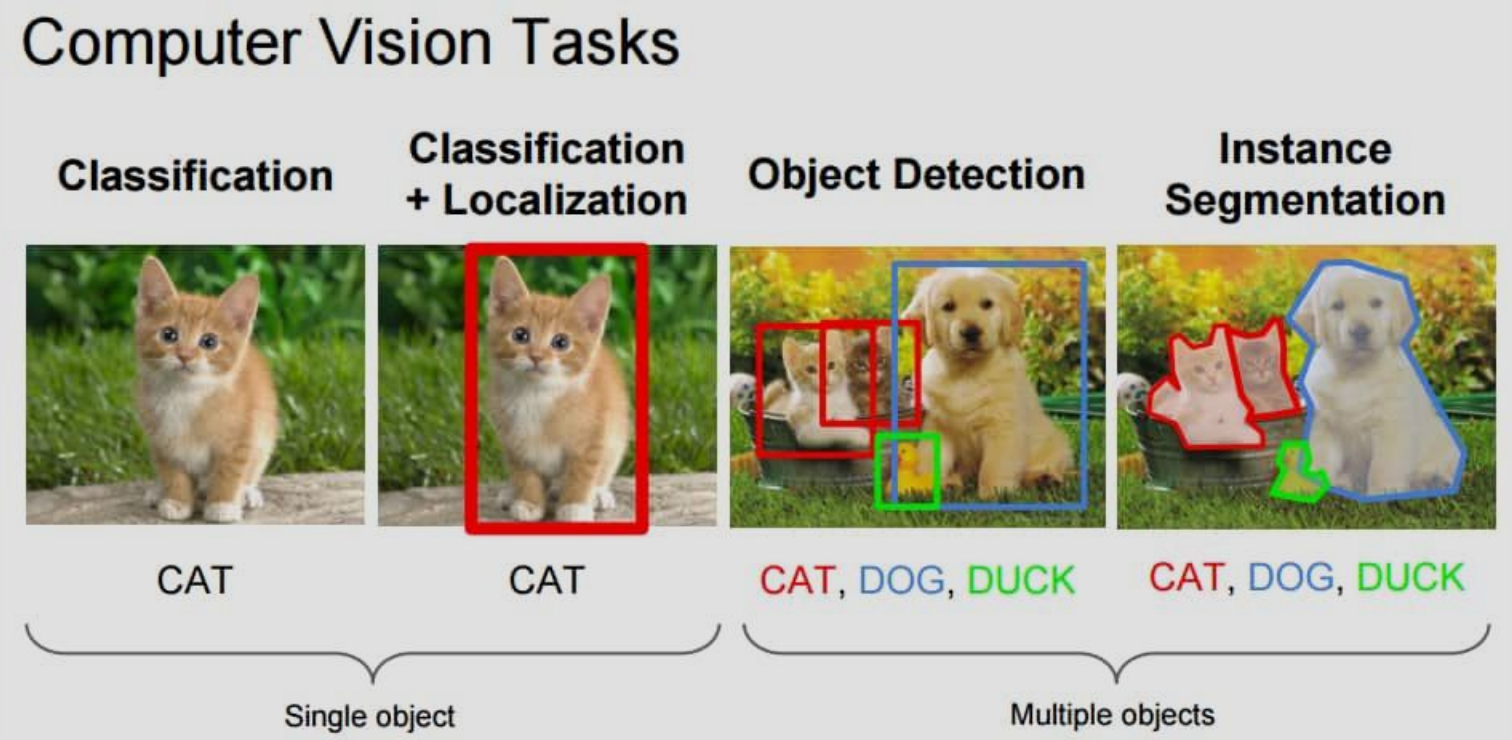
nodeflux

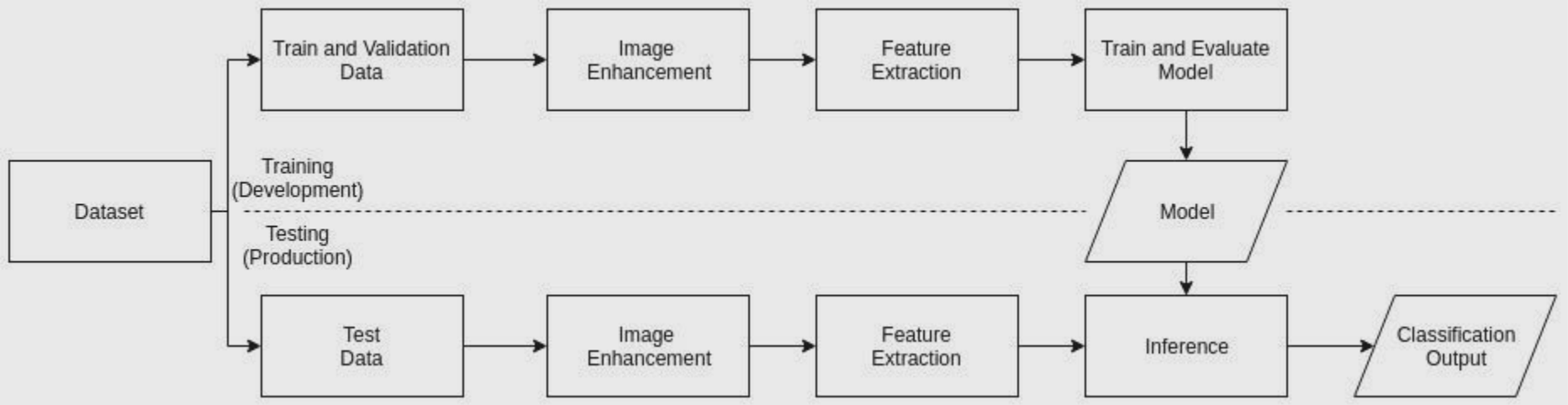
A Distributed Computation Platform

What is Image Classification?



Source: Fei-Fei Li, Andrej Karpathy & Justin Johnson (2016) cs231n, Lecture 8 - Slide 8, *Spatial Localization and Detection* (01/02/2016). Available: http://cs231n.stanford.edu/slides/2016/winter1516_lecture8.pdf

General Pipeline of Image Classification

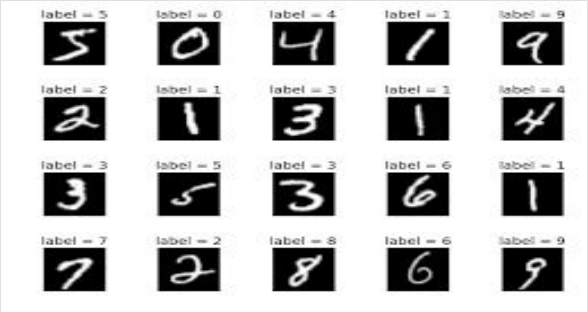


Use Case

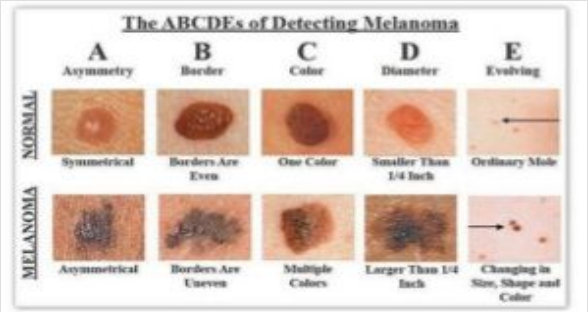
Face Recognition

Product Recognition

Optical Character Recognition



Cancer Detection



Source: google images

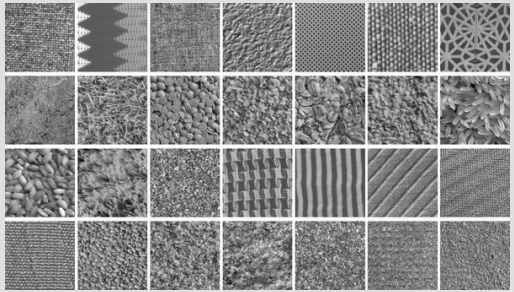
nodeflux

What is Feature?

Color

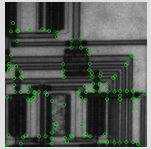
Shape

Texture

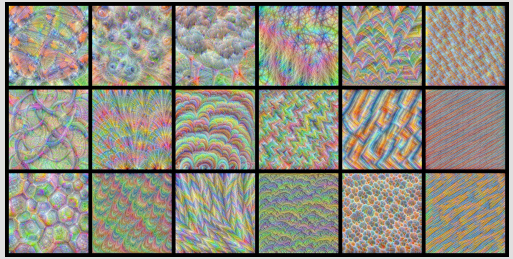


Edge

Corner



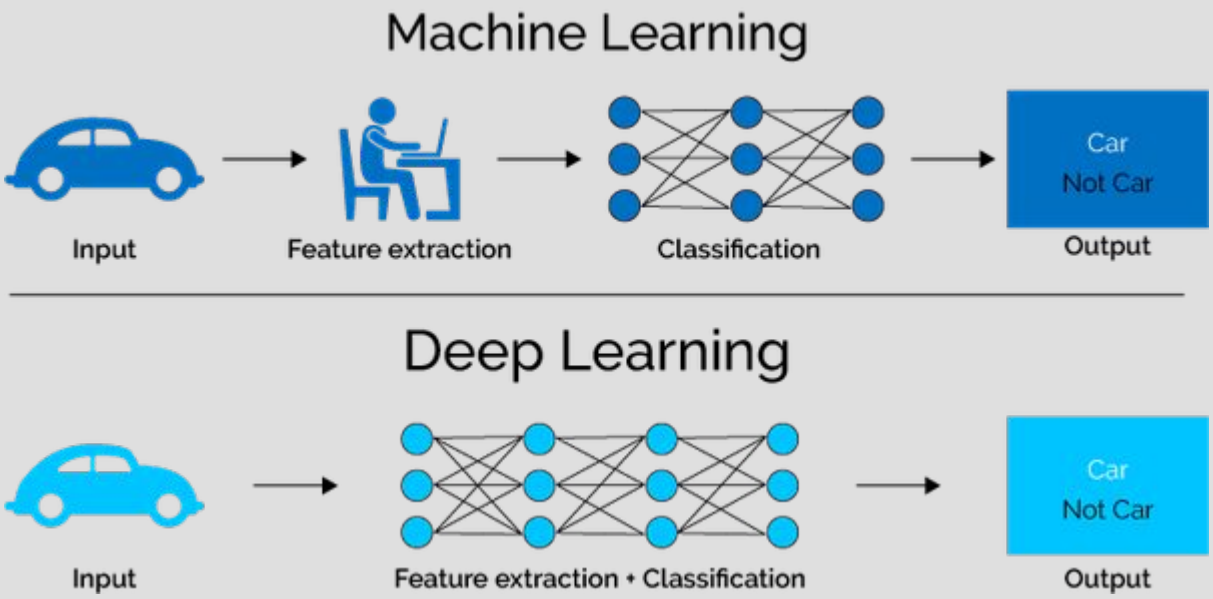
etc



Source: google images

nodeflux

Deep Learning vs Traditional ML



Source:
https://www.google.com/url?sa=i&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwiHu_uCidvgAhVXiHAKHfHEDbMQjB16BAgBEAQ&url=https%3A%2F%2Ftowardsdatascience.com%2Fwhy-deep-learning-is-needed-over-traditional-machine-learning-1b6a99177063&psig=AOvVaw37F9-o0G3Jol_1aHWOSSOp&ust=1551327928244502

nodeflux

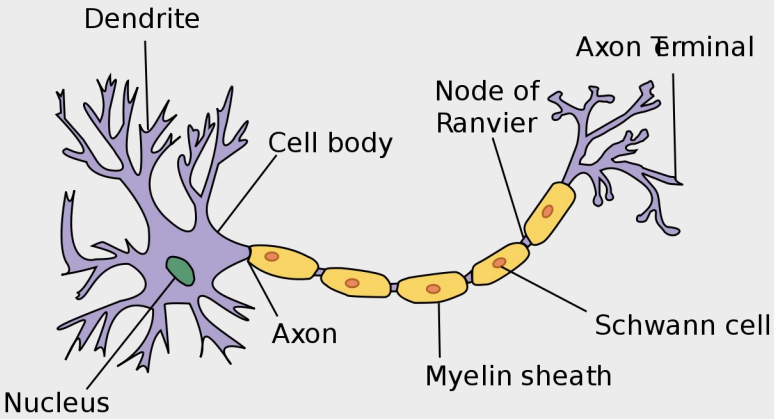
Extending Vision Beyond Imagination

Section 2 - Intro to CNN

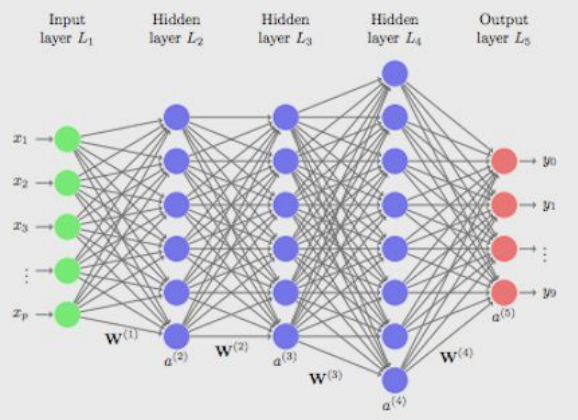
www.nodeflux.io

Neural Network

Inspired by:

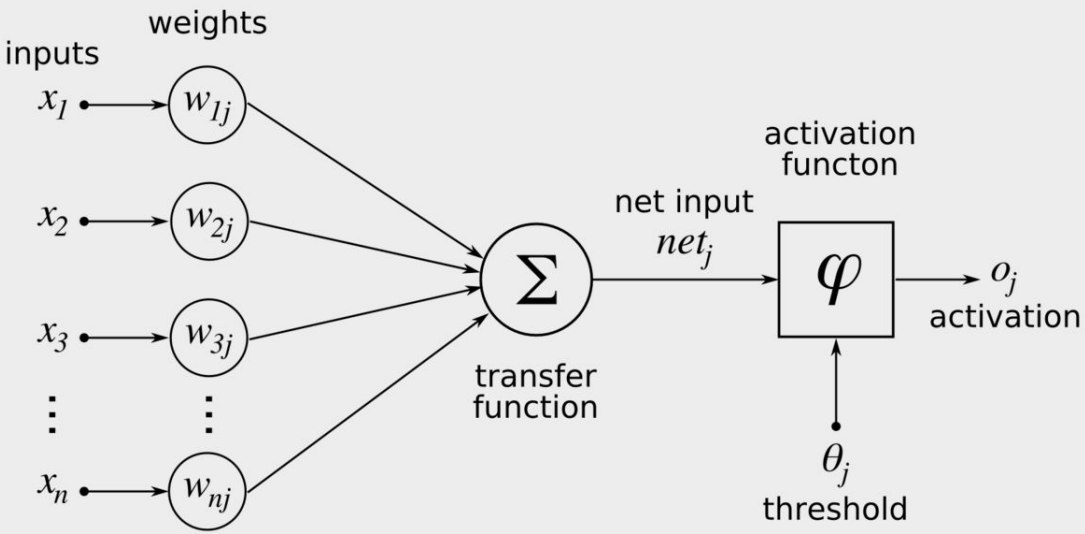


And modelled as this:



Source: google images

Neural Network Behind the Scenes

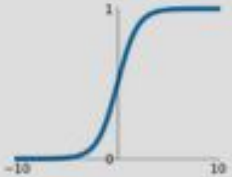


Source: https://evanlariviereblog.wordpress.com/2018/04/09/machine-learning-minimizing-thetas-with-tensorflow/

Activation Function

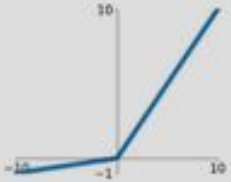
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



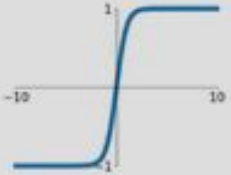
Leaky ReLU

$$\max(0.1x, x)$$



tanh

$$\tanh(x)$$

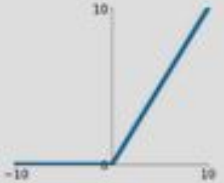


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

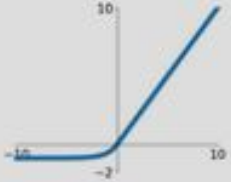
ReLU

$$\max(0, x)$$



ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Softmax

$$\sigma(x_j) = \frac{e^{x_j}}{\sum_i e^{x_i}}$$

Source: https://medium.com/@shrutijadon10104776/survey-on-activation-functions-for-deep-learning-9689331ba092

nodeflux

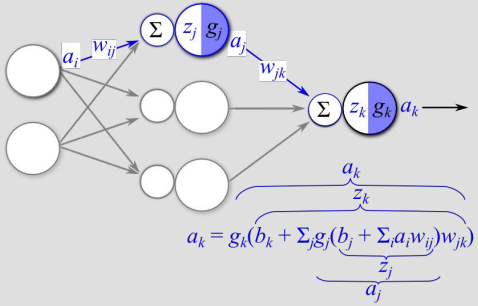
Loss Function

Regression Case	Classification Case
Mean Absolute Error/L1 $$\frac{\sum_{i=1}^{n} \mid y_i - \hat{y}_i \mid}{n}$$ Mean Square Error/L2 $$\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}$$	Cross Entropy Loss $$-(y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i))$$ SVM Loss $$SVMLoss = \sum_{j \neq y_i} max(0, s_j - s_{y_i} + 1)$$

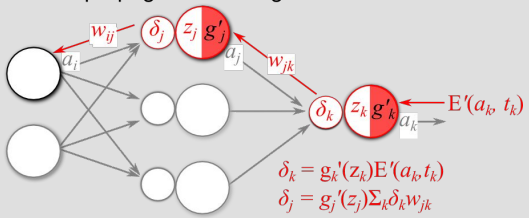
Source: google images

How model learns?

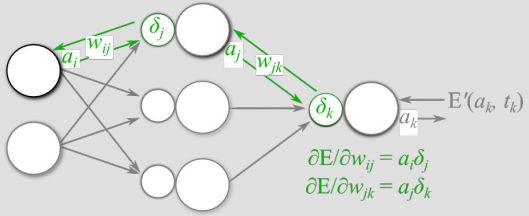
I. Forward-propagate Input Signal



II. Back-propagate Error Signals



III. Calculate Parameter Gradients



IV. Update Parameters

$w_{ij} = w_{ij} - \eta(\partial E/\partial w_{ij})$

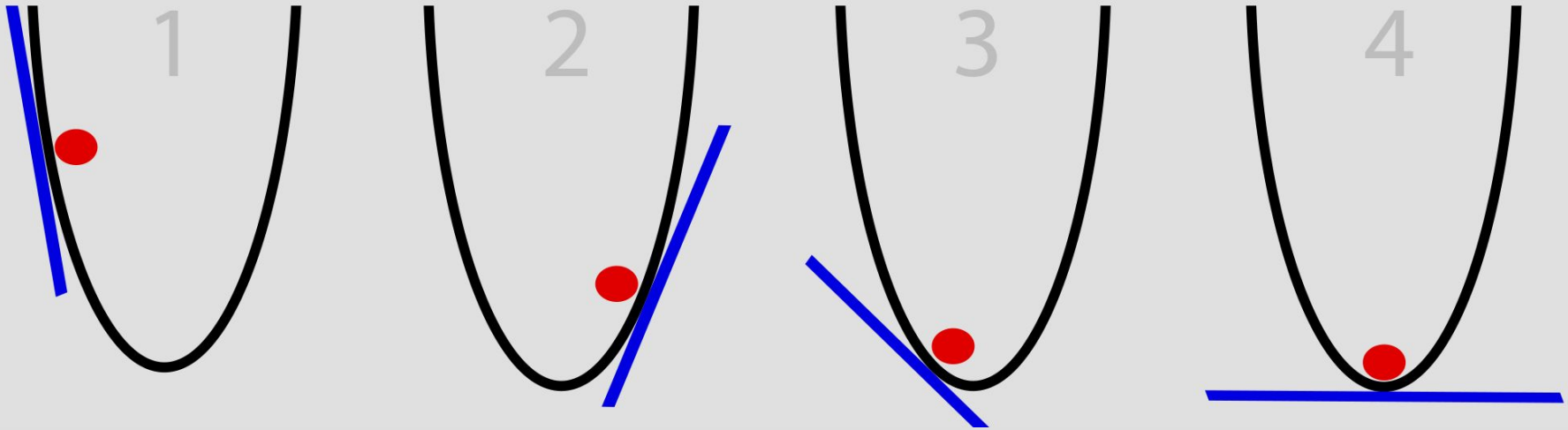
$w_{jk} = w_{jk} - \eta(\partial E/\partial w_{jk})$

for learning rate η

Source: https://theclevermachine.wordpress.com/tag/backpropagation/

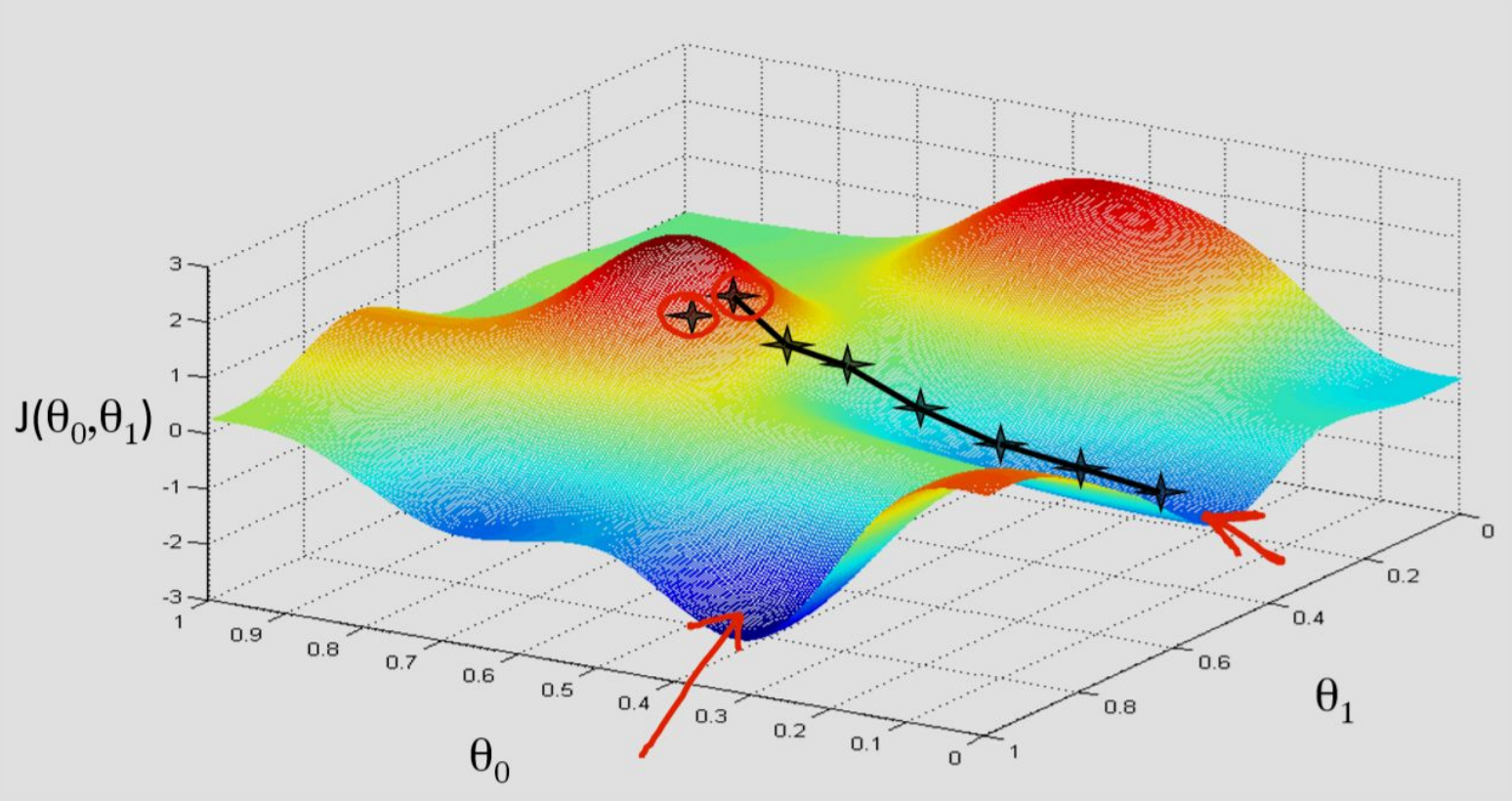
nodeflux

Gradient



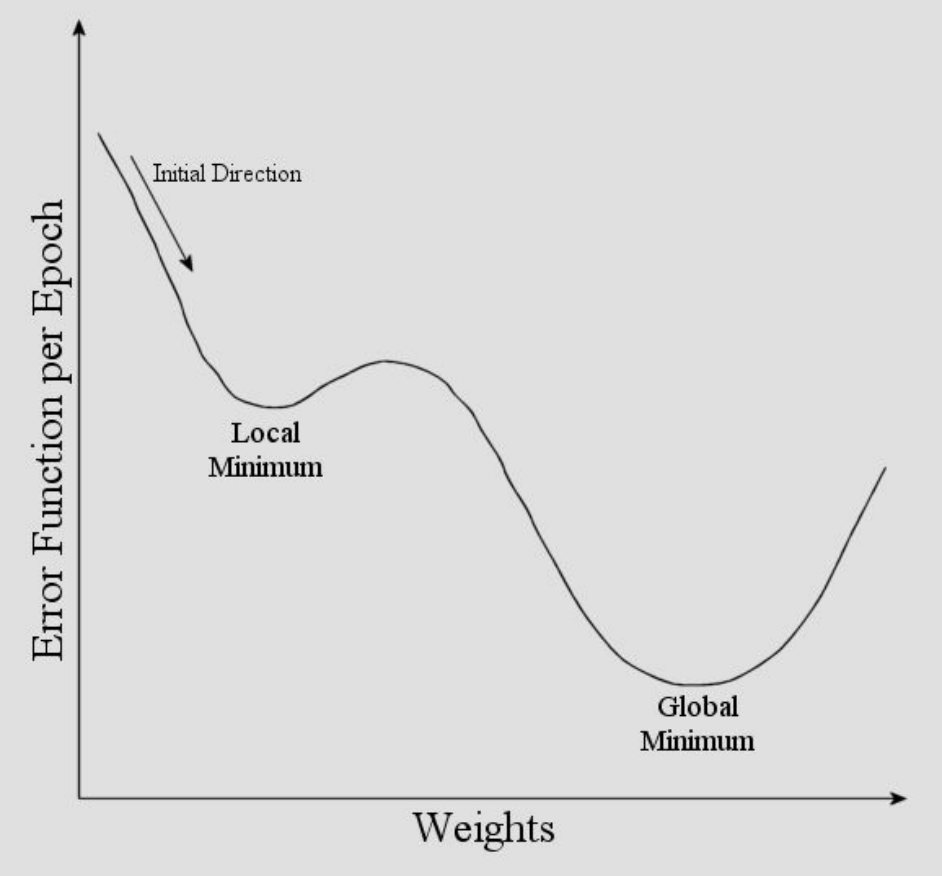
Source: google images

How model learns?



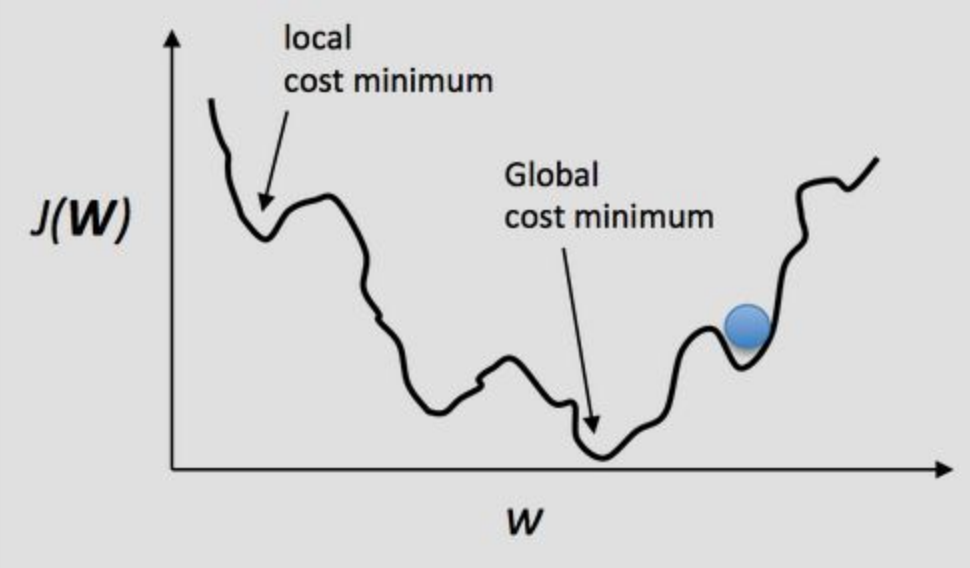
Source: google images

Local Minima vs Global Minima



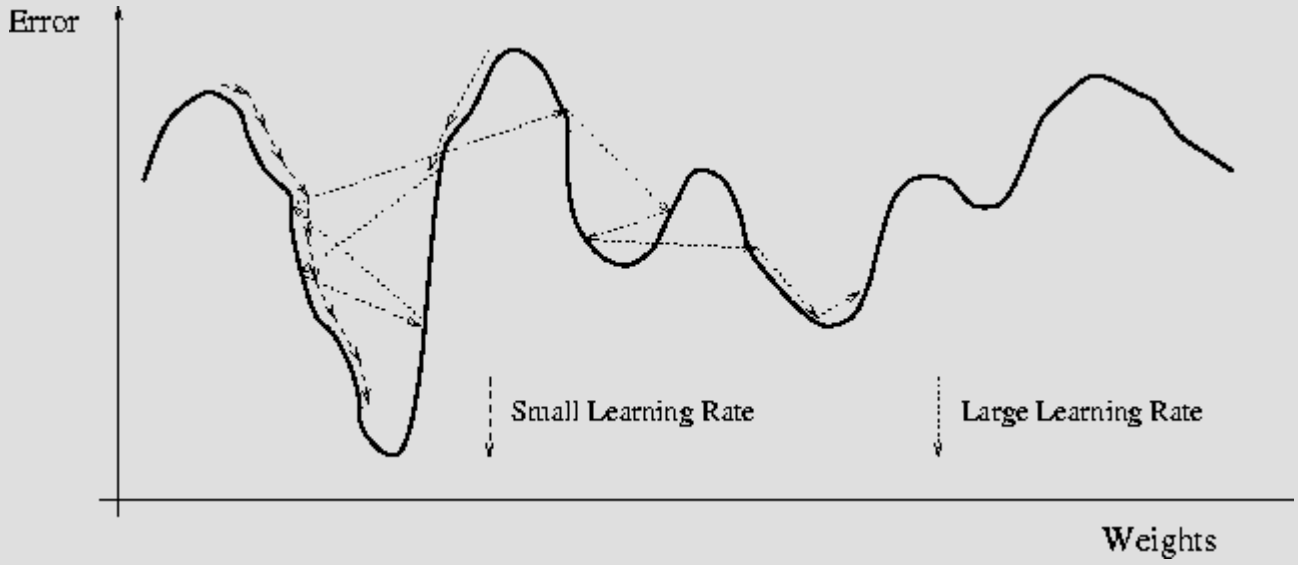
Source: google images

Momentum



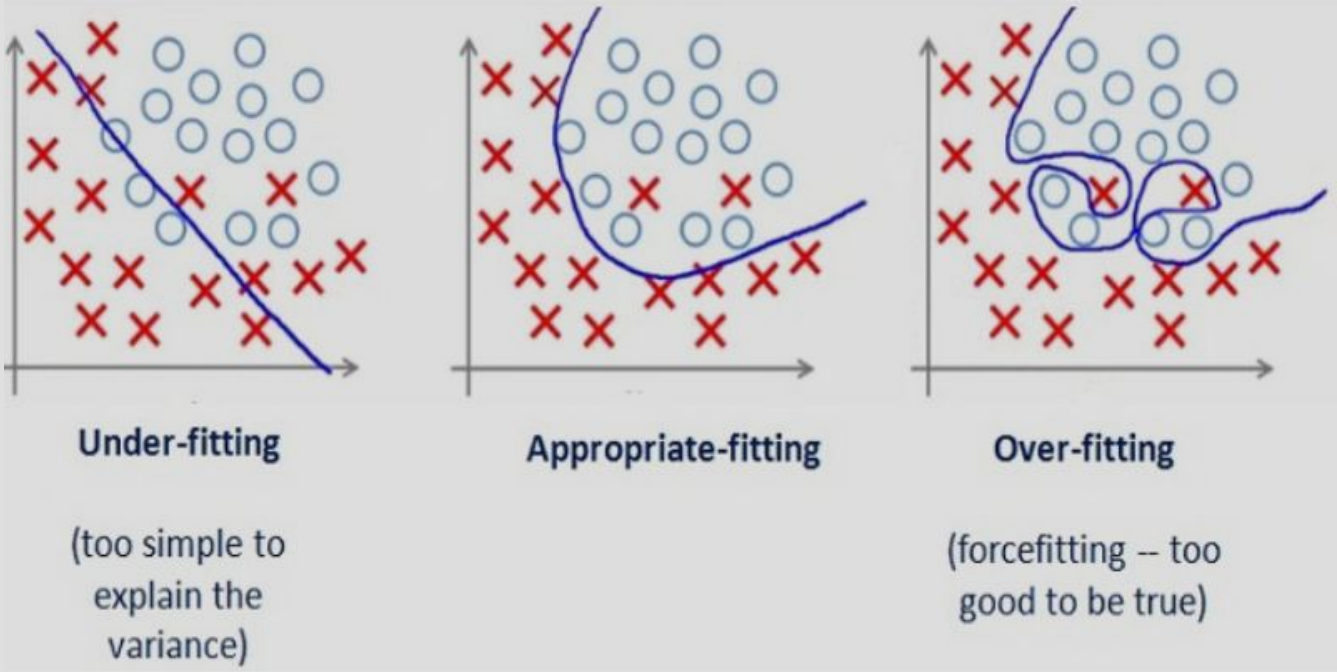
Source: google images

Learning Rate



Source: google images

Overfitting vs Underfitting



Source: https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6803a989c76

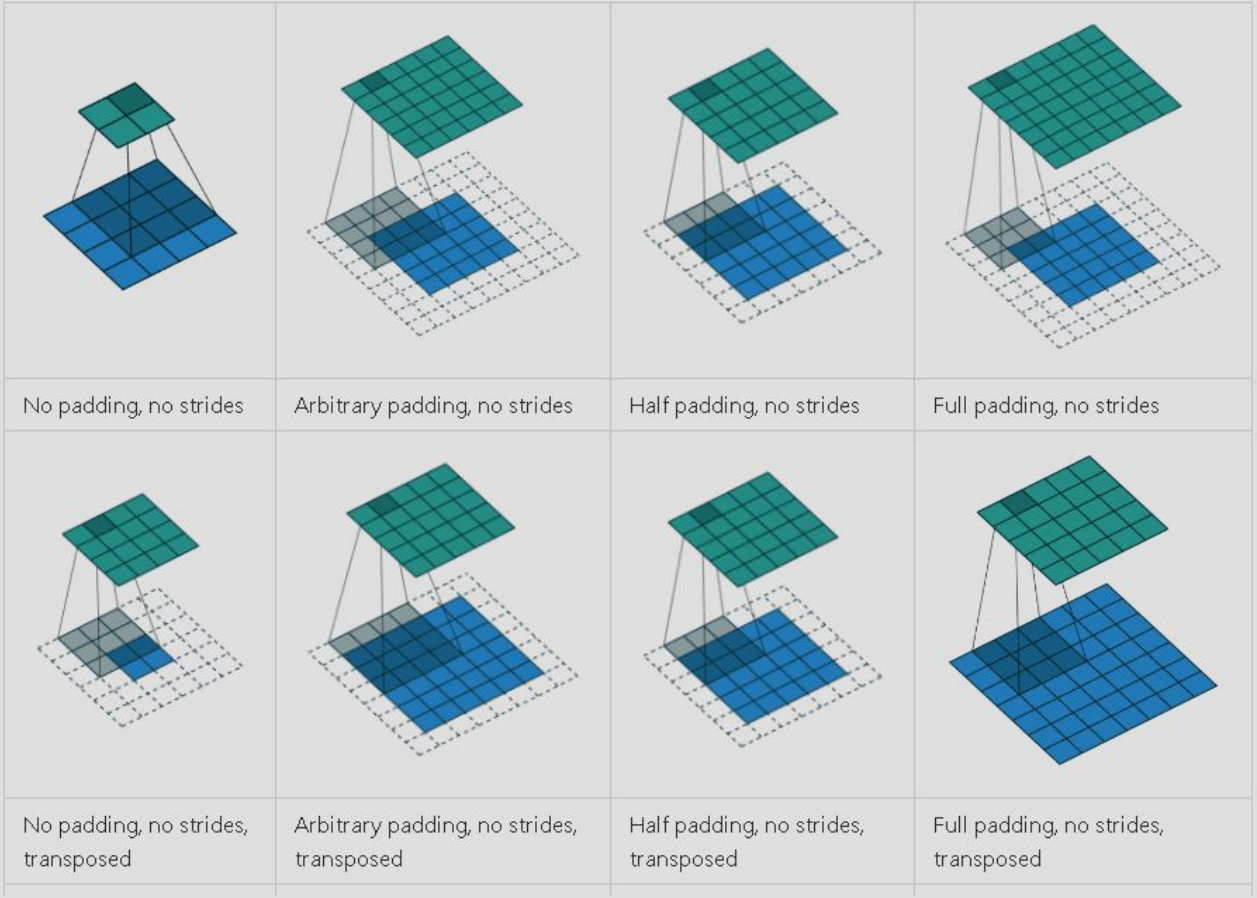
Convolutional Layer

1x1	1x0	1x1	0	0
0x0	1x1	1x0	1	0
0x1	0x0	1x1	1	1
0	0	1	1	0
0	1	1	0	0

4		

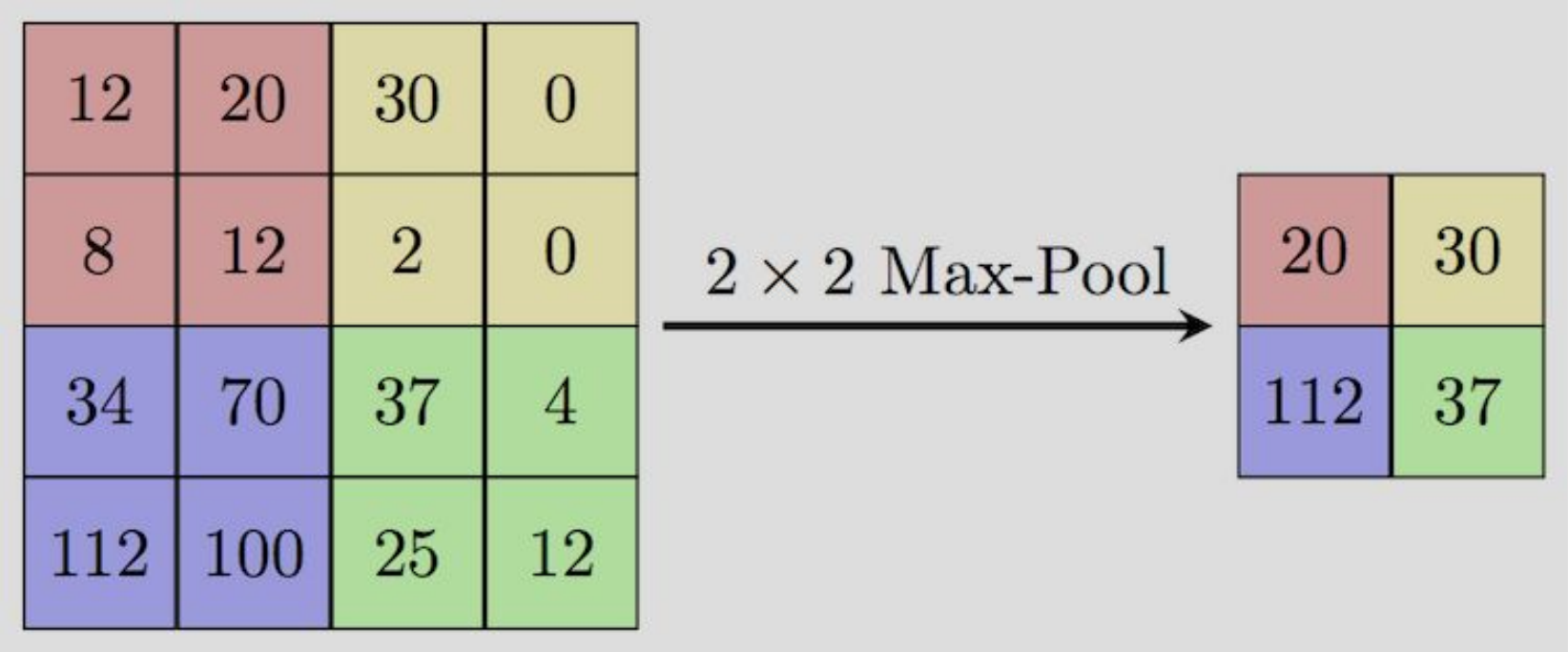
Source: https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

Stride and Padding



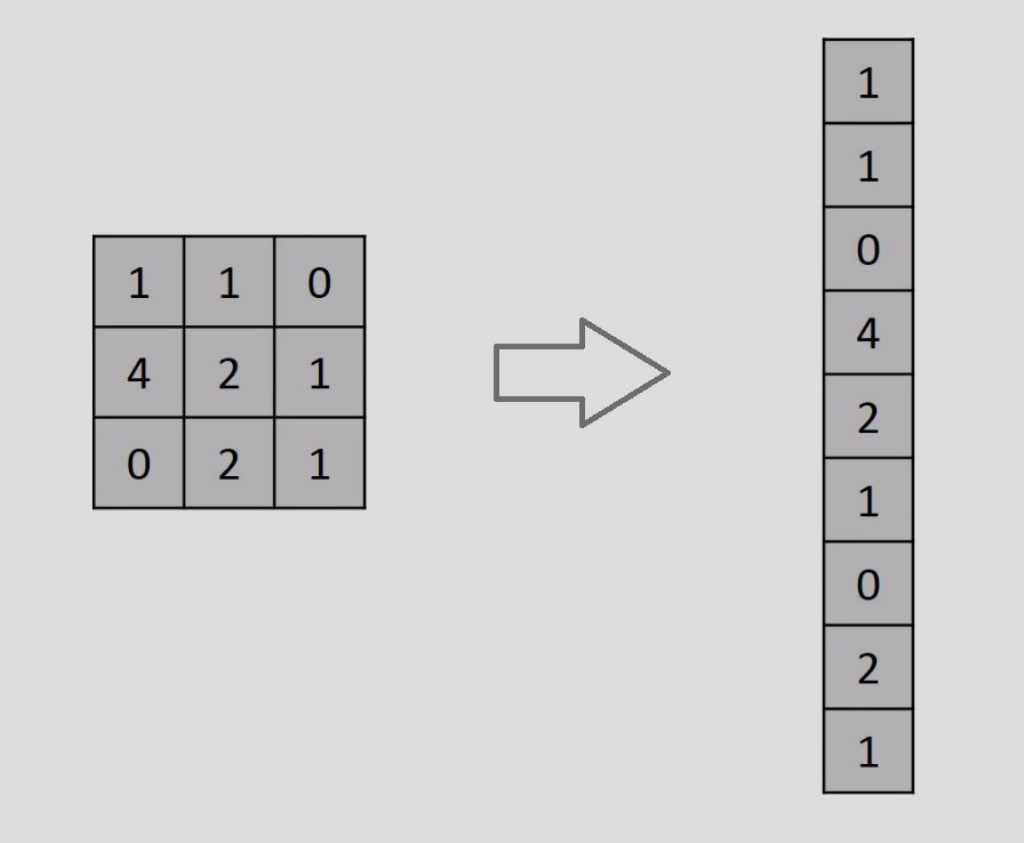
Source: https://github.com/vdumoulin/conv_arithmetic

Pooling Layer



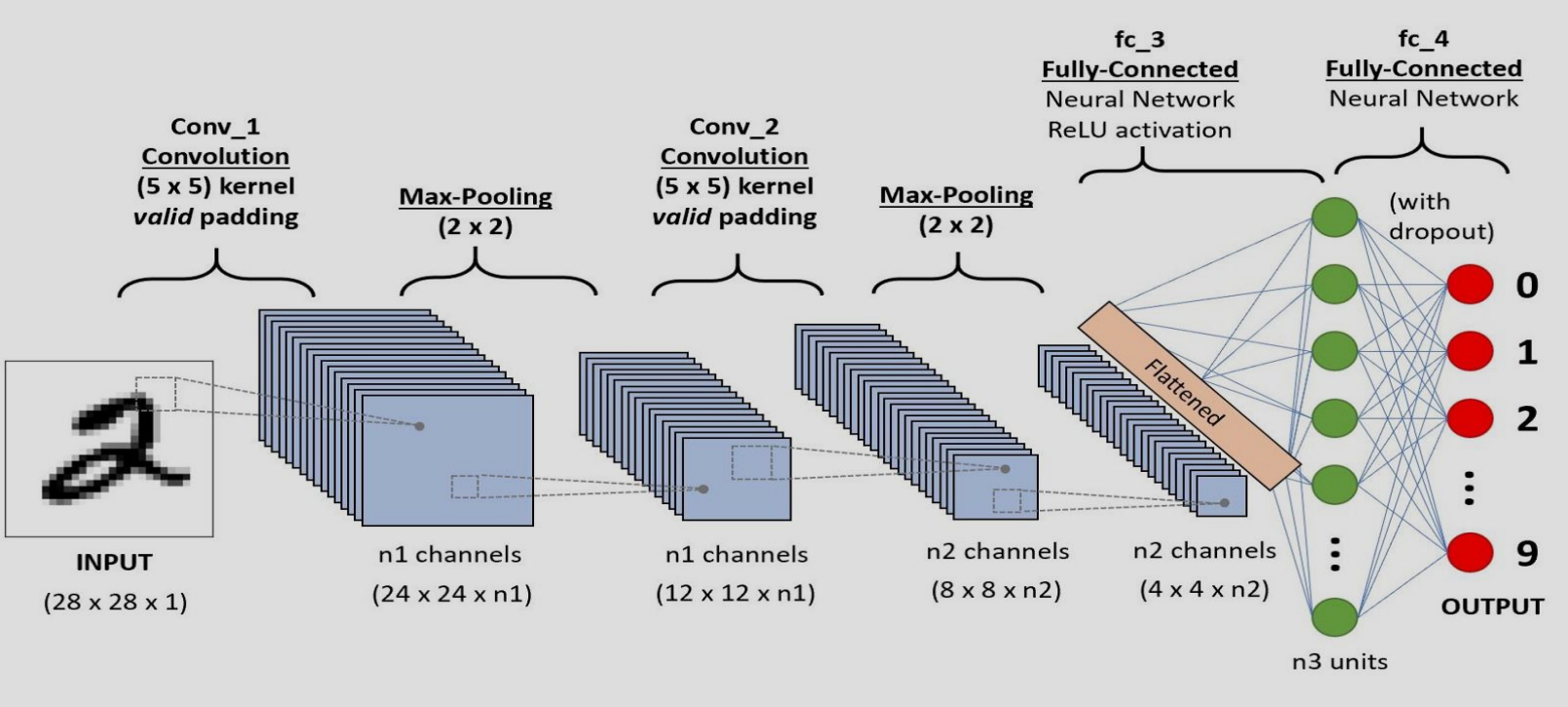
Source: http://cs231n.github.io/convolutional-networks/

Flattened



Source: https://rubikscode.net/2018/02/26/introduction-to-convolutional-neural-networks/

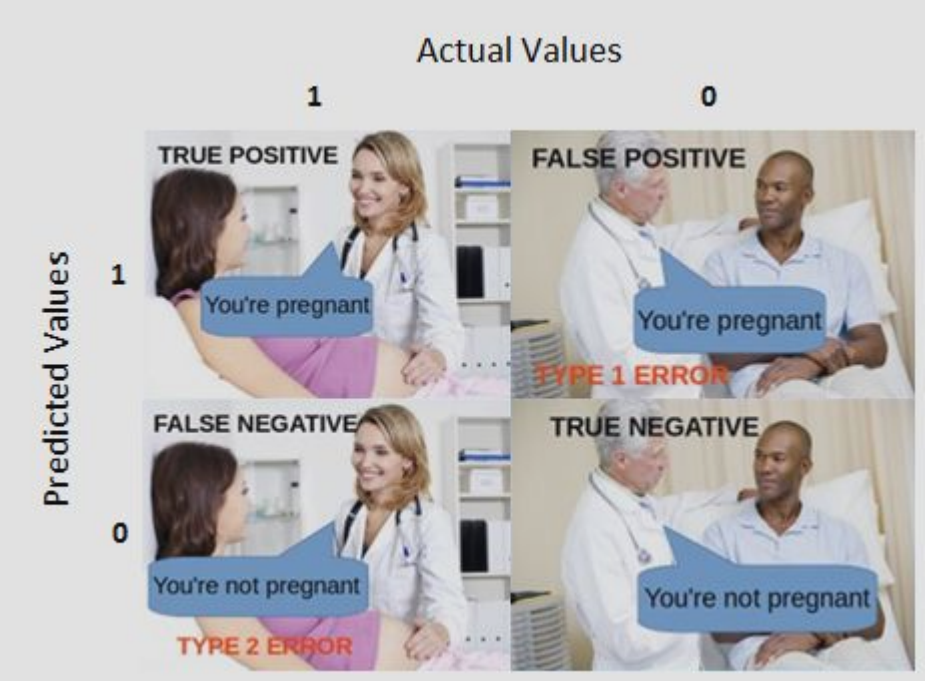
CNN



Source: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Performance Metric

Confusion Matrix



$$Precision = \frac{TP}{TP + FP} = \frac{TP}{\text{all detections}}$$

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{\text{all ground truths}}$$

Source: https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

nodeflux

Extending Vision Beyond Imagination

Section 3 - CNN with Pytorch

www.nodeflux.io

Transfer Learning

bit.ly/nodeflux-image-classification

nodeflux