VIT-AP UNIVERSITY

Computer Vision (Course Code: 4047)

Module-4:Lecture-3: Deep Learning Classifiers - basics

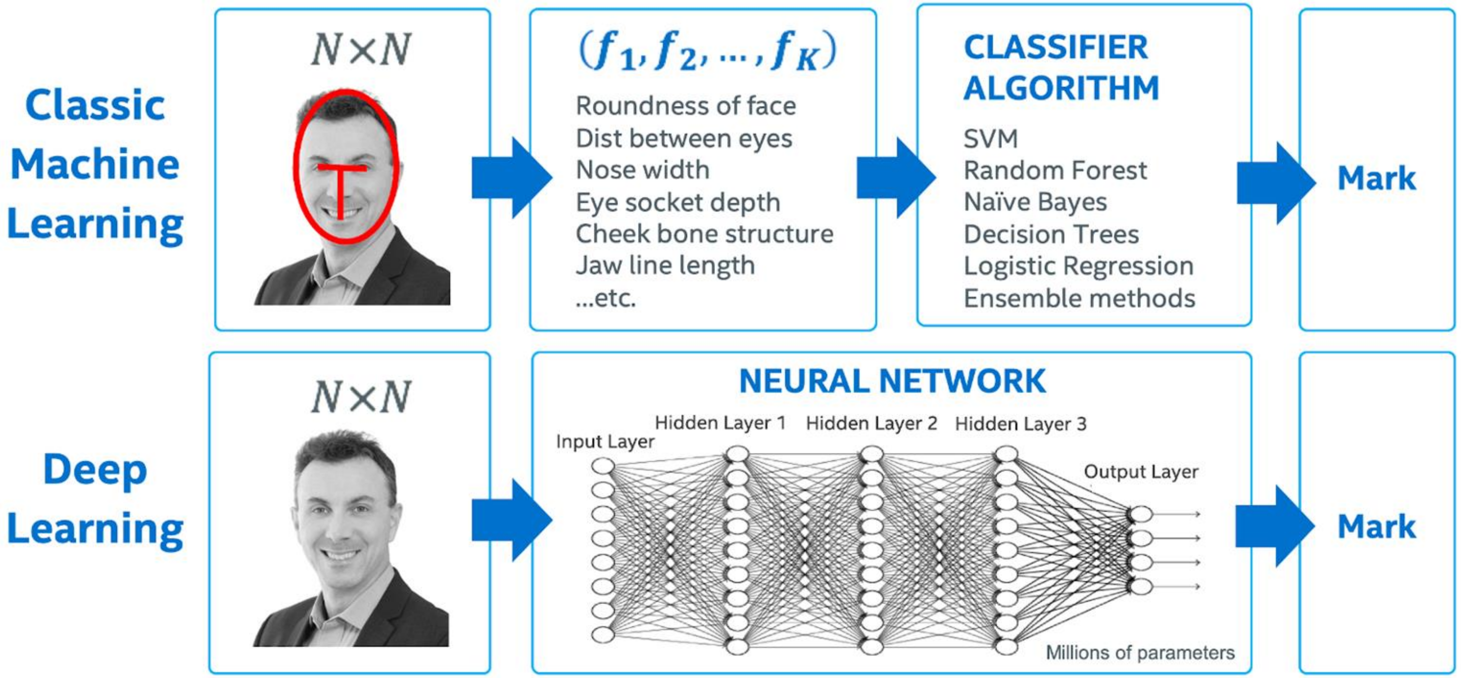
Gundimeda Venugopal, Professor of Practice, SCOPE

Topic Coverage

- Deep Learning
- Why Deep Learning?
- Deep Neural Network Optimization
- Convolutional Neural Network
- CNN Usecases

Deep Learning

- Deep Learning is a subset of Machine Learning based on Neural Networks that permit a machine to train itself to perform a task.
- Deep learning algorithms attempt to learn (multiple levels of) representation by using a hierarchy of multiple layers (> 1 hidden layer).
- Each layer of nodes trains on a distinct set of features based on the previous layer’s output
- The further you advance into the neural net, the more complex the features your nodes can recognize, since they aggregate and recombine features from the previous layer



Face Detection example

- Manually designed features
- Need small amount of training data
- Very good at learning patterns. Small models
- Cannot model complex challenges with high accuracy

- Model complex challenges (Vision, speech, NLP, .. .)
- Automatic Feature Learning. Easy to adapt, fast to learn
- Utilize large amounts of training data
- Requires better hardware resources for training

Source: [Intel AI 101]

Deep Learning Pioneers

- Geoffrey Hinton: Backpropagation, Boltzman machines, Deep Learning, CNN improvements, Capsules, AlexNet (Google)
- Yann LeCun: CNN, Backpropagation improvements (Facebook)
- Joshua Bengio: GAN, word embeddings, sequence models, (Mila)

Andrew Ng (Google Brain, Baidu)

Ian Goodfellow (Apple, GAN)

Fei-Fei Li (Stanford Prof, Google)

Andrej Karpathy (Tesla, ImageNet)

Demis Hassabis (Deep Mind)

Many others ..

Source; acm.org

Why Now?

1952	Stochastic Gradient Descent
1958	Perceptron • Learnable Weights
⋮	
1986	Backpropagation • Multi-Layer Perceptron
1995	Deep Convolutional NN • Digit Recognition
⋮	

Neural Networks date back decades, so why the resurgence?

1. Big Data

- Larger Datasets
- Easier Collection & Storage

IMAGENET

WIKIPEDIA
The Free Encyclopedia

2. Hardware

- Graphics Processing Units (GPUs)
- Massively Parallelizable

3. Software

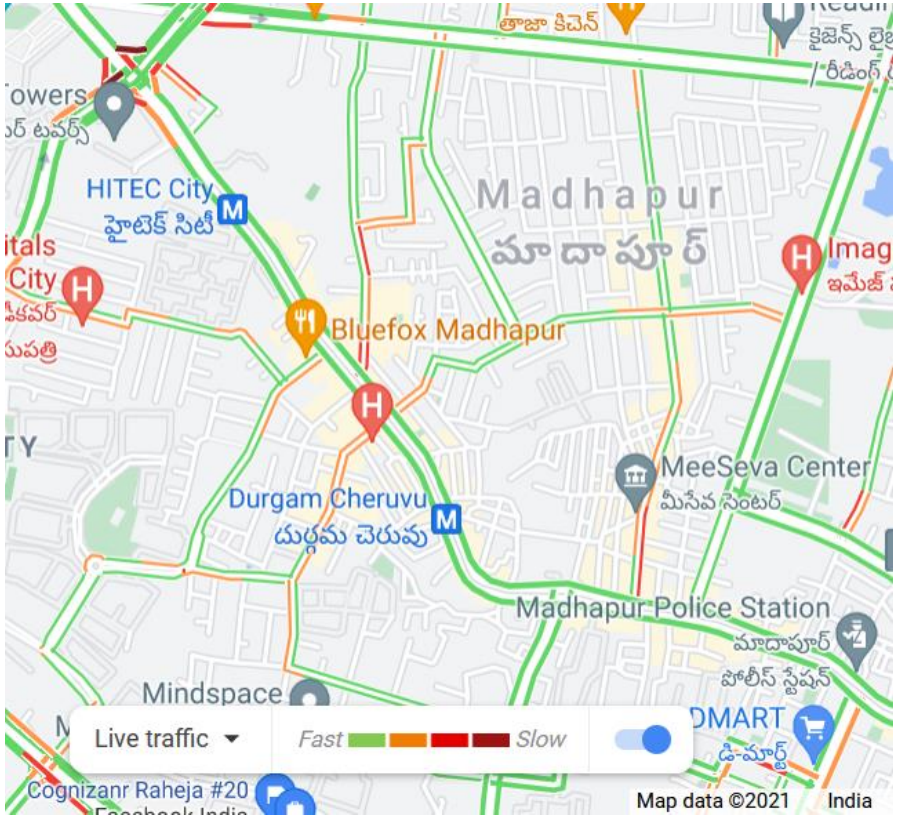
- Improved Techniques
- New Models
- Toolboxes

TensorFlow

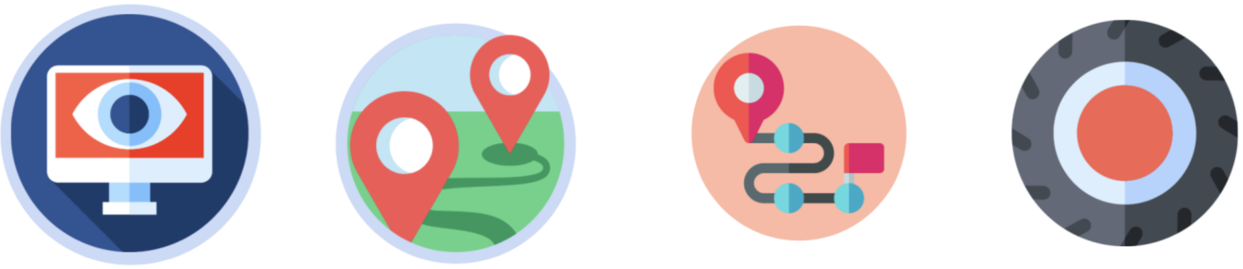
Source: MIT 6.S191 (2020): Introduction to Deep Learning

Real world examples

Google Maps Traffic Prediction



Self driving Cars



In **Perception**, you find the **environment** and **obstacles** around you.

In **Localization**, you define your **position** in the world at 1–3 cm accuracy.

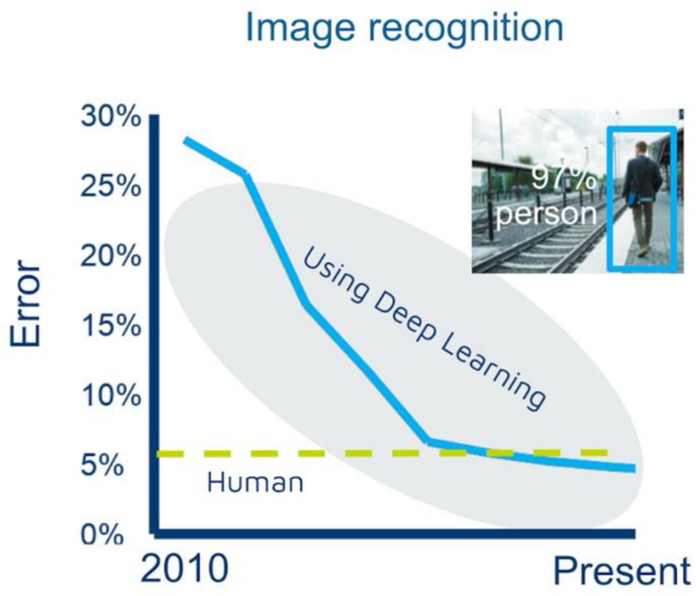
In **Planning**, you define a **trajectory from A to B**, using perception and localization.

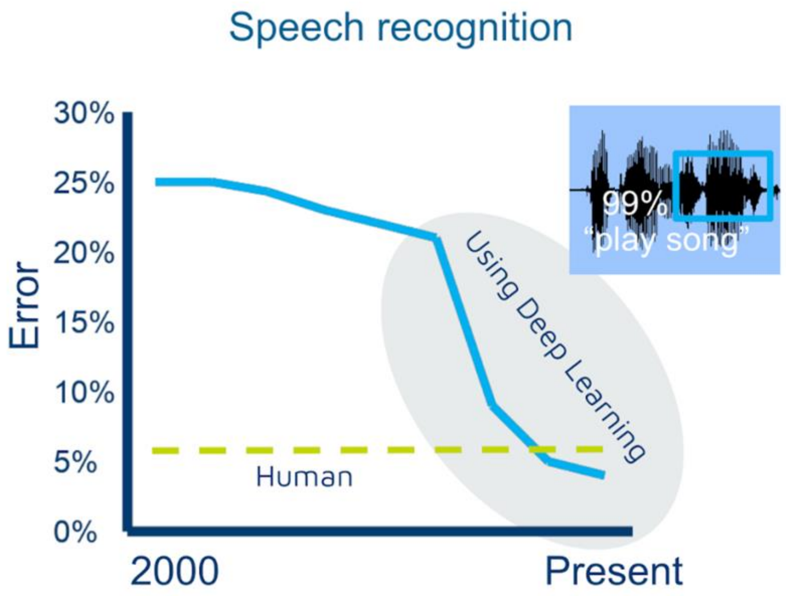
In **Control**, you follow the trajectory by **generating a steering angle and an acceleration value**.

Source: becominghuman.ai

State of the art in Image Recognition, Speech and NLP

In ~2010 Deep Learning started outperforming other Machine Learning techniques first in speech and vision, then NLP





Several big improvements in recent years in NLP

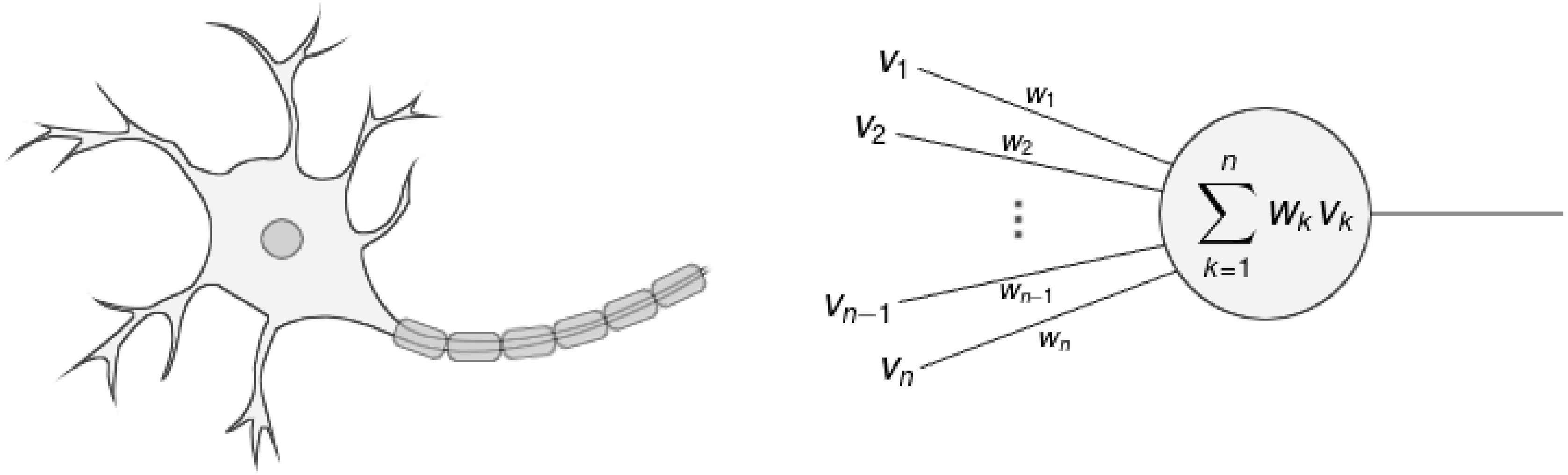
- Machine Translation
- Sentiment Analysis
- Dialogue Agents
- Question Answering
- Text Classification …

Leverage different levels of representation

- words & characters
- syntax & semantics

Source: Intel AI 101

Neuron



Artificial neuron is inspired by biological neuron

Image source: [hackernoon]

Activation function

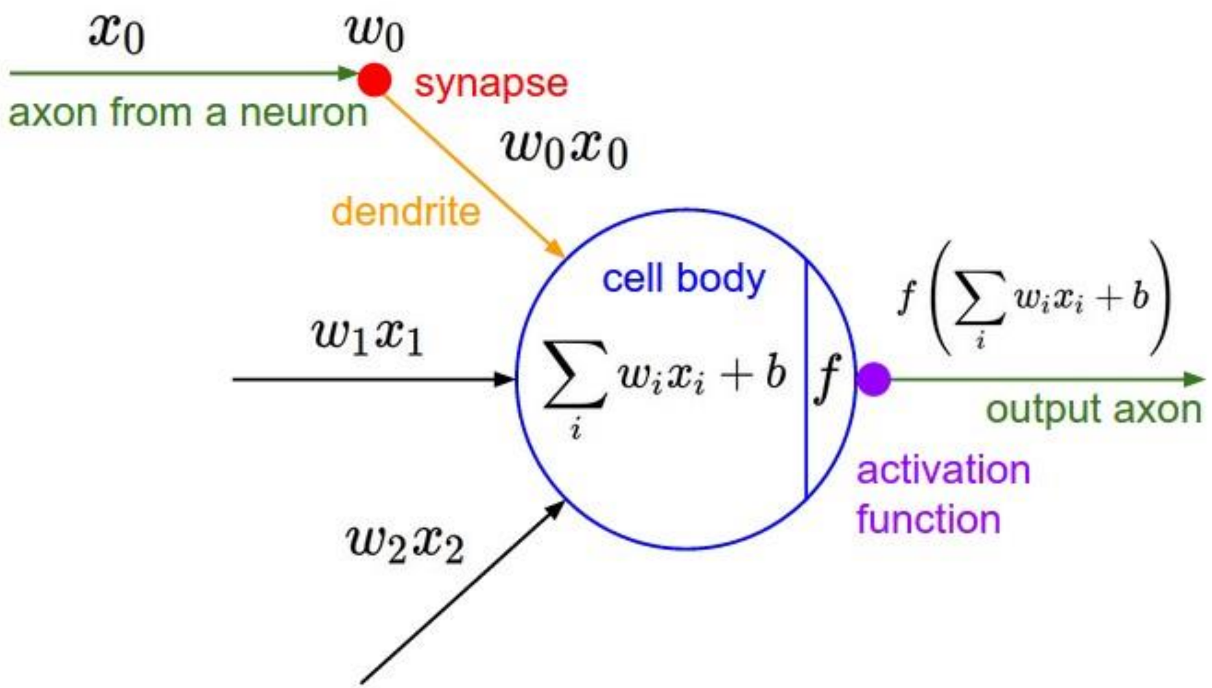
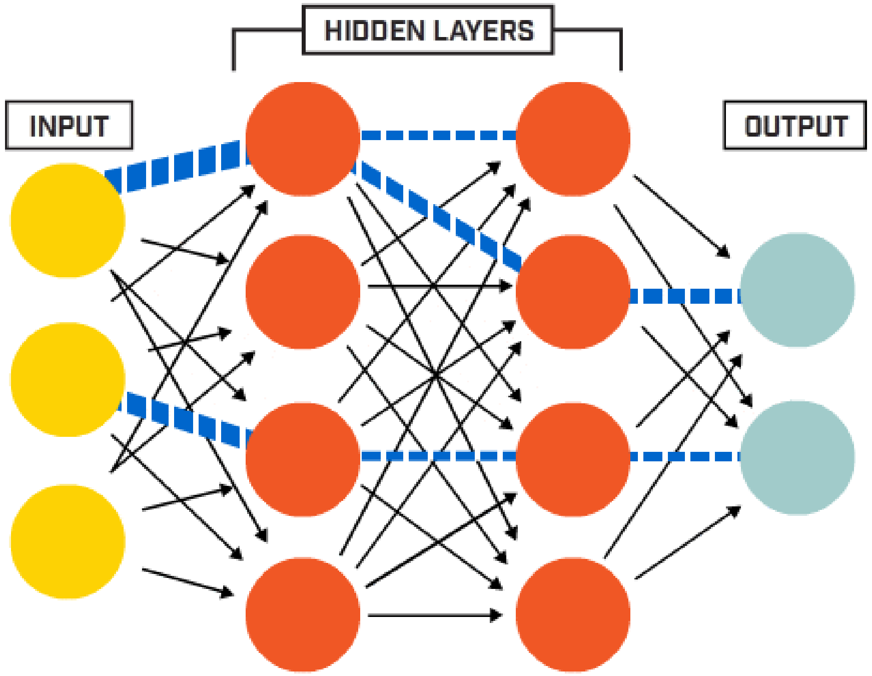


Image source: [cs231n stanford]()

Train a neural network



- Choose network design
- Form a neural network
- Choose hyper parameters
- Compute an estimate value for all samples
- Compute loss
- Reduce loss
- Repeat last three steps

Image source: ptgrey.com

Error and Loss function

- In most learning networks, error is calculated as the difference between the actual output and the predicted output.
- The function that is used to compute this error is known as Loss Function.
- Different loss functions will give different errors for the same prediction, and thus have a considerable effect on the performance of the model.
- One of the most widely used loss function is mean square error, which calculates the square of difference between actual value and predicted value.
- Different loss functions are used to deal with different type of tasks, i.e. regression and classification.

Text source: xpertup

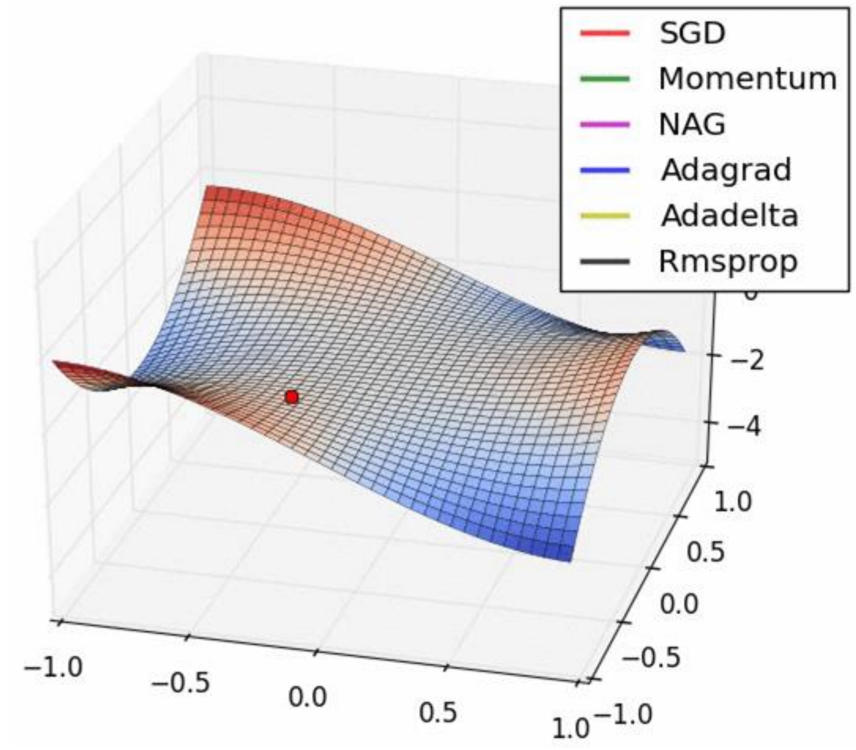
Neural Network Training

Sample labeled data (**batch**) → **Forward** it through the network, get predictions → **Back-propagate** the errors → **Update** the network weights

1. Need to optimize (min. Or max.) the objective/cost function J(W) using an optimizer (e.g, Gradient Descent, SGD, RMSprop, Adam)
2. Generate error signal that measures difference between predictions and target values
3. Use error signal to change the weights and get more accurate predictions
4. Subtracting a fraction of the gradient moves you towards the (local) minimum of the cost function

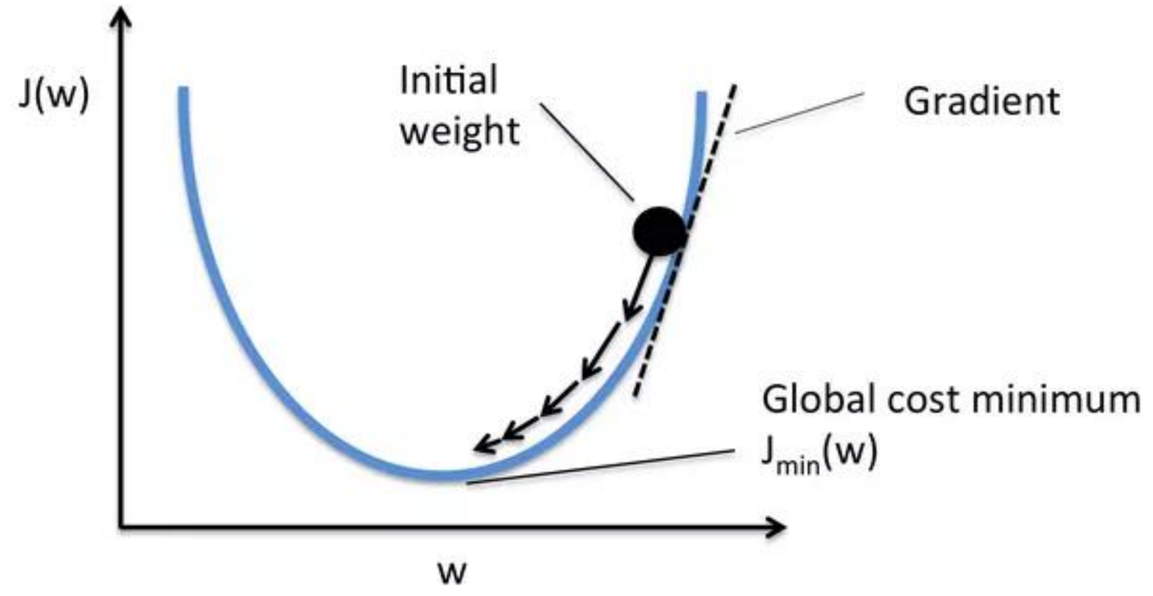
Loss Functions in real life are quite complex. They may have:

- A Global Minima
- Multiple Local Minima
- Saddle points



Source: Medium

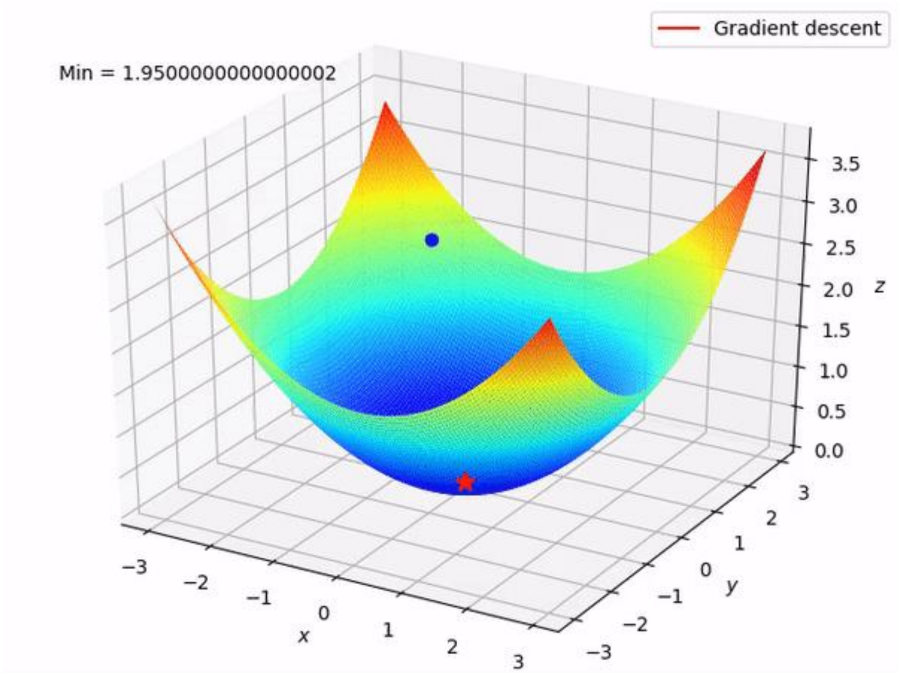
Gradient



Optimisation functions usually calculate the **gradient** i.e. the partial derivative of loss function with respect to weights, and the weights are modified in the opposite direction of the calculated gradient. This cycle is repeated until we reach the minima of loss function.

Image and text source: [xpertup]

Gradient Descent

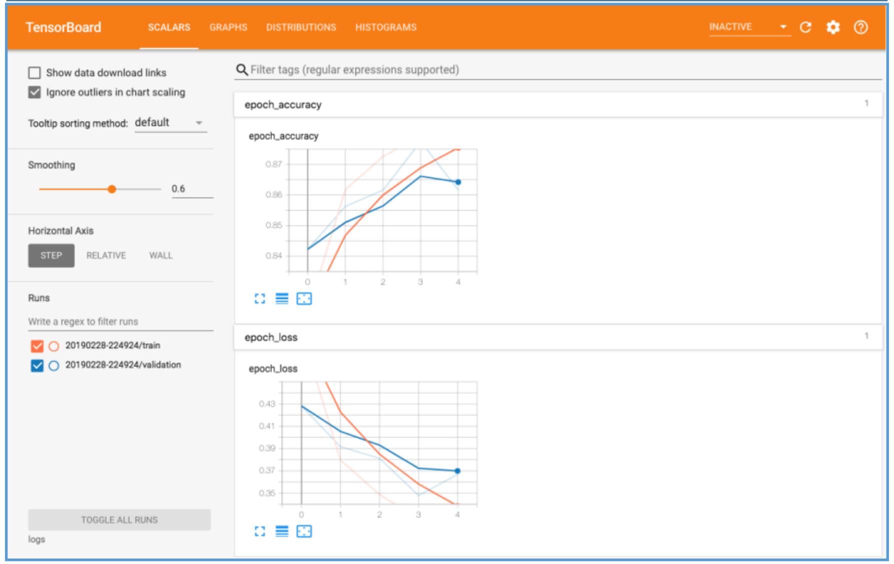


The procedure of repeatedly evaluating the gradient and then performing a parameter update is called Gradient Descent.

Image source: [xpertup]

Google Tools for Better Model Building

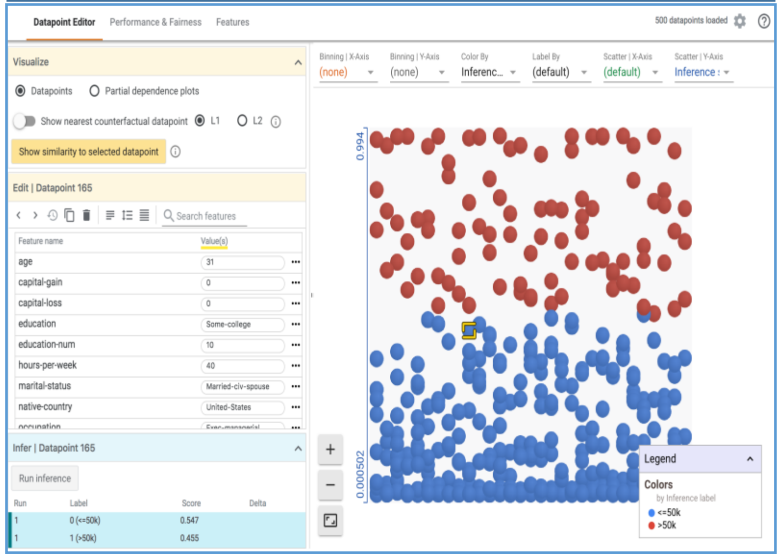
Tensorboard



Source: [Google]()

A Visual Interface to visualize the Model Graph and Training progress. Integrated with what-if and HParams dash board tools.

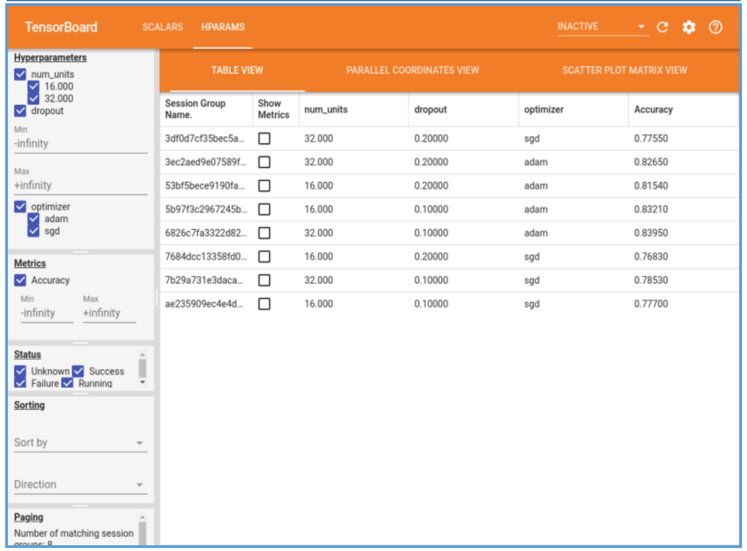
What-if Tool



Source: [Google]()

A Visual Interface that helps to better understand your data sets and the output of machine learning models (Currently, Classification and regression models)

HParams Dashboard

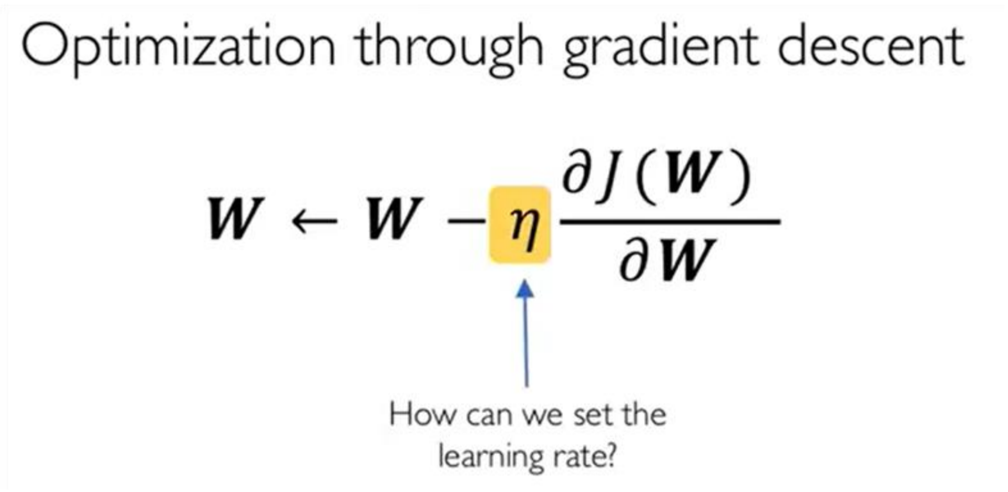


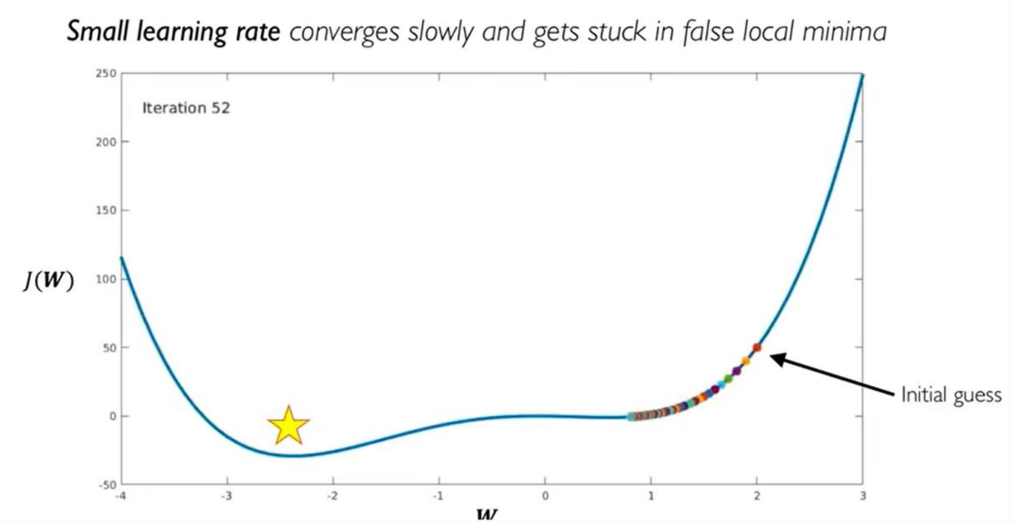
Source: [google]()

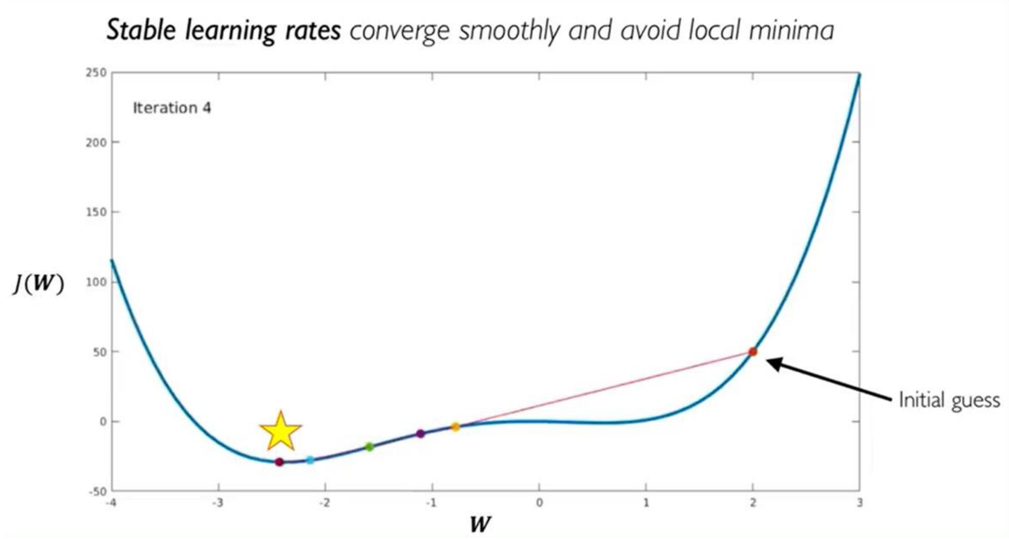
HParams dashboard is a tool for Hyperparameter tuning. It helps with this process of identifying the best experiment or most promising sets of hyperparameters.

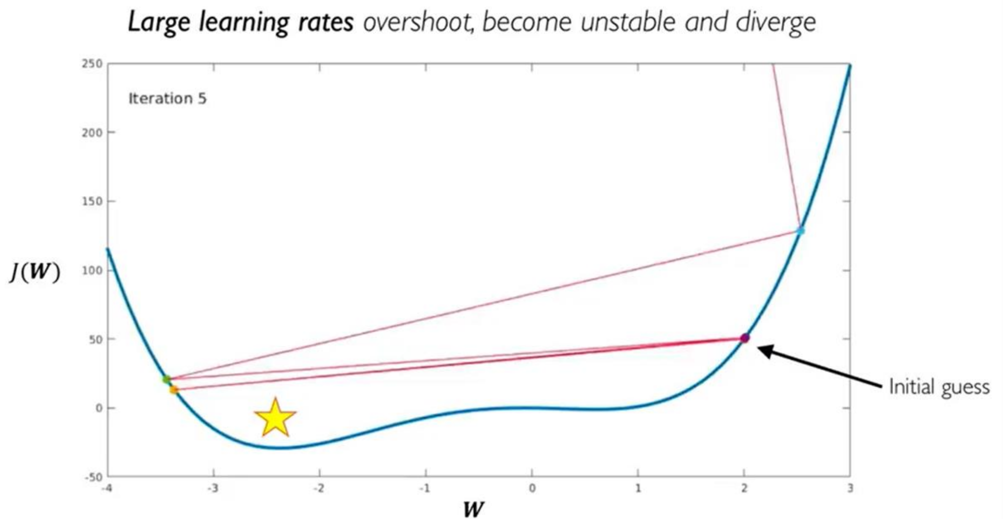
Source: [Google]()

Loss function Optimization - Learning Rate









Source: [MIT 6.S191 (2020): Introduction to Deep Learning]

Learning rate

- With low learning rate the improvements will be linear
- Higher learning rate can decay the loss faster, but it can get stuck
- Initially keep the learning rate higher
- Later decrease the learning rate

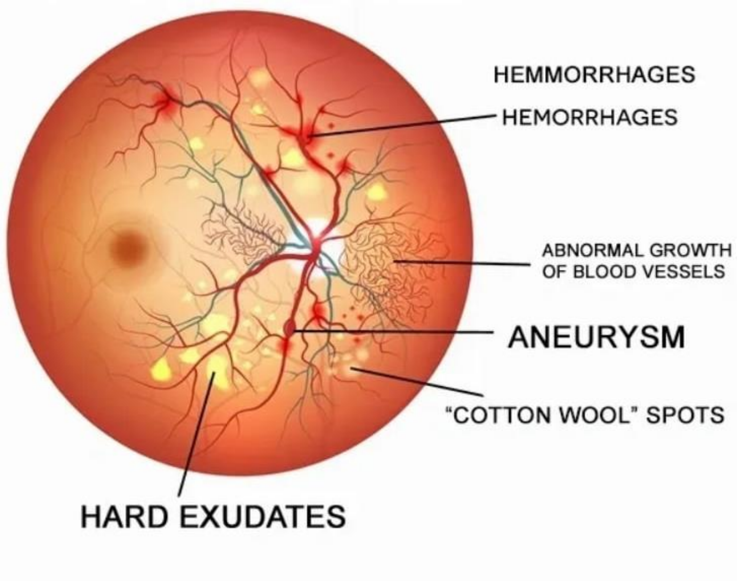
Convolutional Neural Networks

Convolutional Neural Network

- Convolutional Neural Networks (CNNs) are Neural Networks that are designed to work efficiently for Images. Also, CNNs are used for NLP, Speech recognition and other tasks too.
- Few real world examples include:

Fingerprint Recognition

Face Recognition



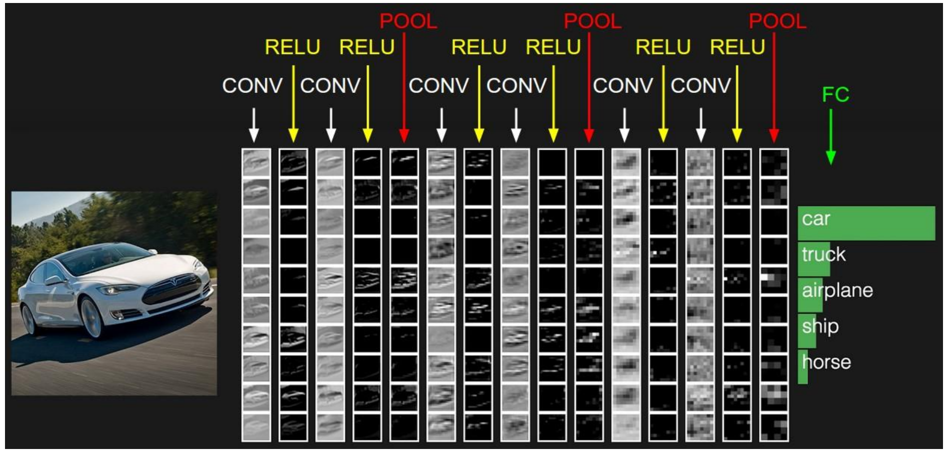
Diabetic Retinopathy Detection

Source: [Convolutional Neural Networks for Visual Recognition]()

Convolutional Neural Network Structure

- Convolutional Neural Networks (CNNs) are Neural Networks that are designed to work efficiently for Images.
- Images can be Black/white, Grayscale and RGB (3 channels).
- A CNN is made up of Layers. Unlike a regular Neural Network, the layers of a CNN have neurons arranged in 3 dimensions: width, height, depth. The depth indicates the number of channels in the input image.
- A CNN architecture is a list of Layers that transform the image volume into an output volume (e.g. holding the class scores)

  e.g., 32 X 32 X 3 input CIFAR-10 image 1 X 1 X 10 size output (10 output classes to choose)
- Multiple types of layers are used to build ConvNet architectures. We will stack these layers to form a full CNN architecture.
  - Convolutional Layer (CONV)
  - Pooling Layer (POOL)
  - Fully-Connected Layer (FC Layer)
  - Activation Layer (e.g., RELU layer)
- Each Layer accepts an input 3D volume and transforms it to an output 3D volume through a differentiable function
- Each Layer may or may not have parameters (e.g. CONV/FC do, RELU/POOL don't)
- Each Layer may or may not have additional hyperparameters (e.g. CONV/FC/POOL do, RELU doesn't)



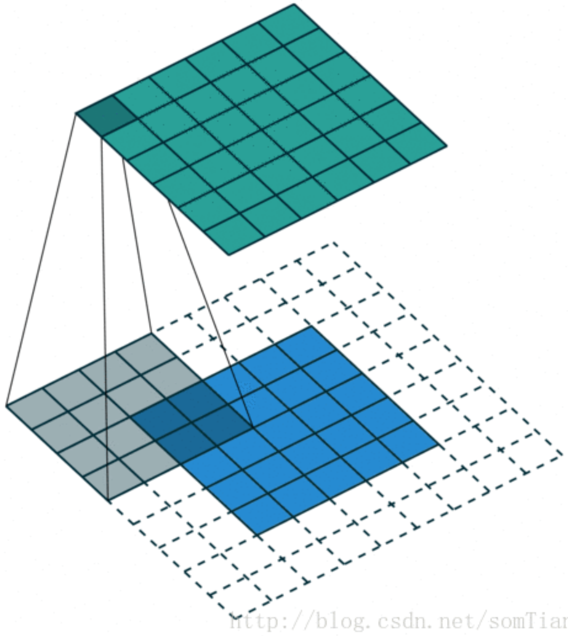
Source: *Convolutional Neural Networks for Visual Recognition*

CIFAR-10 CNN Classification Example

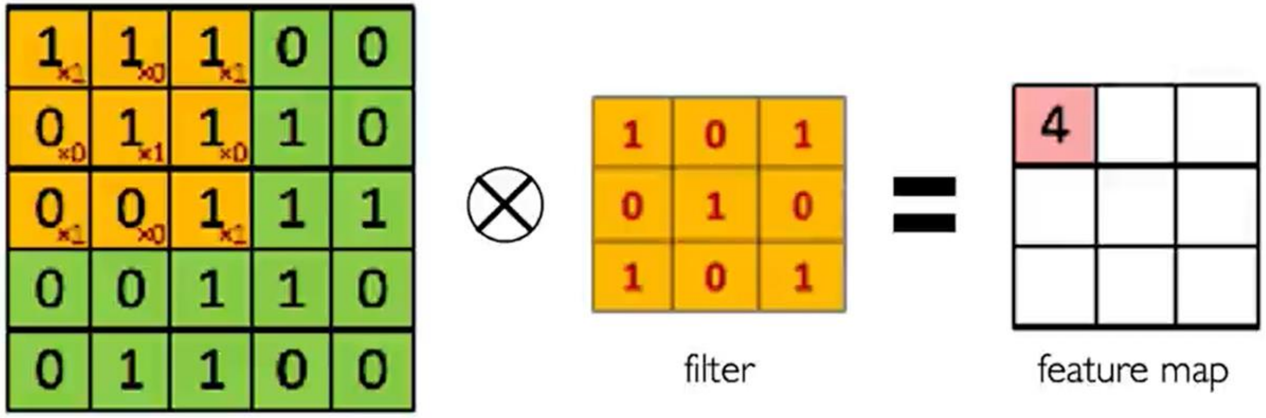
Convolution Operation

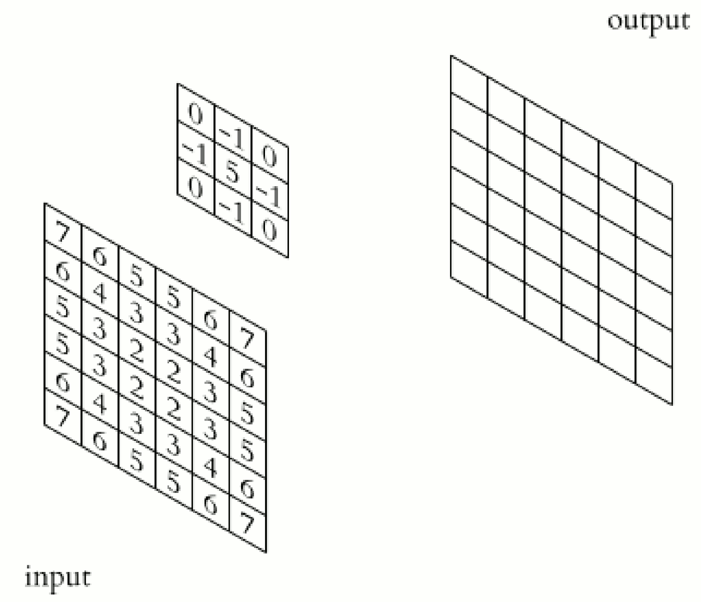
- Convolution is using a ‘kernel’ to extract certain ‘features’ from an input image.
- A kernel is a matrix, which is slid across the image and multiplied with the input such that the output is enhanced in a certain desirable manner. Kernel is also called a filter.
- Images can be Black/white, Grayscale and RGB (3 channels)
- The output of convolutional layers are feature maps

Filter Size = 4 X4, Padding = 2, Stride = 1 Example



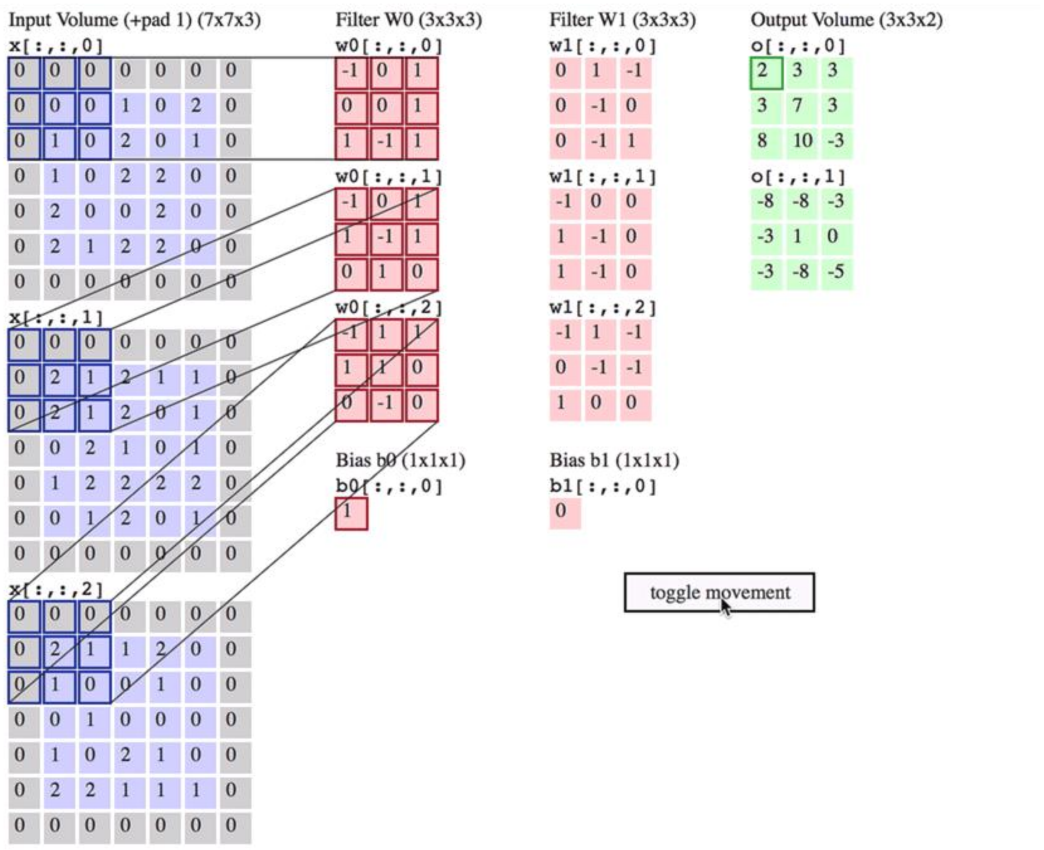
We slide the 3x3 filter over the input image, element-wise multiply, and add the outputs:

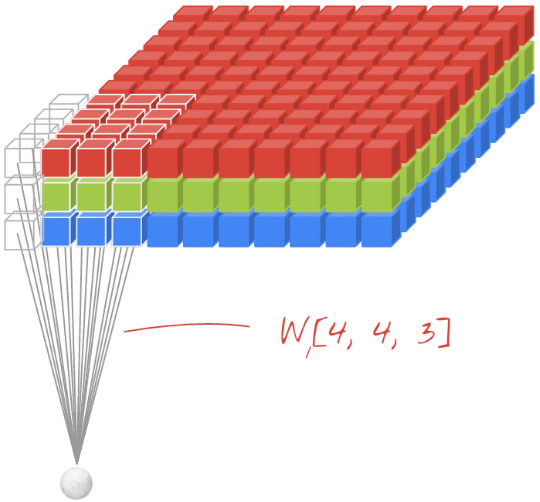




Filter Size = 3X3, Padding = 1, Stride = 1 Example

Convolutional layer – multiple filters example





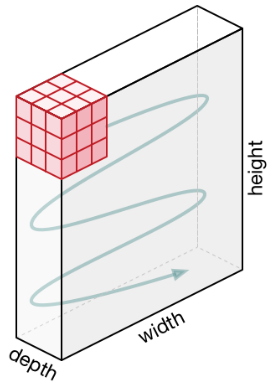
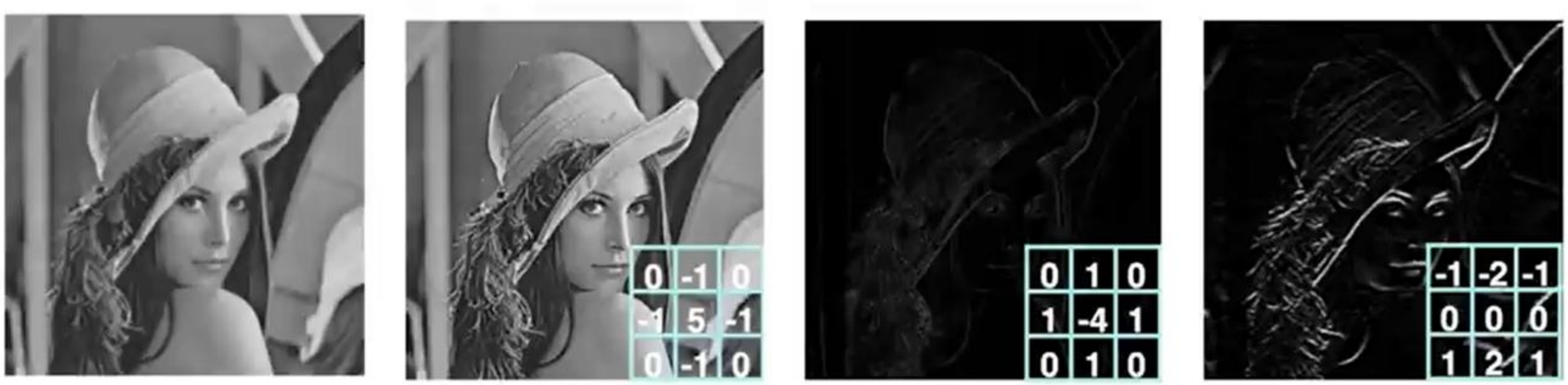


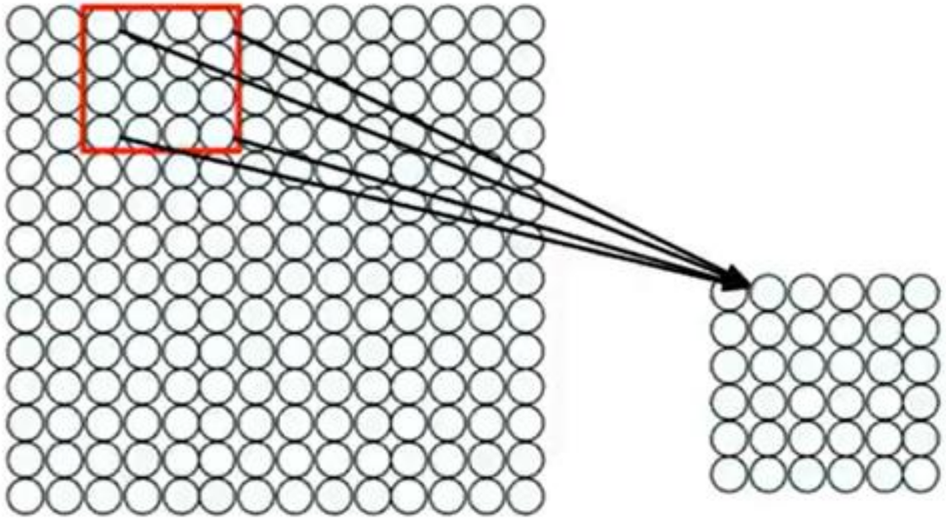
Image source: [cs231 notes]()

Feature Extraction with Convolution - Creating Feature Maps

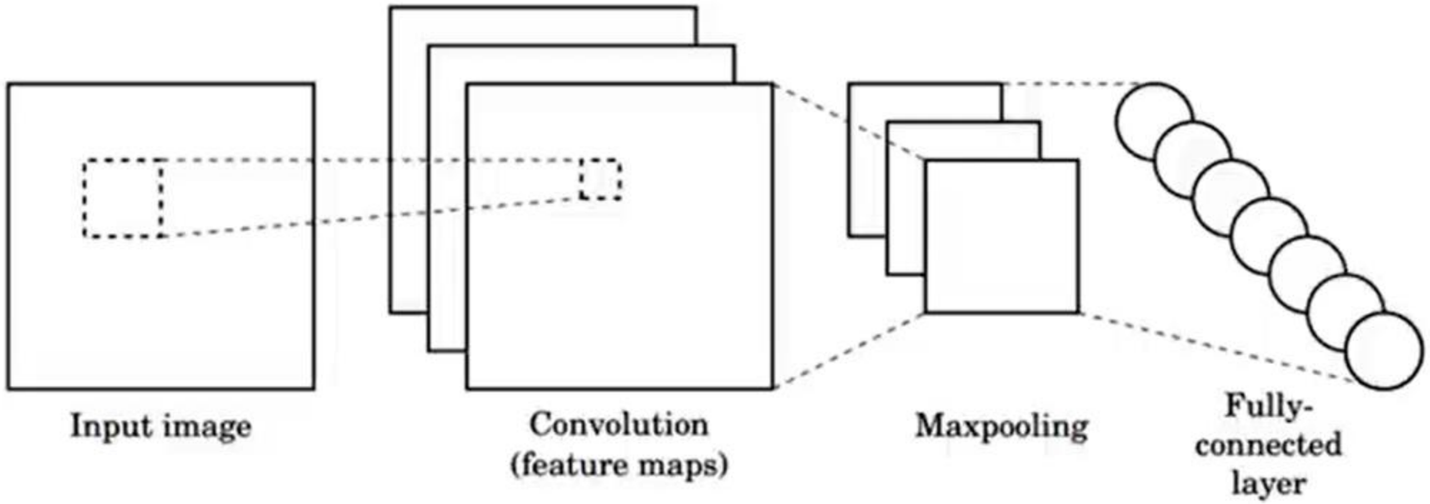


Original | Sharpen | Edge Detect | "Strong" Edge Detect

- Apply a set of weights (Filter) to extract local features
- Different types of Filters are used to extract features.
- Spatially share parameters (weights) of each filter in a convolution layer
- Reducing number of connections between layers



CNNs for Classification



1. **Convolution**: Apply filters to generate feature maps.
2. **Non-linearity**: Often ReLU.
3. **Pooling**: Downsampling operation on each feature map.

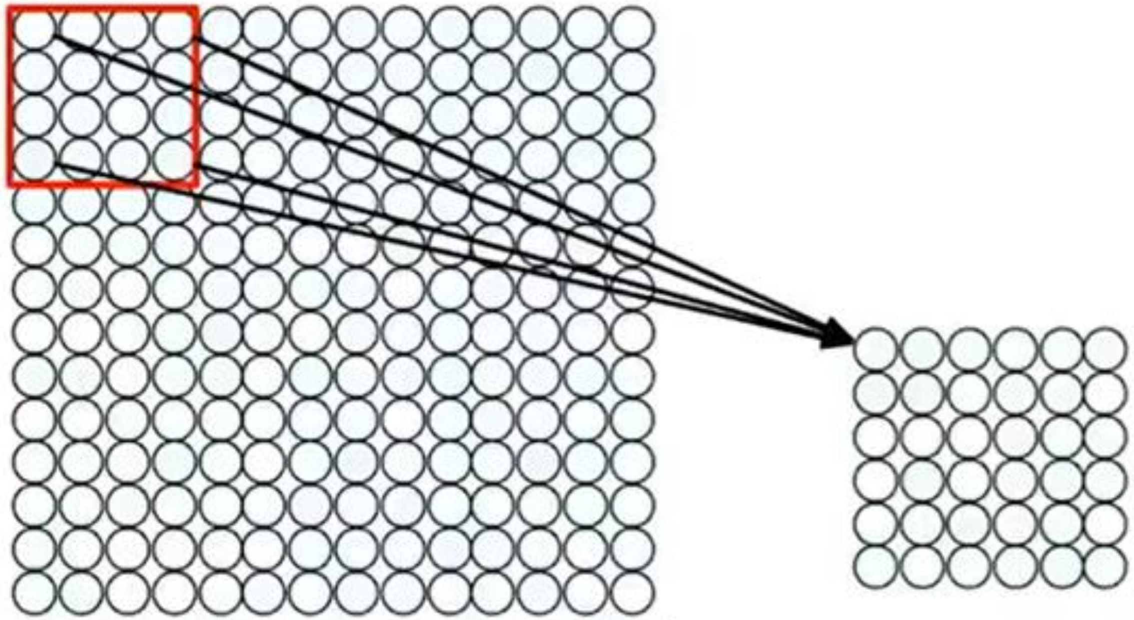
```
tf.keras.layers.Conv2D
tf.keras.activations.*
tf.keras.layers.MaxPool2D
```

Train model with image data.
Learn weights of filters in convolutional layers.

Source: [MIT 6.S191 (2020): Convolutional Neural Networks]

Convolution Layers: Local Connectivity

`tf.keras.layers.Conv2D`



4x4 filter: matrix of weights w_{ij}

$$\sum_{i=1}^{4}\sum_{j=1}^{4} w_{ij}\, x_{i+p,j+q} + b$$

for neuron (p,q) in hidden layer

For a neuron in hidden layer:

- Take inputs from patch
- Compute weighted sum
- Apply bias

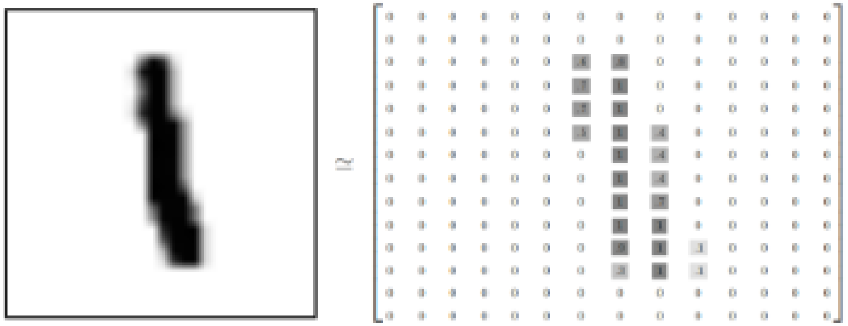
1) applying a window of weights
2) computing linear combinations
3) activating with non-linear function

Note: Rectified Linear Unit (ReLU) is used after every Convolution operation to replace all negative pixel values in the feature map by zero. tanh or sigmoid can also be used instead of ReLU, but ReLU has been found to perform better in most situations.

Source: MIT 6.S191 (2020): Convolutional Neural Networks

MNIST Handwritten Digit Recognition

0–9 handwritten digit recognition:



CNN

Deep Neural Network

“1”

28 x 28

MNIST Data maintained by Yann LeCun: http://yann.lecun.com/exdb/mnist/
Keras provides data sets loading function at http://keras.io/datasets

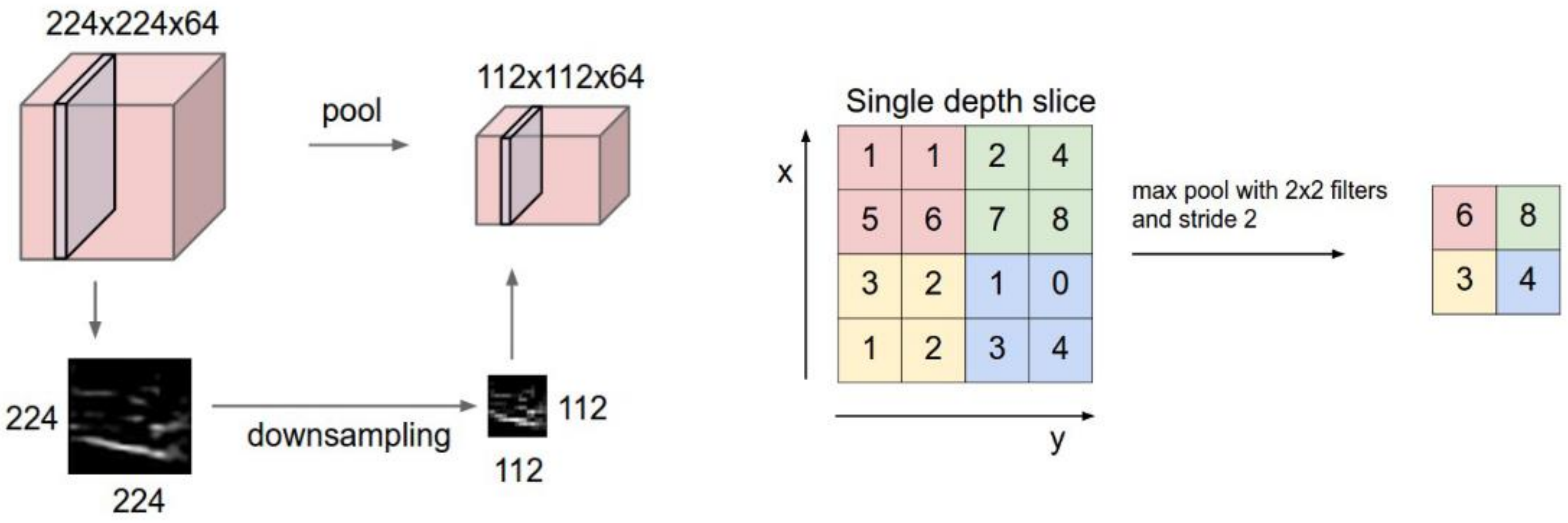
State of the Art Object Classification (Handwritten digit recognition) Results

mnist-cnn-example.py

Source: MNIST

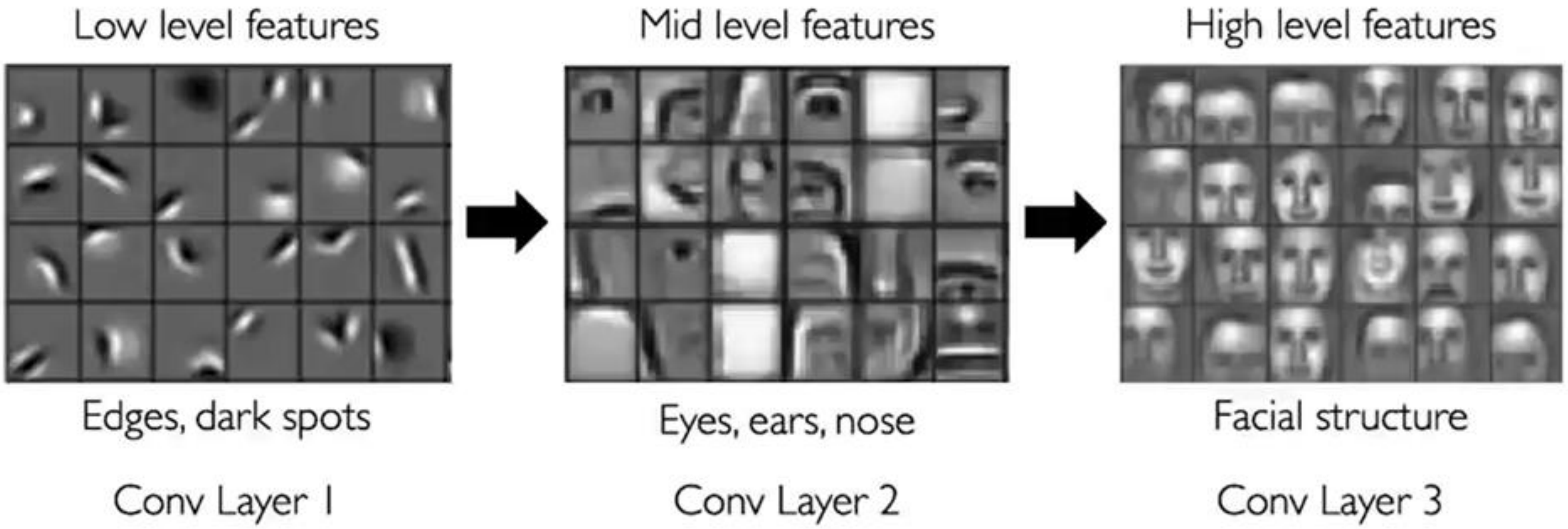
Pooling Layer

- Pooling layer down samples the volume spatially, independently in each depth slice of the input.
- Popular pooling is Max pooling. Mean pooling can also be used.



Source: https://lipyeow.github.io/

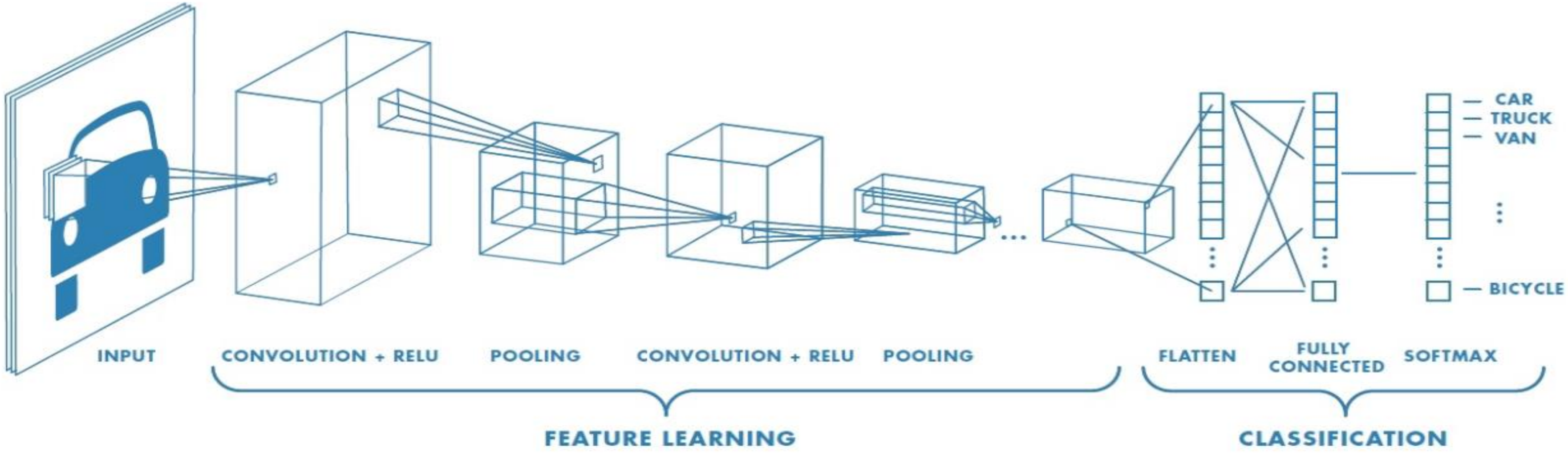
Deep CNNs – Representation Learning



Note: The further you advance into the Deep CNN, the more complex the features your nodes can recognize, since they aggregate and recombine features from the previous layer

Source: MIT 6.S191 (2020): Convolutional Neural Networks

CNNs for Classification: Feature Learning + Classification probabilities



Feature Learning

- Feature Learning in image with convolution
- Non-linearity using activation functions
- Reduce dimensionality and preserve spatial invariance with Pooling
- Convolution and pooling layers extract high level input features.

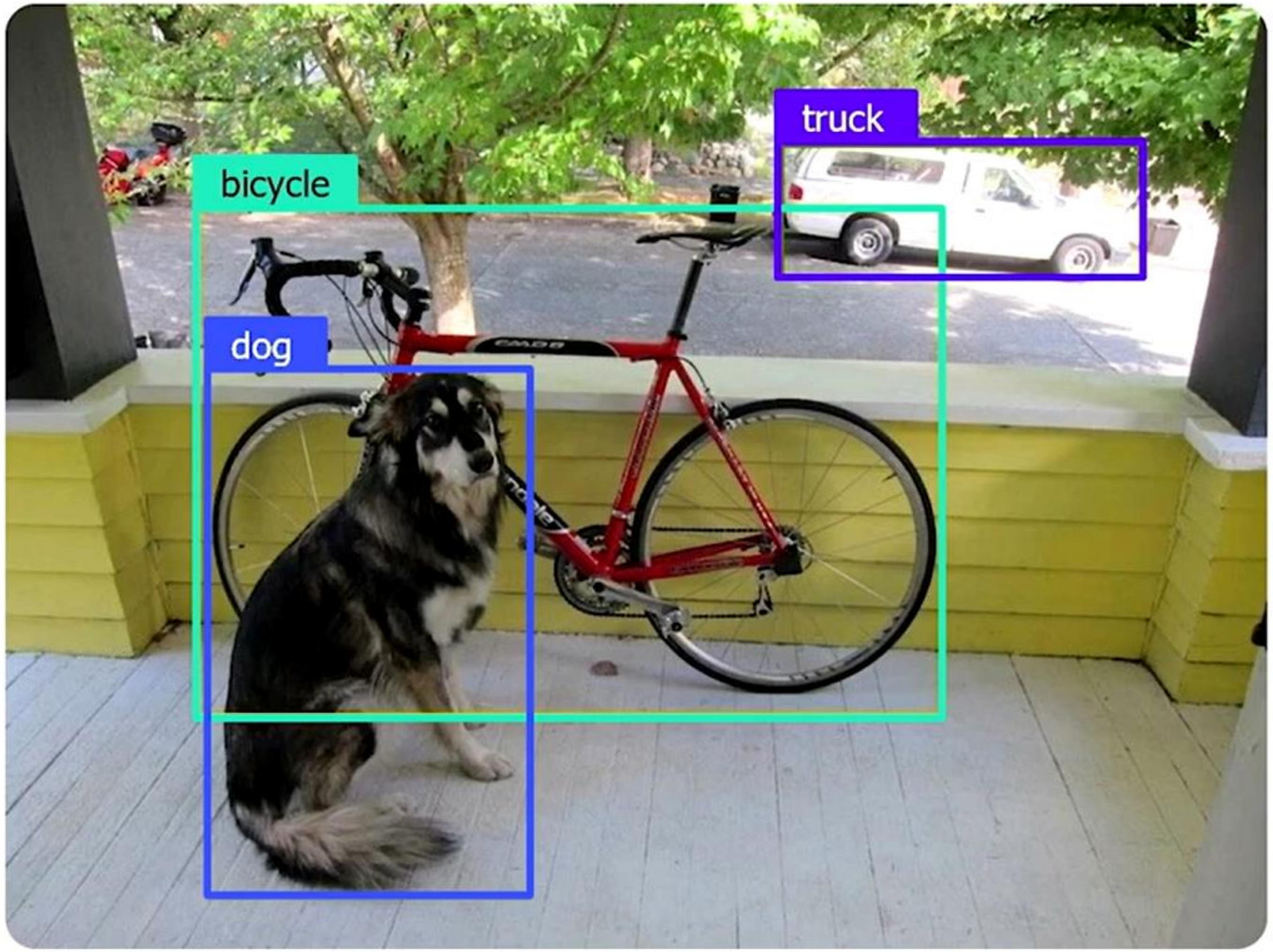
Classification

❖ Fully connected layer uses the extracted features from Feature Learning pipeline.

❖ Express output as probability of image belonging to a particular class with the help of SoftMax function.

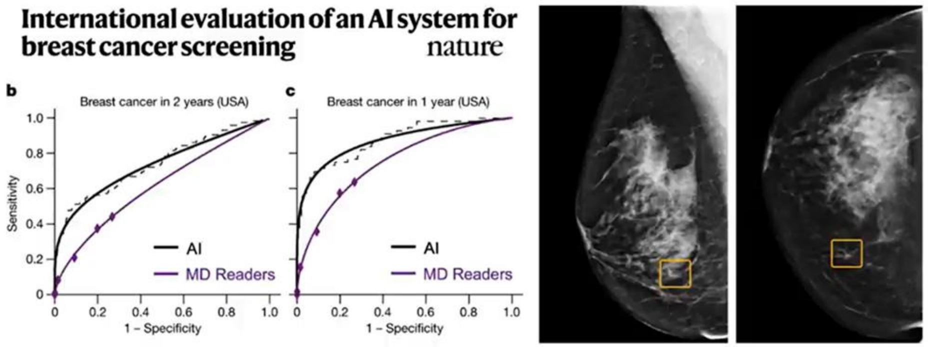
Source: MIT 6.S191 (2020): Convolutional Neural Networks

Object Detection using a Deep learning model (e.g., YOLO)



CNN Architecture: Many applications

Classification: Breast Cancer Screening



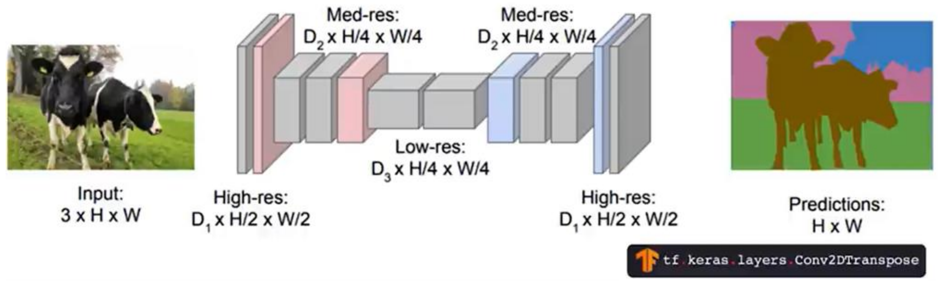
CNN-based system outperformed expert radiologists at detecting breast cancer from mammograms

Breast cancer case missed by radiologist but detected by AI

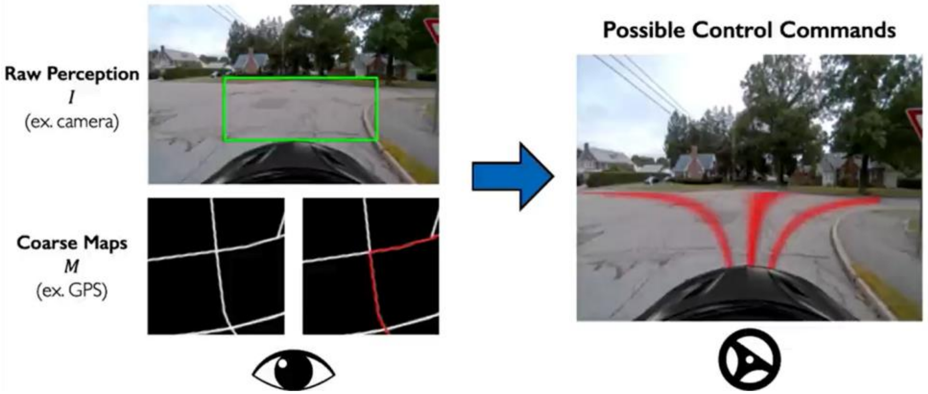
Object Detection

Semantic Segmentation: Fully Convolutional Networks

FCN: Fully Convolutional Network.
Network designed with all convolutional layers, with **downsampling** and **upsampling** operations



Continuous Control: Navigation from Vision



Probabilistic Control

Source: [MIT 6.S191 (2020): Convolutional Neural Networks]