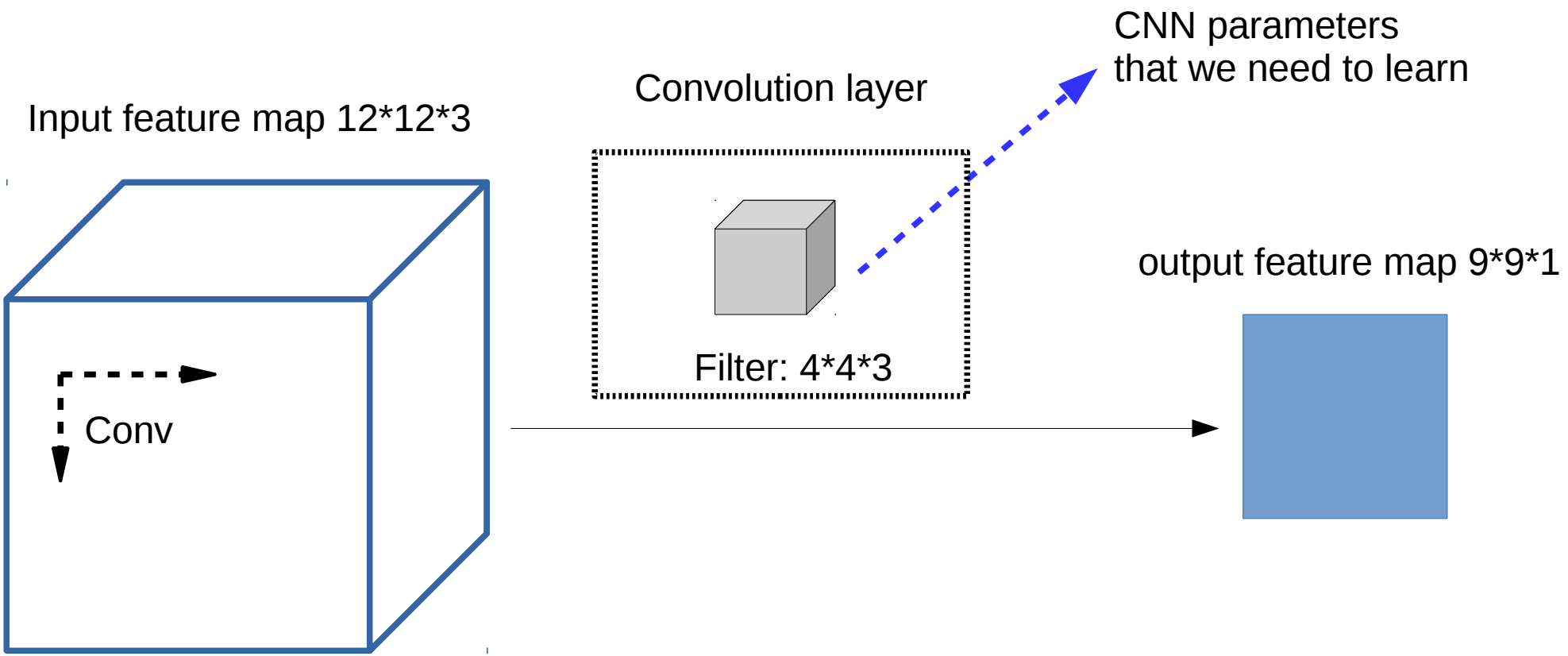


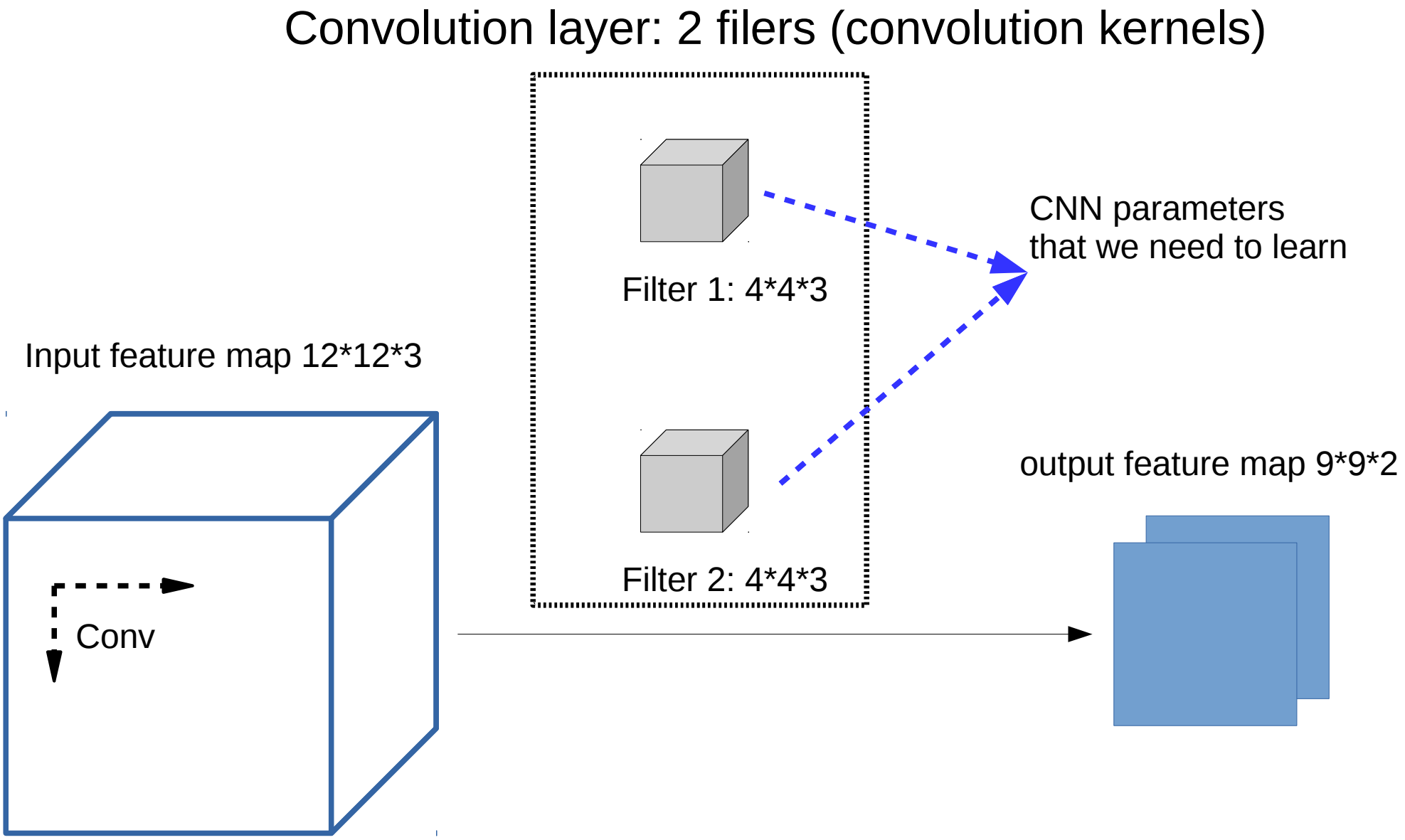
Figure from Marc'Aurelio Ranzato

CNN 2D Convolution



Receptive field size (2D size of the filter): 4*4

CNN layers for 2D Convolution



Large-scale image classification

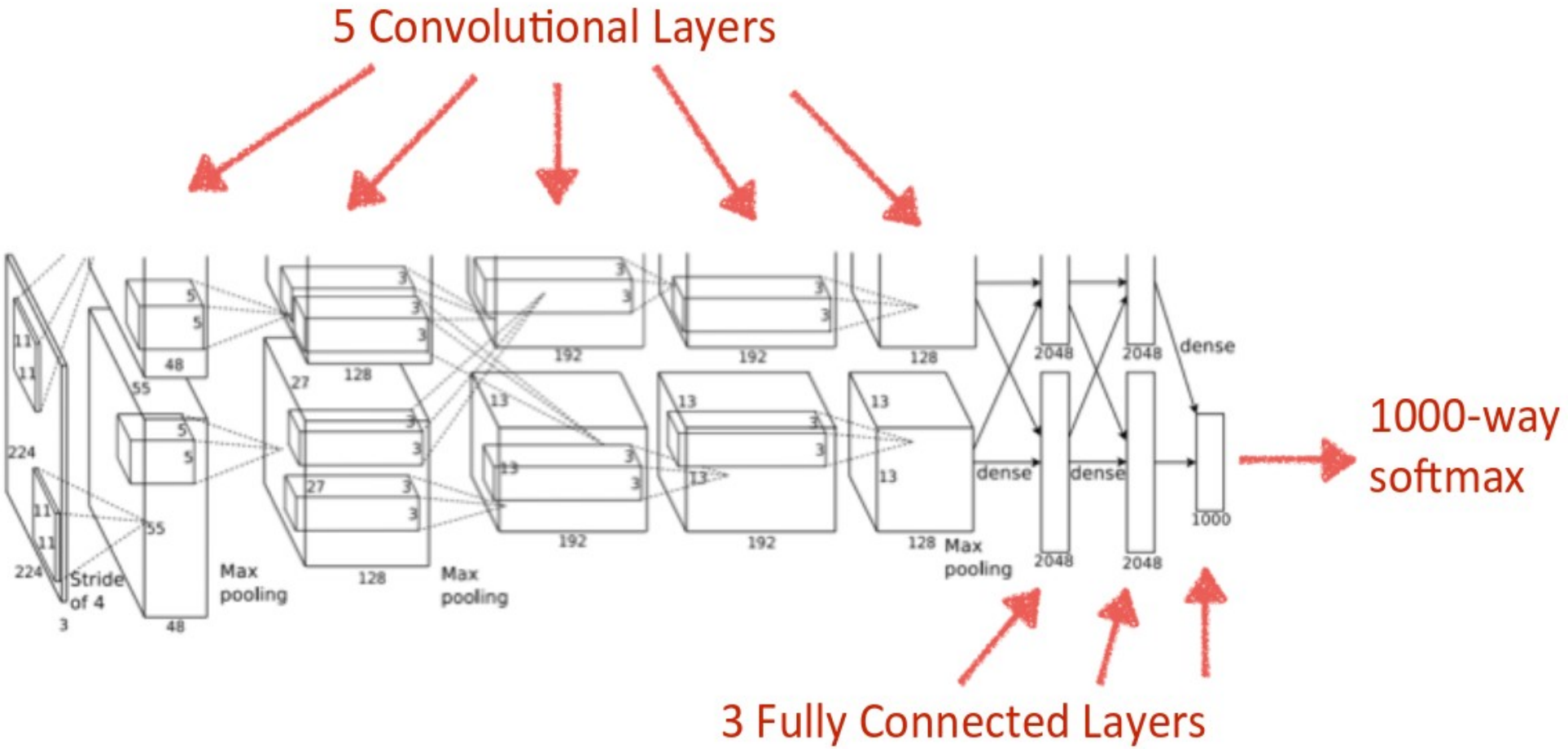
- Image-Net image classification competition:
  - 1000 categories, more than 1 million images
- Breakthrough in computer vision
  - Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton; “ImageNet Classification with Deep Convolutional Neural Networks”, NIPS 2012

Rank	Name	Error rate	Description
1	**U. Toronto**	0.15315	Deep learning
2	U. Tokyo	0.26172	Hand-crafted features and learning models. Bottleneck.
3	U. Oxford	0.26979	
4	Xerox/INRIA	0.27058	

Object recognition over 1,000,000 images and 1,000 categories (2 GPU)

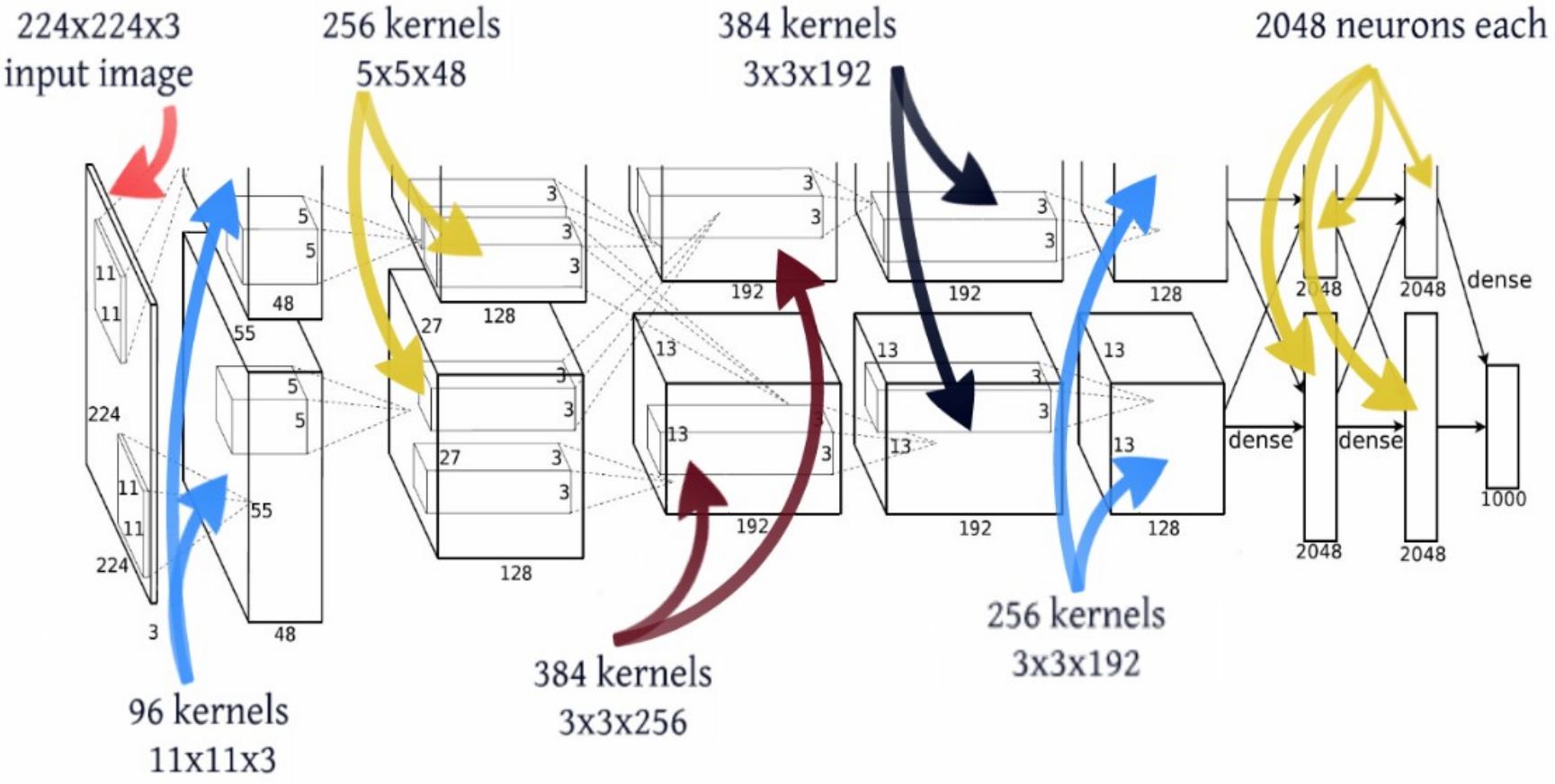
(Image-Net classification challenge ILSVRC 2012 , Table from Xiaogang Wang)

AlexNet



(Figure from Tugce Tasci and Kyunghee Kim)

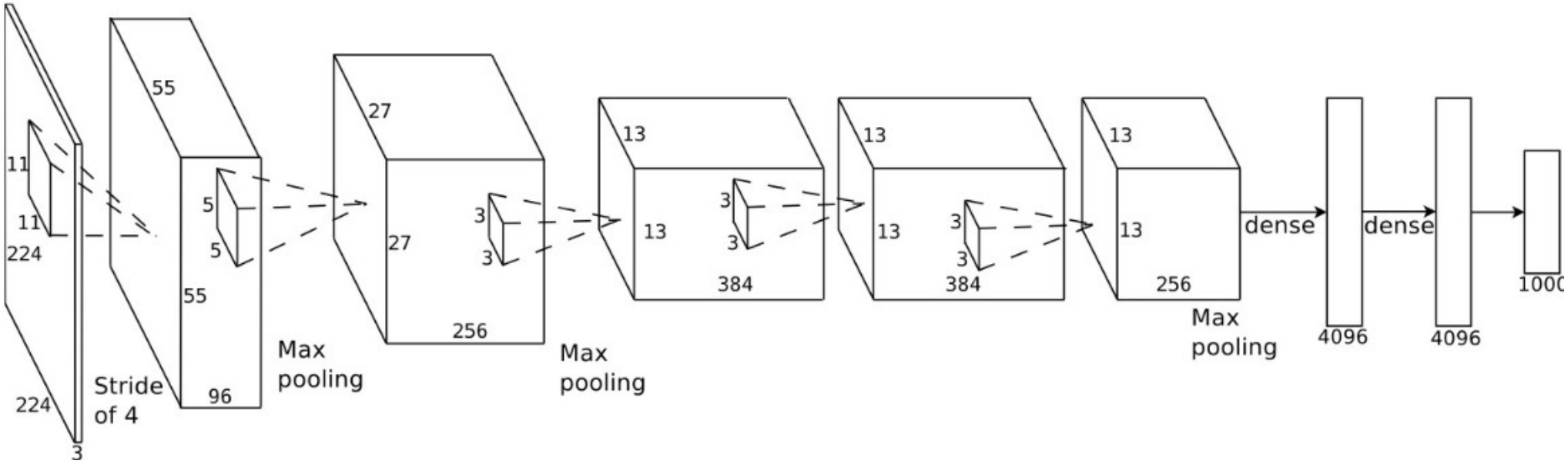
AlexNet



(Figure from Tugce Tasci and Kyunghee Kim)

AlexNet

Architecture for using one GPU
(the actual architecture used in CNN toolboxes)



(Figure from Alex Krizhevsky)

AlexNet

- 8 layers with weights
  - parameters need to learn
  - 60 Million parameters
- Key components:
  - Convolutional layers
    - filters (convolution kernel)
    - Parameters (weights) in filters need to learn
  - Rectified Linear Units (RELU)
  - Overlapping pooling (sliding pooling)
  - Dropout

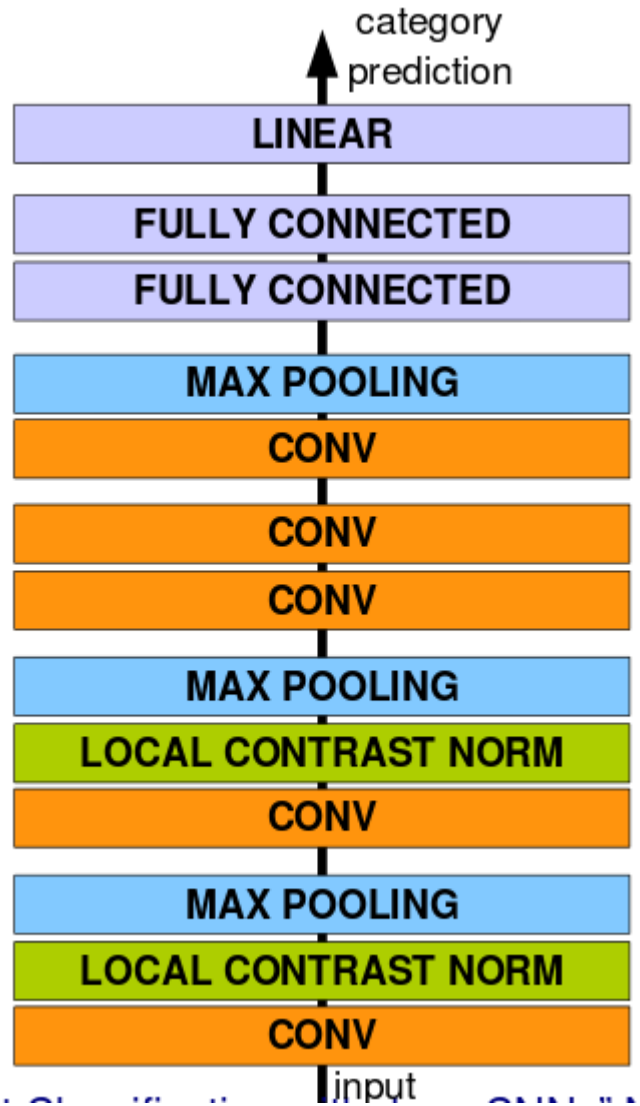
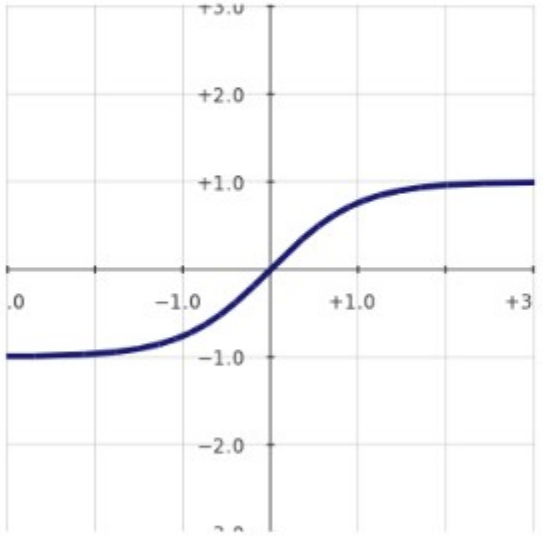


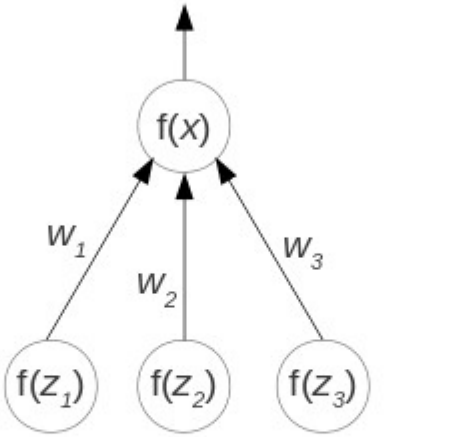
Figure from Marc'Aurelio Ranzato

RELU activation

Neurons

f(x) = tanh(x)

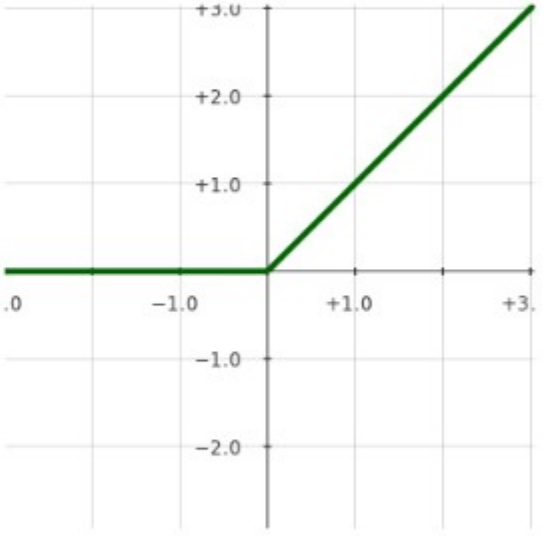




$x = w_1 f(z_1) + w_2 f(z_2) + w_3 f(z_3)$

x is called the total input to the neuron, and f(x) is its output

f(x) = max(0, x)



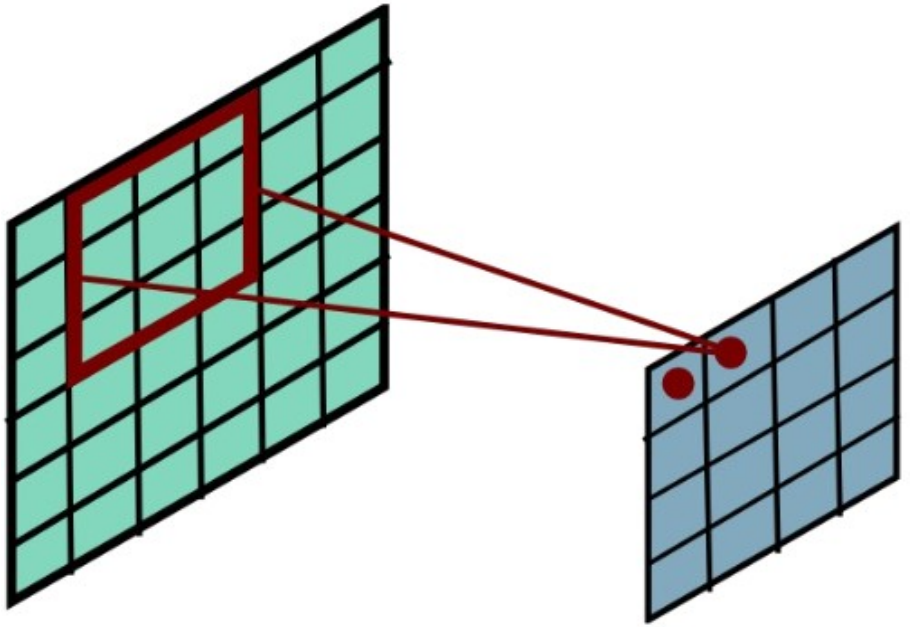
Very bad (slow to train)

Very good (quick to train)

(Slide from Alex Krizhevsky)

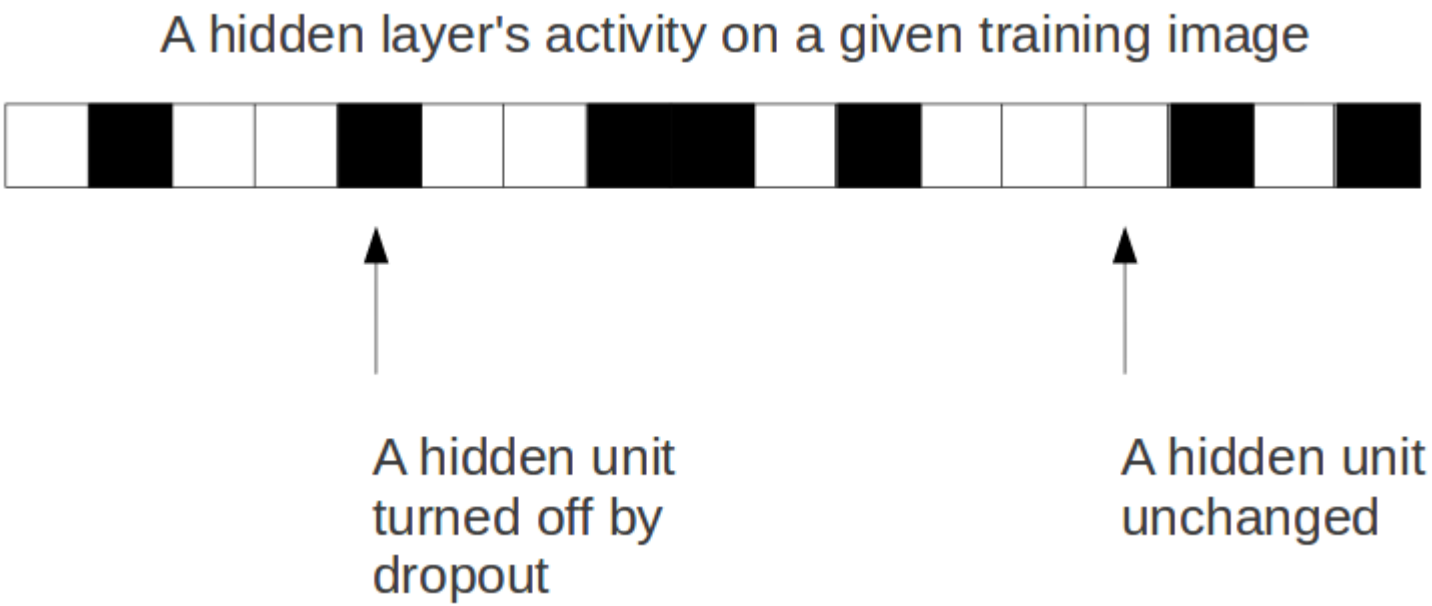
Overlapping pooling

- Overlapping pooling (sliding pooling)
  - Perform pooling in sliding windows (convolutional)
  - Invariant to small transformations
  - Max/average pooling



Dropout

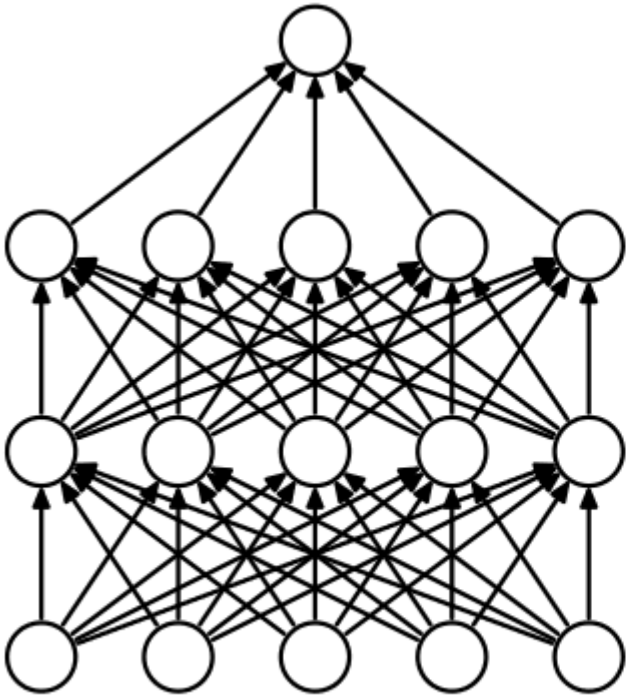
- Apply in the training stage
- Reducing over-fitting
  - Better generalisation performance
- Theoretical analysis: related to model averaging
- Dropout: set the unit output to zero with probability 0.5
- Apply to last two fully connected layers with 4096-dim output



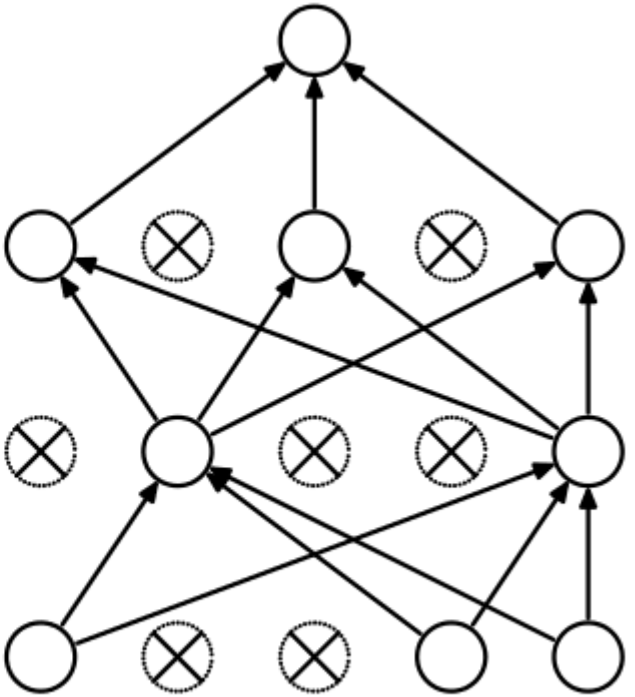
(Slide from Alex Krizhevsky)

Dropout

- Paper: Nitish Srivastava at el. “Dropout: A Simple Way to Prevent Neural Networks from overfitting” JMLR 2014



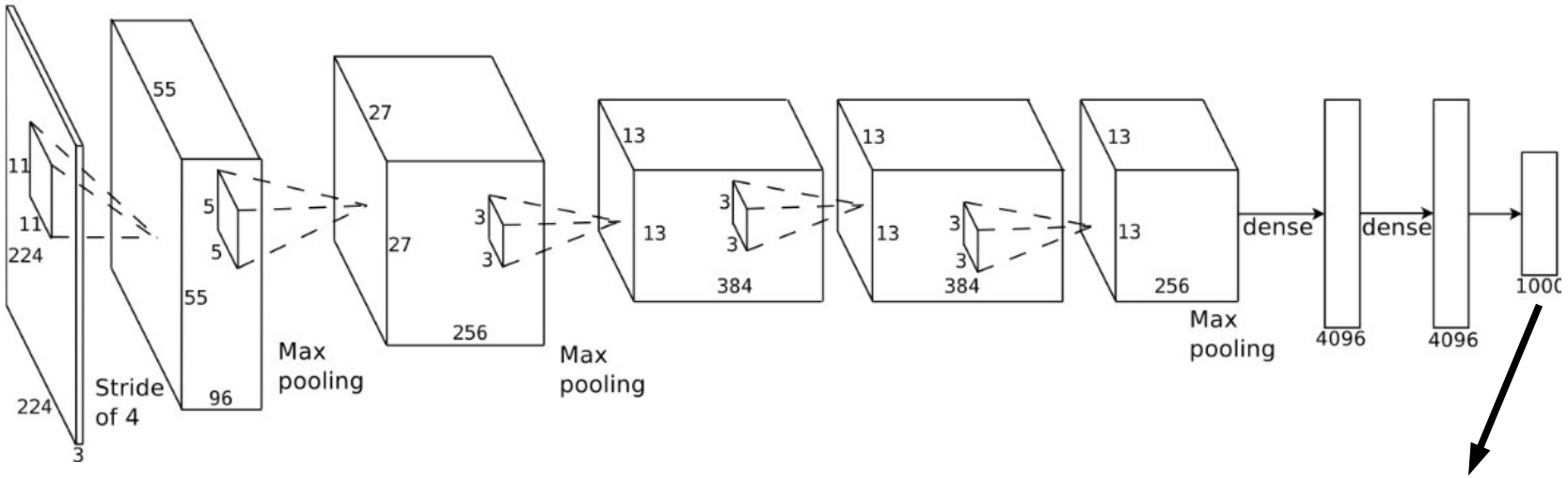
(a) Standard Neural Net



(b) After applying dropout.

Learning CNNs

Learning deep CNNs is very challenging:
Highly non-convex, large number of parameters, large-scale training data.



Add a softmax loss layer (log-loss, cross-entropy loss) for multi-class classification

Reference:
http://ufldl.stanford.edu/tutorial/supervised/SoftmaxRegression/

Softmax loss

x: the output of last layer (1000 dimensions)

Softmax function: $$\sigma_j(\boldsymbol{x}) = \frac{e^{\boldsymbol{w}_j^\top \boldsymbol{x}}}{\sum_{k=1}^{K} e^{\boldsymbol{w}_k^\top \boldsymbol{x}}}.$$

Loss function (W is the filter parameters need to learn):

$$J(\boldsymbol{W}) = -\frac{1}{m}\left[\sum_{i=1}^{M}\sum_{j=1}^{K} 1\{y^{(i)} = j\} \log \sigma_j(\boldsymbol{x})\right] + \frac{\lambda}{2}\|\boldsymbol{W}\|^2$$

$$= E(\boldsymbol{W}) + + \frac{\lambda}{2}\|\boldsymbol{W}\|^2$$

CNN learning with SGD

- Training CNN using stochastic gradient decent (SGD)
  - Mini-batch : a small number of examples for one gradient update
  - Momentum (avoid gradient fluctuation)
  - Calculate gradients of all layers using chain rule (back-propagation)

- Parameter update (t-th iteration of SGD):

$$\boldsymbol{W}(t+1) = \boldsymbol{W}(t) + \Delta\boldsymbol{W}(t)$$

$$\Delta\boldsymbol{W}(t) = -\eta\frac{\partial J(\boldsymbol{W})}{\partial \boldsymbol{W}} + \alpha\Delta\boldsymbol{W}(t-1)$$

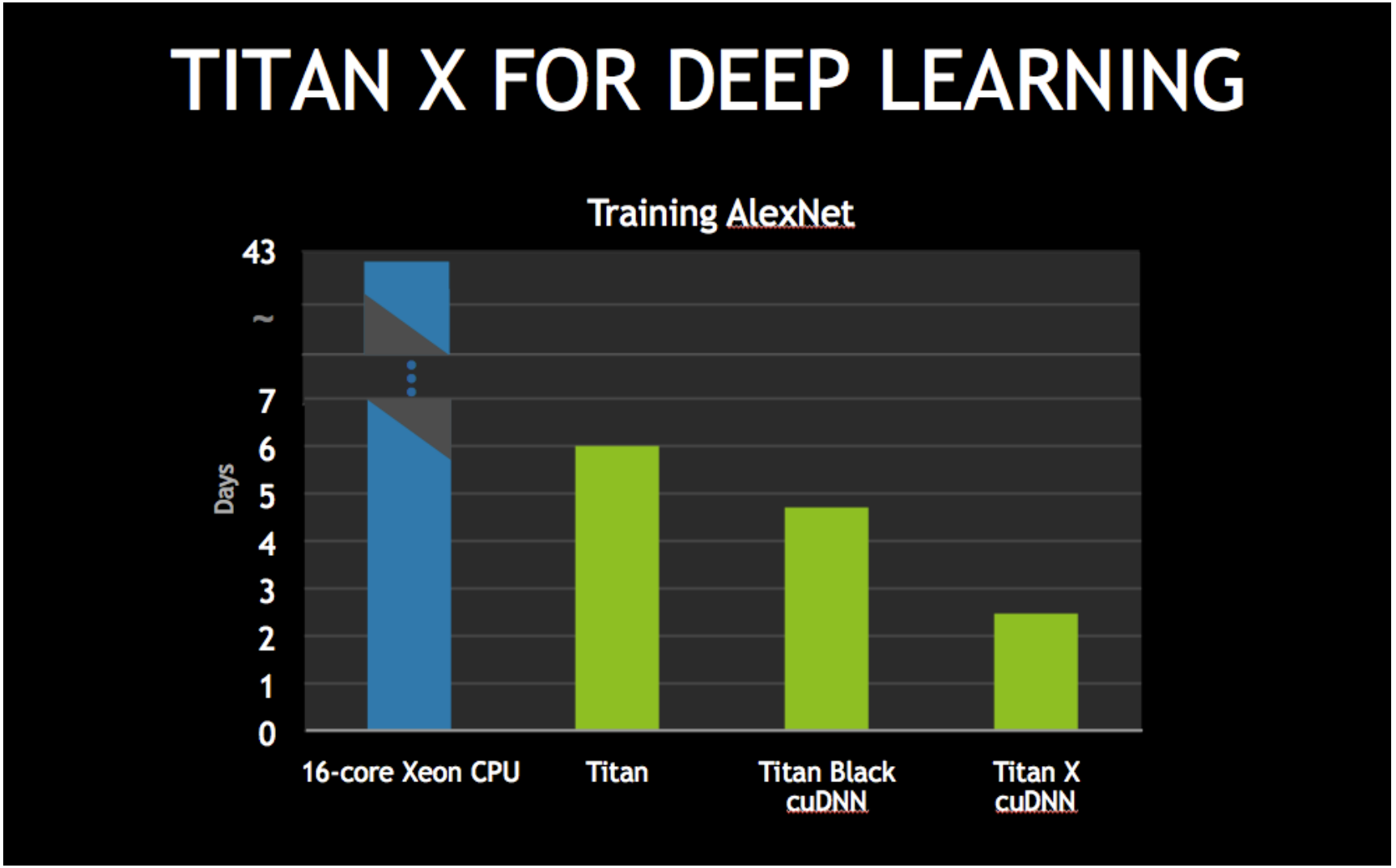
$$= -\eta\frac{\partial E(\boldsymbol{W})}{\partial \boldsymbol{W}} - \eta\lambda\boldsymbol{W} + \alpha\Delta\boldsymbol{W}(t-1).$$

Learning rate

Weight decay

momentum

AlexNet



Training time for AlexNet on ILSVRC2012 (figure from Nvidia)