

Odisee
DE CO-HOGESCHOOL

Introductie



Jens Baetens



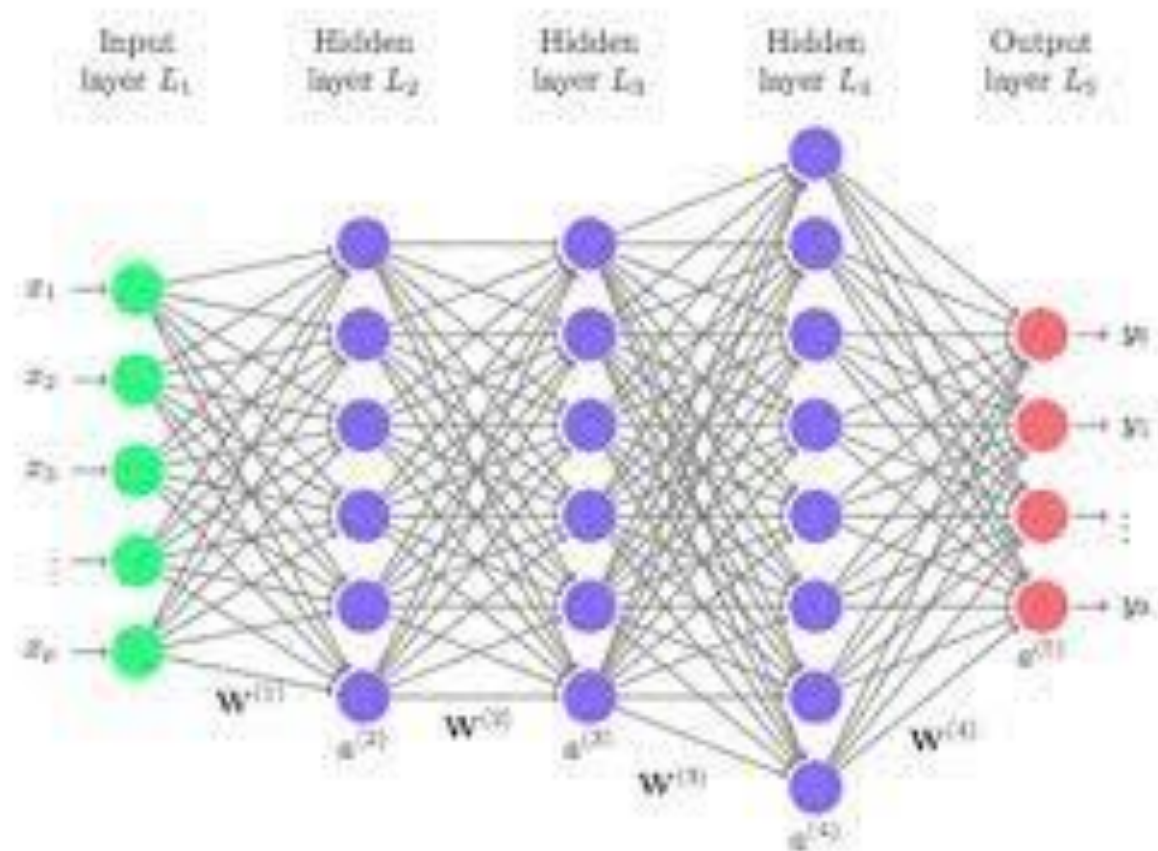
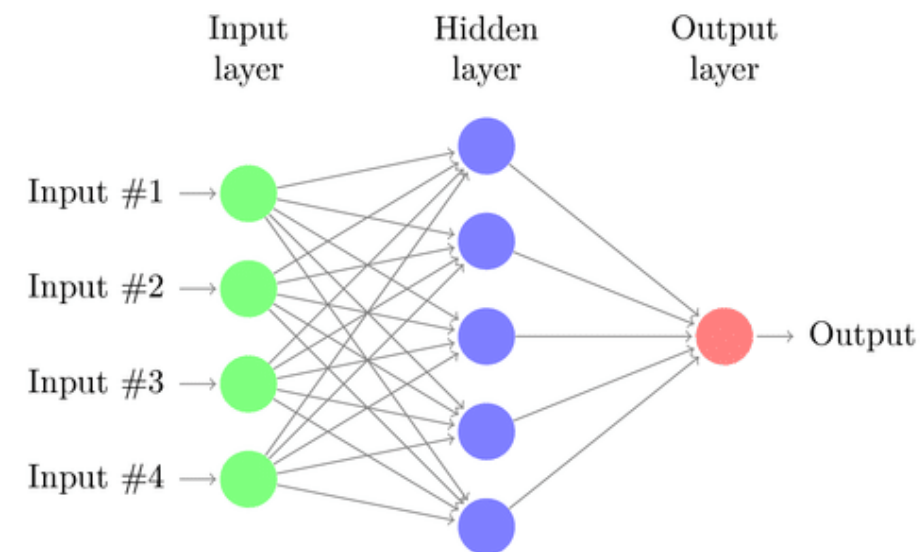
Inhoud

- ▣ Wat gaan we leren tijdens dit vak?
- ▣ Recap Data Science
- ▣ Recap Neurale Netwerken



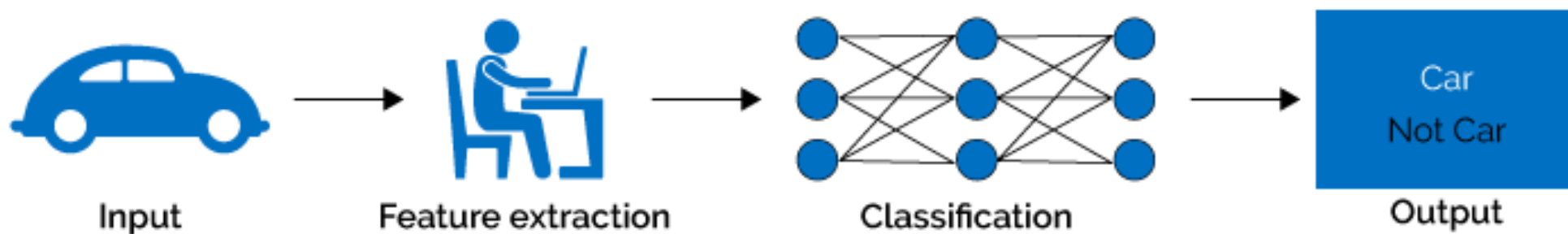
Wat gaan we leren?

Geavanceerde neurale netwerken

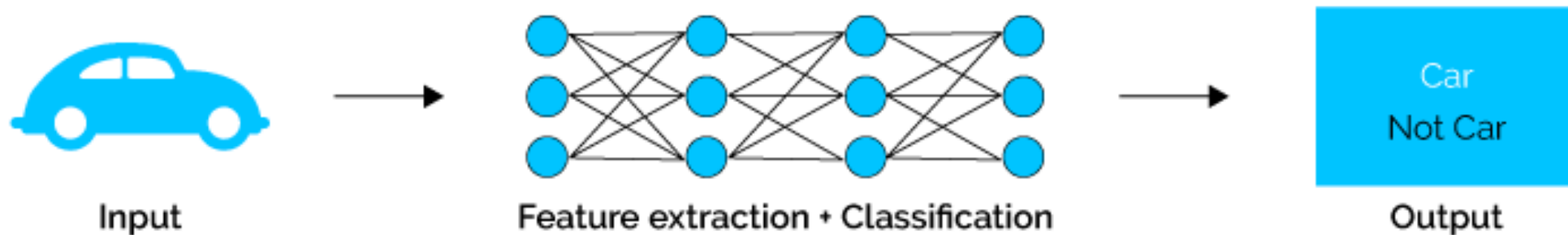


Waarom deep learning?

Machine Learning



Deep Learning



Selfie genomen op drukke plek? Betrapt!

Een nieuw online kunstproject gaat in videobeelden van drukke, openbare plekken op zoek naar mensen die Instagram-selfies nemen. Dat idee kan alleen maar uit de koker van Dries Depoorter komen.

Dominique Deckmyn

Woensdag 14 september 2022 om 3.25 uur



State of the art deep learning – deep fakes



State of the art deep learning

- ▣ Detecting troop positions
 - From satellite
 - From social media images
- ▣ High precision missiles
- ▣ ...



Pentagon AI tools are generating battlefield intelligence for Ukraine. An unusable Russian tank on the Kyiv-Zhytomyr highway after Ukrainian soldiers recaptured the region in early April 2022. METIN AKTAS/ANADOLU AGENCY VIA GETTY IMAGES

SCIENCE & TECH

AI Is Already Learning from Russia's War in Ukraine, DOD Says

Today's battlefield data is helping smart machines model the wars of the future.

<https://www.defenseone.com/technology/2022/04/ai-already-learning-russias-war-ukraine-dod-says/365978/>

AI impact

- ▣ Text to speech
- ▣ Reading brain functions
- ▣ Exoskeletons
- ▣ ...

ARTIFICIAL INTELLIGENCE'S IMPACT ON THE LIVES OF PEOPLE WITH DISABILITIES

ARTIFICIAL INTELLIGENCE LATEST NEWS

by IndustryTrends / September 11, 2022



<https://www.analyticsinsight.net/artificial-intelligences-impact-on-the-lives-of-people-with-disabilities/>

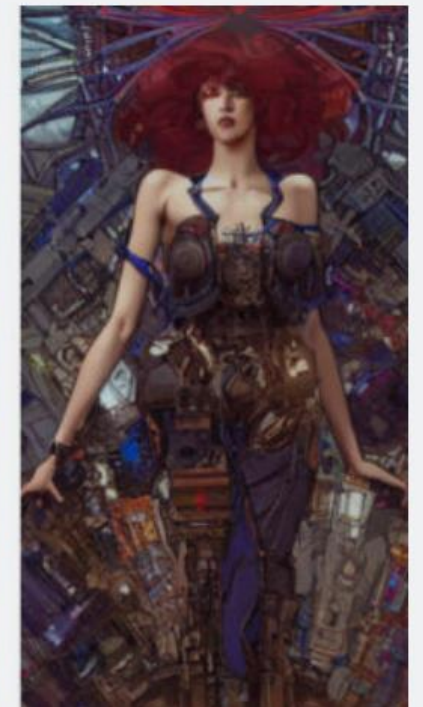
Creative sector

BEGUN, THE CLONE WARS HAVE —

With Stable Diffusion, you may never believe what you see online again

AI image synthesis goes open source, with big implications.

BENJ EDWARDS - 9/6/2022, 3:30 PM



Live examples

▣ Computervisie



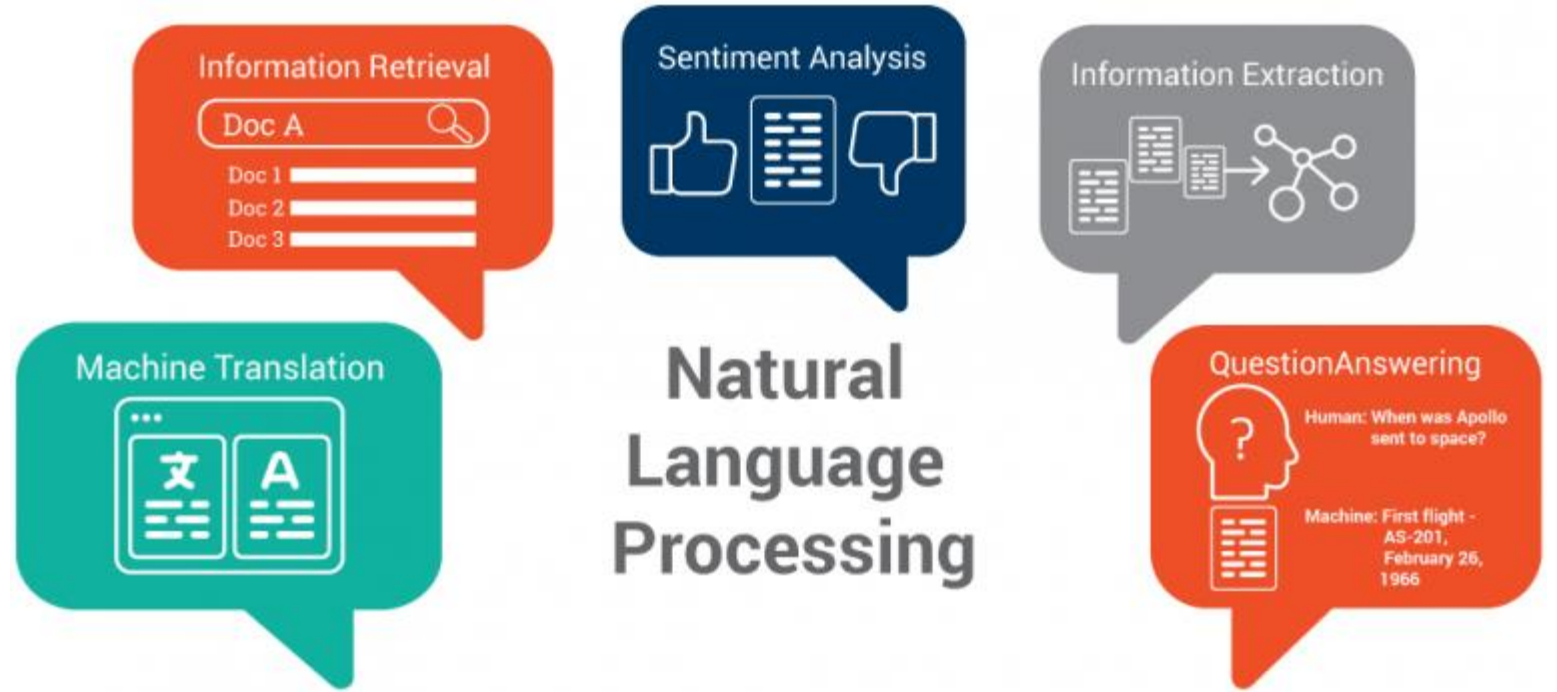
Live examples

- ▣ Computer vision
- ▣ Transformers



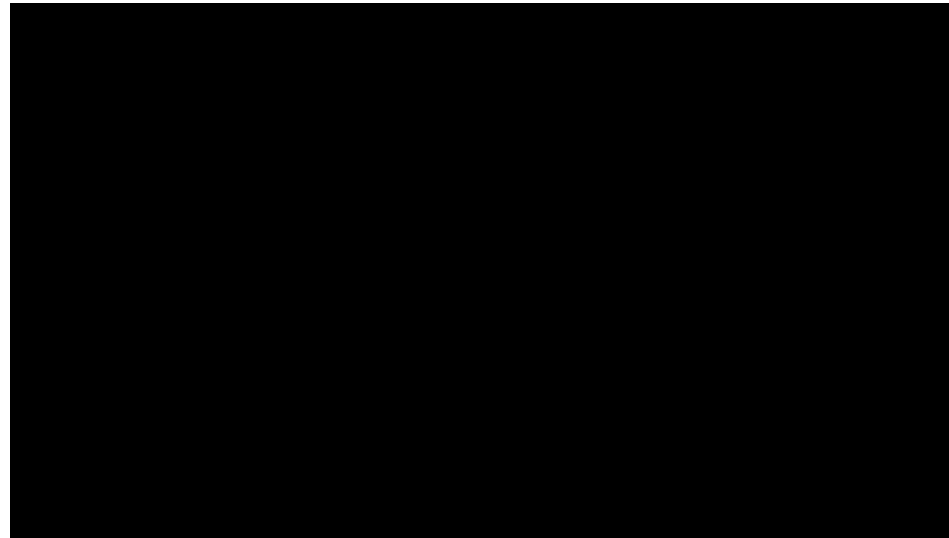
Examples

- ▣ Computer vision
- ▣ Transformers
- ▣ Chatbots

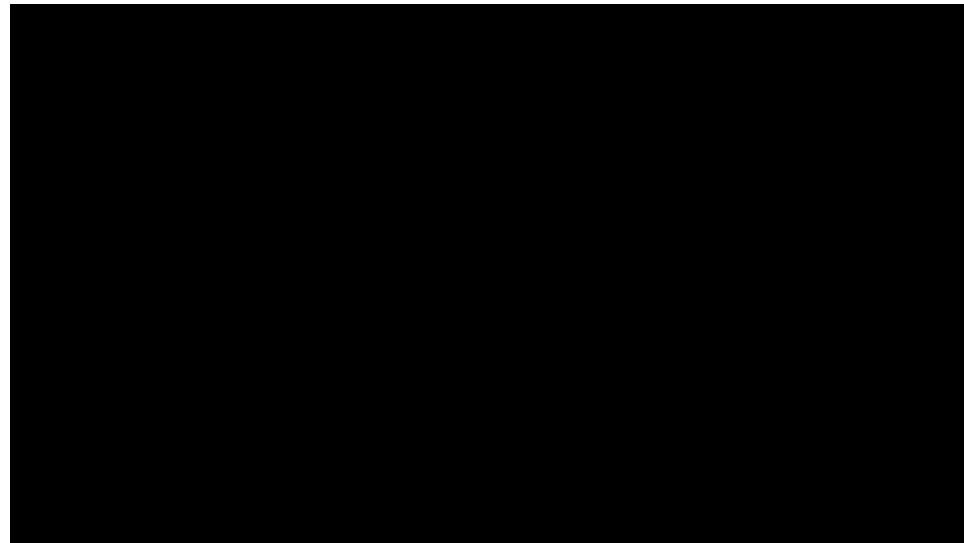


Examples

- ▣ Computer visie
- ▣ Transformers
- ▣ Chatbots
- ▣ Reinforcement learning



<https://www.youtube.com/watch?v=gn4nRCC9TwQ>



<https://www.youtube.com/watch?v=a8Bo2DHrow>



Dall-e

- ▣ <https://openai.com/dall-e-2/>
- ▣ Oil painting of donald duck in the style of picasso



Herhaling Data Science

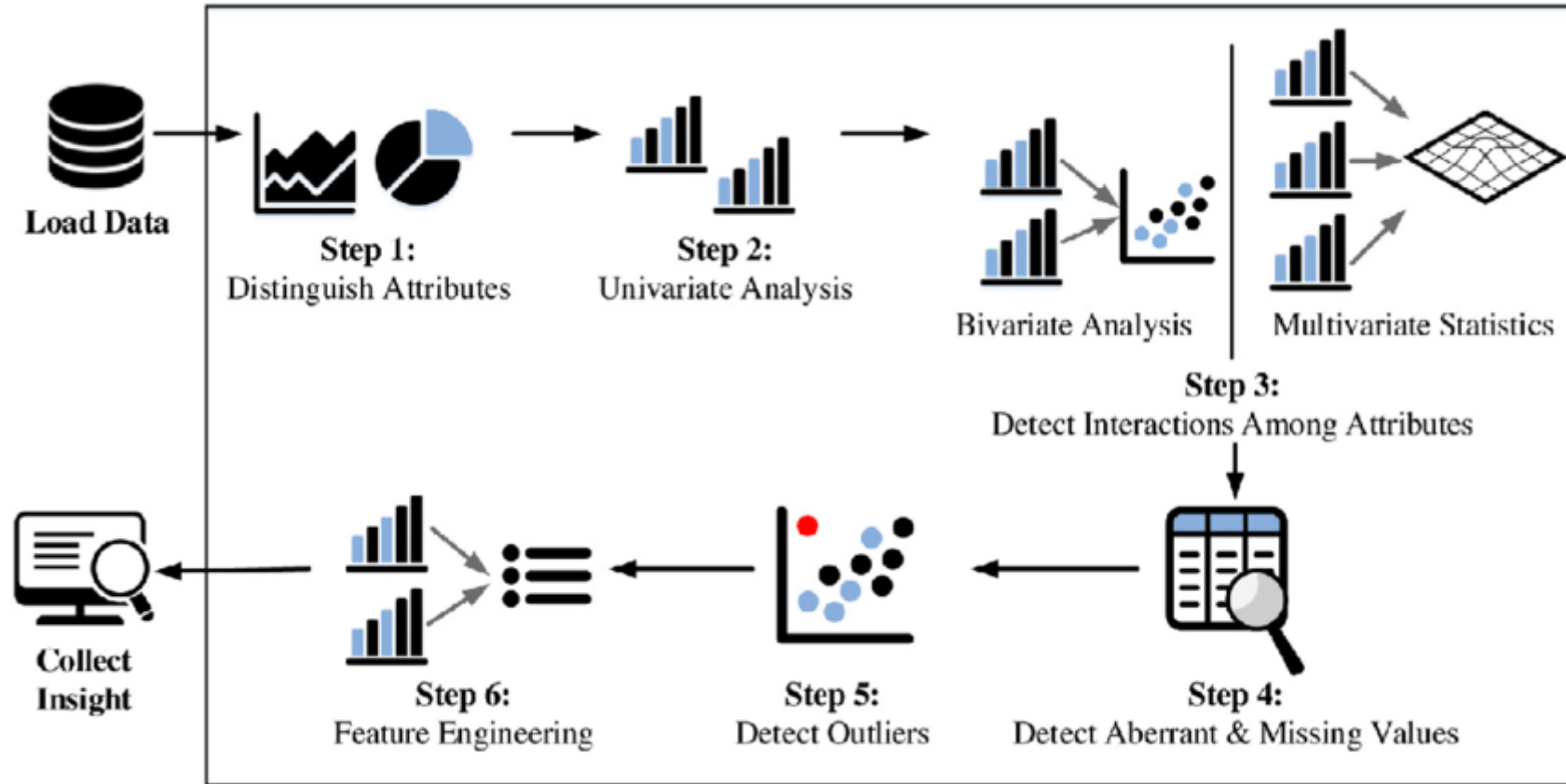
Data Science Life Cycle



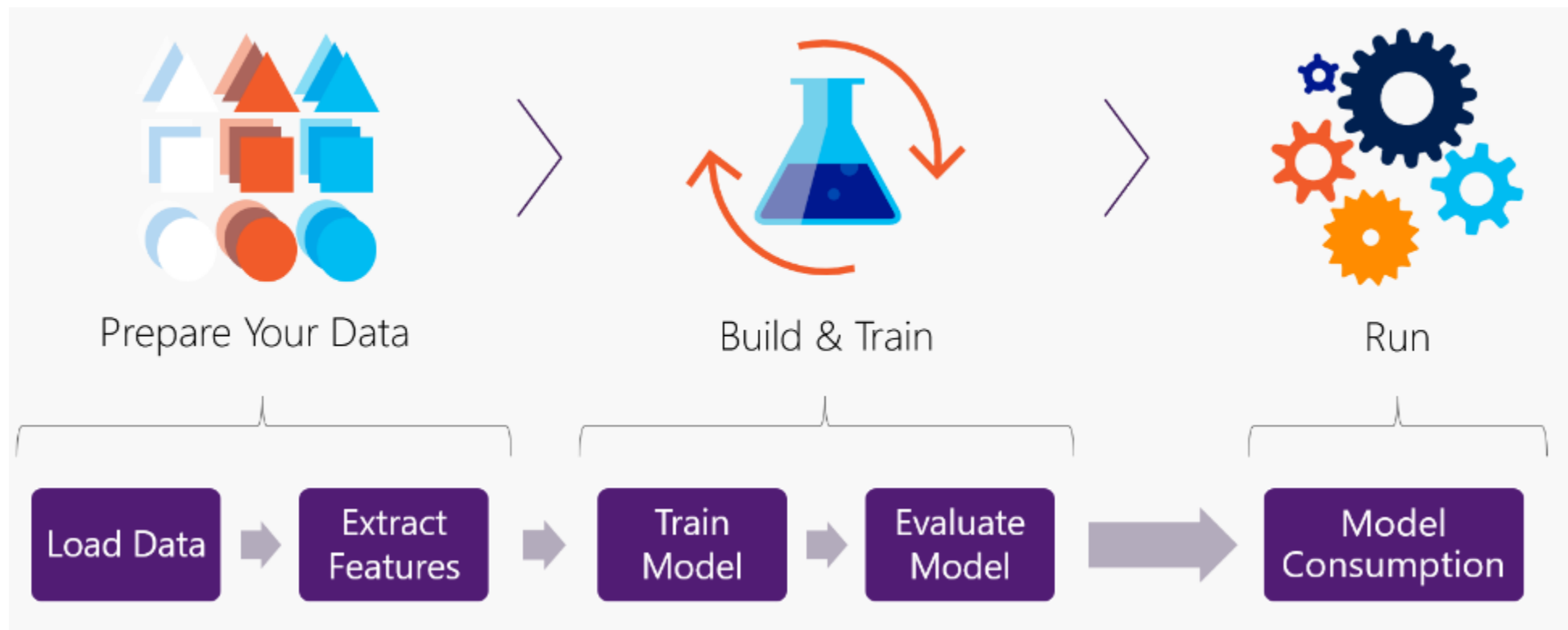
Data Cleaning



Exploratory Data Analysis



Data Modelling



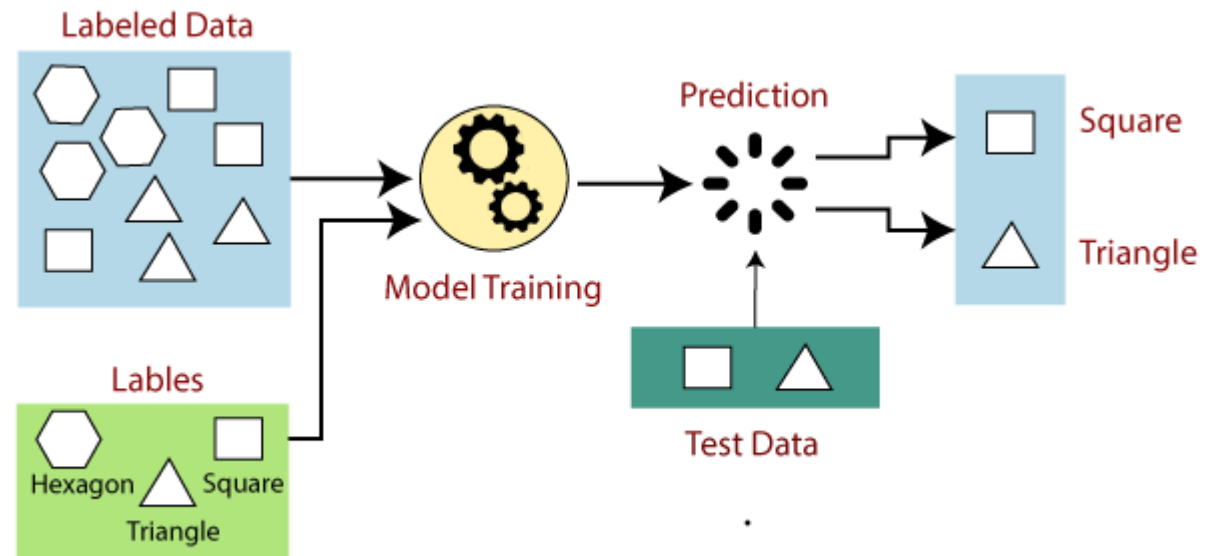
Supervised

▣ Regression

- Lineaire regressie

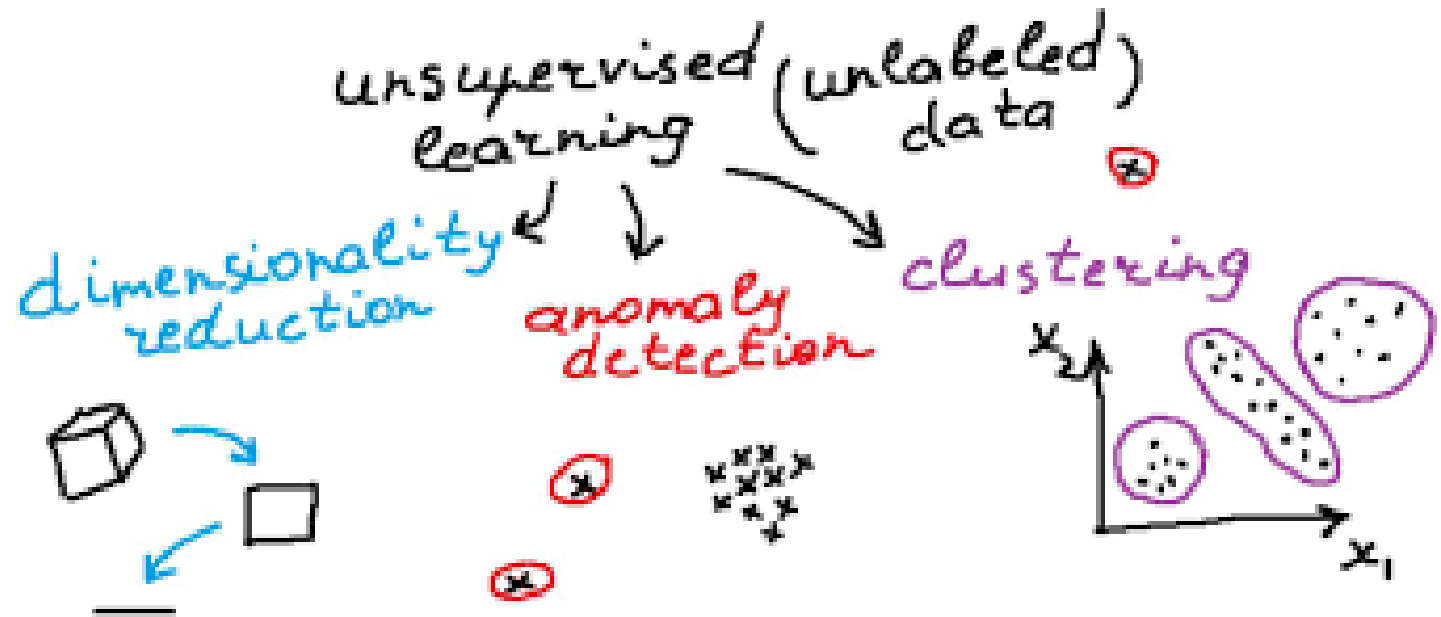
▣ Classification

- Logistische regressie
- SVM
- Naïve Bayes
- Decision Trees
- Random Forests
- K-Nearest Neighbours

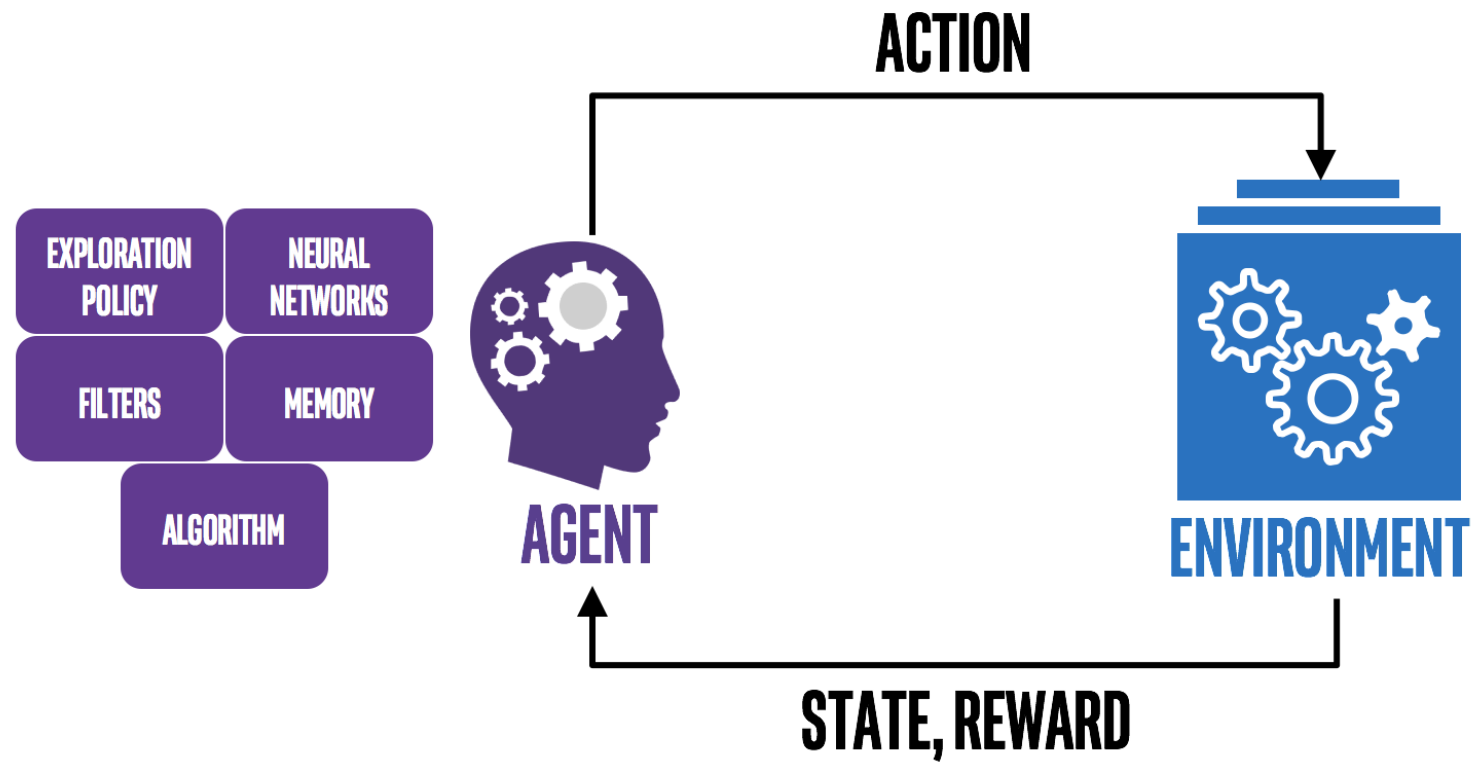


Unsupervised

- Clustering
 - K-Means
 - Mean Shift
- Anomaly/Outlier Detection
- Dimensionality Reduction
 - Principal Component Analysis



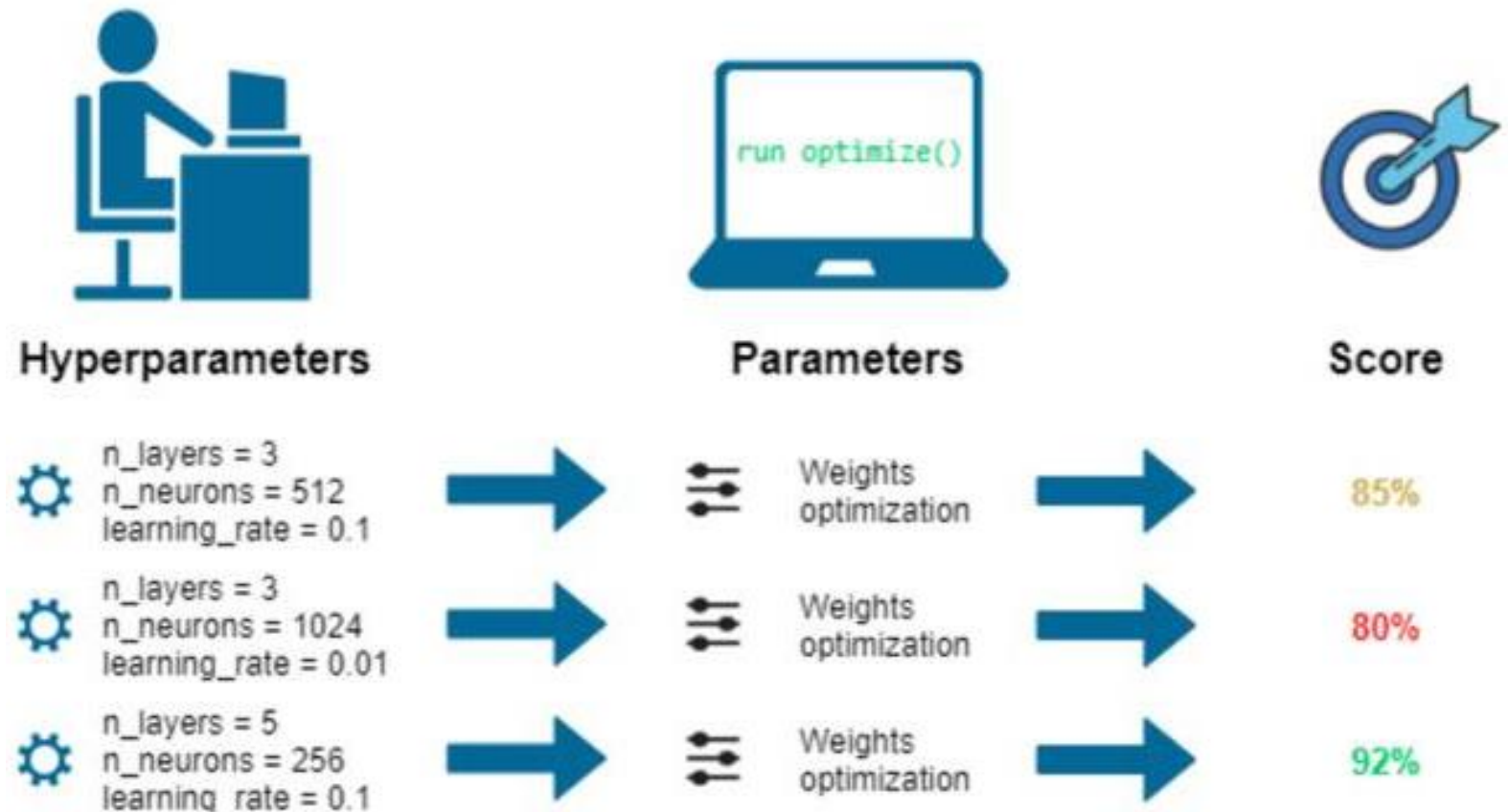
Reinforcement learning



Hyperparameter tuning

▣ Hoe beste parameters kiezen?

- Trial en error?
- Automatisch?
 - Gridsearch



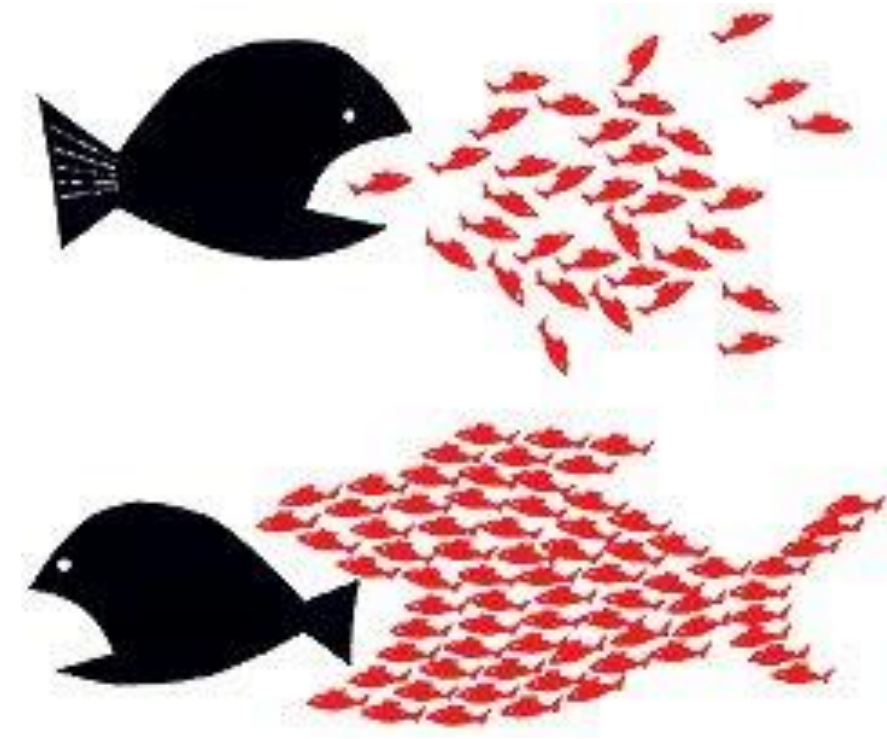
Pipelines

- ▣ Vaste volgorde van uit te voeren stappen
- ▣ Zowel data-cleaning als model trainen
- ▣ Aanpassen eenvoudiger
- ▣ Hyperparameter tuning van een pipeline
 - ▬ Ook parameters van de data-cleaning stappen aan te passen
- ▣ Altijd aangeraden om dit te gebruiken
 - ▬ Geen vergissingen mogelijk door een stap te vergeten/volgorde aan te passen
 - ▬ Kan eenvoudiger omgezet worden naar productie aangezien de preprocessing stappen erin verwerkt zijn.



Ensembles

- ▣ Meerdere modellen werken samen
 - Elk model moet accurater zijn dan 50%
 - Hoe meer modellen hoe beter
 - Vaak gebaseerd op decision trees maar niet verplicht
- ▣ Bagging
 - Alle modellen in parallel op dezelfde data
- ▣ Boosting
 - Sequentieel, meer kans dat fouten getoond worden aan volgende modellen
- ▣ XGBoost



Woordenlijst

▣ Bespreek in je eigen woorden de termen die in de voorgaande slides aan bod zijn gekomen

- ▬ Data Cleaning
- ▬ Data Exploration
- ▬ Data Modelling
- ▬ Supervised learning
- ▬ Unsupervised learning
- ▬ Reinforcement learning
- ▬ Regression
- ▬ Classification
- ▬ Clustering
- ▬ Dimensionality Reduction
- ▬ Anomaly detection
- ▬ Hyperparameter tuning
- ▬ Pipelines
- ▬ Ensembles
- ▬ Bagging
- ▬ Boosting
- ▬ Model evaluation
- ▬ Parameter
- ▬ Hyperparameter



Oefening

- ▣ Maak de notebook om nog eens de vaardigheden uit data science op te frissen



Tensorflow

Wat is het?

- ▣ An end-to-end open source machine learning platform (framework)
- ▣ Vooral voor neurale netwerken
- ▣ Ontwikkeld door onderzoekers van Google

- ▣ Er bestaan goede tutorials/guides
 - ▣ <https://www.tensorflow.org/learn>
- ▣ Er bestaat ook een bibliotheek met pre-trained modellen die gebruikt kunnen worden
 - ▣ Zoals in de demos
 - ▣ <https://www.tensorflow.org/hub>

▣ Ondersteund

- ▬ Multidimensionele rij-gebaseerde numerieke operaties
 - Gelijkaardig aan numpy in python (matrix operaties)
- ▬ GPU-ondersteunde en gedistribueerde berekeningen
- ▬ Automatische berekening van afgeleiden
- ▬ Opbouwen, trainen, evalueren en exporteren van ML-modellen
- ▬ ...



Keras

- Sinds versie 2 wordt er gebruik gemaakt van Keras in Tensorflow
 - ▬ Vlottere manier om modellen te programmeren
 - ▬ Snellere prototypes van deep learning models
 - ▬ Tensorflow bied nog de mogelijkheid om alle fijnere details te controleren en wijzigen
- Library boven tensorflow (en een aantal andere platformen)

Tensorflow basics: Tensor

- ▣ Multidimensionele rij
- ▣ Immutable / Onwijzigbaar
- ▣ Bewerkingen kunnen op de GPU uitgevoerd worden
- ▣ Meer info + code voorbeelden <https://www.tensorflow.org/guide/tensor>

```
import tensorflow as tf

x = tf.constant([[1., 2., 3.],
                 [4., 5., 6.]])
```

```
x + x
```

```
5 * x
```

```
if tf.config.list_physical_devices('GPU'):
    print("TensorFlow **IS** using the GPU")
```

Tensorflow basics: Tensor

▣ Enkele belangrijke termen

- ▬ Axis: Een dimensie van een tensor
- ▬ Shape: aantal elementen op elke as/dimensie van de tensor
- ▬ Rank: Aantal assen/dimensies van de tensor
 - Een getal heeft rang 0
 - Een rij heeft rang 1, matrix rang 2
- ▬ Size: Totaal aantal elementen in de tensor

▣ Indexing op de python manier met slicing

▣ Sparse tensor: Efficiëntere manier om een grote tensor met veel nullen op te slaan

Tensorflow variables

- ▣ State die kan veranderen kan niet bewaard zijn in een tensor

- Bvb gewichten van de modellen
- Gebruik een variabele (tf.Variable)
- Waarde kan toegekend worden met assign

```
var = tf.Variable([0.0, 0.0, 0.0])  
var.assign([1, 2, 3])  
var.assign_add([1, 1, 1])
```

- ▣ <https://www.tensorflow.org/guide/variable>



Tensorflow basics

■ Eager execution

- ▬ Operaties uitgevoerd door python, operatie per operatie
- ▬ Resultaten steeds teruggegeven aan python

■ Graph execution

- ▬ Flexibeler omdat de graph gebruikt kan worden waar geen python mogelijk is
 - Mobile, IoT, embedded applications, ...
- ▬ Vaak performanter
- ▬ Berekeningen uitgevoerd als een graaf (stappenplan)

■ https://www.tensorflow.org/guide/intro_to_graphs#what_are_graphs

Graph Execution

■ Maak gebruik van tf.function

- ▬ Graph execution automatisch ook gebruikt voor alle functies die de functie oproept
- ▬ Ook standaard python code (if, for, ...) omgezet naar graph
- ▬ Argumenten en datatypes van de argumenten belangrijk
 - Wordt ook de signature of input signature genoemd

```
# Define a Python function.
def a_regular_function(x, y, b):
    x = tf.matmul(x, y)
    x = x + b
    return x

# `a_function_that_uses_a_graph` is a TensorFlow `Function`.
a_function_that_uses_a_graph = tf.function(a_regular_function)
```

Graph Execution

▣ Niet alles wordt uitgevoerd


- Enkel de operaties die een gevolg/observeerbaar effect hebben uitgevoerd
 - Return value van de functie
 - Gedocumenteerde side-effects zoals
 - ▾ `tf.print()` (en niet de standaard `print`)
 - ▾ `tf.debugging`
 - ▾ Wijzigingen aan een `tf.Variable` object

Tf.function best practices

- ▣ Experimenteer met `@tf.function` decoraties om ervaring met graph executions te krijgen
- ▣ Wissel vaak tussen eager en graph execution om vast te stellen op welk punt er verschil is
- ▣ Maak de `tf.Variables` aan buiten de functies en wijzig ze binnen de functies
 - ▢ Ook voor layers, models, optimizers,
- ▣ Vermijd global standaard python variabelen te gebruiken.
- ▣ Probeer geen standaard python variabelen te gebruiken als input
- ▣ Plaats zoveel mogelijk berekeningen in een `tf.function` voor zo hoog mogelijke performance boost



Vorbereitung volgende les

- 
- Bestudeer de tensorflow guide van de basics die in deze les behandeld zijn
 - ▬ <https://www.tensorflow.org/guide/basics>
 - Bekijk ook het gedeelte rond modules, layers en models
 - ▬ https://www.tensorflow.org/guide/intro_to_modules
 - ▬ https://www.tensorflow.org/guide/basic_training_loops
 - ▬ Nog niet gezien in de les
 - ▬ Noteer je vragen, zaken waar je niet zeker over bent
 - ▬ Volgende les is er een moment waarbij ik je vragen bekijk
 - ▬ Geef hier voldoende aandacht aan om volgende les vlot te kunnen volgen