Deep Learning

Redes Neuronales con TensorFlow

Agosto 29

Deep Learning

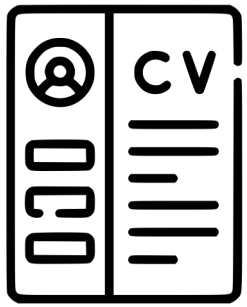
Redes Neuronales con TensorFlow

Agosto 29

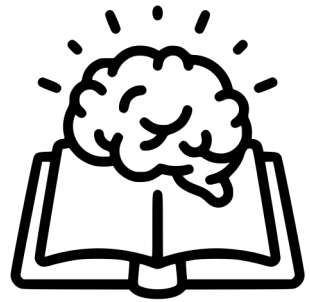
- Introducción a la clase
- ¿Por qué Deep Learning?
  - ML vs DL vs AI
  - Materiales del curso
  - Google Colab

Emilio Sandoval Palomino

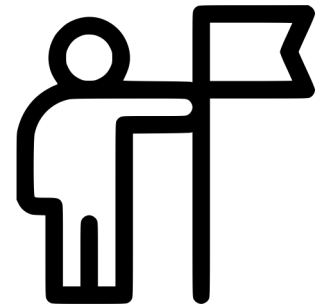
Machine learning Engineer



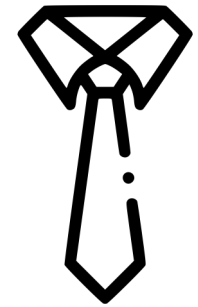
Resúmen general



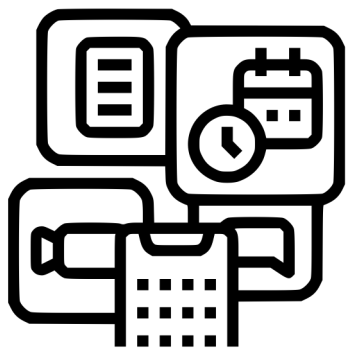
Conocimientos y cursos previos



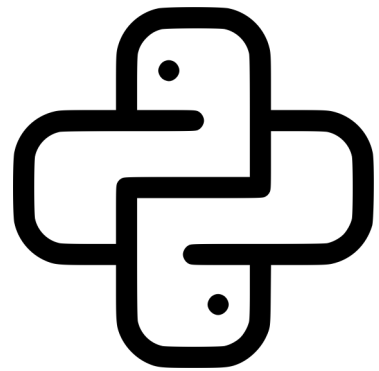
Expectativas del curso



Utilidad en su vida profesional

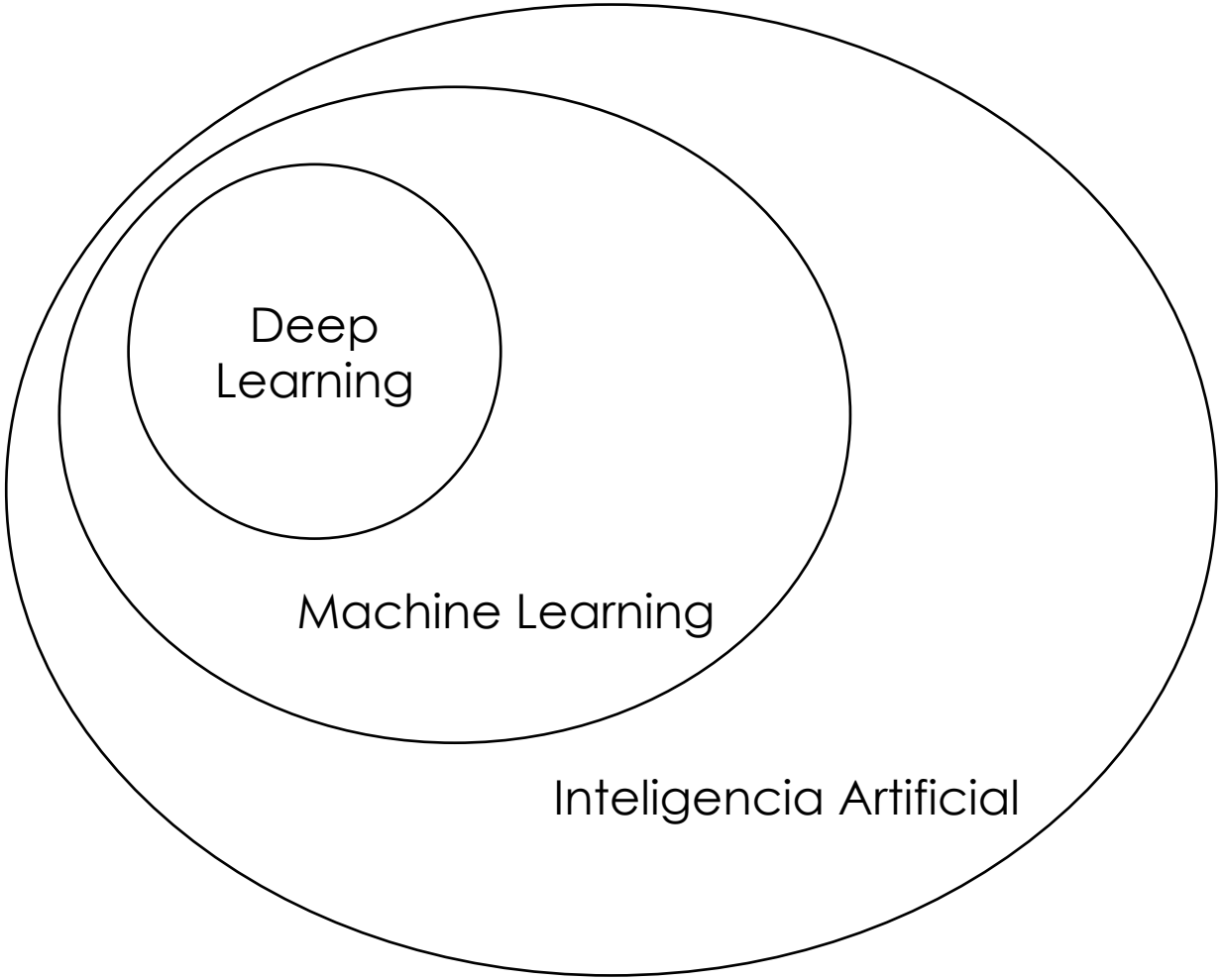


Aplicaciones que han escuchado y/o desean hacer



Dominio Python

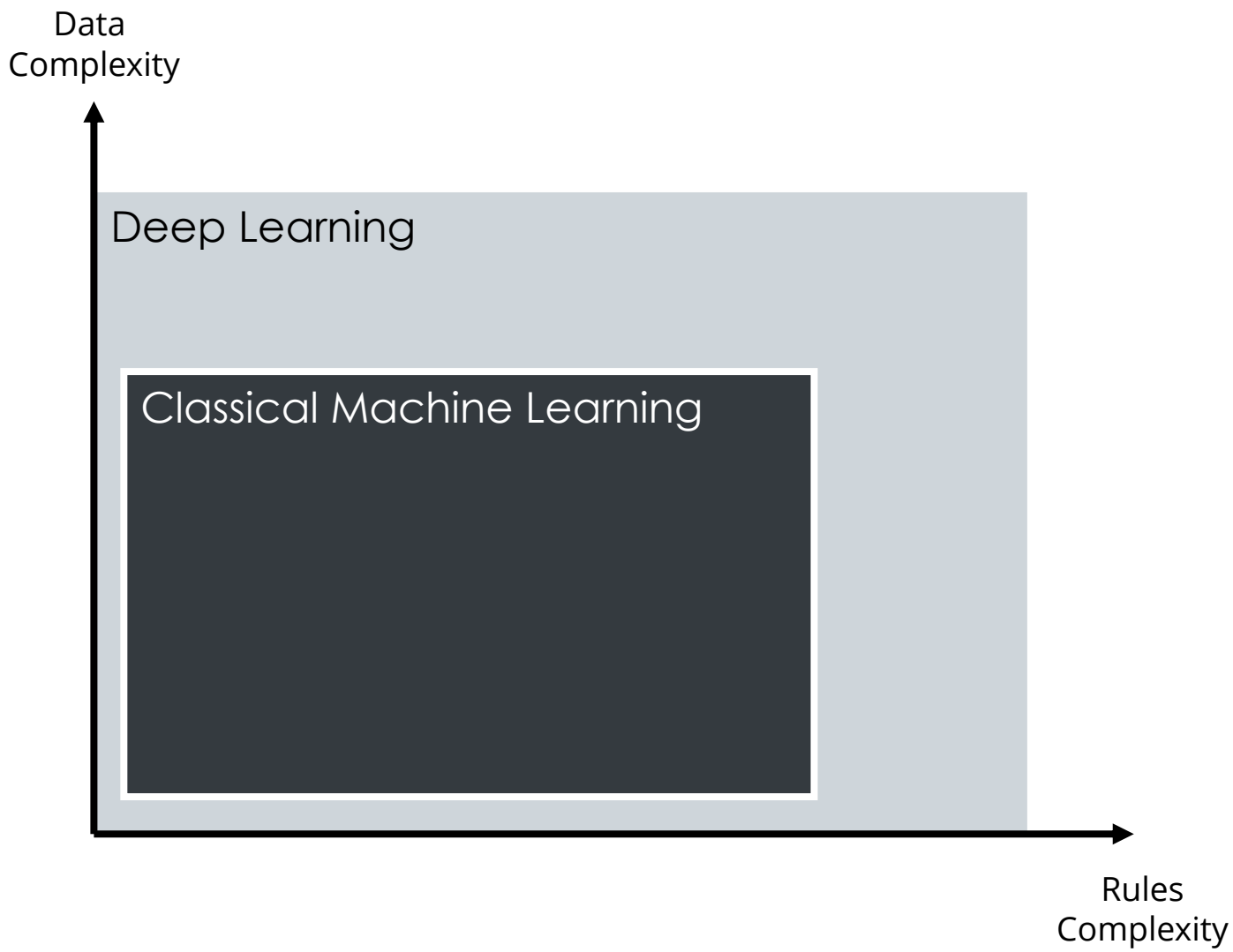
Mejor forma de aprender


Deep
Learning
Machine Learning
Inteligencia Artificial

Problemas lineales

Problemas no lineales

Problemas no lineales



*Tomado de Meor Amer (2022)

DeepLearning.AI

HD INSTITUTO HUMAN DELTA MR

- Intro to deep learning
- Deep Sequence Model
- Intro to TF: Music Generation
- Computer Vision
- Generative Modelling
- Recognition Systems
- Reinforcement Learning
- Self-Driving
- Speech Recognition

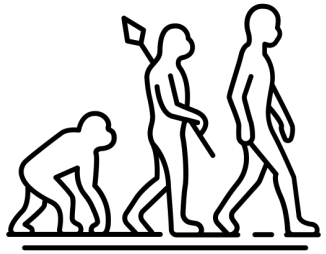
- Neural Networks and Deep Learning
- Hyperparameter tunning, Regularization & Optimization
- Structuring Machine Learning Projects
- Convolutional Neural Networks
- Sequence Models

- Regression
- Image Classification (convolutionals)
- Transfer Learning
- Natural Language Processing
- Time Series, sequences and predictions

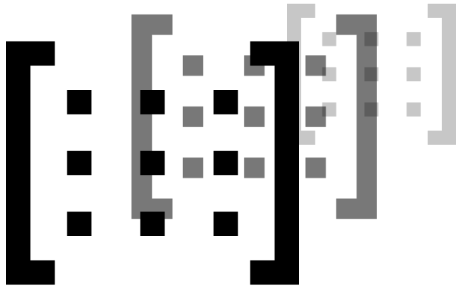
- Fundamentos teóricos de Deep Learning
- Introducción a los tensores
- Regression
- Classification
- Convolutional Networks
- Transfer Learning
- Recurrent Networks

HD INSTITUTO HUMAN DELTA MR

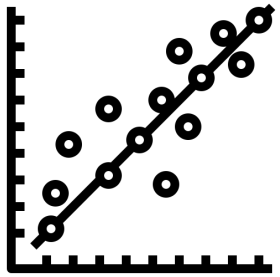
HD INSTITUTO HUMAN DELTA MR



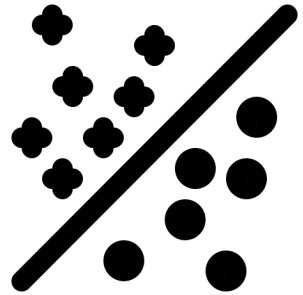
Introducción teórica al Deep Learning



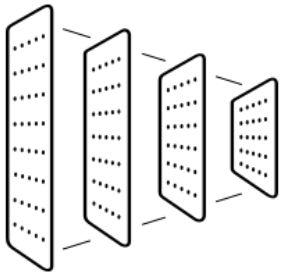
Tensores



Regresión

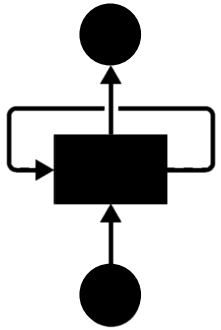


Clasificación



Redes convolucionales

Transfer learning



Redes recurrentes

Herramientas a utilizar

TensorFlow

SHAP

scikit learn

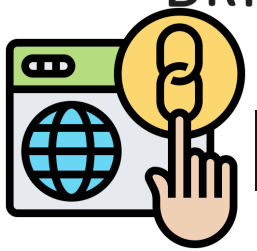
gradio

Weights & Biases

GitHub

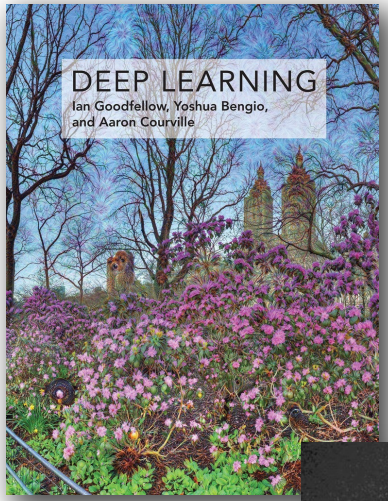
kaggle

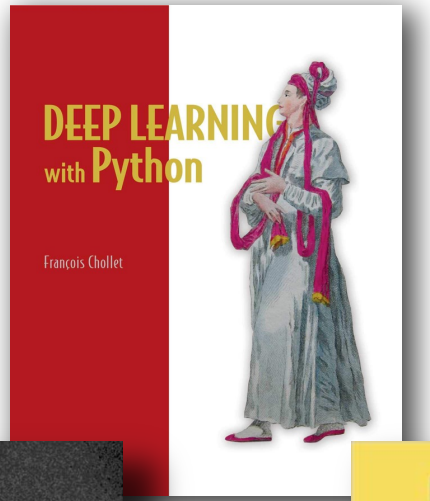
BRILLIANT

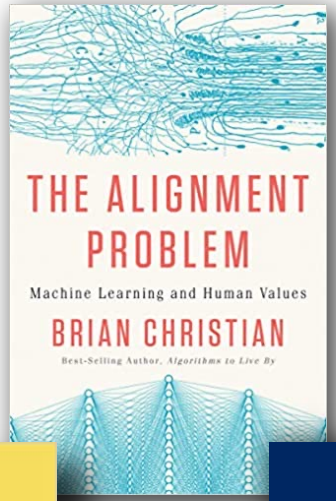


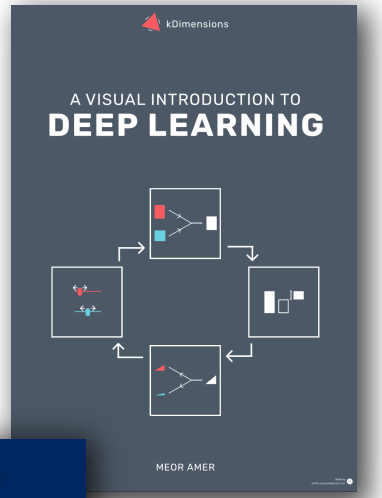
bnomial

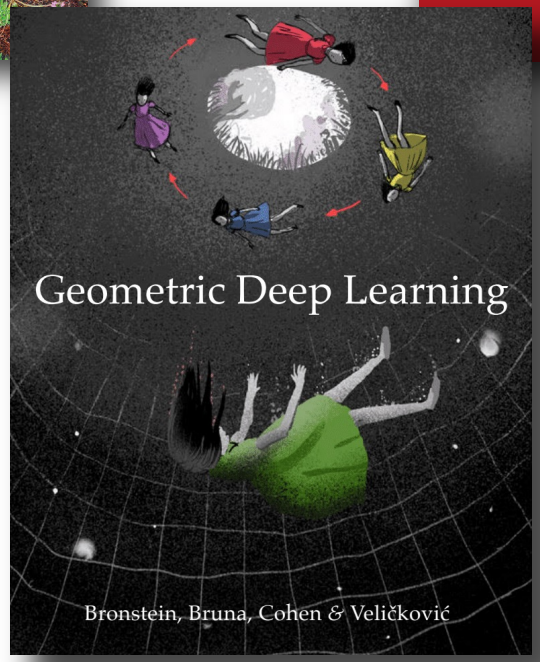
Libros a utilizar

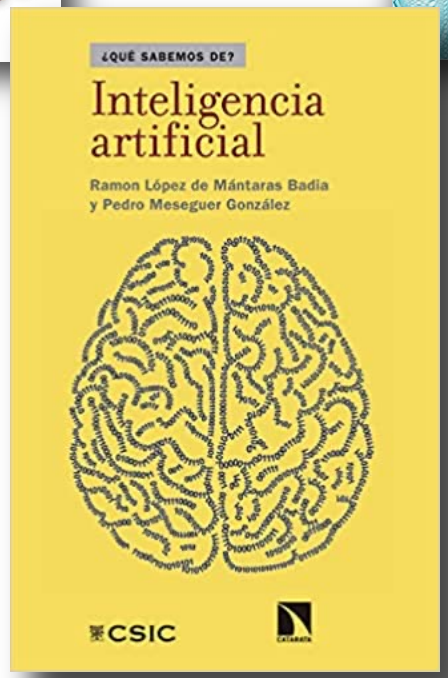


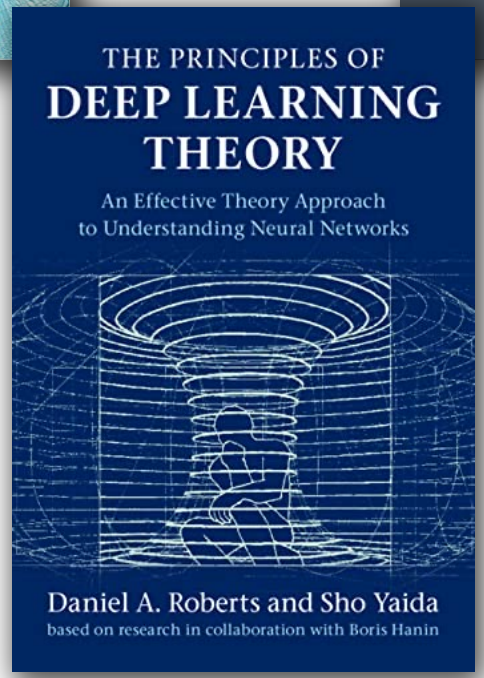












Historia y Teoría

Código

Matemáticas

Estado del Arte

Prerrequisitos

Python

- Nociones de Programación Orientada a objetos
- Nociones de Programación funcional
- Conocimiento de principales librerías

Matemáticas

- Nociones de Álgebra Lineal y Probabilidad y estadística

Especialidad en Machine Learning y Deep Learning

Python Primer

Mathematics Primer

Primer Cuatrimestre

- Fundamentos de Inteligencia Artificial
- Matemáticas para Machine Learning
- Aprendizaje Supervisado

Segundo Cuatrimestre

- Bases de Datos y Análisis Exploratorio
- Statistical Learning
- Deep Learning

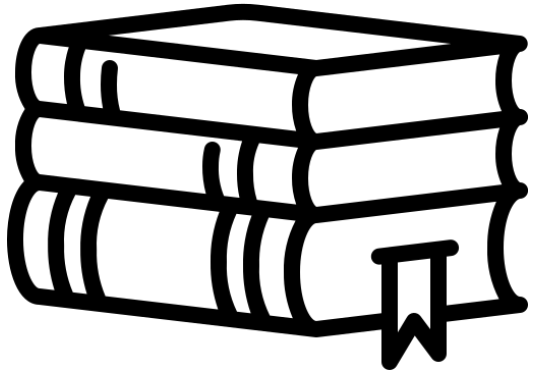
Tercer Cuatrimestre

- Aprendizaje Semisupervisado y no supervisado
- Procesamiento de Lenguaje Natural
- Computer Vision

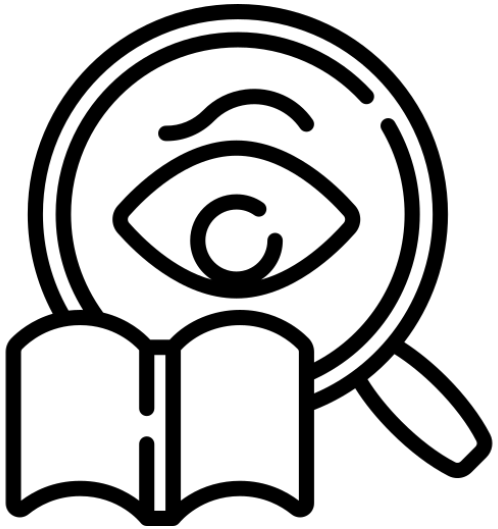
Cuarto Cuatrimestre

- Reinforcement Learning
- Deep Learning Avanzado
- MLOps
- Seminario

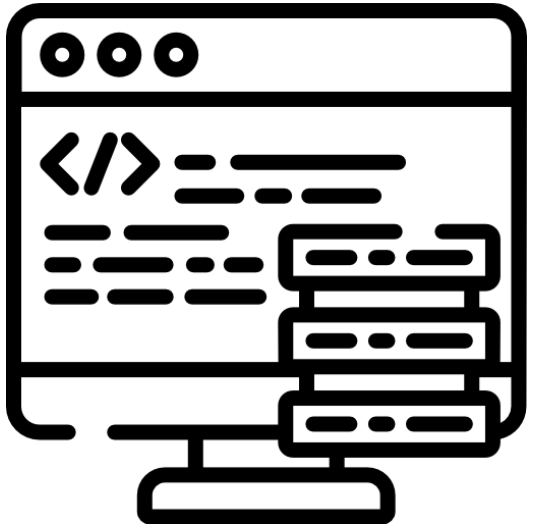
Filosofía de la clase



Leer

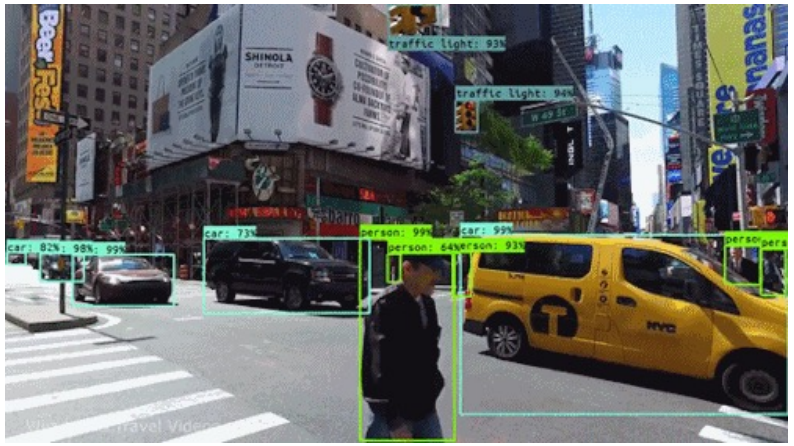


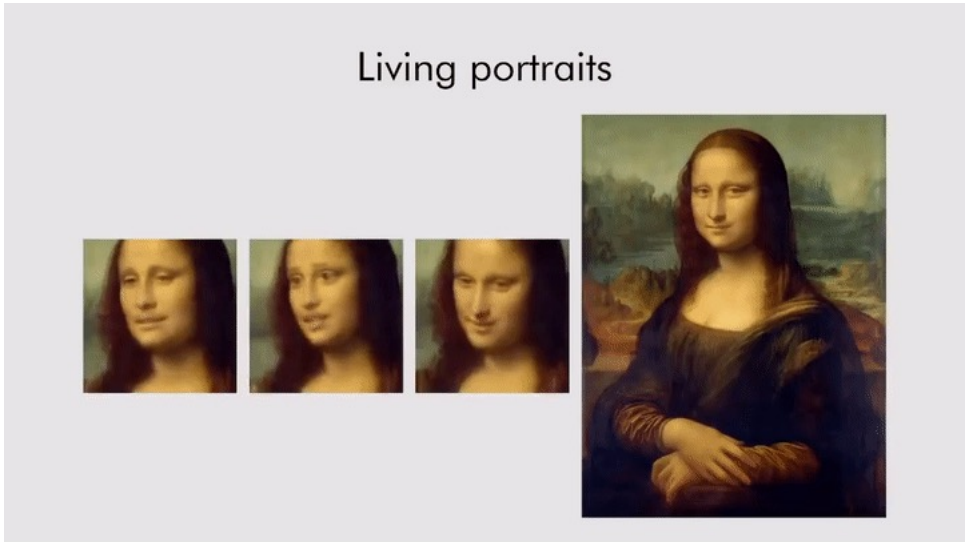
Ser crítico

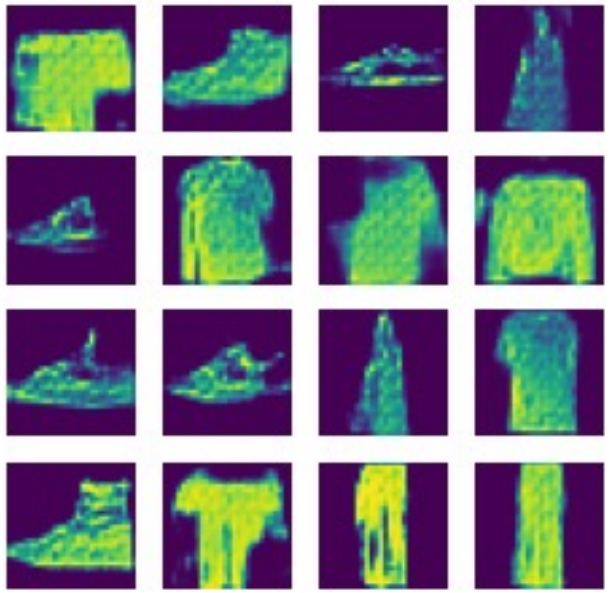


Crear

Automatizar vs aprender: apliaciones reales







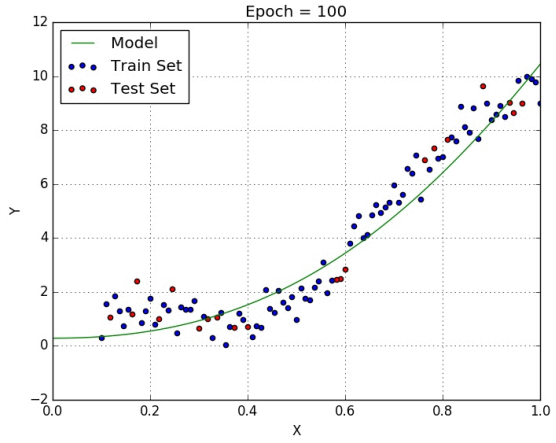
Input Prompt: `Recite the first law of robotics`

GPT-3

Output:

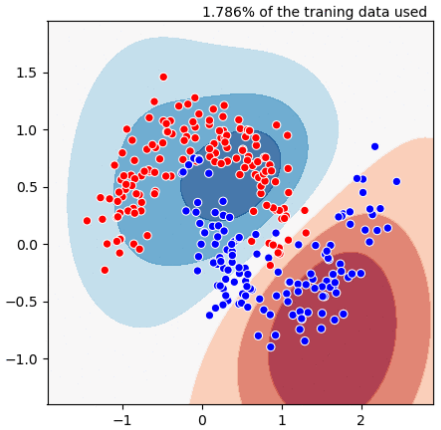
¿Qué problemas resuelve el Deep Learning?

Regresión



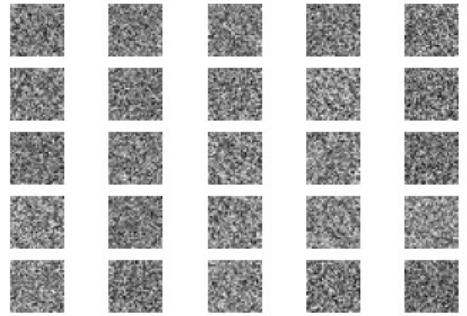
- Precio de casas
- Predicción de tarifas

Clasificación



- Clasificación de Spams
- Clasificación de imágenes
- Detección de anomalías en series de tiempo

Generación



- Generación de texto
- Color en fotos en B/N
- Aplicaciones de estilos de pinturas en fotos
- Restauración

TensorFlow: Código a alto nivel

TensorFlow:
Large-Scale Machine Learning on Heterogeneous Distributed Systems
(Preliminary White Paper, November 9, 2015)

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng
Google Research*

Abstract

TensorFlow [1] is an interface for expressing machine learning algorithms, and an implementation for executing such algorithms. A computation expressed using TensorFlow can be executed with little or no change on a wide variety of heterogeneous systems, ranging from mobile devices such as phones and tablets up to large-scale distributed systems of hundreds of machines and thousands of computational devices such as GPU cards. The system is flexible and can be used to express a wide variety of algorithms, including training and inference algorithms for deep neural network models, and it has been used for conducting research and for deploying machine learning systems into production across more than a dozen areas of computer science and other fields, including speech recognition, computer vision, robotics, information retrieval, natural language processing, geographic information extraction, and computational drug discovery. This paper describes the TensorFlow interface and an implementation of that interface that we have built at Google. The TensorFlow API and a reference implementation were released as an open-source package under the Apache 2.0 license in November, 2015 and are available at www.tensorflow.org.

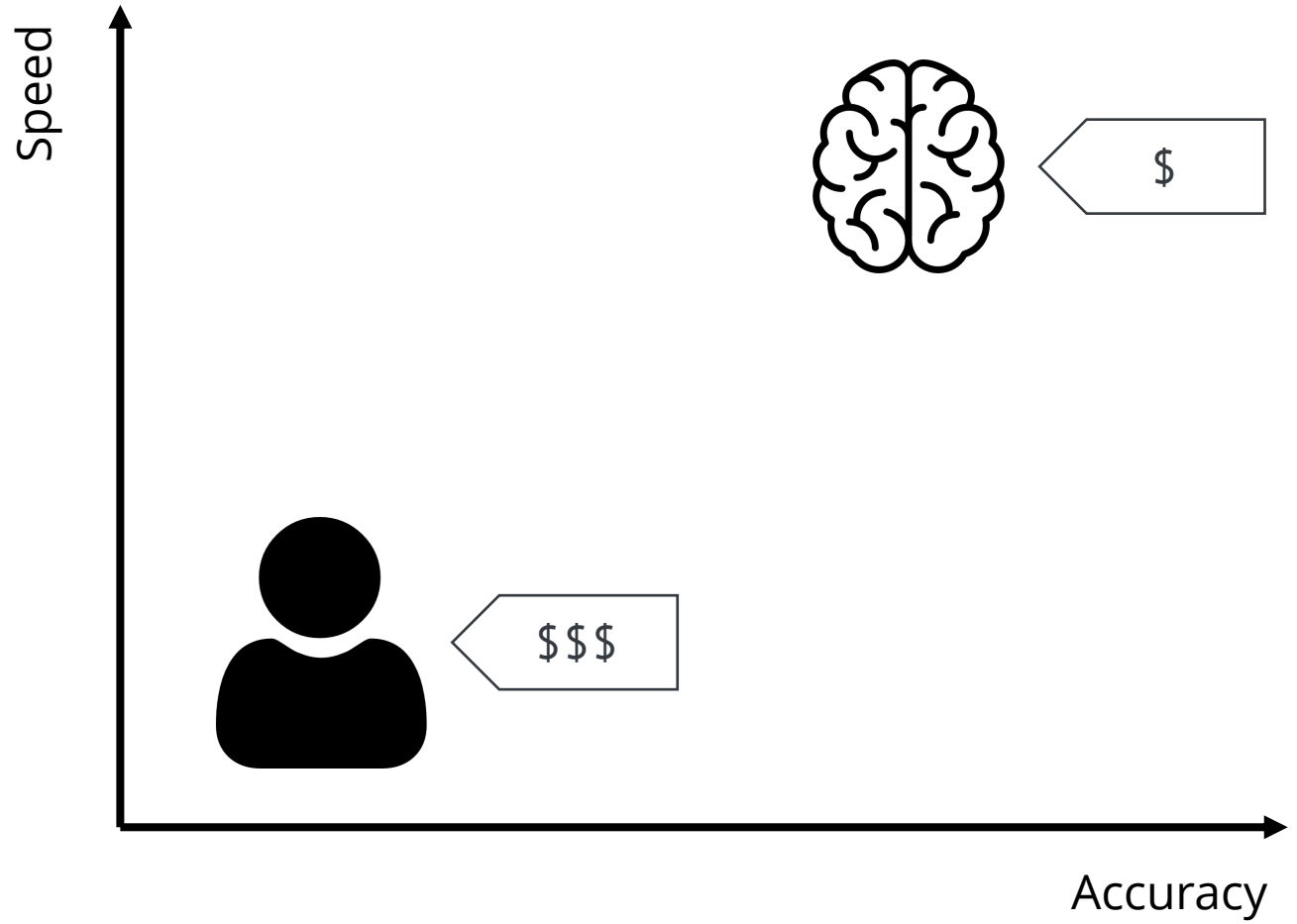
1 Introduction

The Google Brain project started in 2011 to explore the use of very-large-scale deep neural networks, both for research and for use in Google's products. As part of the early work in this project, we built DistBelief, our first-generation scalable distributed training and inference system [14], and this system has served us well. We and others at Google have performed a wide variety of research using DistBelief including work on unsupervised learning [31], language representation [35, 52], models for image classification and object detection [16, 48], video classification [27], speech recognition [56, 21, 20], sequence prediction [47], move selection for Go [34], pedestrian detection [2], reinforcement learning [38], and other areas [17, 5]. In addition, often in close collaboration with the Google Brain team, more than 50 teams at Google and other Alphabet companies have deployed deep neural networks using DistBelief in a wide variety of products, including Google Search [11], our advertising products, our speech recognition systems [50, 6, 46], Google Photos [43], Google Maps and StreetView [19], Google Translate [18], YouTube, and many others.

Based on our experience with DistBelief and a more complete understanding of the desirable system properties and requirements for training and using neural networks, we have built TensorFlow, our second-generation system for the implementation and deployment of large-scale machine learning models. TensorFlow takes computations described using a dataflow-like model and maps them onto a wide variety of different hardware platforms, ranging from running inference on mobile device platforms such as Android and iOS to modest-sized training and inference systems using single machines containing one or many GPU cards to large-scale training systems running on hundreds of specialized machines with thousands of GPUs. Having a single system that can span such a broad range of platforms significantly simplifies the real-world use of machine learning system, as we have found that having separate systems for large-scale training and small-scale deployment leads to significant maintenance burdens and leaky abstractions. TensorFlow computations are expressed as stateful dataflow graphs (described in more detail in Section 2), and we have focused on making the system both flexible enough for quickly experimenting with new models for research purposes and sufficiently high performance and robust for production training and deployment of machine learning models. For scaling neural network training to larger deployments, TensorFlow allows clients to easily express various kinds of parallelism through replication and parallel execution of a core model dataflow

*Corresponding authors: Jeffrey Dean and Rajat Monga: {jeff,rajatmonga}@google.com

Predicción



*Tomado de Meor Amer (2022)

Subcampos

COMPUTER VISION

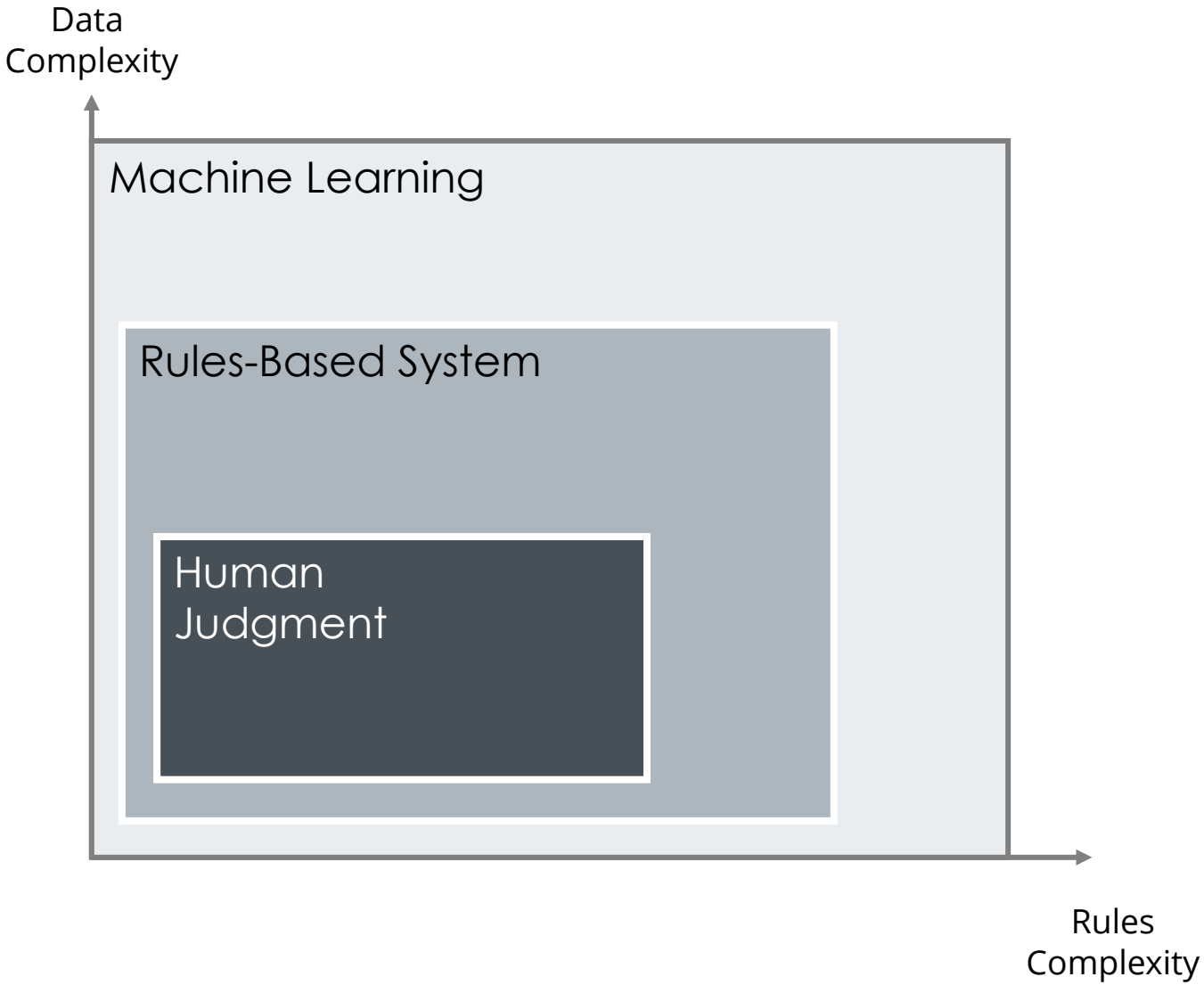
- Classificación de imágenes
- Detección y localización de objetos
- Reconocimiento facial
- Segmentación de imágenes
- Generación de imágenes
- Transformación de imágenes

NATURAL LANGUAGE PROCESSING

- Traducción
- Generación de texto
- Análisis de sentimientos
- Transcripción de texto

SECUENCIAL

- Series de Tiempo
- Video
- Audio



*Tomado de Meor Amer (2022)

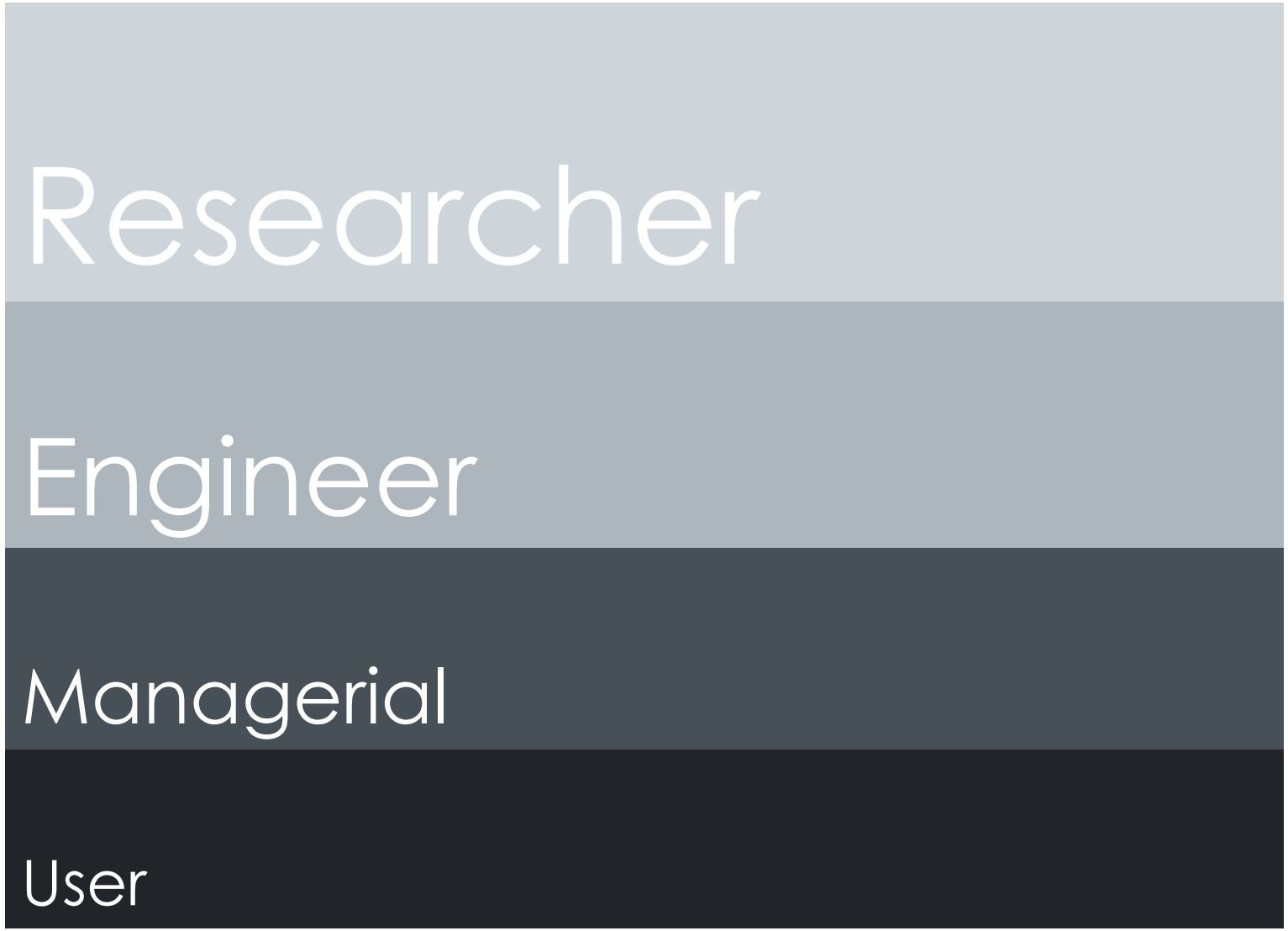
Papers With Code

Paradoja de Moravec

"It is comparatively easy to make computers ***exhibit adult level performance on intelligence tests or playing checkers****, and difficult or* ***impossible*** *to give them the skills of a* ***one-year-old*** *when it comes to perception and mobility."*

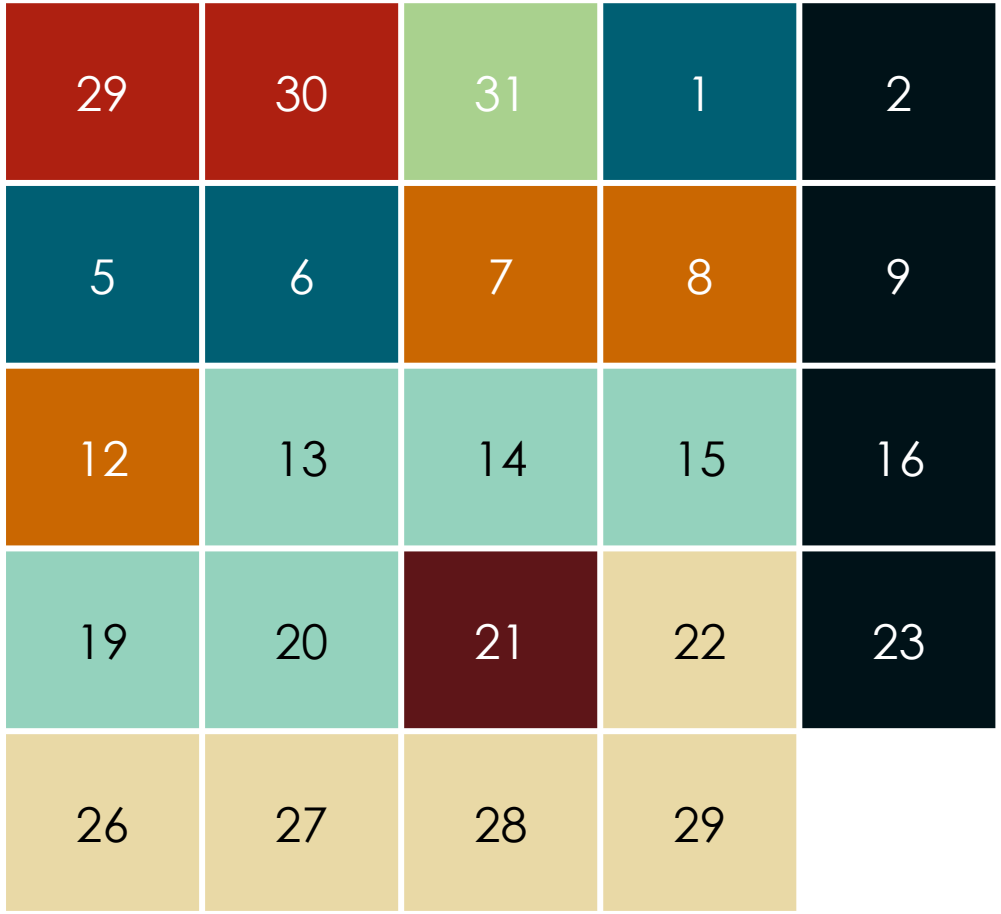
- Hans Moravec

Niveles



40

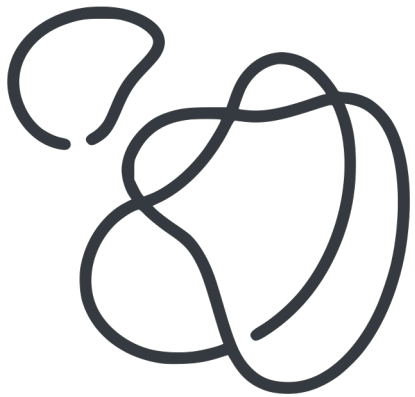
Horas de clase



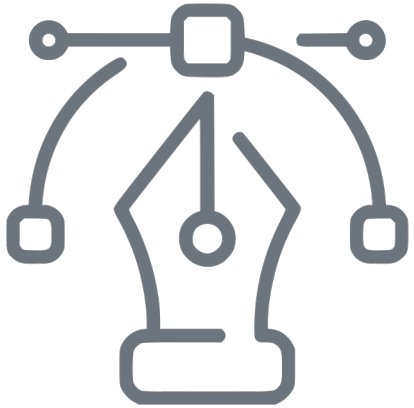
Introducción teórica
Tensores
Regresión
Clasificación
Convolucionales
Transfer Learning
Recurrentes

HD INSTITUTO HUMAN DELTA

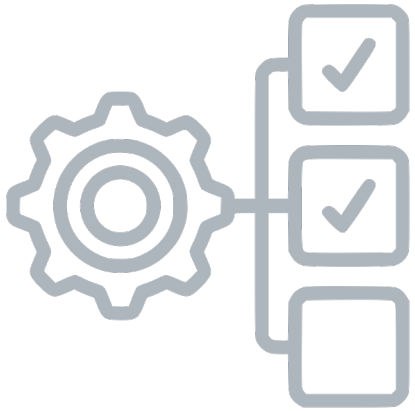
Resumen



El campo del Deep Learning tiene una conexión más amplia con lo abstracto y las tareas más complejas



Existen diferentes herramientas para aprender DL, la cuestión es utilizar la mejor combinación posible de ellas



No todas las tareas del DL se resuelven de la misma forma

Referencias

- Foto de portada: Oakley Smith
- Meor Amer, (2022), **“A visual Introduction to Deep Learning”**
- Roy Keyes, (2022), **“Deep Learning”**

GRACIAS

emilio.sandpal@gmail.com

milioe