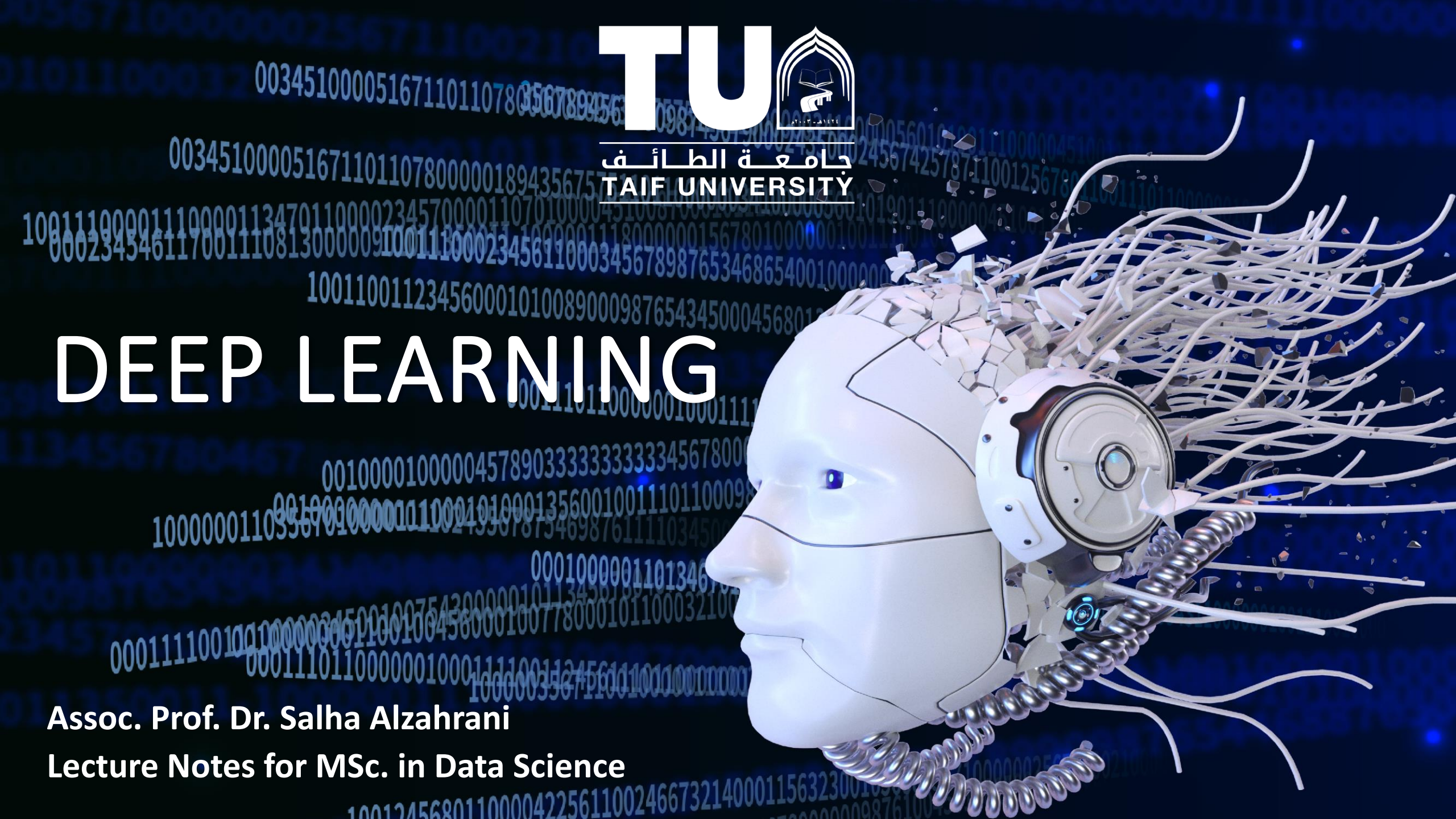
جامعة الطائف
TAIF UNIVERSITY

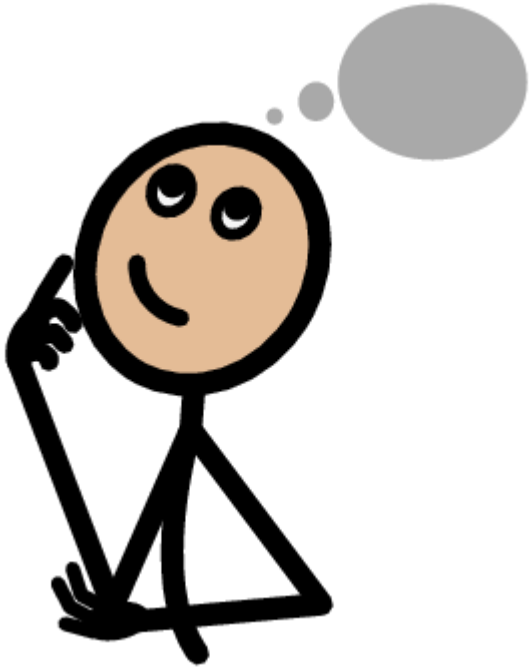
DEEP LEARNING



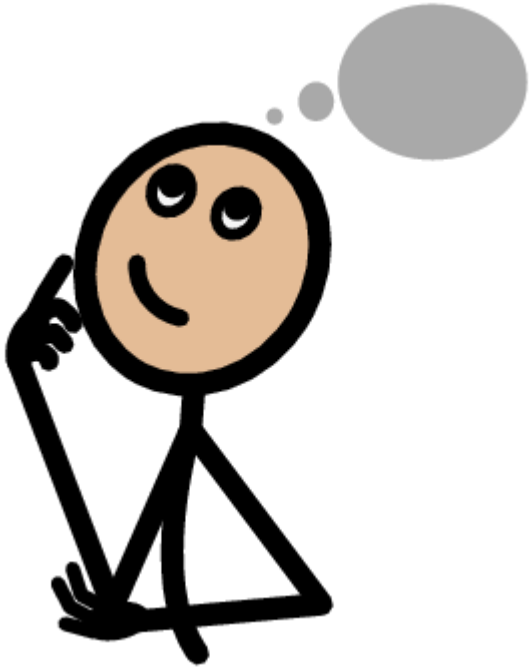
Assoc. Prof. Dr. Salha Alzahrani

Lecture Notes for MSc. in Data Science

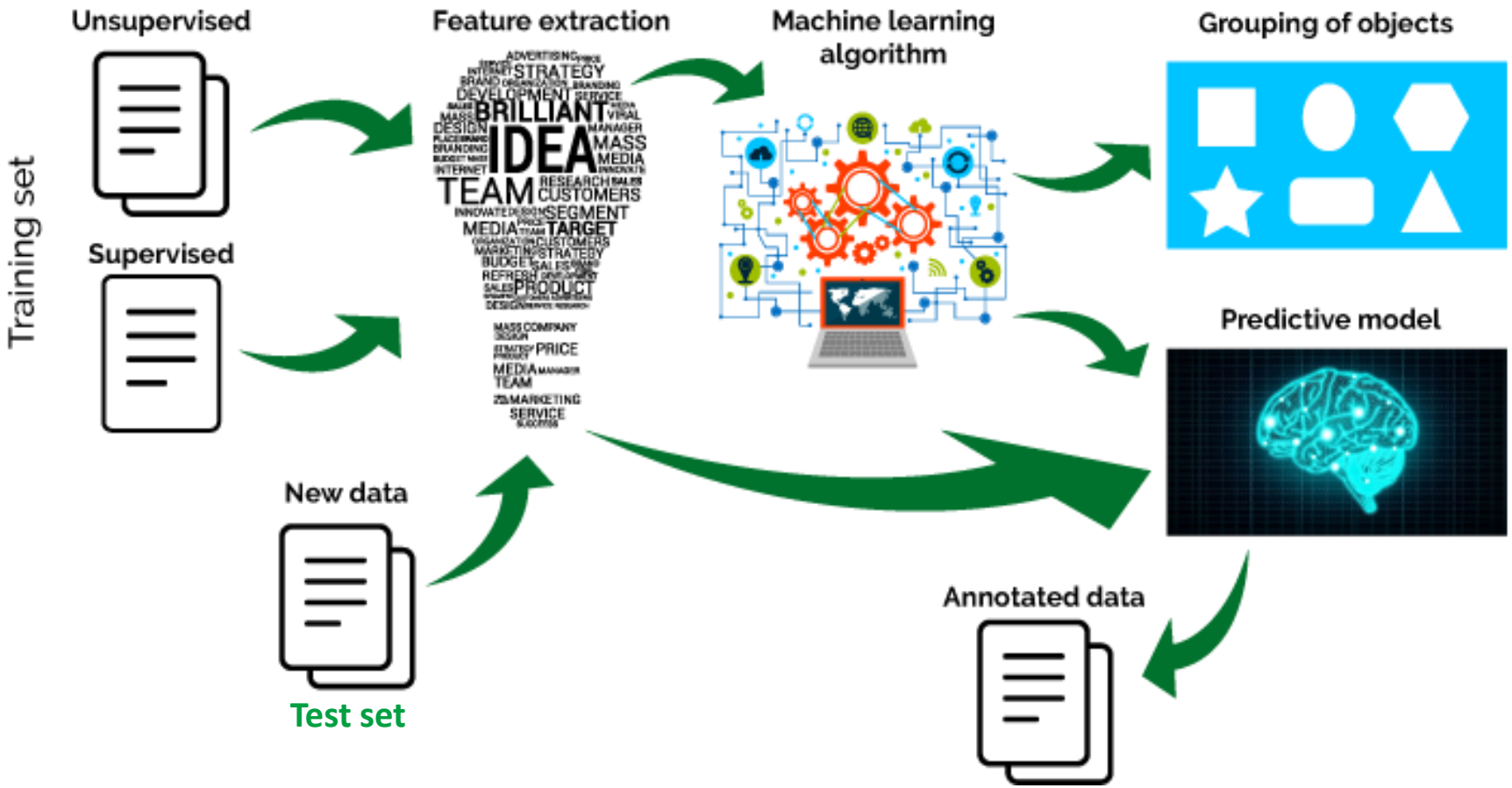
1st Question….
What is deep learning?



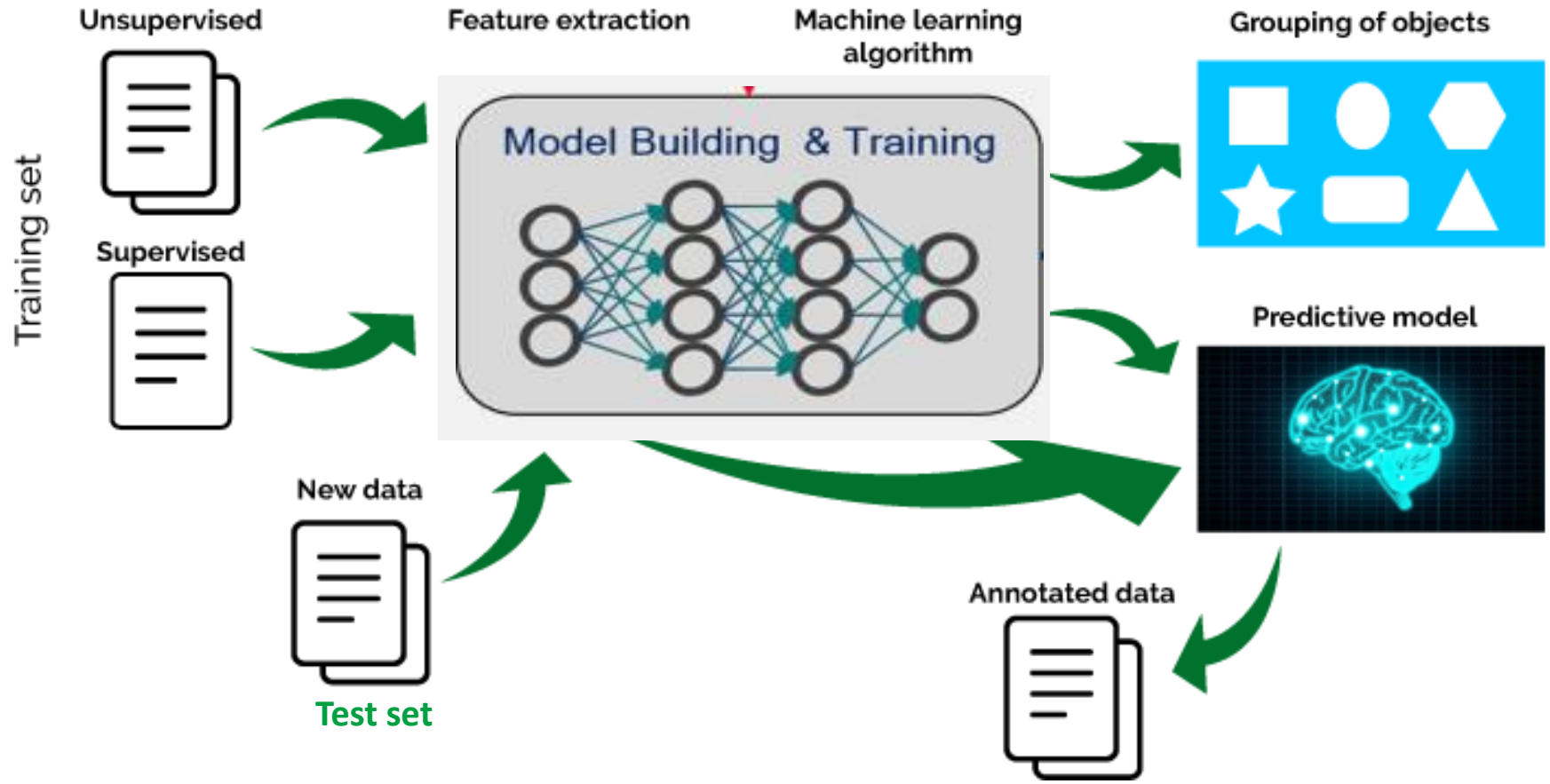
2nd Question…
What is the difference between deep learning and machine learning?



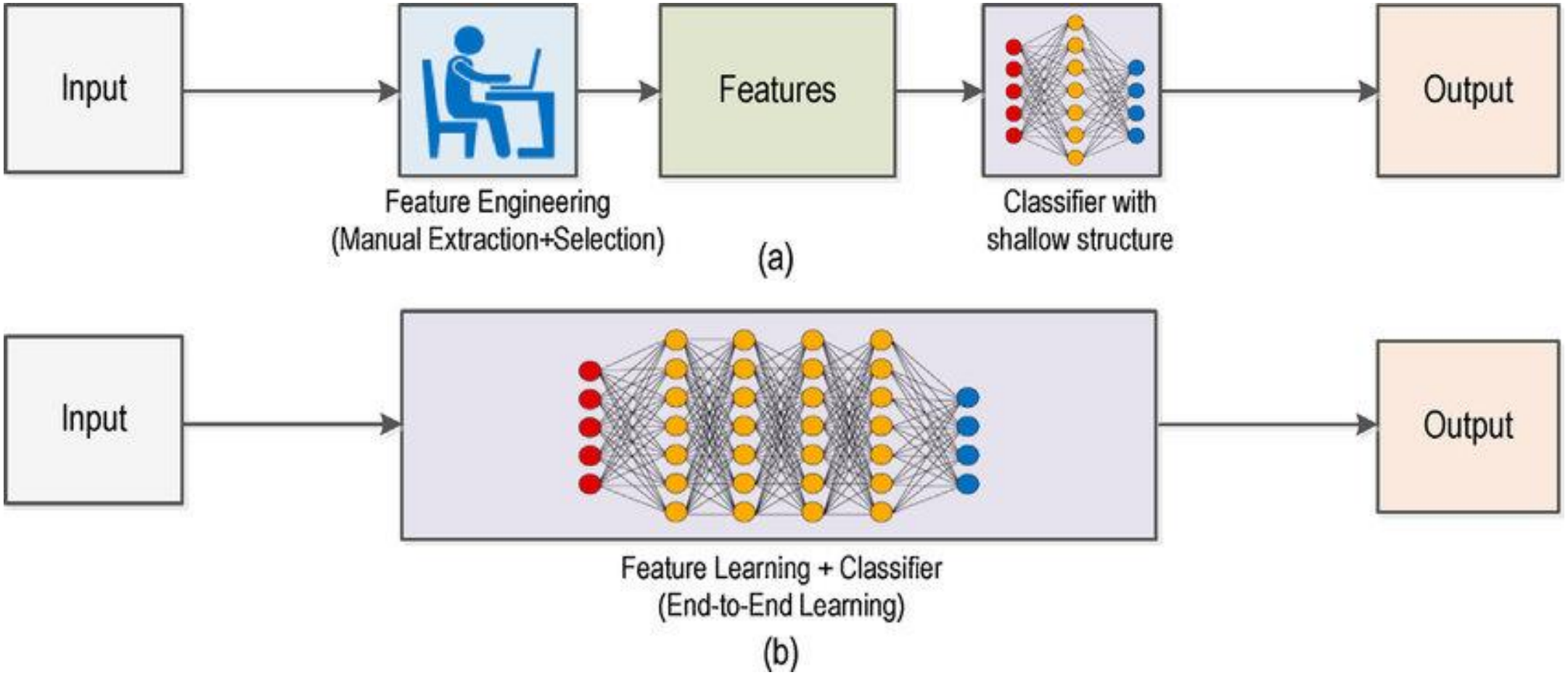
Overall Framework for Machine Learning



Overall Framework for Deep Learning



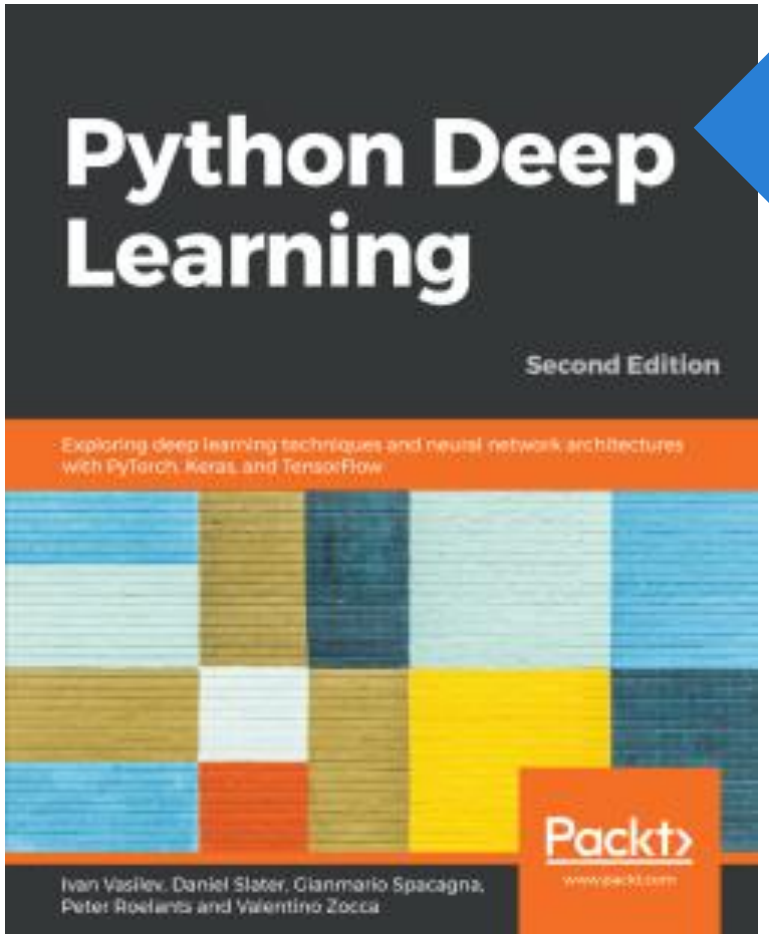
Overall Framework for Machine Learning Using NN vs. Deep Learning



Course Description and Learning Outcome!

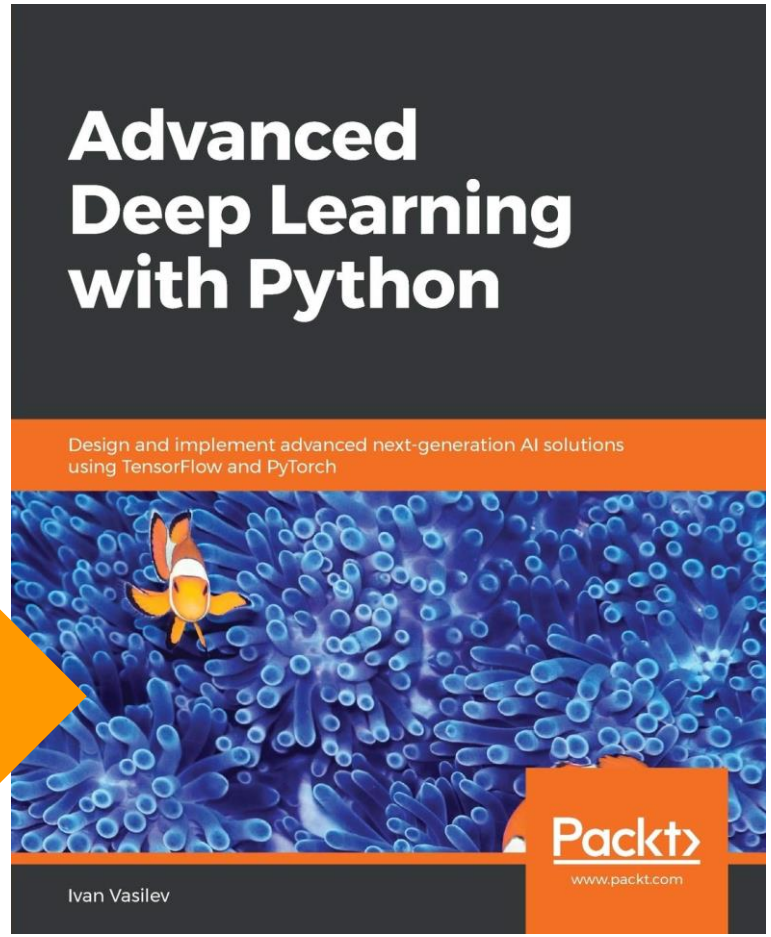
This course will introduce the students to neural networks, and how to train them with high-performance algorithms using different neural network architectures, such as convolutional networks, recurrent neural networks, long short-term memory (LSTM) networks, and capsule networks. Students will learn how to solve problems in the fields of computer vision, natural language processing (NLP), and speech recognition. Students will learn generative model approaches such as variational autoencoders and Generative Adversarial Networks (GANs) to generate images. Finally, the course will introduce the students to the state-of-the-art algorithms of reinforcement learning that are the main components behind popular game applications.

The book!

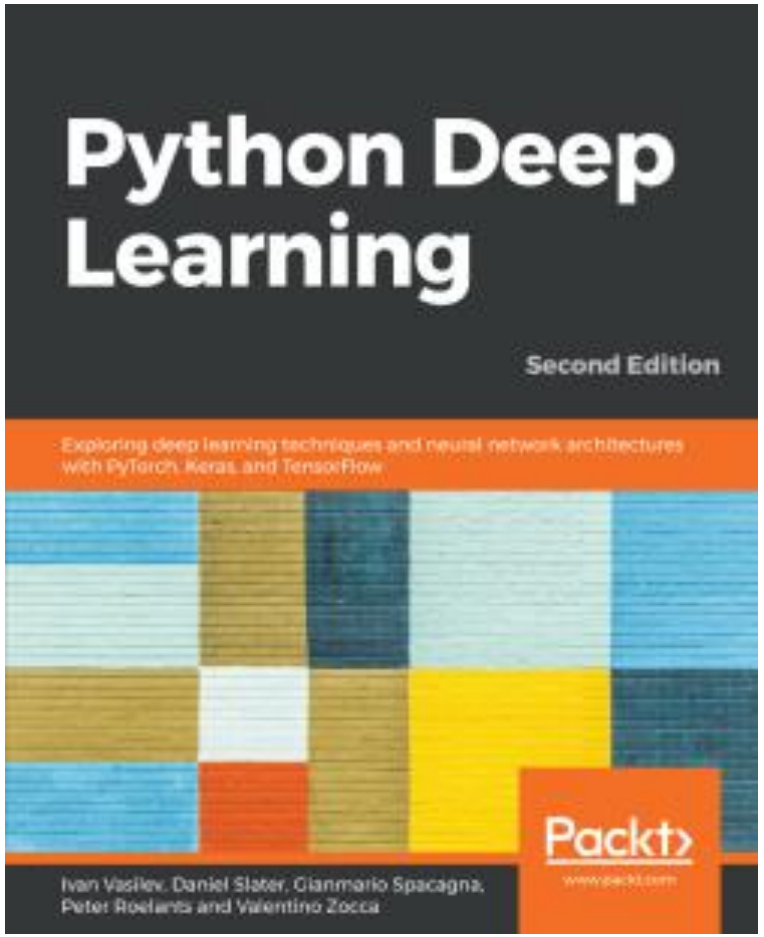


Your Present

Your Future



The book!



1. Machine Learning - an Introduction
2. Neural Networks
3. Deep Learning Fundamentals
4. Computer Vision with Convolutional Networks
5. Advanced Computer Vision
6. Generating Images with GANs and VAEs
7. Recurrent Neural Networks and Language Models
8. Reinforcement Learning Theory
9. Deep Reinforcement Learning for Games
10. Deep Learning in Autonomous Vehicles

https://www.packtpub.com/product/python-deep-learning

What this book covers!

Chapter 2, *Neural Networks*, will introduce you to the mathematics of neural networks. We'll learn about their structure, how they make predictions (that's the feedforward part), and how to train them using gradient descent and backpropagation. The chapter will also discuss how to represent operations with neural networks as vector operations.

Chapter 3, *Deep Learning Fundamentals*, will explain the rationale behind using *deep* neural networks (as opposed to shallow ones). It will take an overview of the most popular DL libraries and real-world applications of DL.

Chapter 4, *Computer Vision with Convolutional Networks*, teaches you about convolutional neural networks (the most popular type of neural network for computer vision tasks). We'll learn about their architecture and building blocks (the convolutional, pooling, and capsule layers) and how to use a convolutional network for an image classification task.

Chapter 5, *Advanced Computer Vision*, will build on the previous chapter and cover more advanced computer vision topics. You will learn not only how to classify images, but also how to detect an object's location and segment every pixel of an image. We'll learn about advanced convolutional network architectures and the useful practical technique of transfer learning.

Chapter 6, *Generating Images with GANs and VAEs*, will introduce generative models (as opposed to discriminative models, which is what we'll have covered up until this point). You will learn about two of the most popular unsupervised generative model approaches, VAEs and GANs, as well some of their exciting applications.

What this book covers!

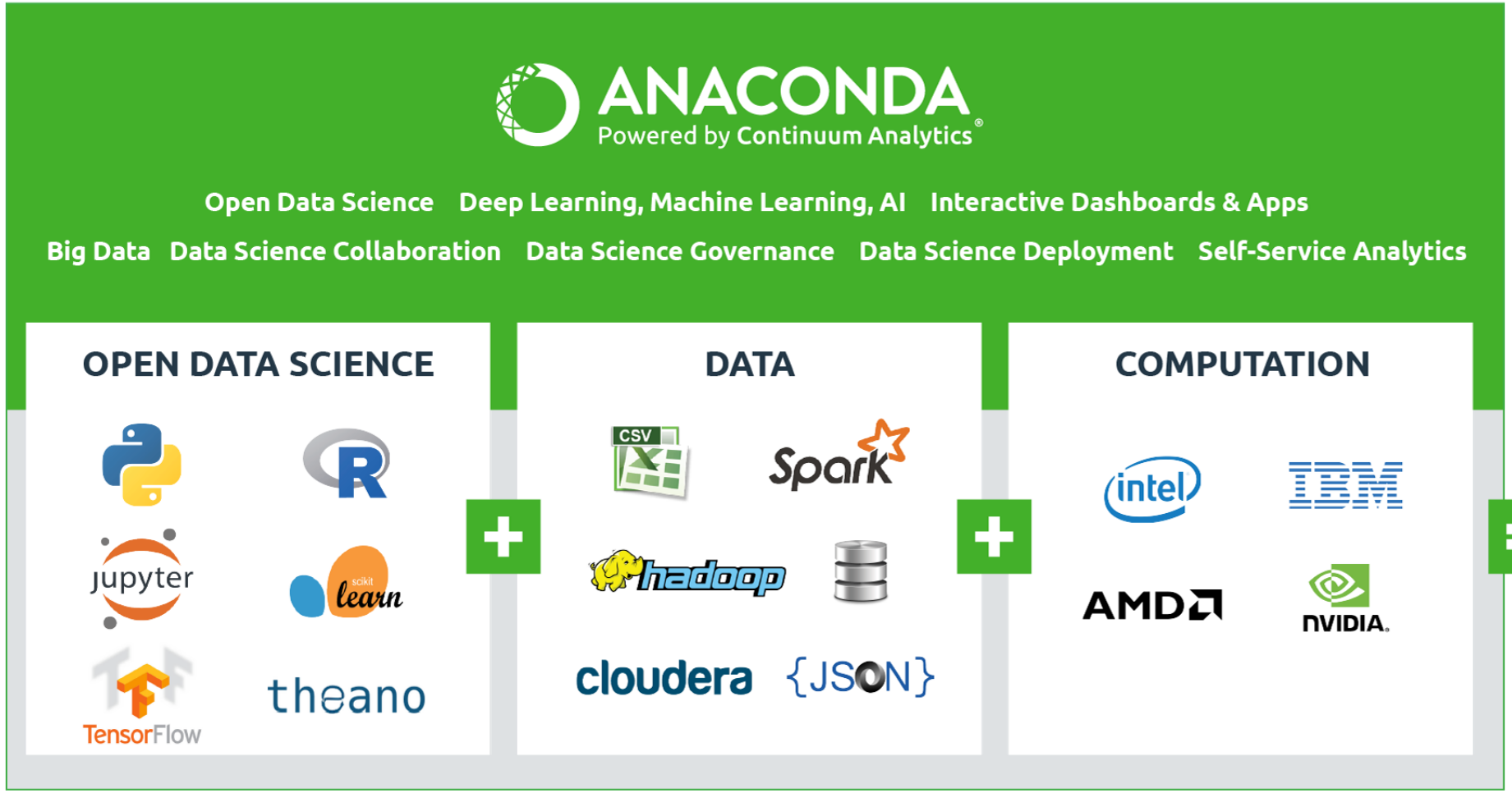
Chapter 7, *Recurrent Neural Networks and Language Models*, will introduce you to the most popular recurrent network architectures: LSTM and **gated recurrent unit** (GRU). We'll learn about the paradigms of NLP with recurrent neural networks and the latest algorithms and architectures to solve NLP problems. We'll also learn the basics of speech-to-text recognition.

Chapter 8, *Reinforcement Learning Theory*, will introduce you to the main paradigms and terms of RL, a separate ML field. You will learn about the most important RL algorithms. We'll also learn about the link between DL and RL. Throughout the chapter, we will use toy examples to better demonstrate the concepts of RL.

Chapter 9, *Deep Reinforcement Learning for Games*, you will understand some real-world applications of RL algorithms, such as playing board games and computer games. We'll learn how to combine the knowledge from the previous parts of the book to create better-than-human computer players on some popular games.

Chapter 10, *Deep Learning in Autonomous vehicles,* we'll discuss what sensors autonomous vehicles use, so they can create the 3D model of the environment. These include cameras, radar sensors, ultrasound sensors, Lidar, as well as accurate GPS positioning. We'll talk about how to apply deep learning algorithms for processing the input of these sensors. For example, we can use instance segmentation and object detection to detect pedestrians and vehicles using the vehicle cameras. We'll also make an overview of some of the approaches vehicle manufacturers use to solve this problem (for example Audi, Tesla, and so on).

Practical Examples!



RESULTS

- Deploying Data Science models and visualizations easily
- Building interactive visualizations and web applications
- Collaboration and sharing of results easily
- Scaling up & out Data Science to meet performance goals
- Reproducibility and governance for Data Science assets

TensorFlow Install Learn API More Search English GitHub Sign in

An end-to-end open source machine learning platform

TensorFlow | For JavaScript | For Mobile & Edge | For Production

The core open source library to help you develop and train ML models. Get started quickly by running Colab notebooks directly in your browser.

Get started with TensorFlow

Keras

Simple. Flexible. Powerful.

Get started | API docs | Guides | Examples

```
from tensorflow import keras
from tensorflow.keras import layers

# Instantiate a trained vision model
vision_model = keras.applications.ResNet50()

# This is our video encoding branch using the trained vision_model
video_input = keras.Input(shape=(100, None, None, 3))
encoded_frame_sequence = layers.TimeDistributed(vision_model)(video_input)
encoded_video = layers.LSTM(256)(encoded_frame_sequence)

# This is our text-processing branch for the question input
question_input = keras.Input(shape=(100,), dtype='int32')
embedded_question = layers.Embedding(10000, 256)(question_input)
encoded_question = layers.LSTM(256)(embedded_question)

# And this is our video question answering model:
merged = keras.layers.concatenate([encoded_video, encoded_question])
output = keras.layers.Dense(1000, activation='softmax')(merged)
video_qa_model = keras.Model(inputs=[video_input, question_input],
                             outputs=output)
```

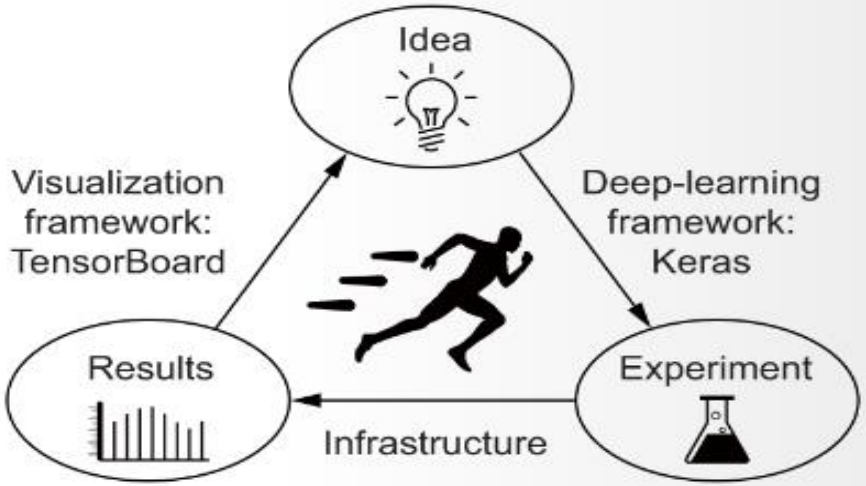
Deep learning for humans.

Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear & actionable error messages. It also has extensive documentation and developer guides.

Keras

Iterate at the speed of thought.

Keras is the most used deep learning framework among top-5 winning teams on Kaggle. Because Keras makes it easier to run new experiments, it empowers you to try more ideas than your competition, faster. And this is how you win.



Keras

Exascale machine learning.

Built on top of TensorFlow 2, Keras is an industry-strength framework that can scale to large clusters of GPUs or an entire TPU pod. It's not only possible; it's easy.

Watch this video: What is Deep Learning in 5 Minutes!


You don't have to be great to start, but you have to start to be great.
Zig Ziglar
quotefancy
@SalhaAlzahrani