# Introduction to Deep Neural Networks

**Speaker:**

Francisco Erivaldo Fernandes Junior
Ph.D. Candidate in Electrical Engineering
feferna@okstate.edu

**Instructor:**

Dr. Gary G. Yen
Regents Professor
gyen@okstate.edu

**School of Electrical and Computer Engineering**
**Oklahoma State University**

October 29, 2019

# Introduction and Motivation

# Brief History – First Developments and AI Winter

- **McCulloh and Pitts (1943):** First mathematical description of how neuron cells might work.
- **Hebb (1949):** First description of learning in animal brains.
- **Widrow and Hoff (1960):** First application of neural networks to solve a real world problem.
- **Widrow and Hoff (1962):** Development of the first learning algorithm for use with neural networks.
- **Minksy (1969):** Presented rigorous mathematical analysis showing that early neural networks could not solve linearly separable problems, such as a XOR function.
- **Between 1969 and 1982:** The interest in neural networks quickly died off due to limitations in the types of problems they could solve. This period is sometimes called the AI winter.

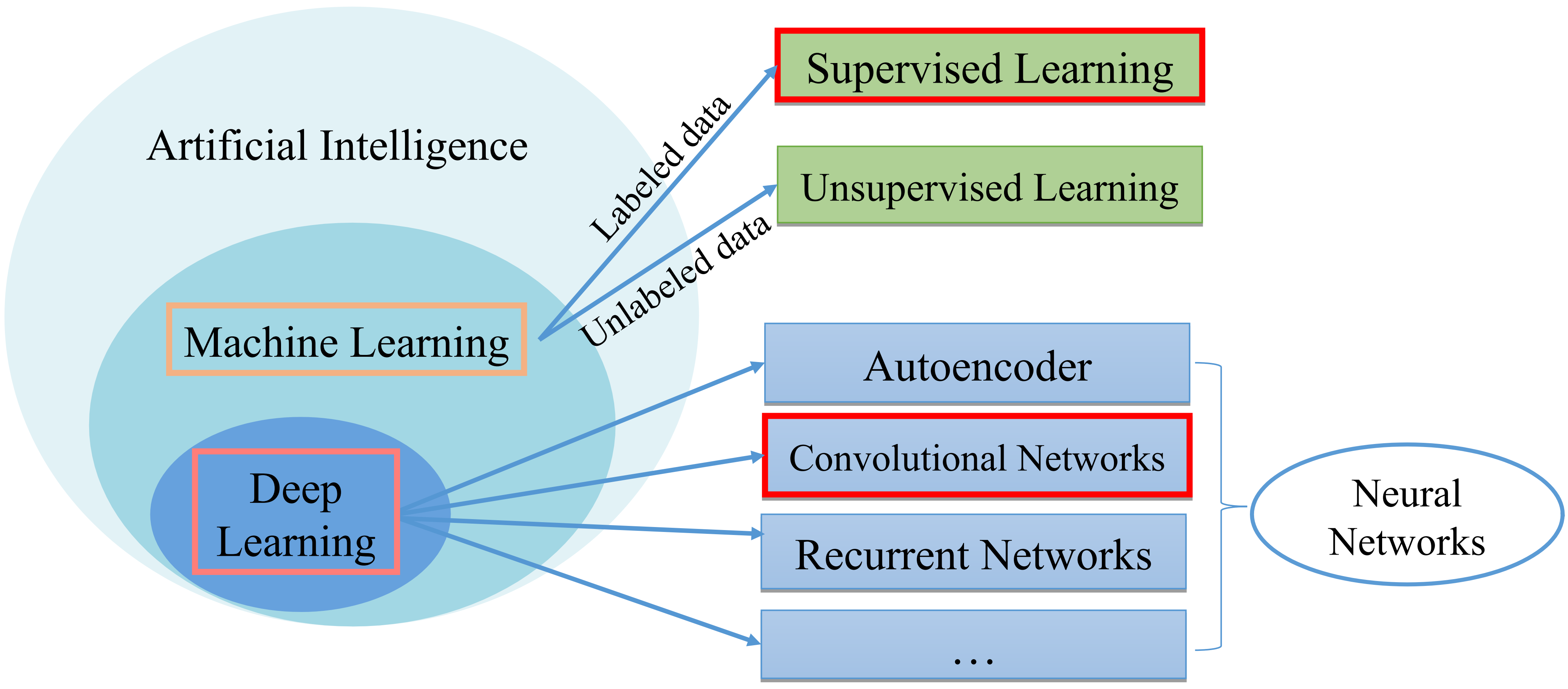
# Brief History – Neural Network Renaissance

- **From 1982 to 1985:** First development of the backpropagation algorithm allowing the automatic training of multi-layer neural networks.
- **Lecun (1998):** Development of the Convolutional Neural Network (CNNs) to solve pattern recognition problems.
- **From 1985 to 2000:** Most of the current theory used in the field of deep learning was developed during this time period, such as training algorithms and recurrent neural networks.
- However, the years **from 2000 to 2010** were another slow decade in the field of neural networks due to the lack of data and processing power to train networks in challenging real-world problems.

# Brief History – The Deep Learning Era

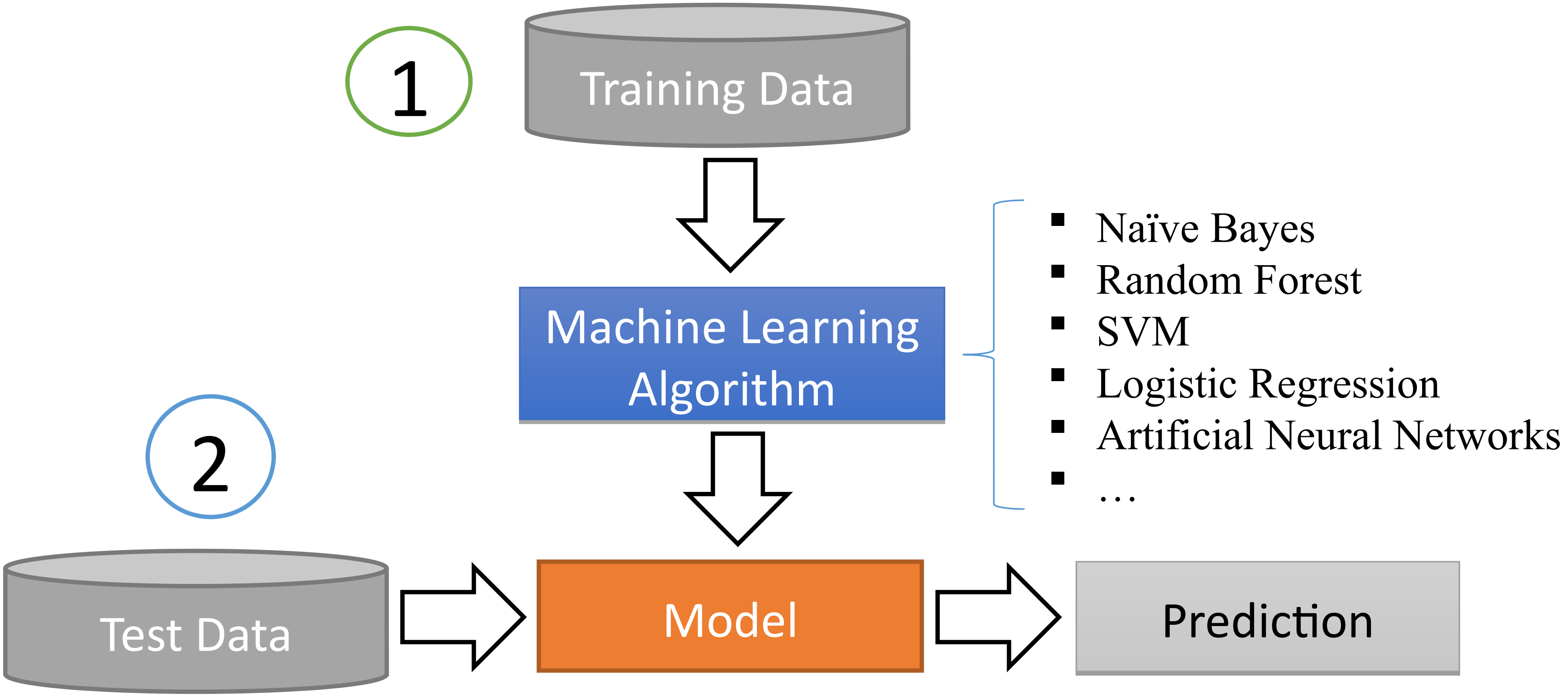
- **Nvidia (2007):** Development of CUDA which allows the programming of Graphical Computer Units (GPUs) to perform general computations.
- **Big Data (around 2008):** The deployment of high-speed internet connections around the globe made it possible to amass data in amounts never seen in human history before.
- **ImageNet Dataset (2009):** One of the first datasets used for image classification tasks with millions of data points. No machine learning model at the time was capable of achieving more than 75% of accuracy.
- **Krizhevsky (2012):** First Deep Convolutional Neural Network model to win the ImageNet Challenge.
- **From 2012 to Now:** Further availability of data and processing power allowed the development of huge deep neural network (DNNs) models, such as Residual Networks, Dense Networks, and Recurrent Networks.

# Big Picture

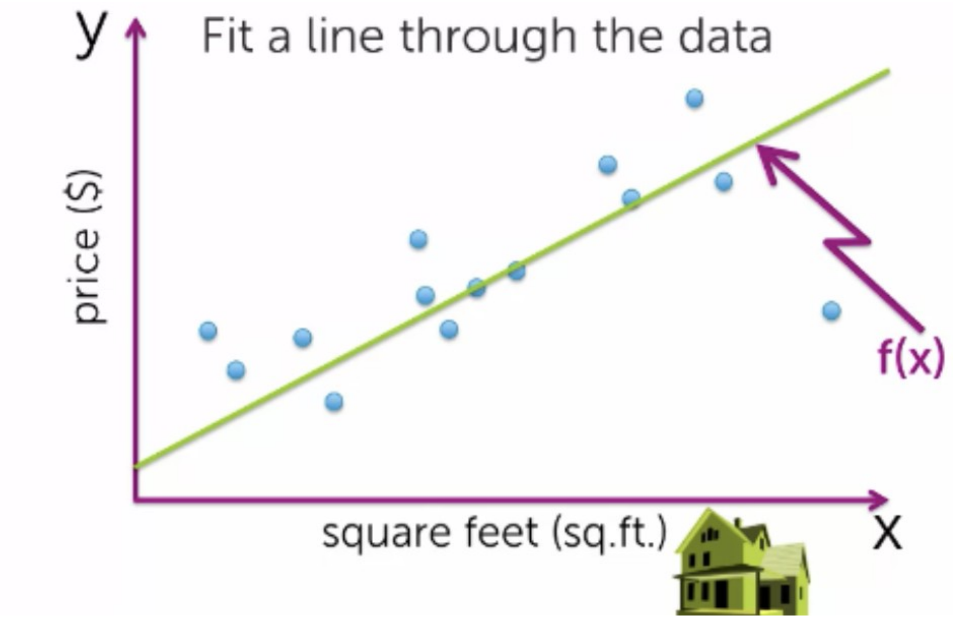


Artificial Intelligence
Machine Learning
Deep Learning
Labeled data
Unlabeled data
Supervised Learning
Unsupervised Learning
Autoencoder
Convolutional Networks
Recurrent Networks
...
Neural Networks

# Supervised Learning




Collect data -> split it to train/test sets -> training a model -> evaluate & deploy




The **goal** of supervised learning is to learn a model from training data, and use the learned model to predict unseen data (test data).

Let: ***x*** represents *square feet*, and ***y*** represents *price,* then we have:

**Training data:**

$$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$$

**Test data:**

$$(x_{n+1}, y_{n+1}), (x_{n+2}, y_{n+2}), \ldots, (x_{n+m}, y_{n+m})$$

Thus, this linear model can be represented as:

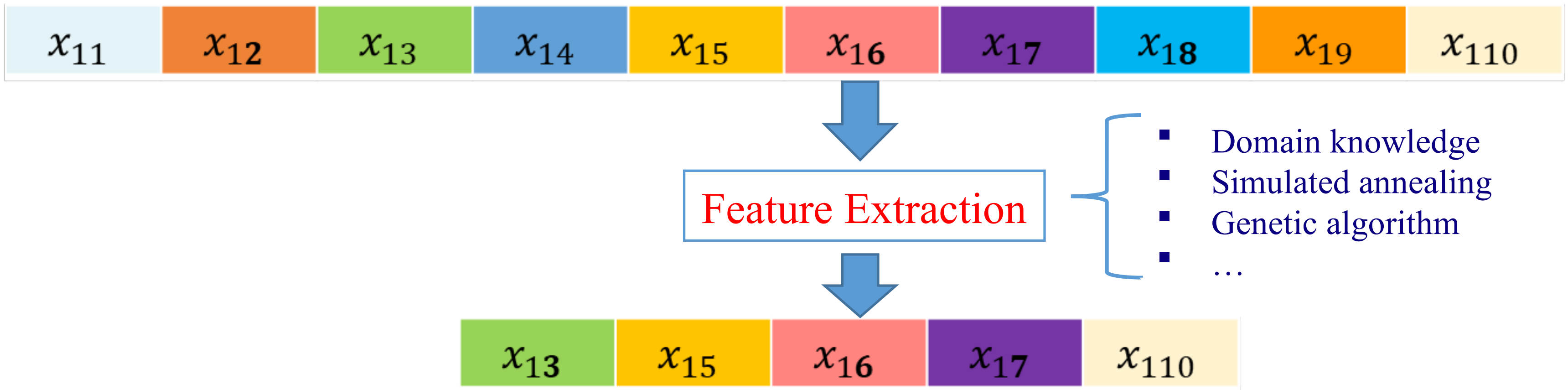
$$y_i = f(x_i) = w \cdot x_i + b$$

- **Features** may be **redundant** or **irrelevant** resulting in **poor performance** of a machine learning model.

| Square feet | Beds | Pets, Garden plants, Garbage can, ... | Price |
|---|---|---|---|
| $x_{11}$ | $x_{12}$ | ... | $y_1$ |
| $x_{21}$ | $x_{22}$ | ... | $y_2$ |
| ... | ... | ... | ... |

Data:

$$([x_{11}, x_{12}, \ldots, x_{1i}], y_1),$$
$$([x_{21}, x_{22}, \ldots, x_{2i}], y_2),$$
$$\cdots,$$
$$([x_{n1}, x_{n2}, \ldots, x_{ni}], y_n),$$

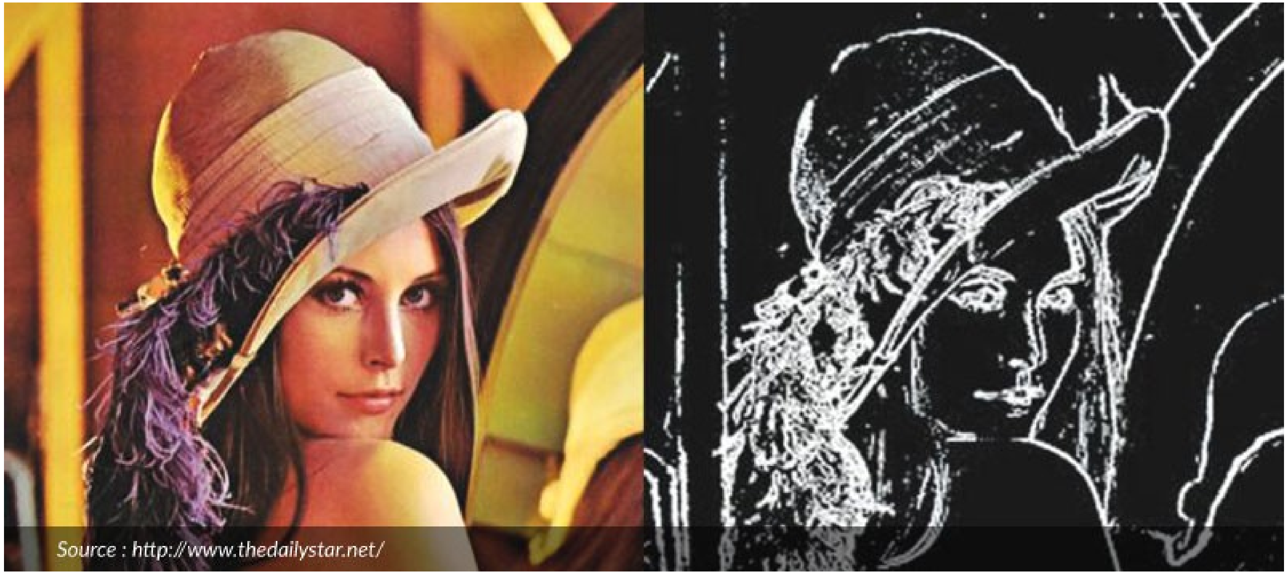


# Challenge for High Dimensional Data

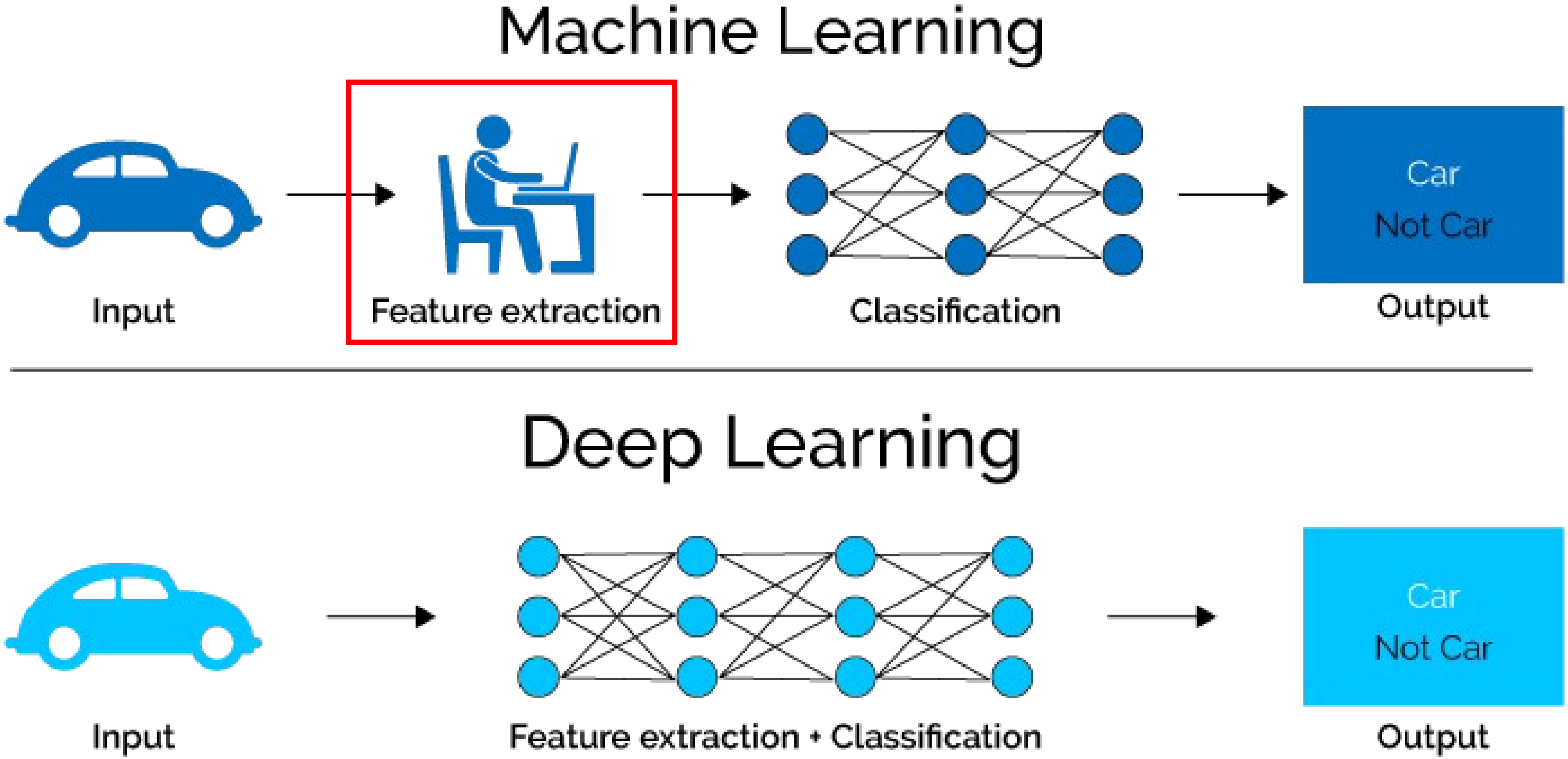
- Images are examples of highly dimensional data.

Input Image

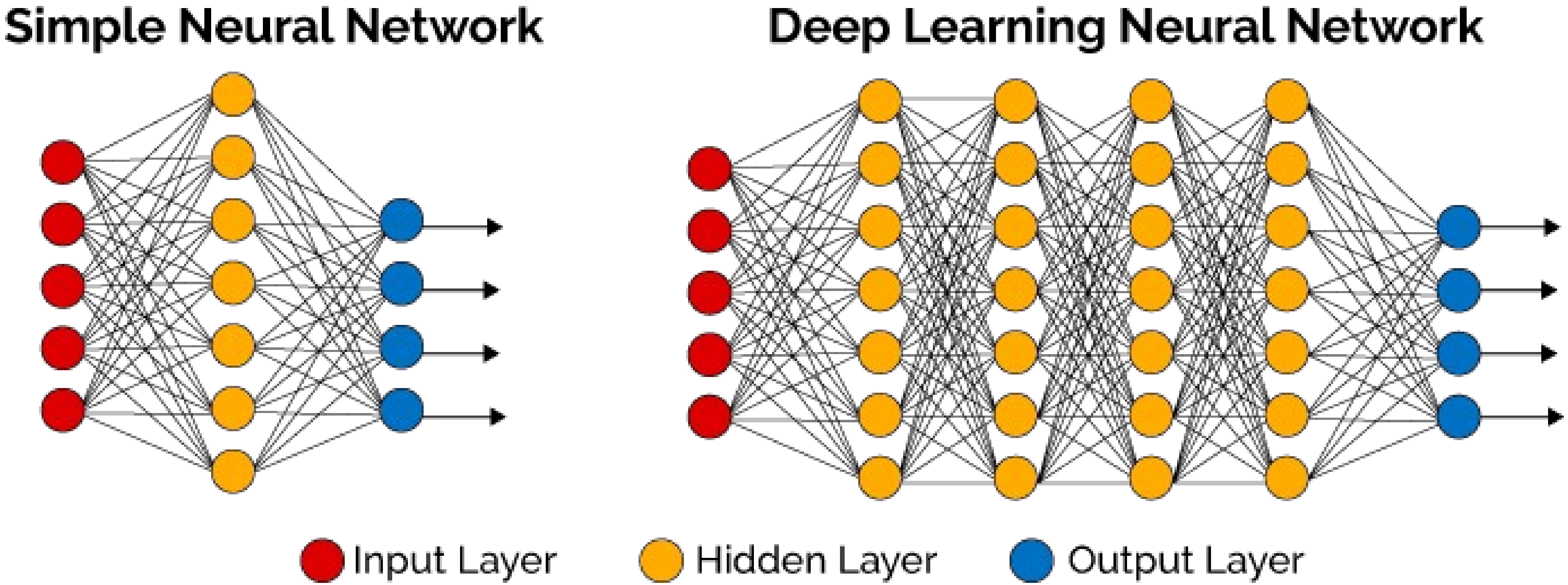
Desired features (edges)



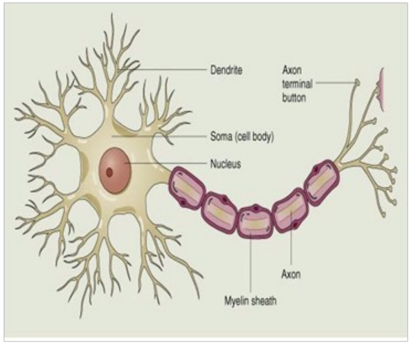
# Deep Learning - Motivation

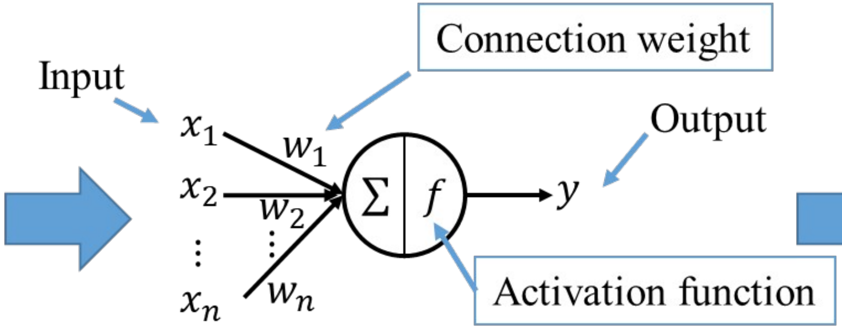


# Deep Neural Networks vs Shallow Neural Networks

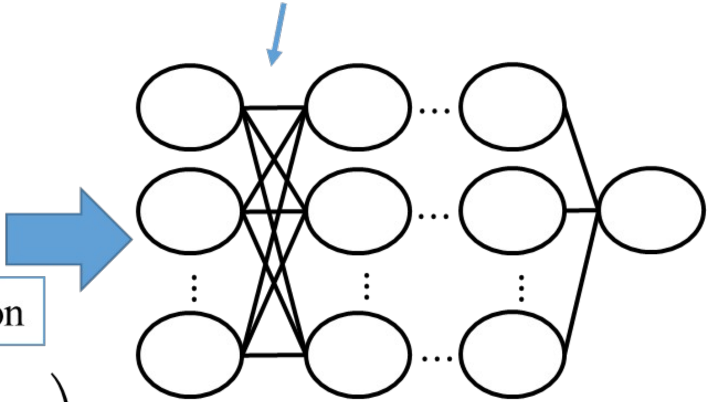


- Bio-inspired computational models




$$y = f(w_1 x_1 + w_2 x_2 + \cdots + w_n x_n) = f\left(\sum_{i=1}^{n} w_i x_i\right)$$




$$y = f_n(f_{\ldots}(f_2(f_1(x, w^1), w^2), w^{\cdots}), w^n)$$

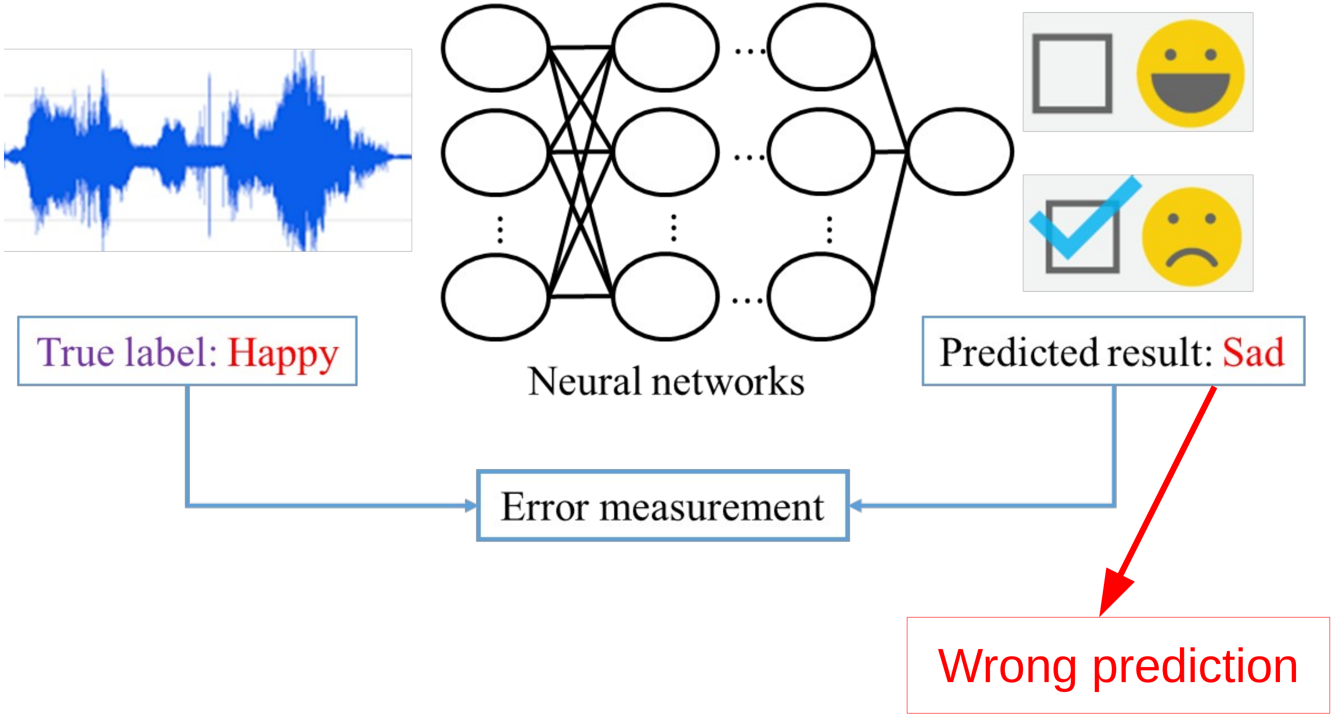
Biological neuron in brain

Computational model of a neuron

A neural network with $n$ layers

# Learning in Neural Networks

- Voice sentiment analysis




**Gradient-based Learning:**

1) Initialize connection weights.
2) Compute output (predicted result) for given input data.
3) Measure error between predicted result and true label.
4) Calculate gradients of the error w.r.t. connection weights.
5) Update weights of NN using gradients to decrease error until converge (error back propagation).

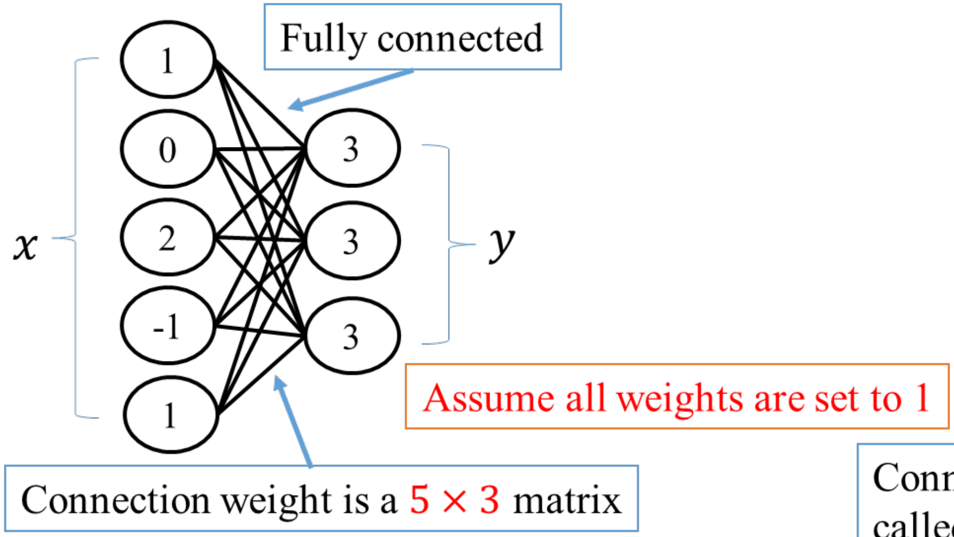
**With given data $x$ and its label $\overline{y}$ from a dataset**

1. Initialize connection weights.
2. Compute output (predicted result) for given input data (forward pass).
3. Measure error between predicted result and true label.
4. Calculate gradients of the error w.r.t. connection weights (backward pass).
5. Update weights of NN using gradients Until the error converge.

1. Initialize $w_1, w_2, \cdots, w_n$
2. $y = f_n(f_{...}(f_2(f_1(x, w_1), w_2), w_{...}), w_n)$
3. $error = ||y - \overline{y}||$
4. $\Delta w_i = \dfrac{\partial error}{\partial w_i}$ (backpropagation)
5. $w_i = w_i - \alpha \cdot \Delta w_i$

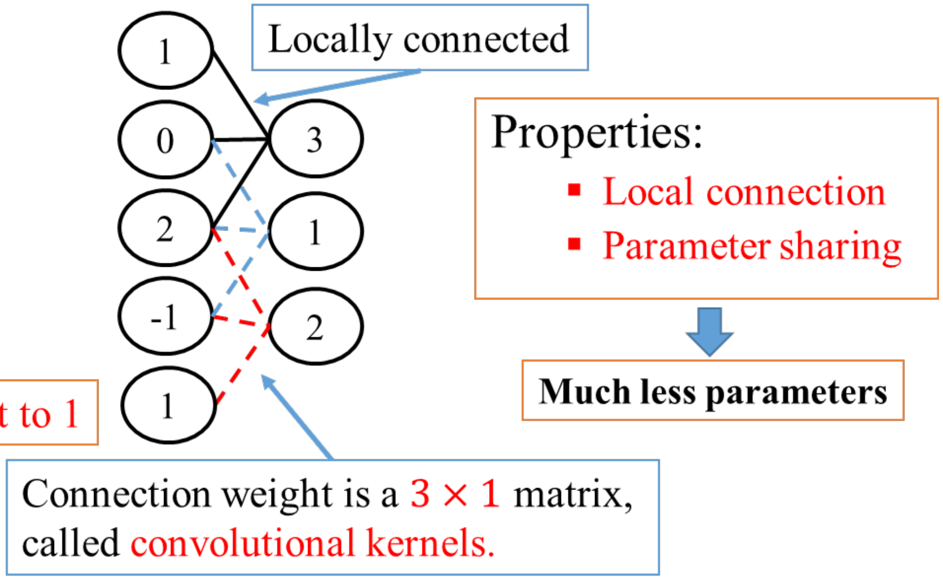
# Convolutional Neural Networks (CNNs)

- **Properties of CNN:** local connection and parameter sharing




$$y_j = \sum_{i=1}^{n} w_{ij} x_i$$

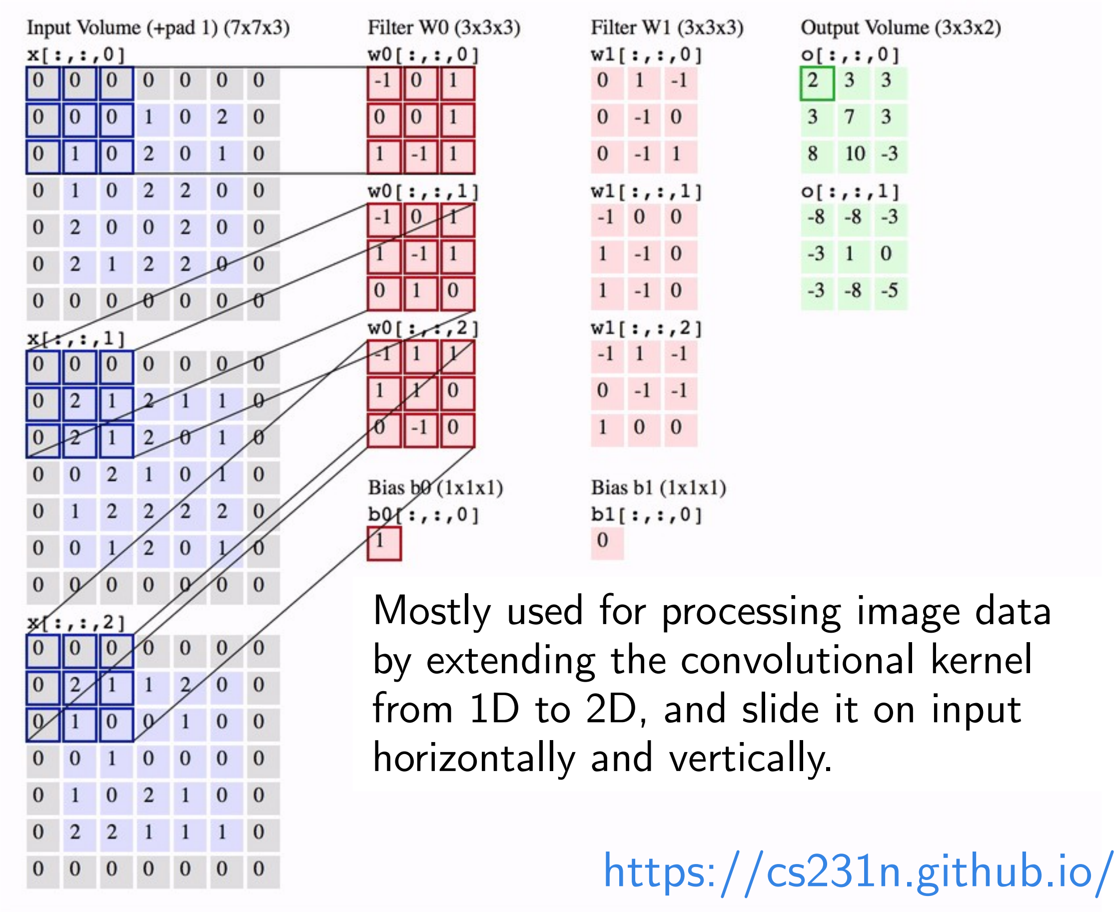
Fully connected neural network




$$y_j = \sum_{i=1}^{k} w_{i1} x_{j+i-1}$$

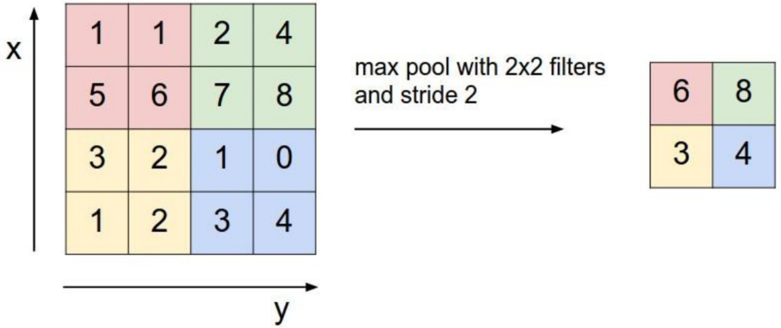
1D convolutional neural network

## 2D convolutional operations




Mostly used for processing image data by extending the convolutional kernel from 1D to 2D, and slide it on input horizontally and vertically.

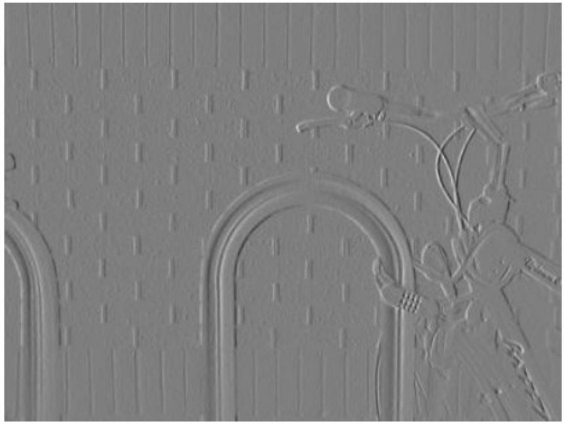
## Max pooling (Nonparametric)



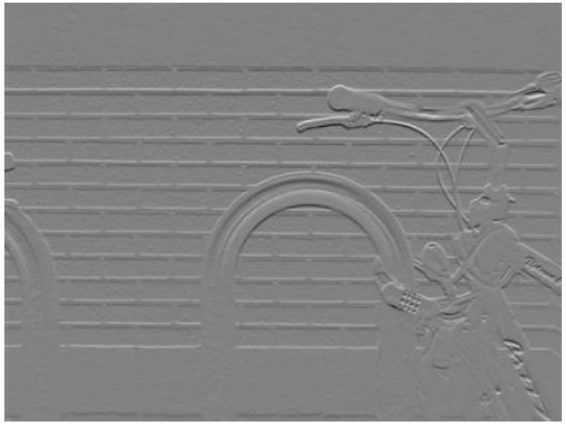

https://cs231n.github.io/convolutional-networks/

# Convolution Kernels (Filters)

Original image



Convolution for finding vertical edges



Convolution for finding horizontal edges

Vertical edge filter:

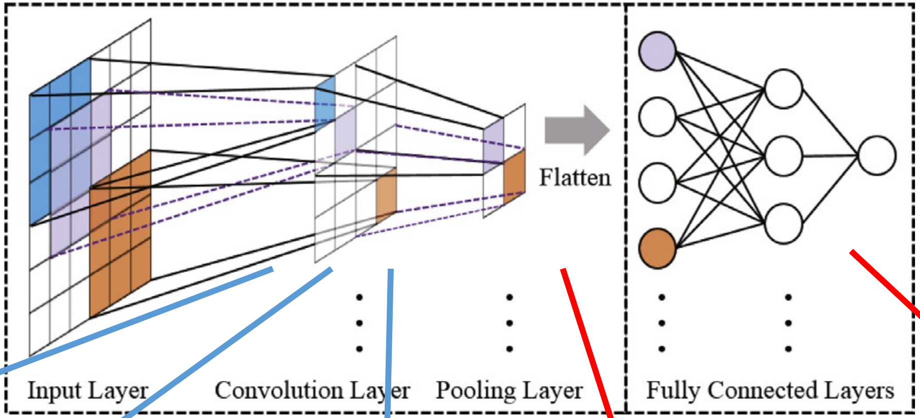
```
[[-1,  0,  1],
 [-2,  0,  2],
 [-1,  0,  1]]
```

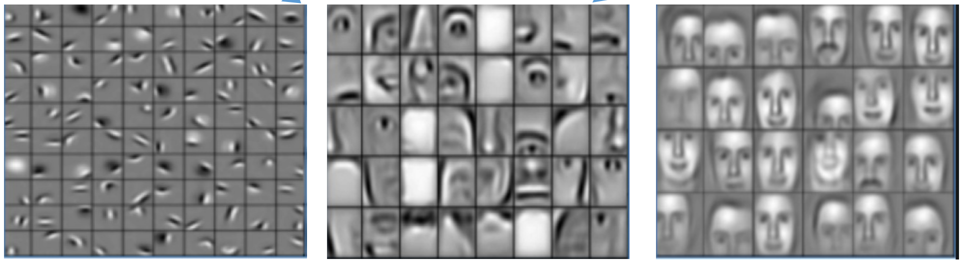
Horizontal edge filter:

```
[[-1, -2, -1],
 [ 0,  0,  0],
 [ 1,  2,  1]]
```

# Convolution Kernels Can Be Found from Data

Gradient based learning



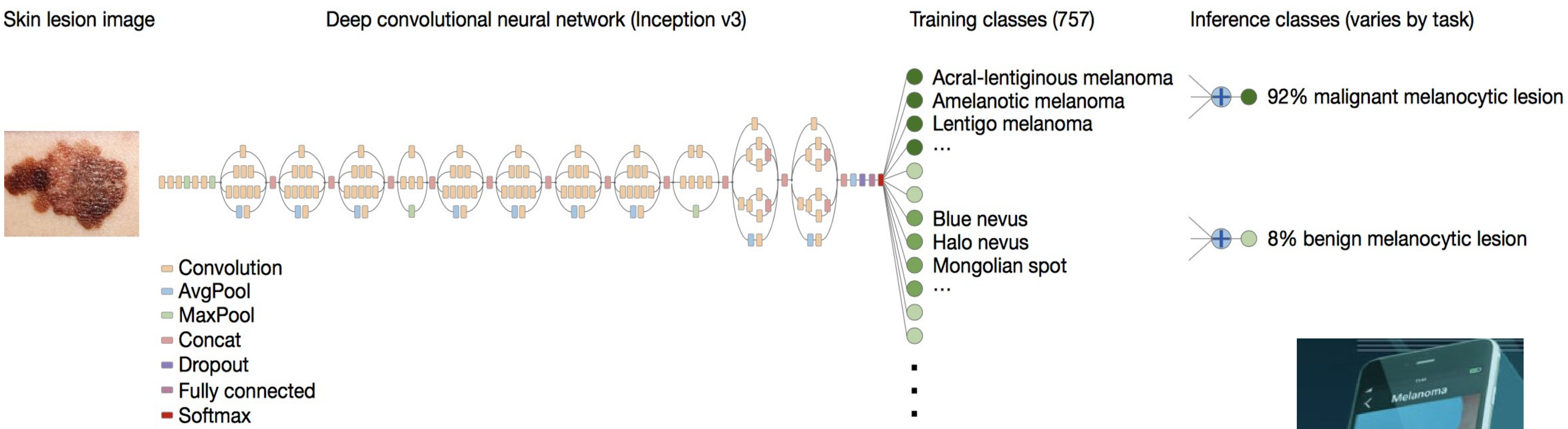
Hierarchical features in CNN layers (e.g., face detection).

Convolution layers learn to extract features

Fully-Connected layers learn to classify features

# Deep Neural Networks Applications

# Diagnose skin cancer at dermatologist-level

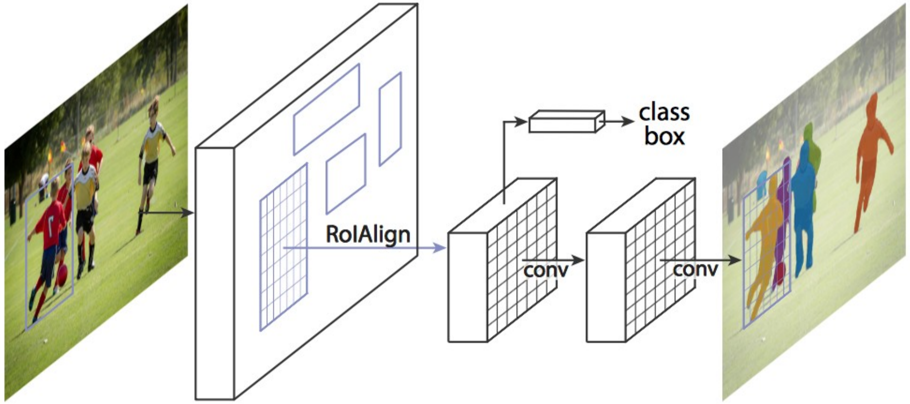



130,000 skin lesion images comprised of over 2,000 diseases



Esteva A, Kuprel B, Novoa R A, et al. Dermatologist-level classification of skin cancer with deep neural networks[J]***Nature***, 2017, 542(7639): 115-118.

# Object detection & segmentation

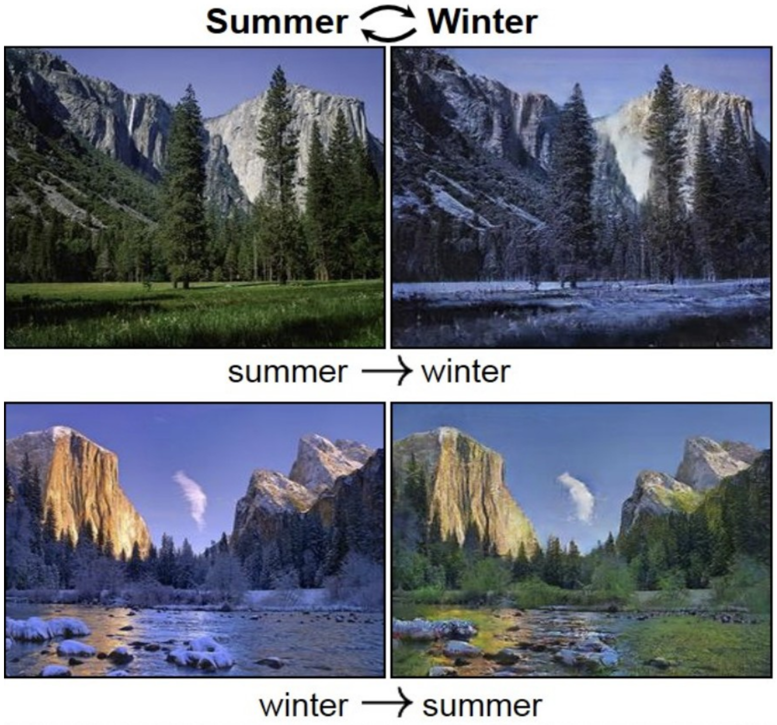



identify each object in pixel-level

He K, Gkioxari G, Dollár P, et al. Mask r-cnn[J]. ICCV 2017.



http://selfdrivingcars.mit.edu




Zhu J Y, Park T, Isola P, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks[J]. arXiv preprint arXiv:1703.10593, 2017.

# Hand-crafted State-of-the-Art DNN Architectures

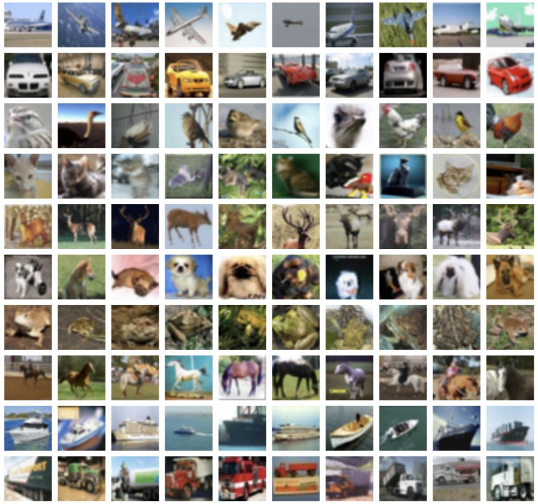
# Benchmark Datasets

**MNIST**

**Images:** 70,000
**Categories:** 10

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

**CIFAR-10**



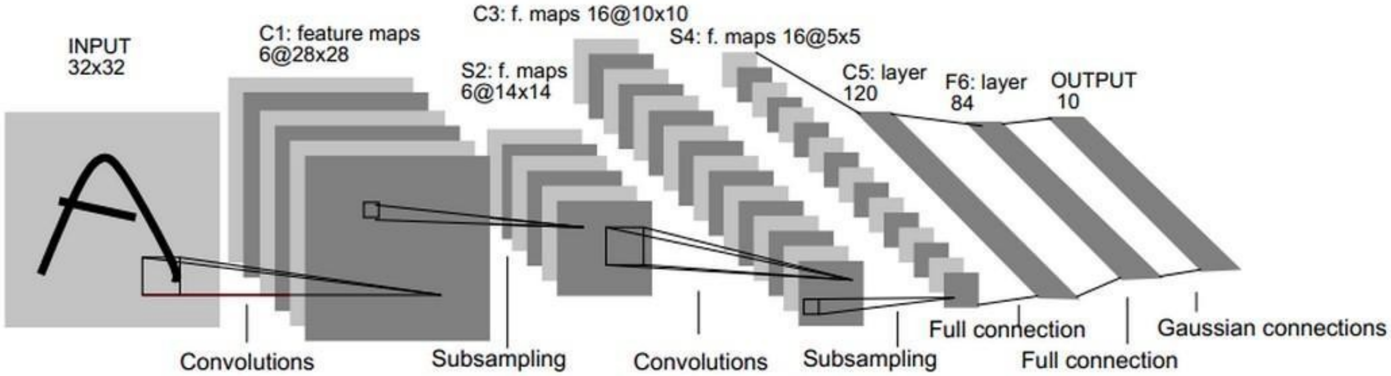
**Images:** 60,000
**Categories:** 10

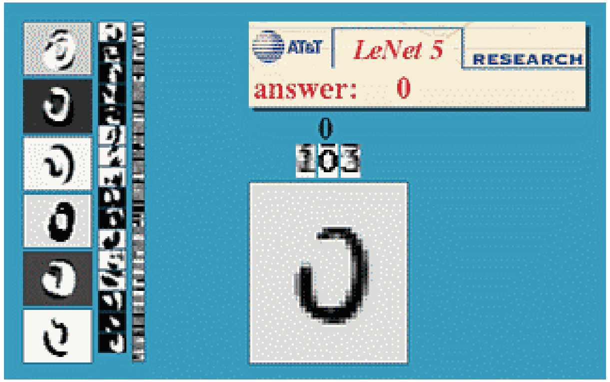
https://www.cs.toronto.edu/~kriz/cifar.html

**ImageNet**

**Images:** 14,197,122
**Categories:** 1,000

http://image-net.org



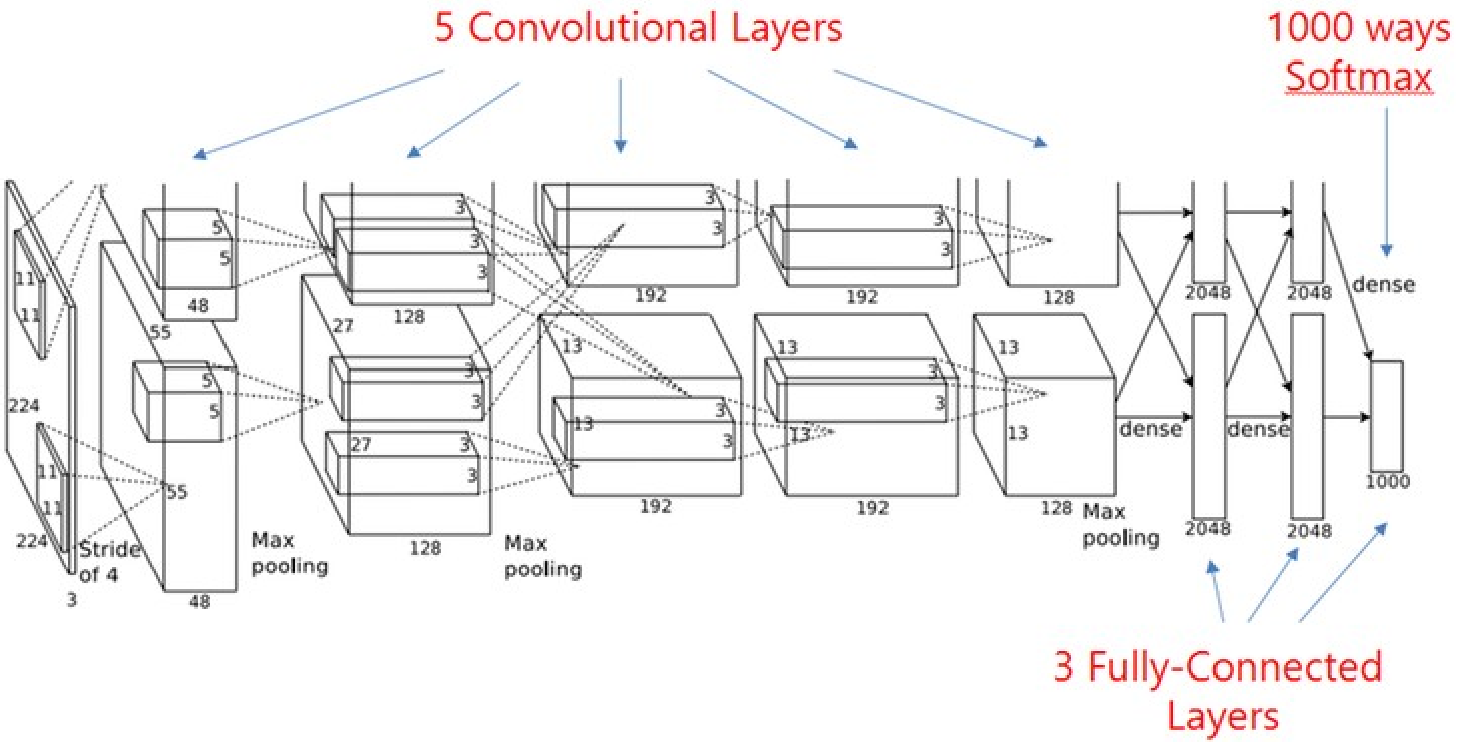




Only two convolution layers

Lenet-5 for hand-written digits recognition

**LeCun Y**, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. *Proceedings of the IEEE*, 1998, 86(11): 2278-2324.

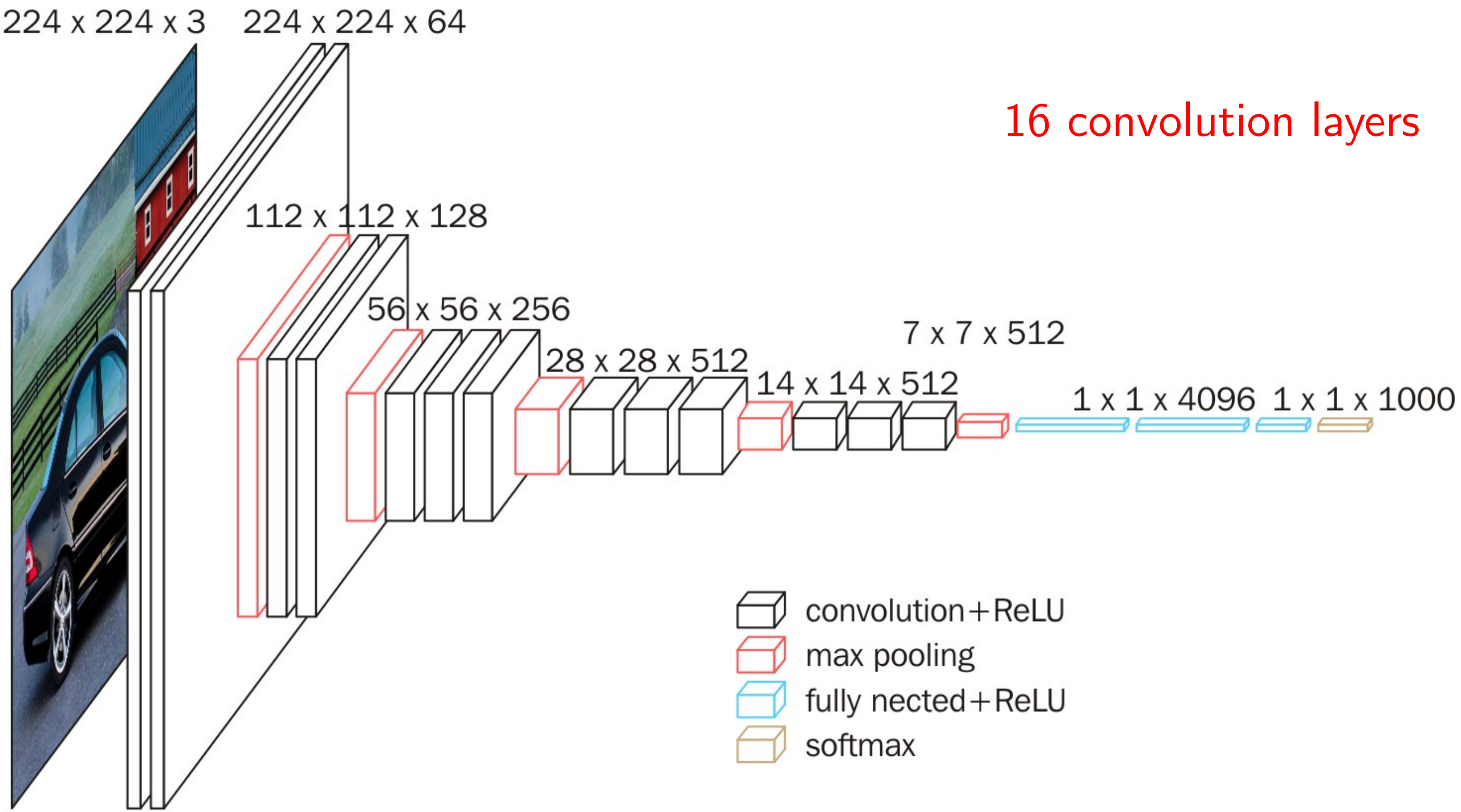
# AlexNet




Won the ImageNet Challenge in **2012**

Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]/Advances in Neural Information Processing Systems. 2012: 1097-1105.

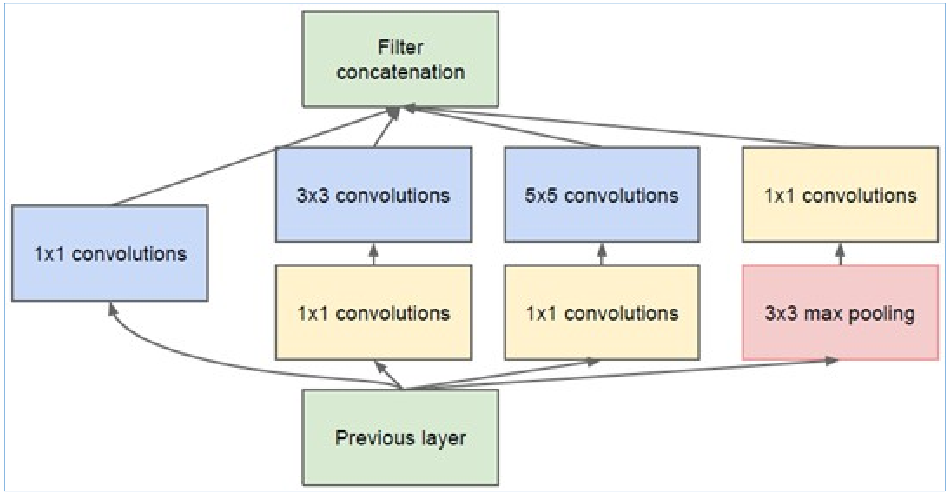
# VGG16



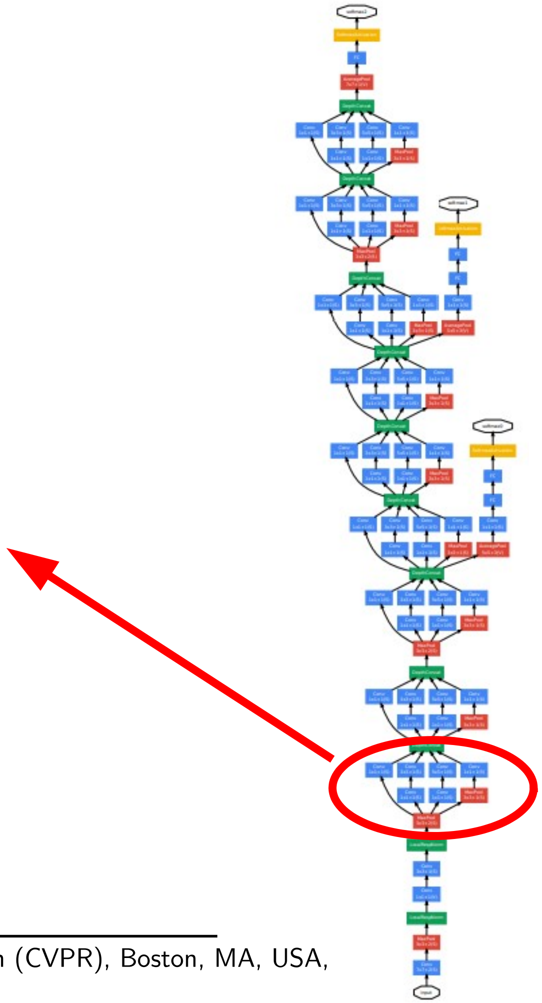

16 convolution layers

K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv:1409.1556, Sep. 2014.

# GoogLeNet




Inception Module



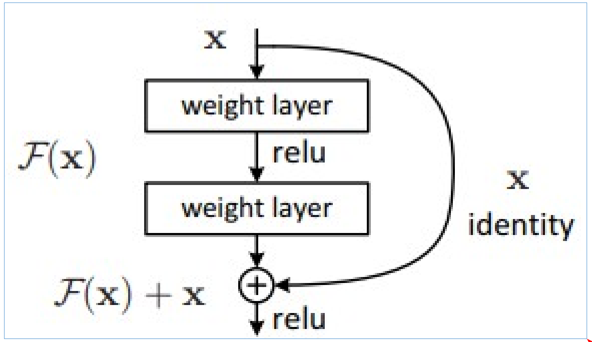
Won the ImageNet Challenge in **2015**

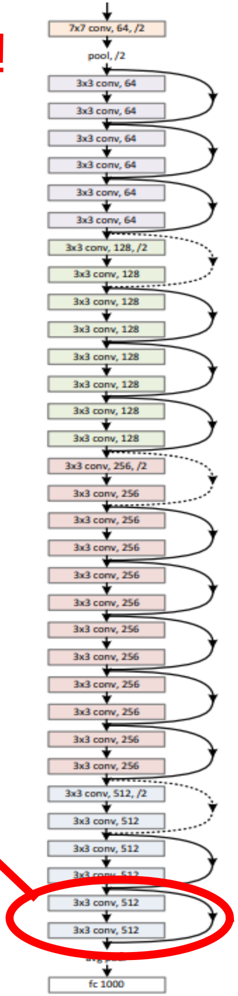
C. Szegedy et al., "Going deeper with convolutions," in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, pp. 1–9.

# ResNet

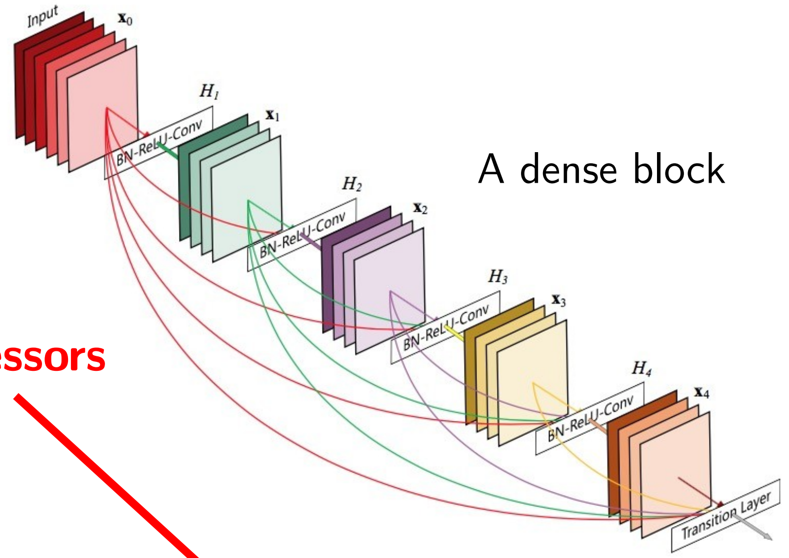
1000 convolution layers!

Won the ImageNet Challenge in **2016**



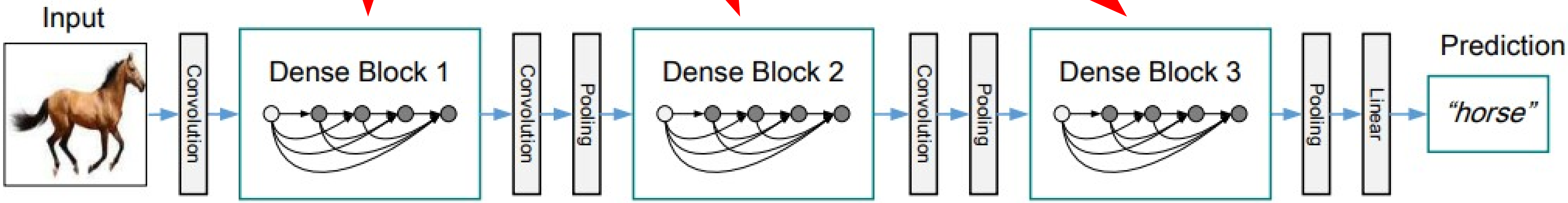




K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 770–778.




A dense block

**Densely connected between each layer and its predecessors**




G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 2261–2269.

# Conclusions

- Currently, Deep Neural Network (DNNs) models are very popular in the field of machine learning.
  - These models are capable of extracting usable information from raw data automatically.
  - They can be used to solve supervised and unsupervised problems.
- However, the development of DNN-based solutions requires lots of expert knowledge about the problem at hand.
- Another problem, when developing DNN-based solutions, is the amount of computational power required to train and deploy such models.

- **Next class:**
  - How the use of Evolutionary Computation algorithms can help us reduce the problems faced by researchers and experts when developing DNN-based solutions.

# Acknowledgments


- I wish to thank Yao Zhou, Ph.D. Candidate at Sichuan University, China, for allowing the use of his slides for this class.
  - Email: zy3381@gmail.com