algoritma
learn data science by building

Convolutional Neural Network
For Product Quality Inspection

Data Science Series
February 2022

algoritma
learn data science by building

Instructor at Algoritma Data Science School 2021

Tech Stack
Python, R, SQL
Deep Learning, Plotly, Dash App, Flask, GCP

Master of Physics, Universitas Gadjah Mada in 2020

DWI GUSTIN NURDIALIT

github.com/dwiknrd

linkedin.com/in/dwi-gustin

dwi@algorit.ma

Teaching Assistants

Yosia

yosiaazarya

yosiaazarya

Data Analysis

Kevin

kevbow

Kevin Wibowo

Machine Learning

Google Classroom

We'll be using Google Classroom for our learning platform. All materials, assignments, additional materials, and announcement will all be posted through class. Join the Classroom by entering your class code: **t3pjpff**

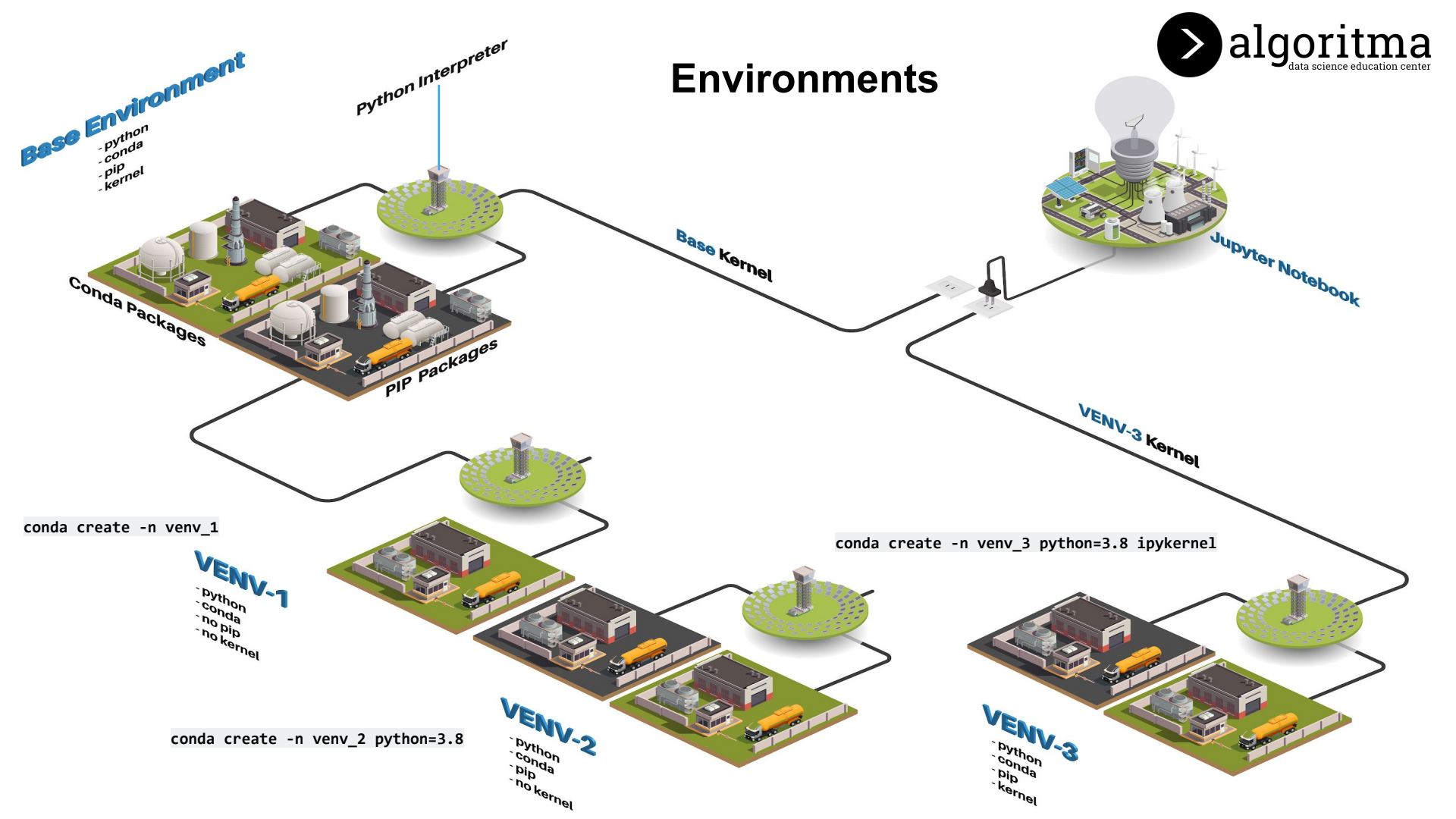
- Download material
- Pay attention to assignment deadline
- Don't miss additional material posted on Stream

PS: Google Classroom mobile application is available on Android and iOS

Training Objective

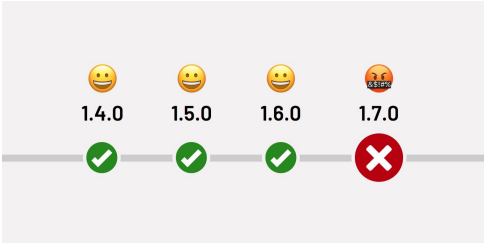
- **Python Programming Basics:**
  - Working with Conda Environment
  - Introduction to Python for data science
- **Neural Network Architecture:**
  - Layer and neurons
  - Activation and cost function
  - Feedforward
  - Backpropagation
- **Convolutional Neural Networks:**
  - Convolution concept: kernel convolutinals, strides, padding, and filter
  - Convolutional Neural Network Architecture
- **Case Study: Product Quality Inspection**
  - Load the data images and apply data augmentation techniques
  - Visualize the images
  - Training with validation: define the architecture, compile the model, model fitting and evaluation
  - Testing on unseen images

Environments

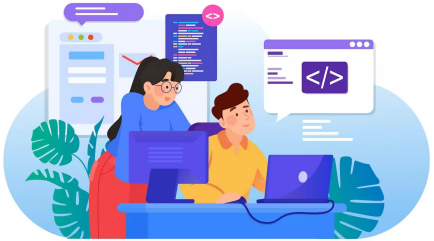


Why do we need Python environments?

You might ask: shouldn't I just install the latest Python version?



Isolate package versions to avoid breaking changes

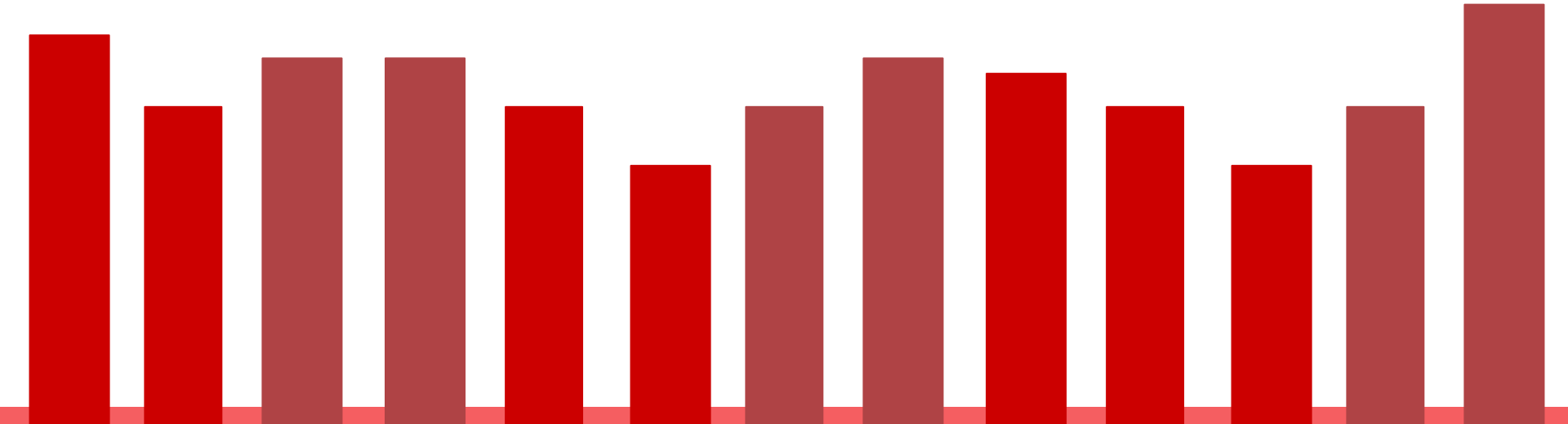


Sharing virtual environment to enable project collaboration

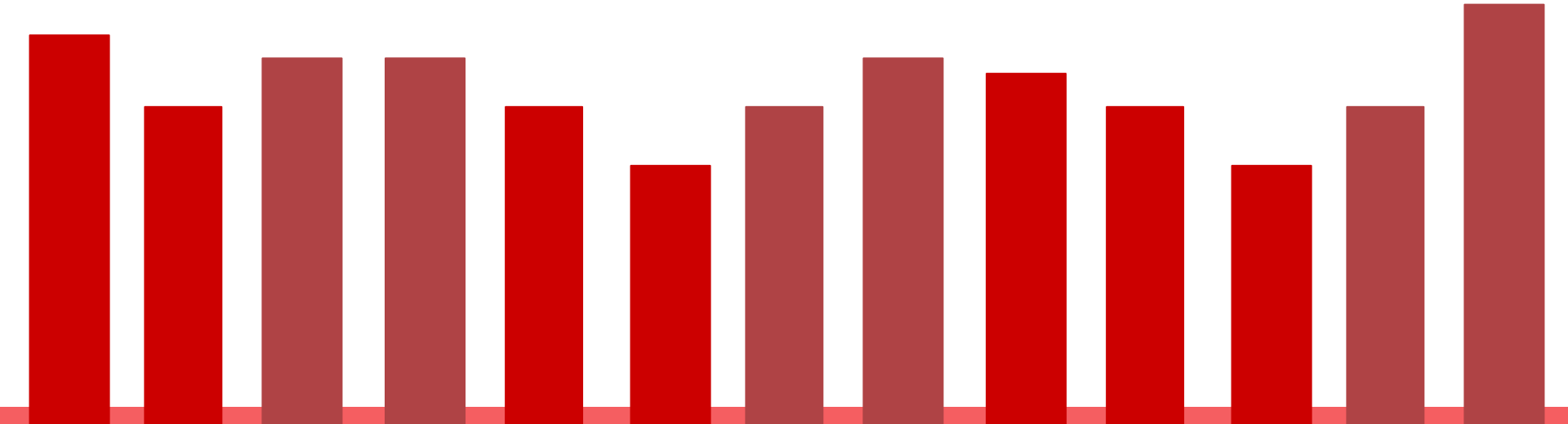
Publishing or deploying an application requires setting up an environment

algoritma
learn data science by building

Neural Network and Deep learning Basics



What do you think about Neural Network and Deep learning?



Age of AI

Artificial Intelligence (AI)

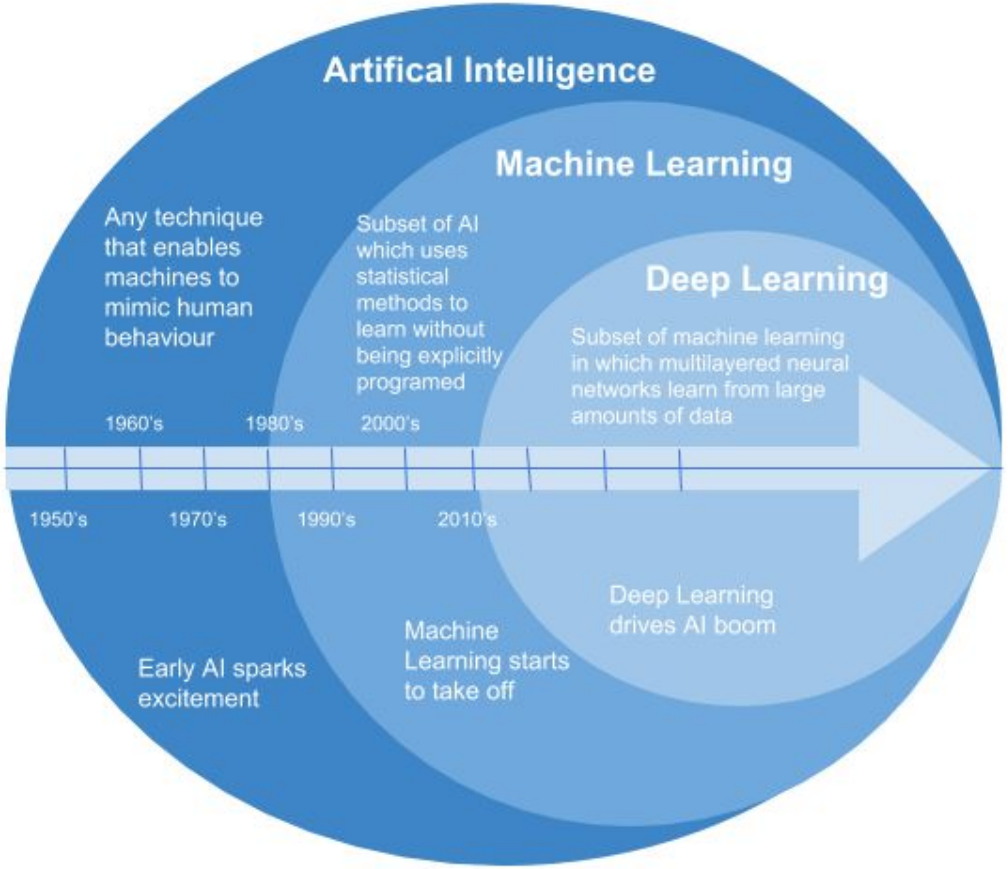
Segala bentuk program untuk mengotomatisasi tugas-tugas intelektual yang biasa dilakukan oleh manusia

Machine Learning (ML)

Salah satu bentuk AI yang membuat keputusan/prediksi berdasarkan pola yang dipelajari dari data.

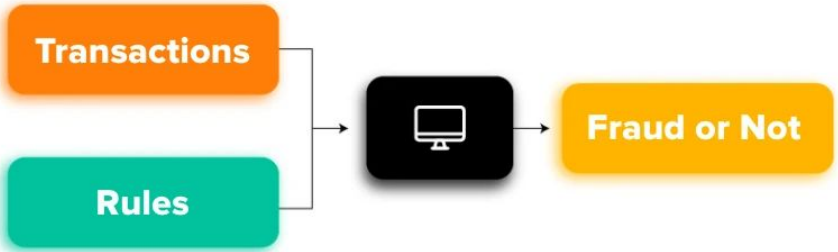
Deep Learning (DL)

Salah satu kelompok algoritma ML yang memanfaatkan arsitektur neural network yang terinspirasi dari cara kerja jaringan syaraf manusia.



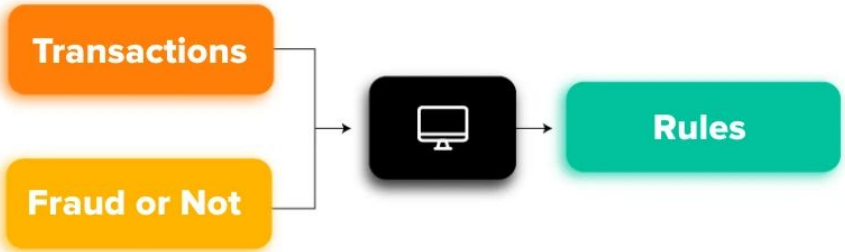
Machine Learning

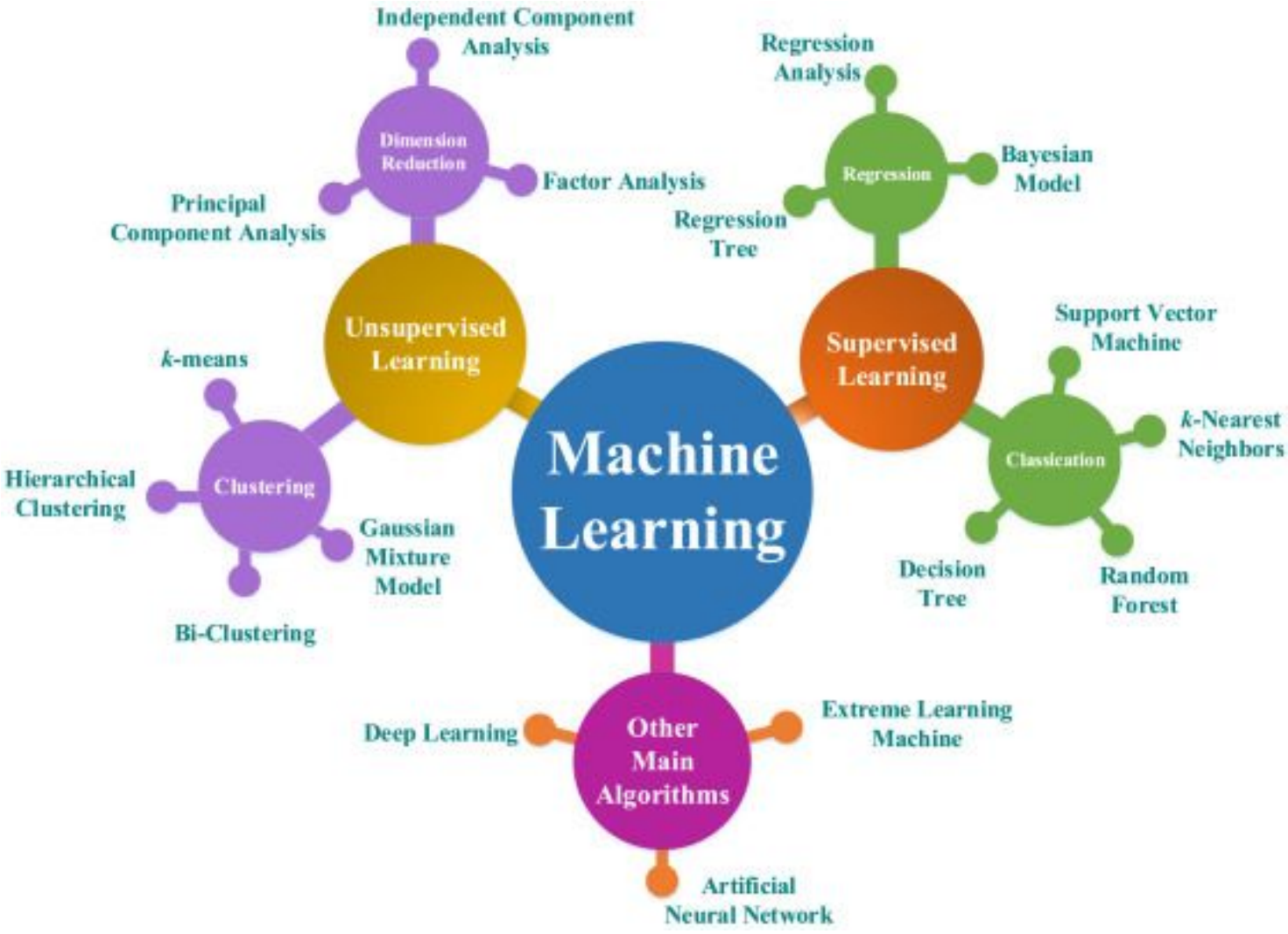
TRADITIONAL PROGRAMMING



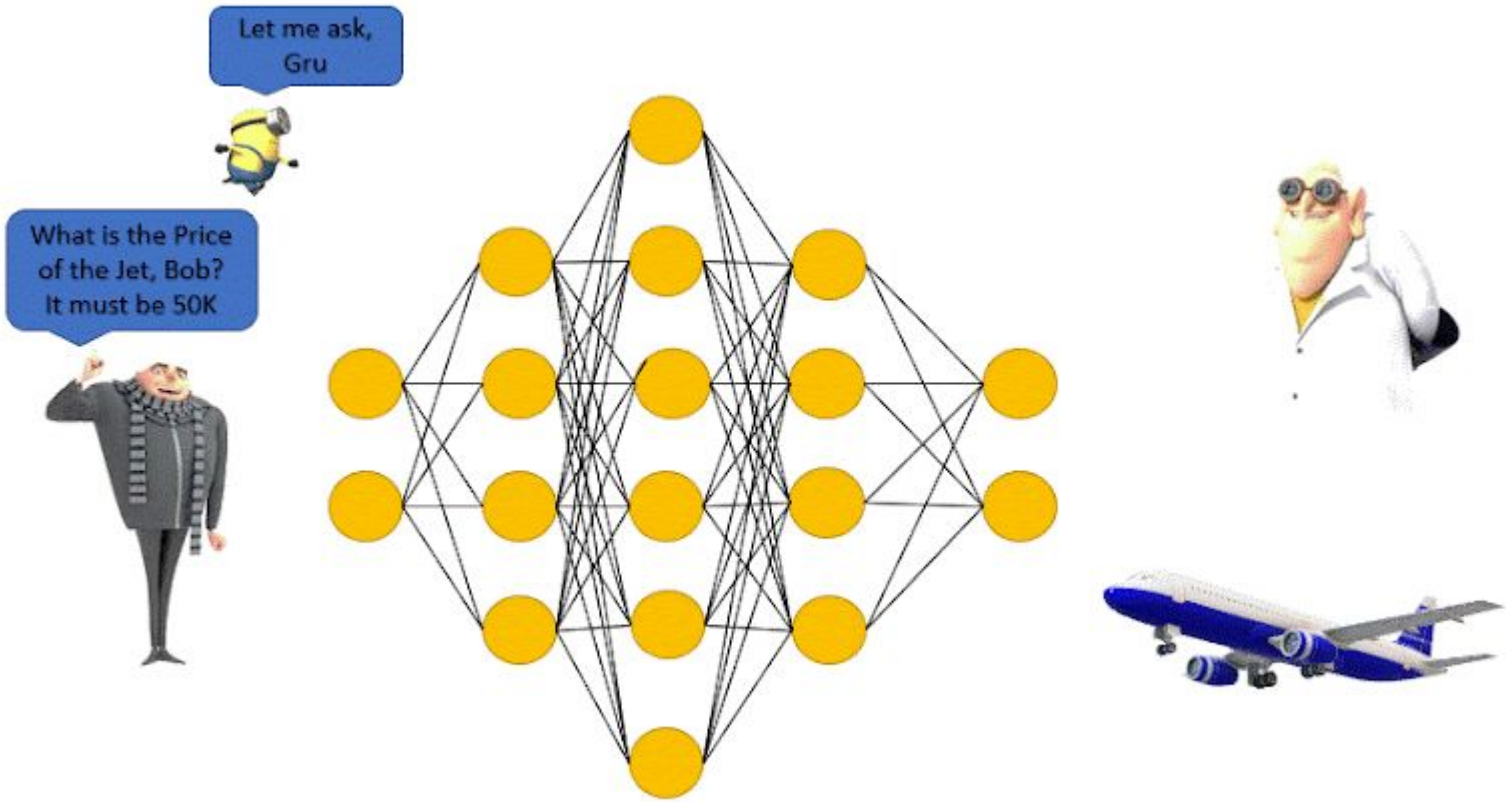
Rule1. Claim time - Submit time < 1 h
Rule2. Agreement review time > 5 m
Rule3. ...

MACHINE LEARNING




Machine Learning
Unsupervised Learning
Supervised Learning
Other Main Algorithms
Dimension Reduction
Clustering
Regression
Classication
Independent Component Analysis
Factor Analysis
Principal Component Analysis
k-means
Hierarchical Clustering
Gaussian Mixture Model
Bi-Clustering
Regression Analysis
Bayesian Model
Regression Tree
Support Vector Machine
k-Nearest Neighbors
Decision Tree
Random Forest
Deep Learning
Extreme Learning Machine
Artificial Neural Network

Deep Learning



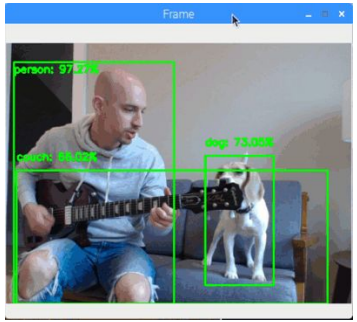
Why Deep Learning is Popular?

algoritma
learn data science by building

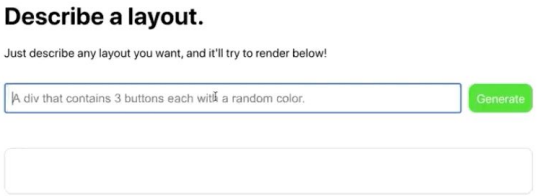
Deep Learning sangat fleksibel untuk menyelesaikan berbagai permasalahan prediksi, terutama untuk data yang tidak terstruktur.

Gambar

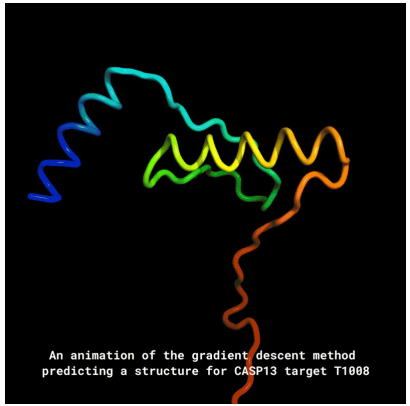
Video



Teks



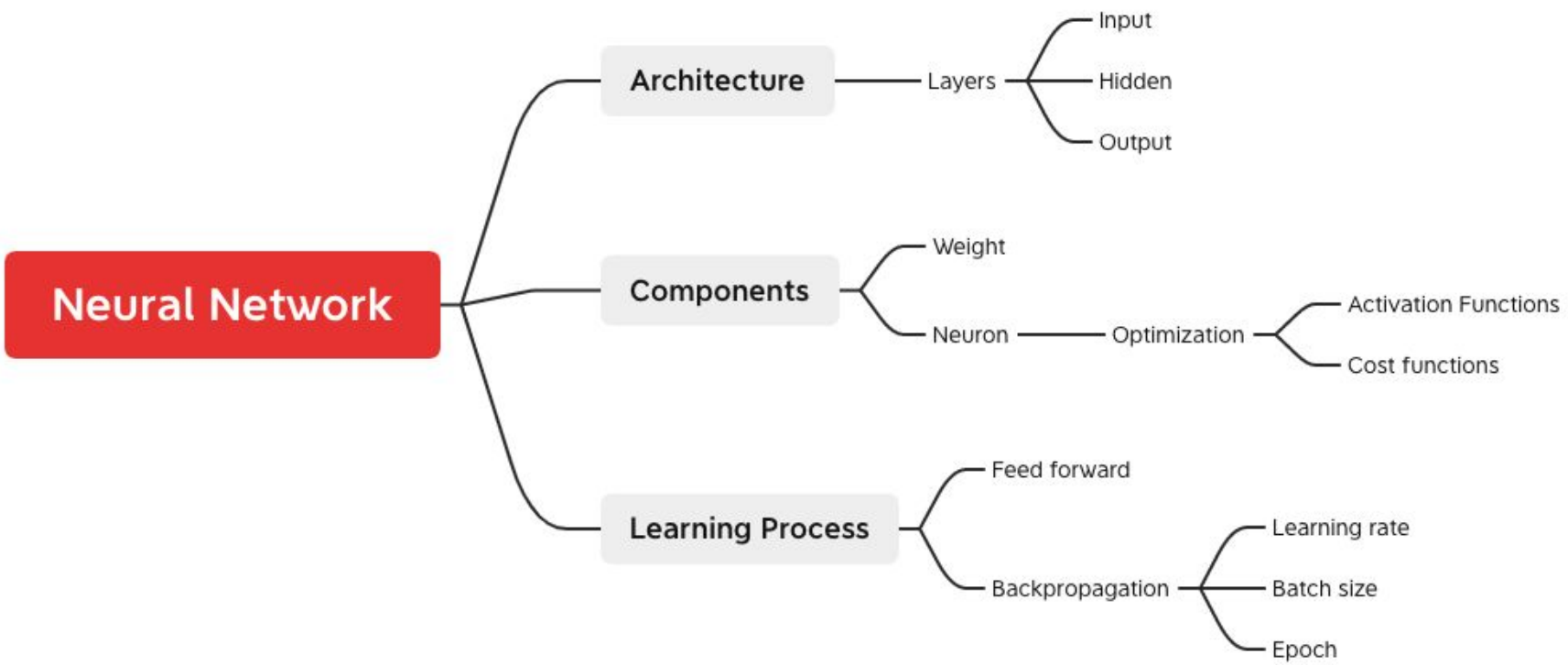
3D Modeling



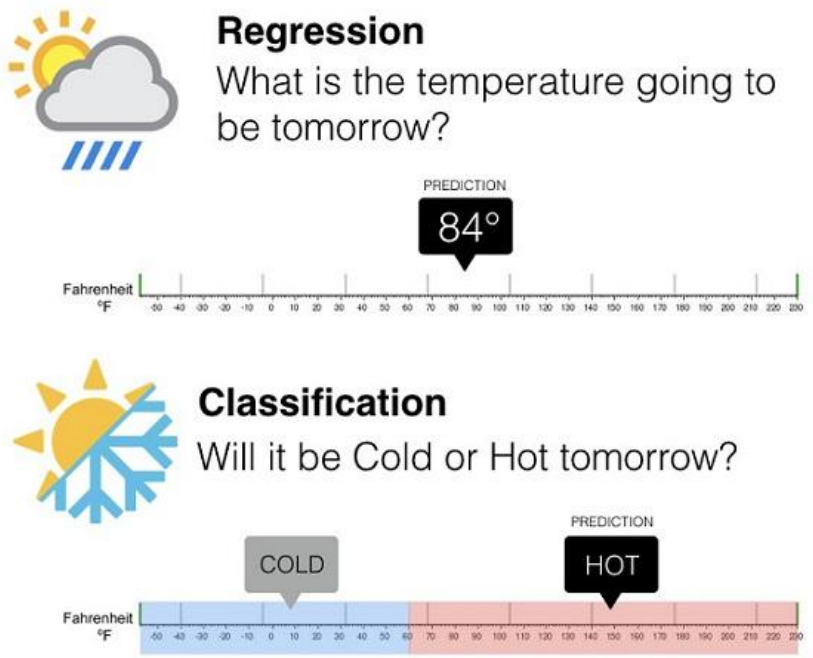
Recommended Video

The Age of A.I.
1 season • 9 episodes

Training Objective



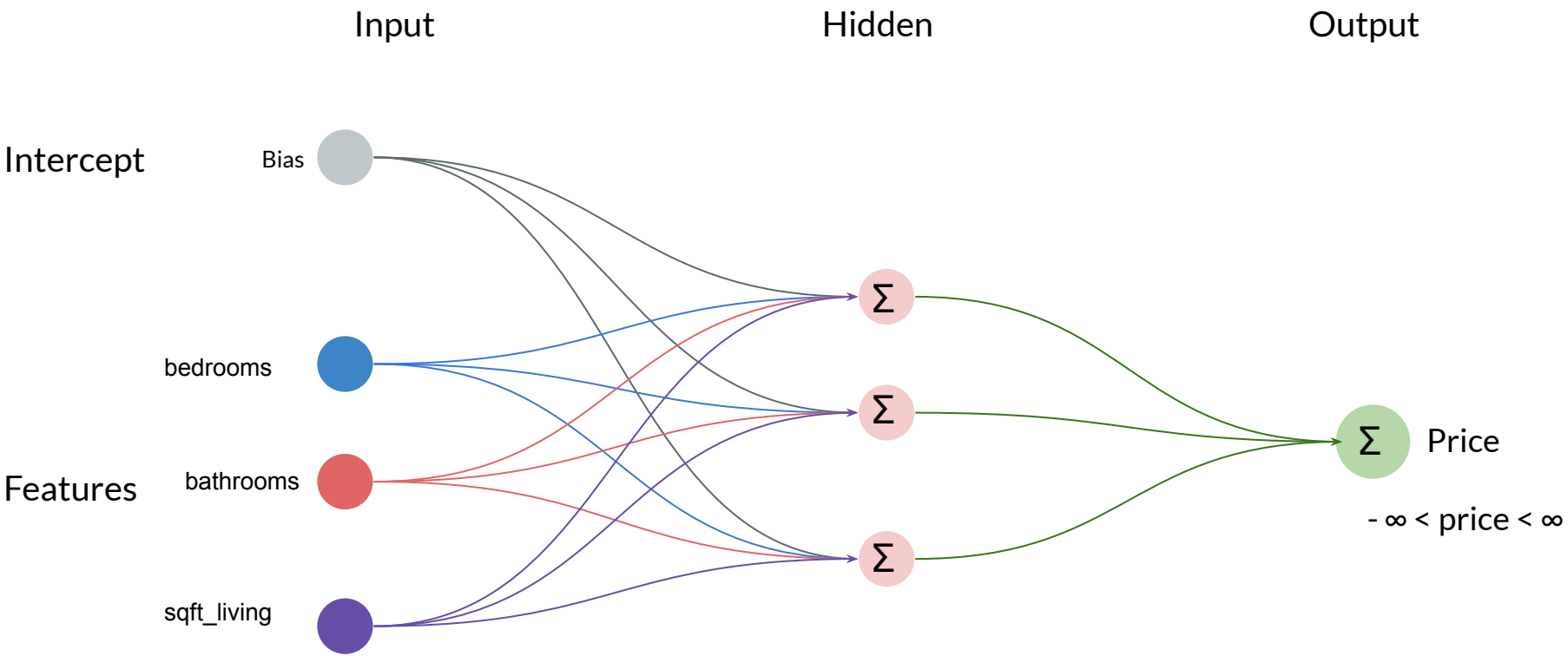
What Do You Think of Weather Prediction?



Source: Kohani A.R., 2017

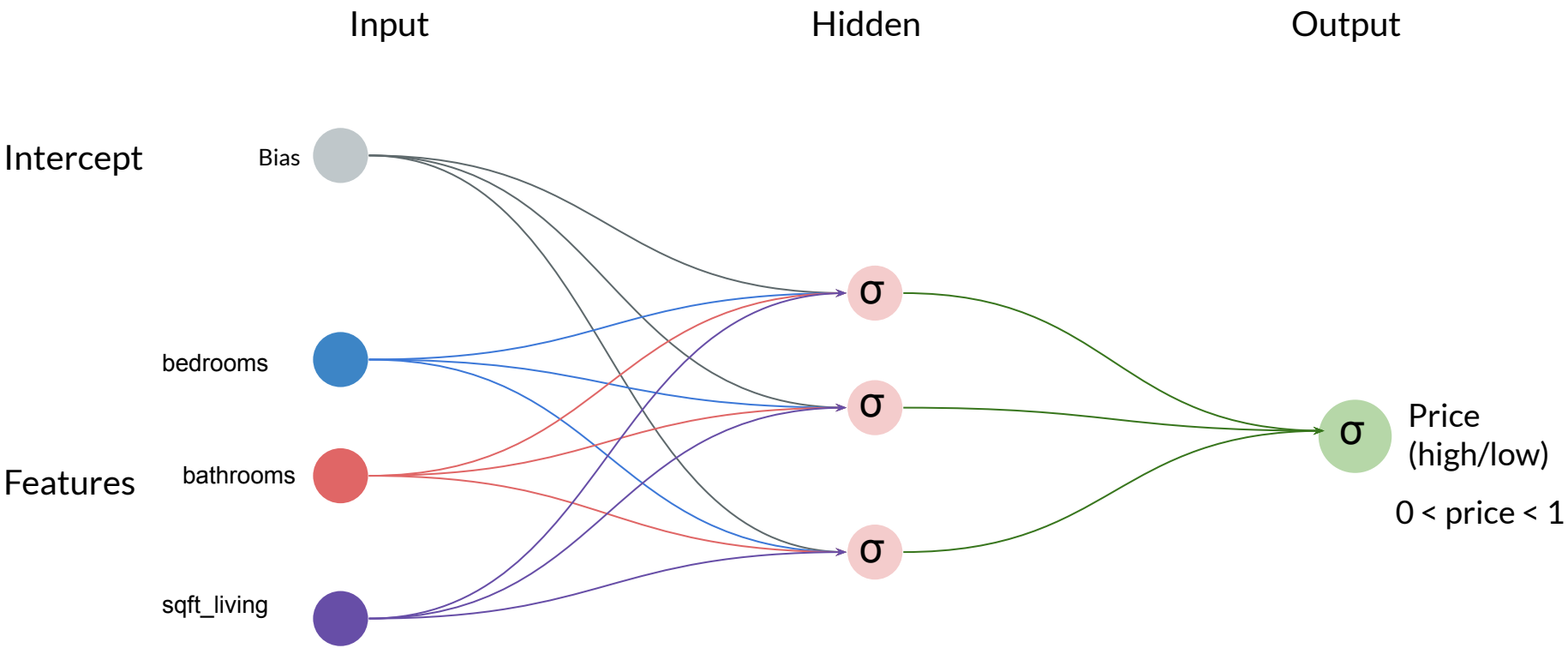
Arsitektur Neural Network untuk regresi

Neural Network memiliki 1 lapisan (layer) tambahan di antara input dan output, disebut dengan **hidden layer**



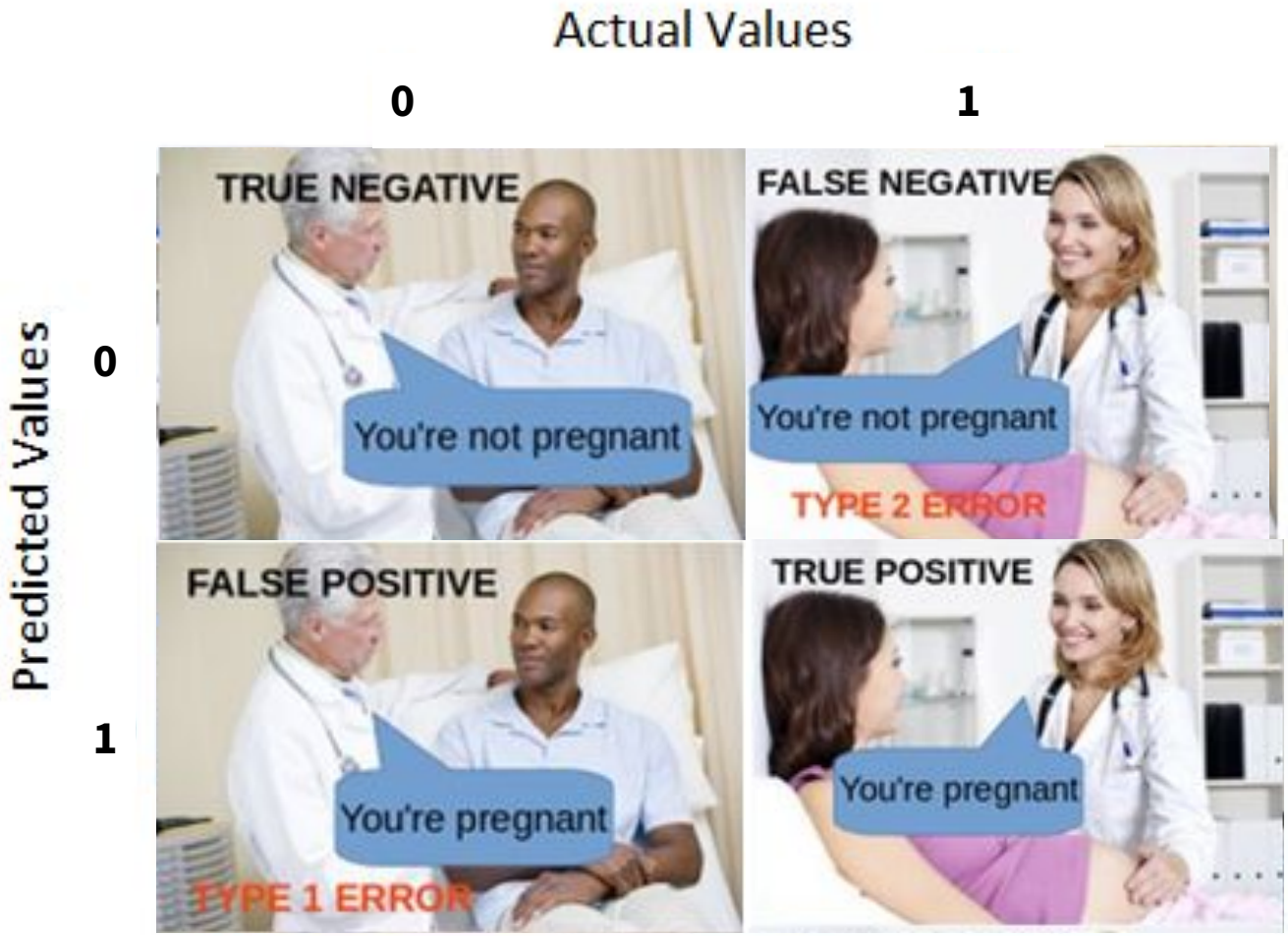
Arsitektur Neural Network untuk klasifikasi

Neural Network memiliki 1 lapisan (layer) tambahan di antara input dan output, disebut dengan **hidden layer**



algoritma
learn data science by building

Confusion Matrix


Actual Values
0
1
Predicted Values
0
1
TRUE NEGATIVE
You're not pregnant
FALSE NEGATIVE
You're not pregnant
TYPE 2 ERROR
FALSE POSITIVE
You're pregnant
TYPE 1 ERROR
TRUE POSITIVE
You're pregnant

Predicted	Actual	
	0	1
0	86	58
1	75	93

TP diprediksi positif, dan benar

TN diprediksi negatif, dan benar

FP diprediksi positif, namun salah

FN diprediksi negatif, namun salah

accuracy

$$\frac{TP + TN}{total}$$

recall

$$\frac{TP}{TP + FN}$$

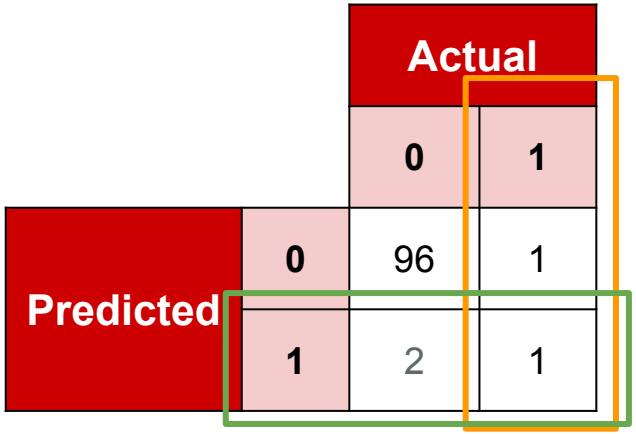
precision

$$\frac{TP}{TP + FP}$$

specificity

When Accuracy does not really matters...

Number of observations: 100 | Number of non-covid: 98 | Number of covid: 2



Accuracy:

(96 + 1) / (96 + 2 + 1 + 1) = 97%

How many did it predict right for the **targeted class** (1)?

Recall/Sensitivity: true positive/actual = 1/(1 + 1) = 50%

Precision: true positive/predicted = 1 / (1 + 2) = 33%

Recall vs Precision

Case 1: Cancer Detection

45%

FP — Akan pemeriksaan lanjutan,
kalau sakit maka diobati,
kalau sehat maka pulang

FN — Tidak diperiksa lanjutan,
kalau sakit maka makin parah

Recall: ↑

$$\frac{TP}{TP + FN}$$

Precision: ↓

$$\frac{TP}{TP + FP}$$

Positive: Kanker
Tindak lanjut: Pemeriksaan lanjutan

Recall vs Precision

Case 2: Spam Classifier

55%

FP Email penting masuk folder Spam

FN Email spam masuk folder Inbox

Recall:

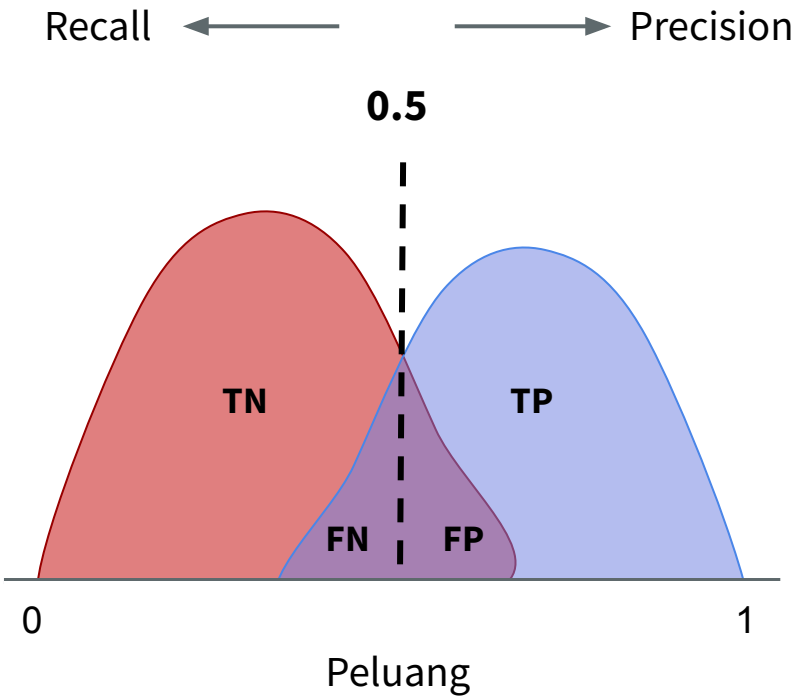
$$\frac{TP}{TP + FN}$$

Precision:

$$\frac{TP}{TP + FP}$$

Positive: Spam
Tindak lanjut: Email Spam masuk folder Spam,
Email Ham masuk Inbox

Recall / Precision Threshold



Recall/Sensitivity:

$$\frac{TP}{TP + FN}$$

Kita mau **sebanyak mungkin** mengambil kemungkinan kelas positif

Precision:

$$\frac{TP}{TP + FP}$$

Kita mau **seselektif /presisi mungkin** dalam memasukkan ke kelas positif

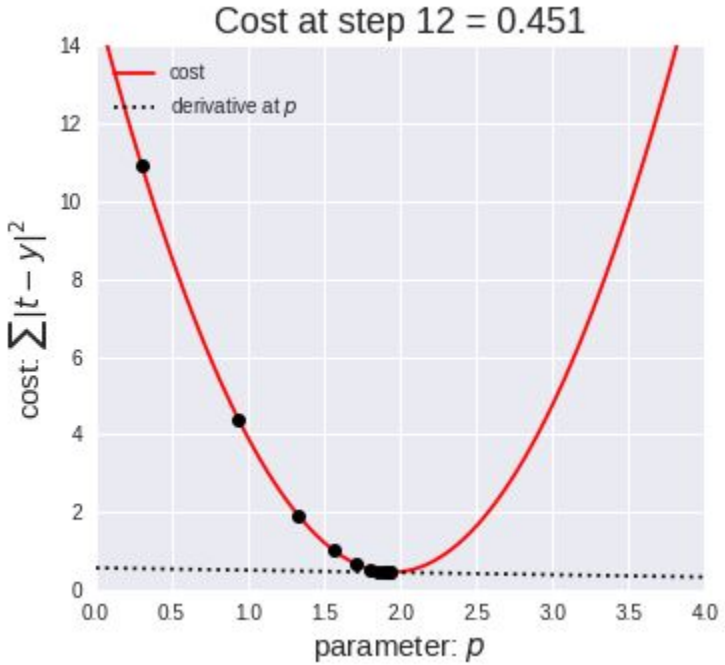
algoritma
learn data science by building

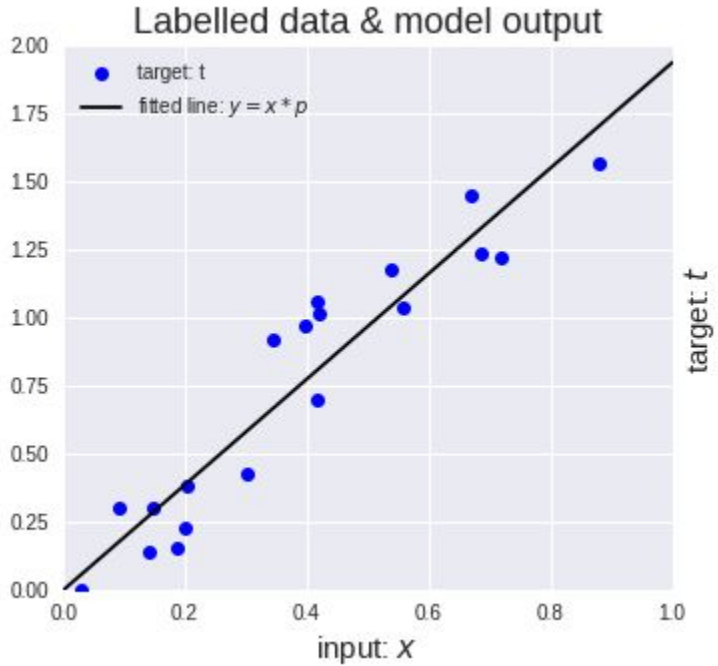
How Neural Networks Learn

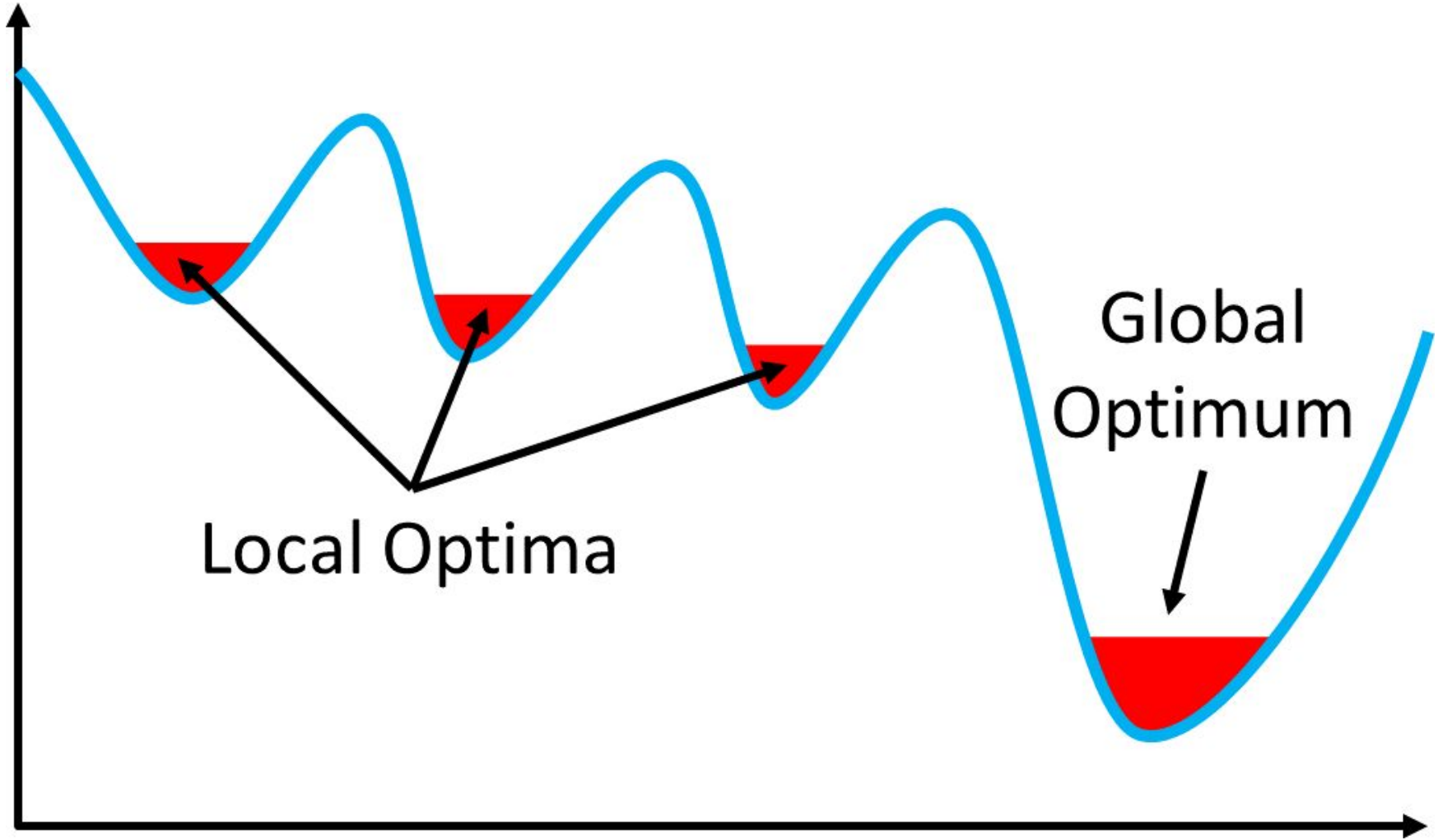
Back Propagation

Algoritma dasar yang digunakan untuk mengupdate bobot adalah **gradient descent**

Mengupdate bobot dan bias yang bisa memberikan error terkecil






Global
Optimum
Local Optima

Back Propagation

Menghitung nilai error yang dihasilkan.

Untuk kasus klasifikasi menggunakan Cross-Entropy

$$Binary\ Cross - Entropy = -p(x)\ log\ q(x) + (1 - p(x))\ log\ (1 - q(x))$$

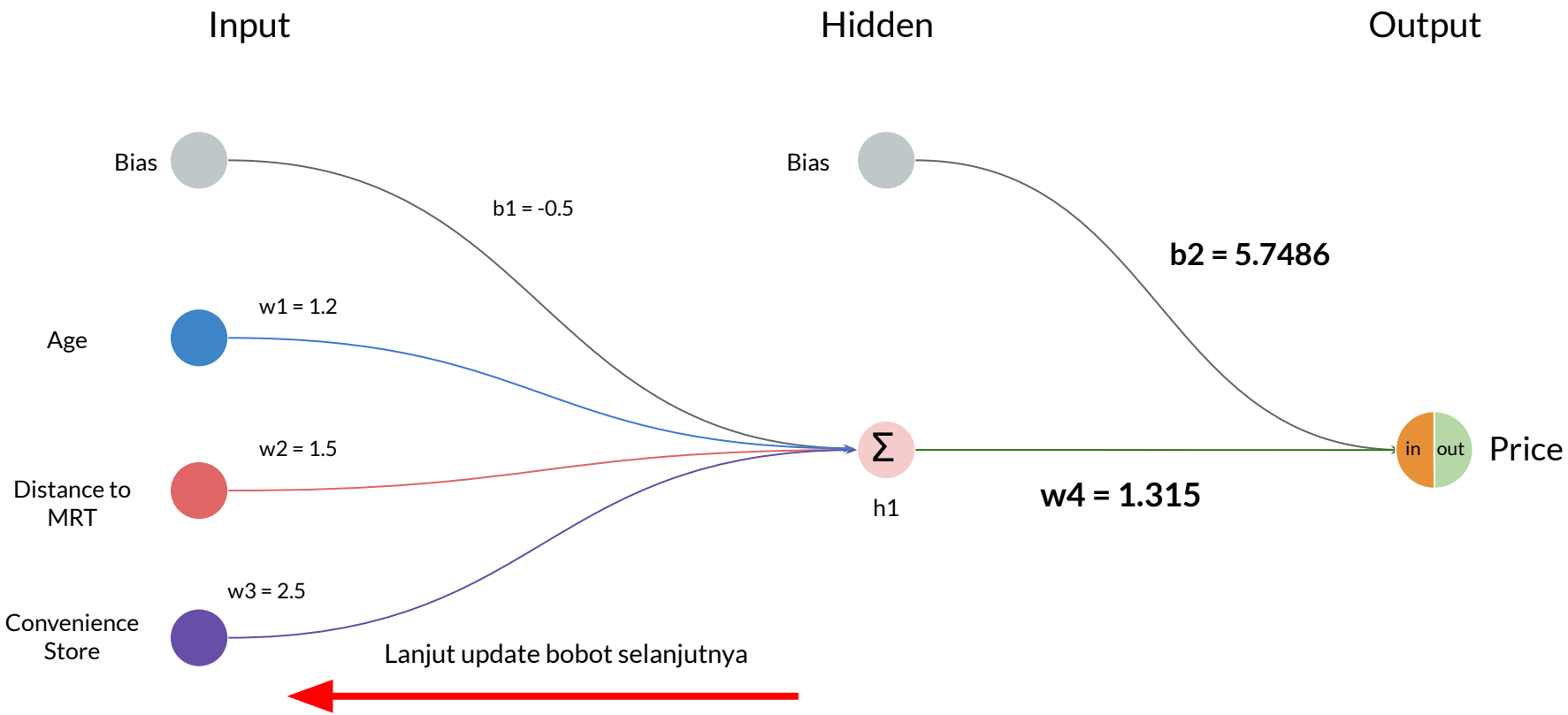
Untuk kasus regresi menggunakan Sum of Squared Error

$$SSE = \frac{1}{2}\Sigma(y - \hat{y})^2$$

SSE = ½ (61.5 - (-0.986))^2 = 1952.25

Back Propagation

Setelah mengupdate bobot dan bias dari hidden layer ke output layer, bisa dilanjutkan menuju layer berikutnya

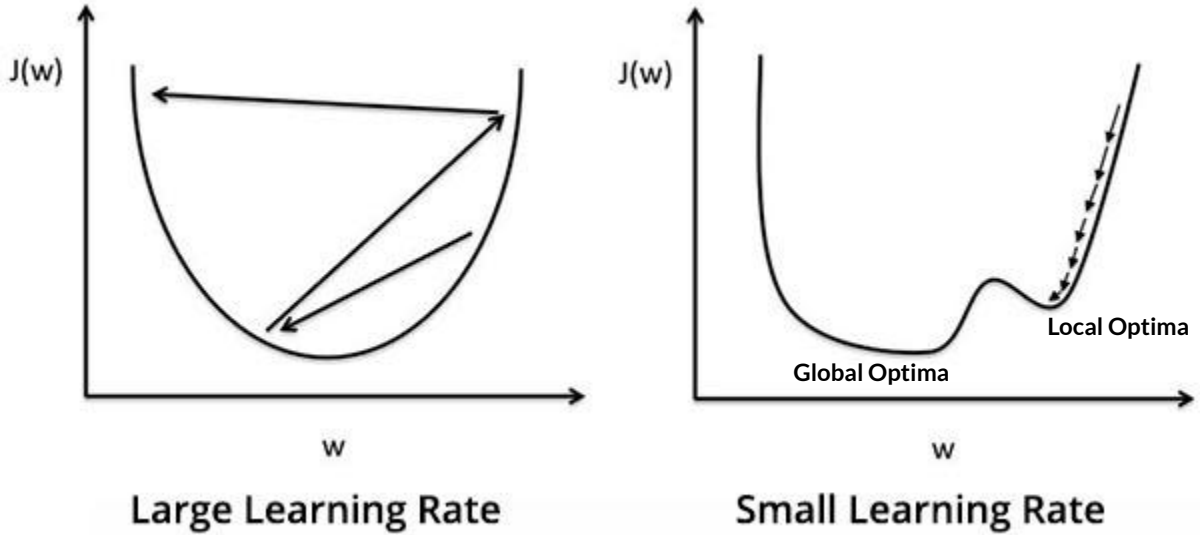


Learning Rate

Learning Rate mengatur seberapa besar perubahan terhadap bobot

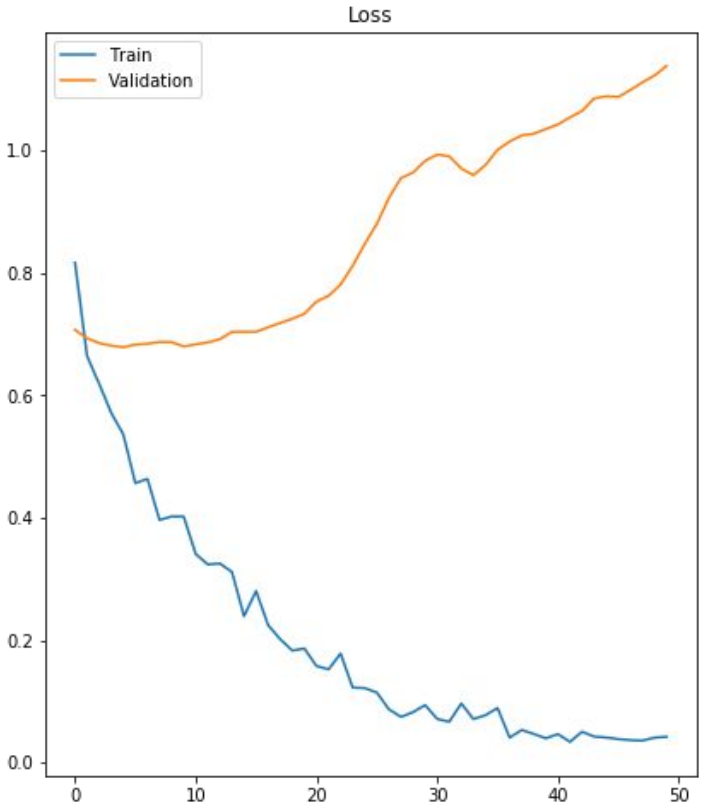
- Learning rate yang besar akan membuat model belajar lebih cepat dengan jumlah step yang lebih sedikit, namun kemungkinan besar akan melewatkan titik terendahnya sehingga tidak bisa mencapai nilai error yang optimal.

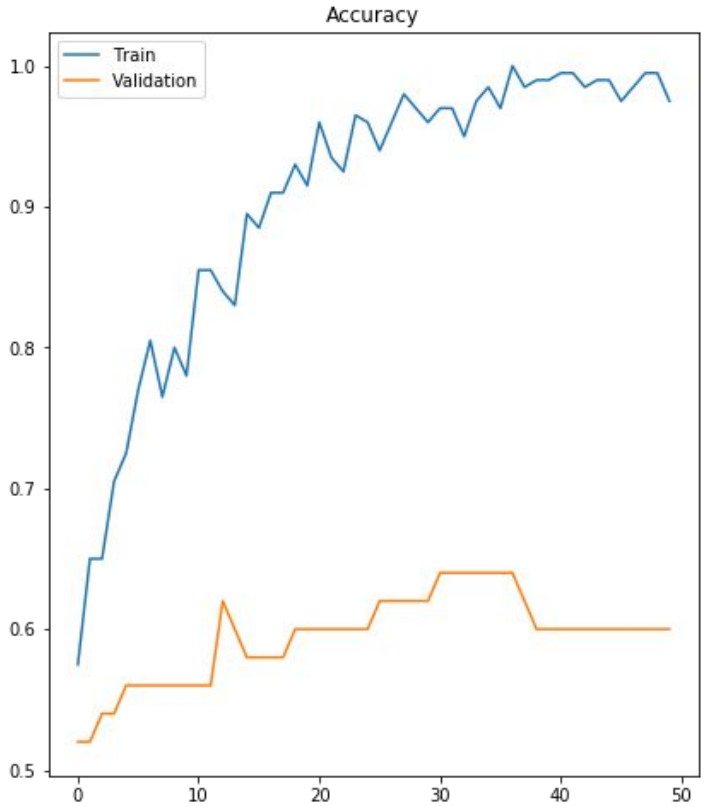
- Learning rate yang kecil akan membuat model mengupdate bobot sedikit demi sedikit sehingga kemungkinan untuk melewatkan titik terendah/titik optimalnya lebih kecil. Namun, model dapat terjebak ke **local optima** dan tidak bisa mencapai **global optima**



Overfit

Terjadi ketika loss (error) dari **training data** jauh lebih kecil daripada **validation/testing data**

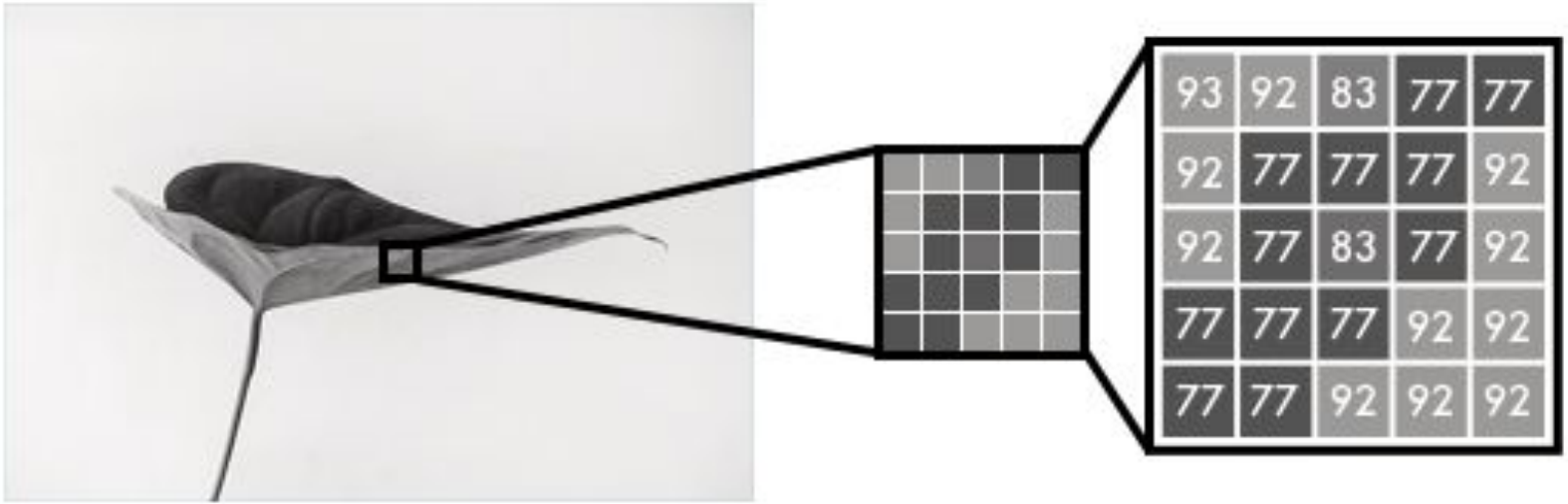




algoritma
learn data science by building

Neural Networks for Image Classification

How does computer see an image?



Gambar direpresentasikan dalam bentuk **matriks** berisikan **nilai pixel**

Nilai pixel 0 = semakin gelap (hitam)

Nilai pixel 255 = semakin terang (putih)

How does computer see an image?

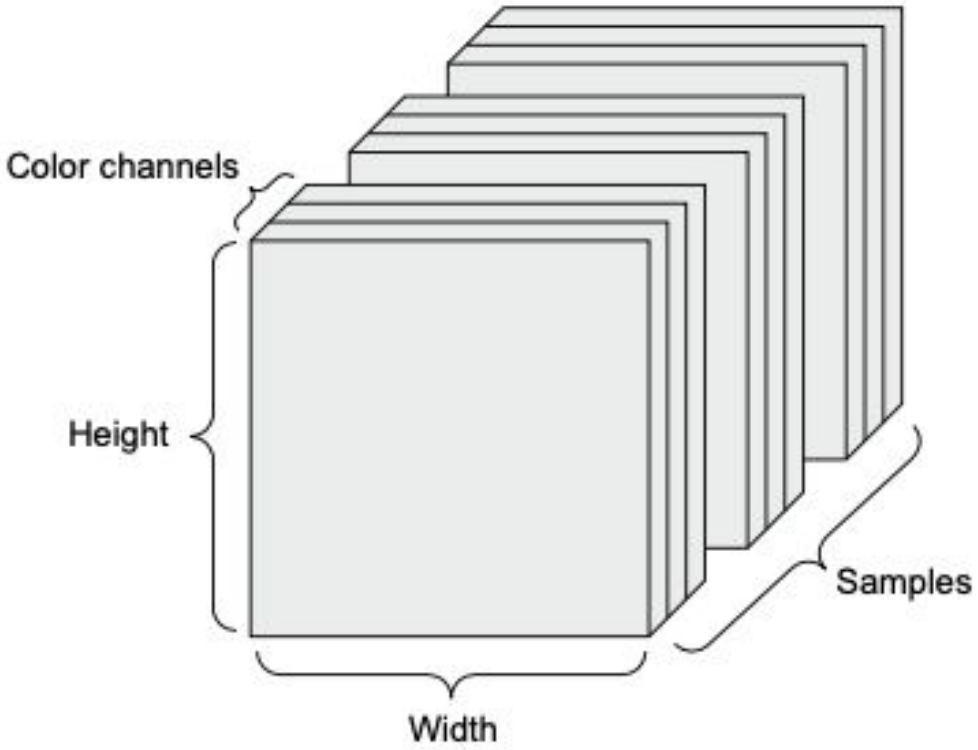
R	B	G		pixel
30	30	255	=	
0	0	255	=	
0	0	210	=	

Gambar berwarna direpresentasikan dalam 3 channel (Red, Green, Blue) berisikan **nilai pixel**

Nilai pixel 0 = semakin gelap / tidak ada warna

Nilai pixel 255 = semakin terang / ada warna

Dimension of Image Data



How does human see an image?

algoritma
learn data science by building

Tebak apa nama hewan pada gambar di bawah ini?

How does human see an image?

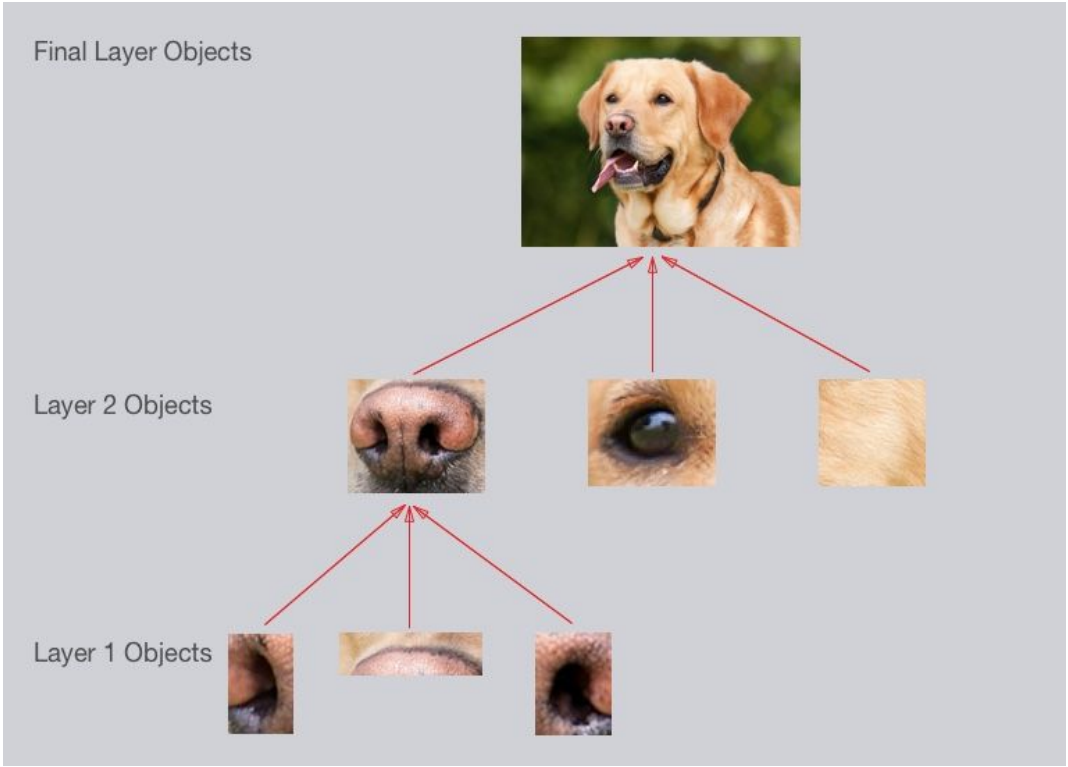
Bagaimana manusia mengenali sebuah gambar? Apakah:

a. Melihat secara detail satu per satu titik piksel

ATAU

b. Melihat bentuk objek pada gambar secara umum?

Hal ini yang diadaptasi oleh layer Convolution pada CNN



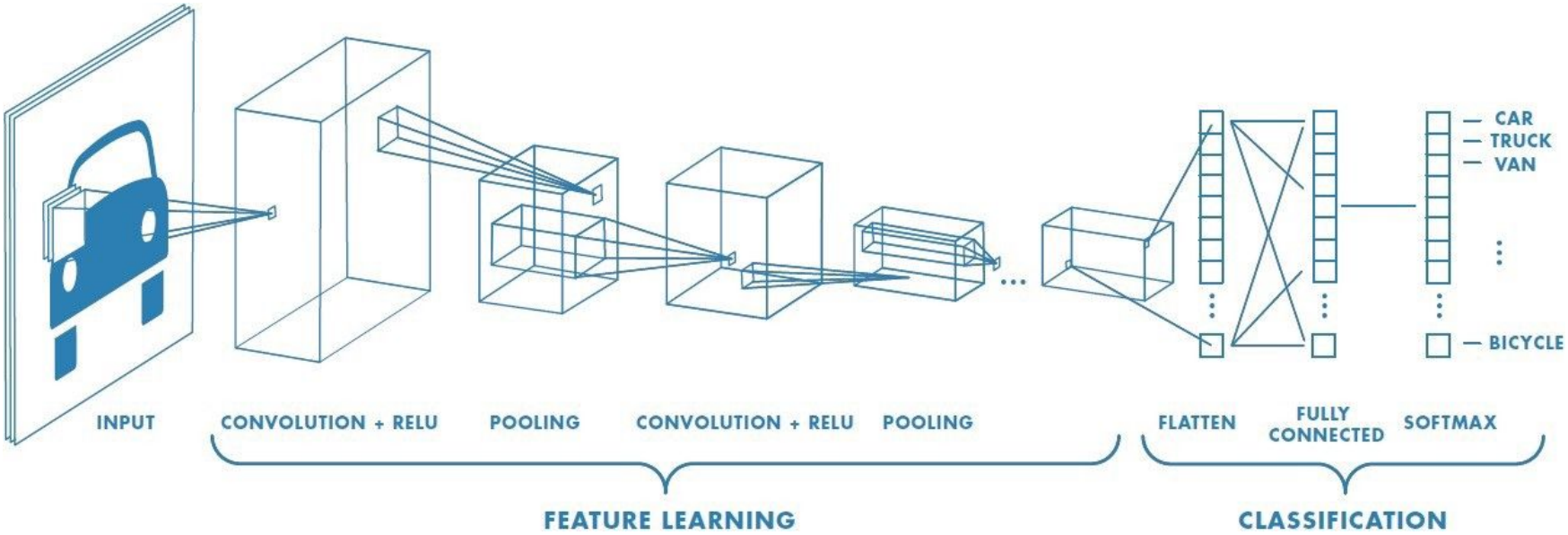
algoritma
learn data science by building

Convolutional Neural Network (CNN)

Layers in CNN

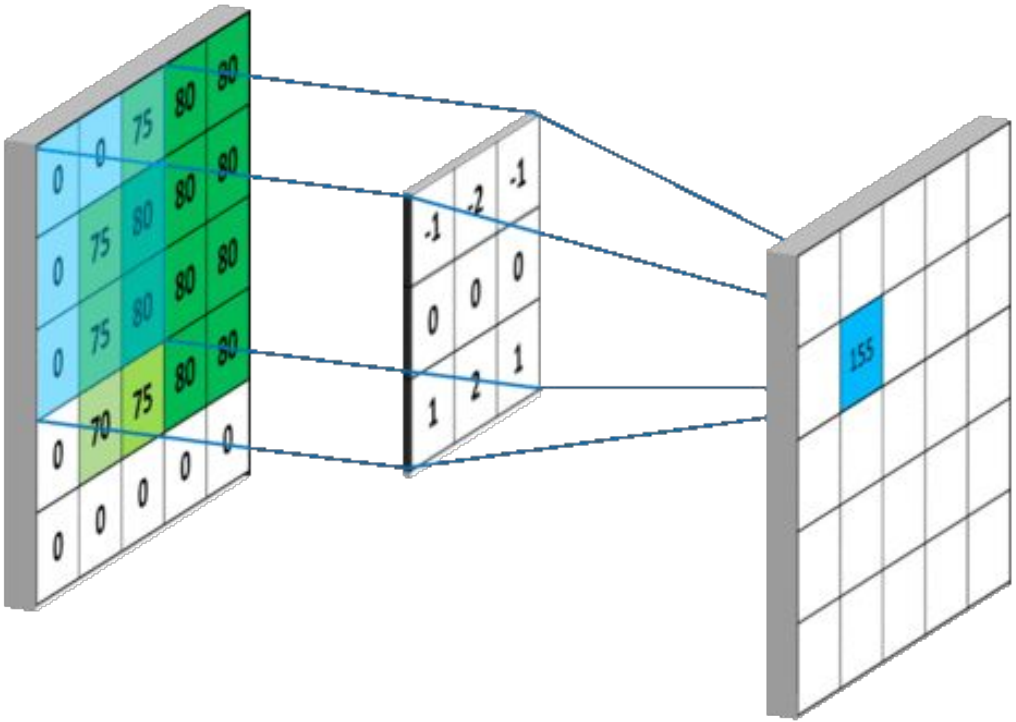
Pada umumnya kita menggunakan 4 jenis layer pada arsitektur CNN:

1. **Convolutional layer**
2. **Pooling layer**
3. Flattening
4. Fully-connected layer (sama seperti NN sebelumnya)



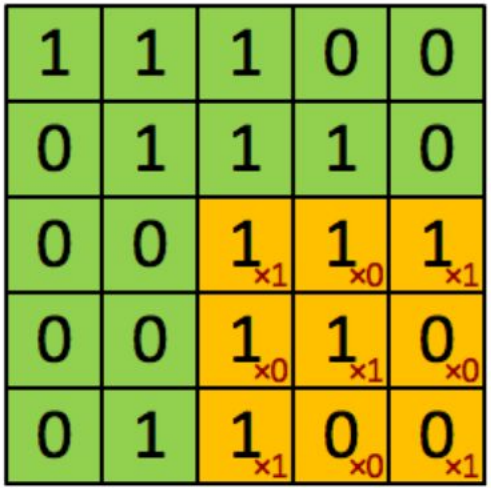
1. Convolutional Layer

- Ekstraksi fitur dari pixel gambar
- Parameter jauh lebih sedikit karena direpresentasikan dalam bentuk **filter (atau kernel)**



1. Convolutional Layer

Perhitungan manual operasi convolution:



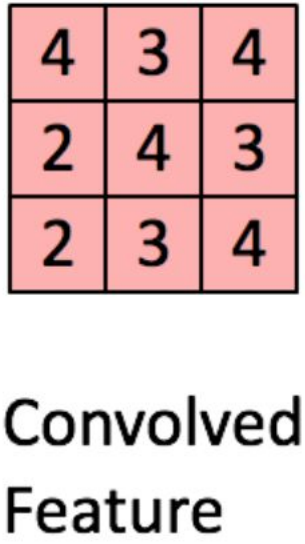
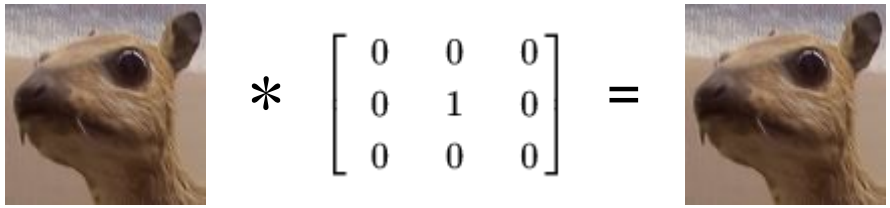


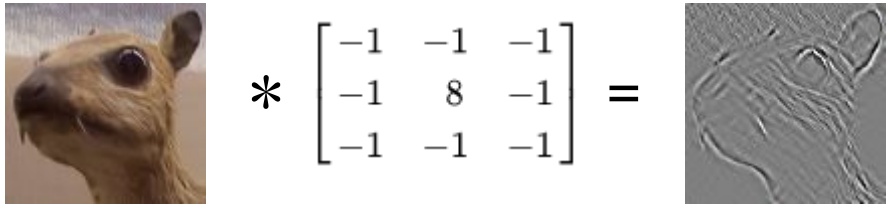
Image Filtering: What does the filter do?

Image filtering mengubah nilai pixel, sehingga **warna gambar** diubah tanpa mengubah posisi pixel
Berikut adalah contoh dari efek beberapa matriks filter:



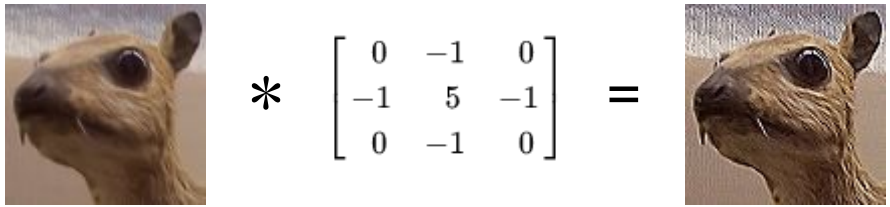
$$* \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} =$$

IDENTITY FILTER



$$* \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} =$$

RIDGE DETECTION FILTER

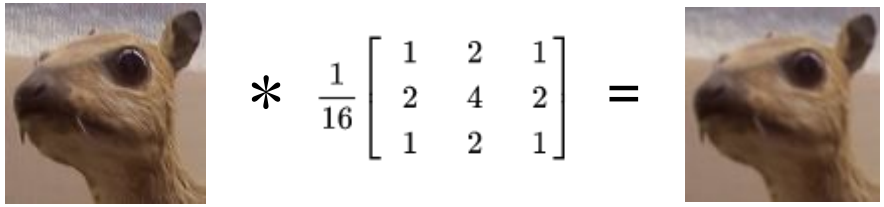


$$* \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} =$$

SHARPENING FILTER

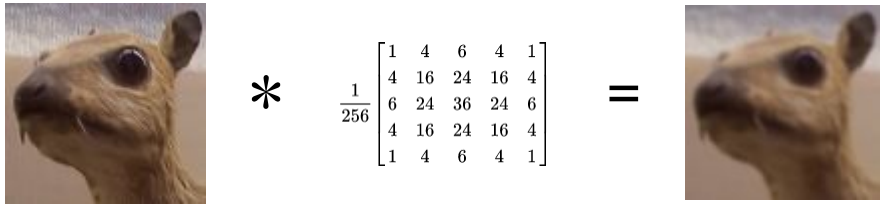
Image Filtering: What does the filter do?

Matriks filter dengan ukuran yang berbeda akan memberikan efek yang berbeda pula



$$* \quad \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} =$$

BLUR FILTER (3 x 3)



$$* \quad \frac{1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} =$$

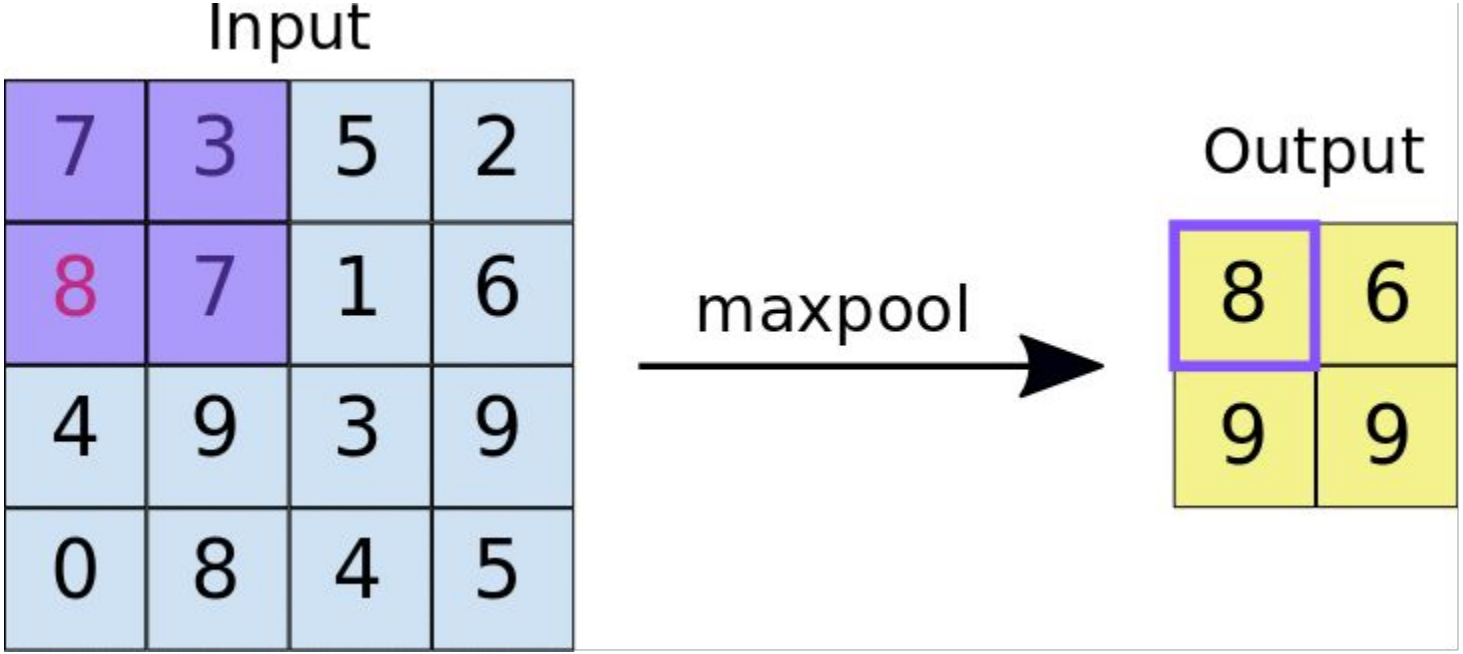
BLUR FILTER (5 x 5)

Key Takeaways:

- Convolutional layer akan berusaha **mencari sendiri filter** yang optimal (Ingat konsep backpropagation)
- Ukuran filter adalah **hyperparameter** dari Convolutional Neural Network

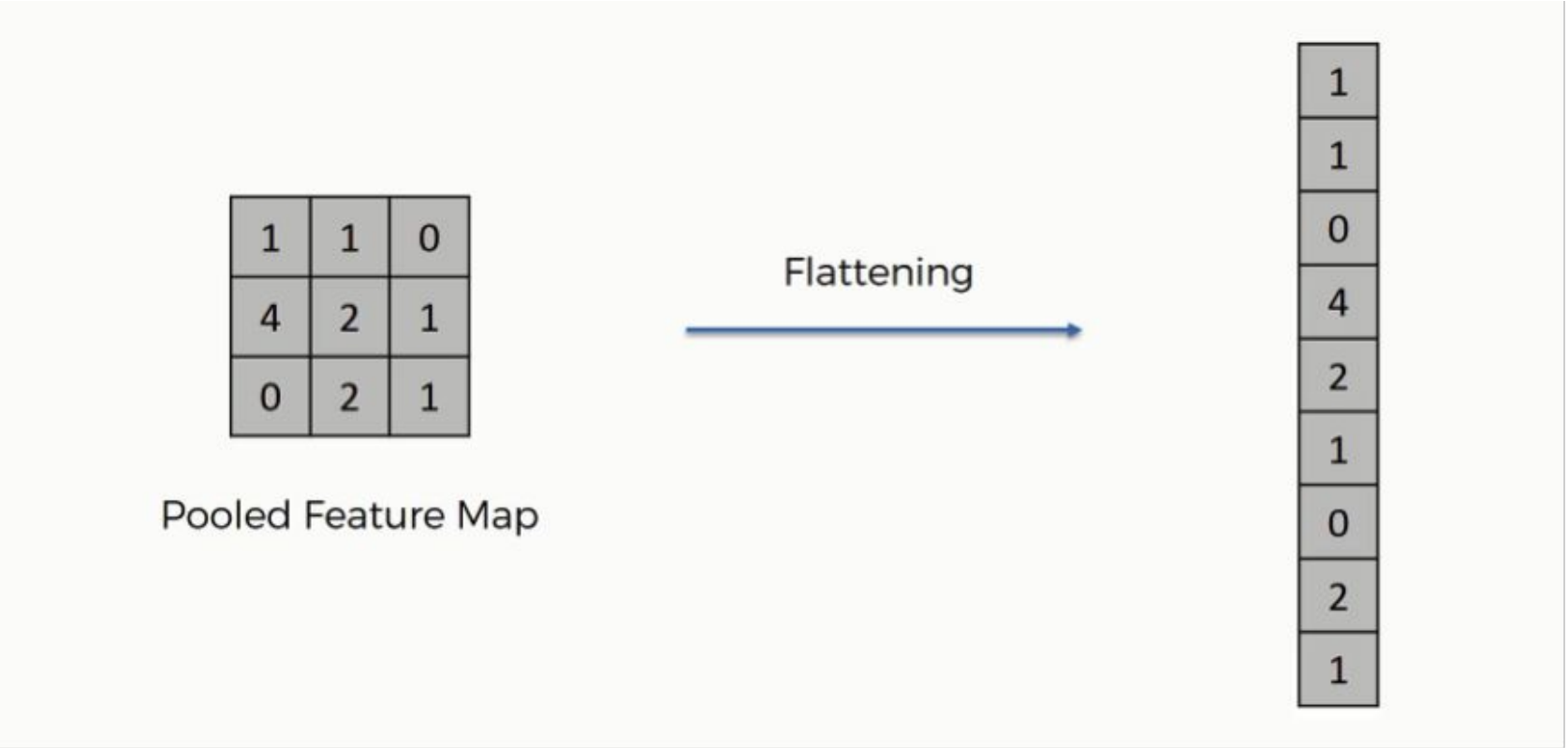
2. Pooling Layer

- Mereduksi ukuran gambar dengan merangkum pixel



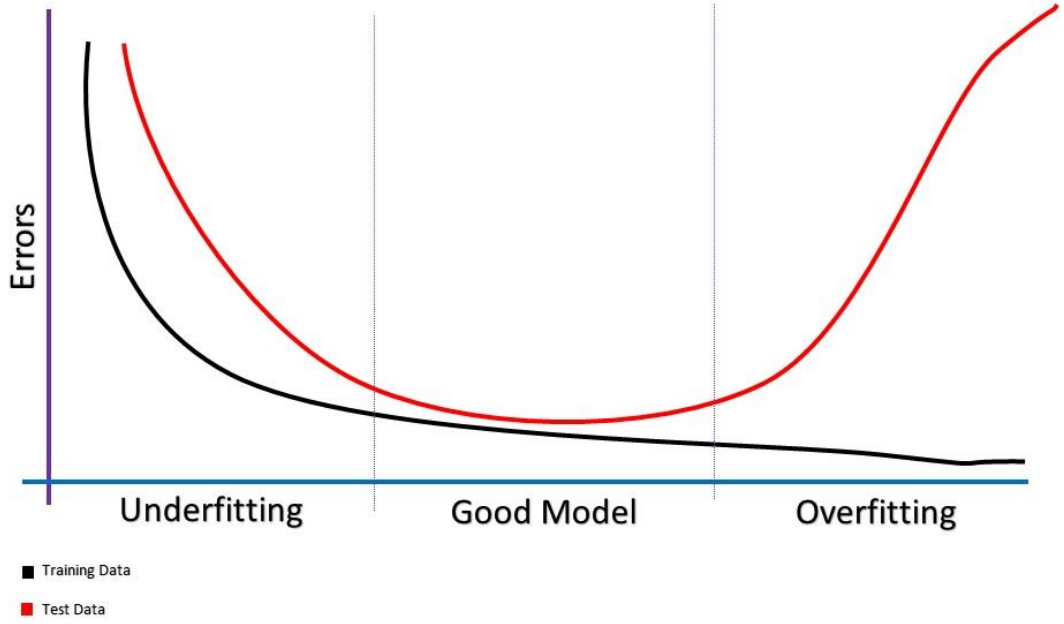
3. Flattening Layer

- Mengubah dimensi data, yang sebelumnya 2 dimensi menjadi 1 dimensi



Fitting of a model

- Underfitting: model terlalu simple
- Overfitting: model terlalu kompleks



Data Augmentation

- Meminimalisir overfitting (kondisi model terlalu “menghafal” pola data)
- Memperbanyak jumlah dan variasi dari data training