

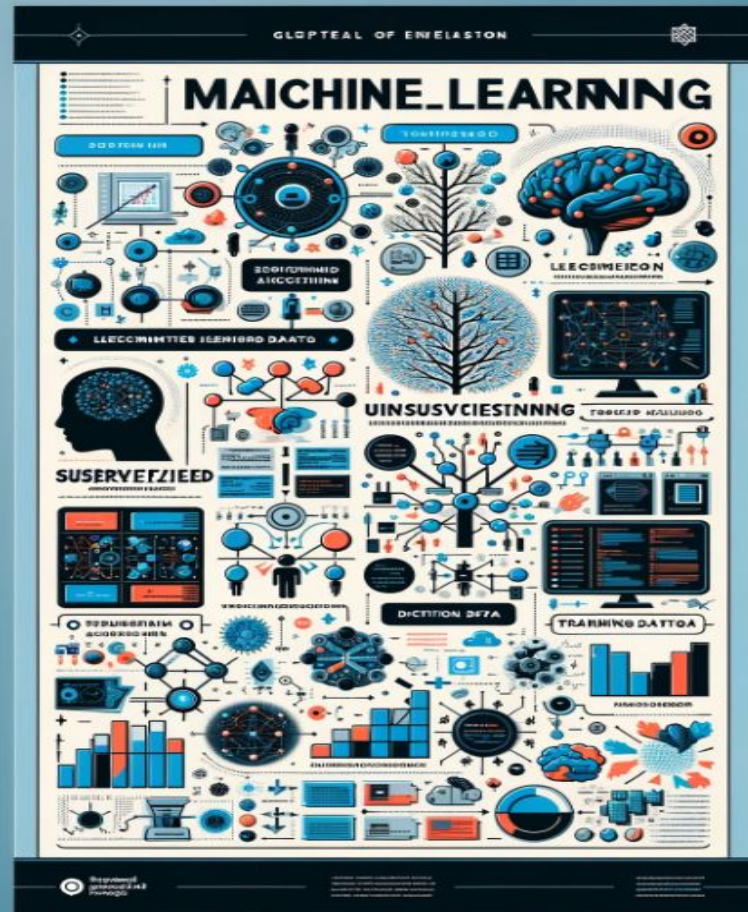
PPGEEC2318

Machine Learning

Fundamentals and first steps

Ivanovitch Silva

ivanovitch.silva@ufrn.br



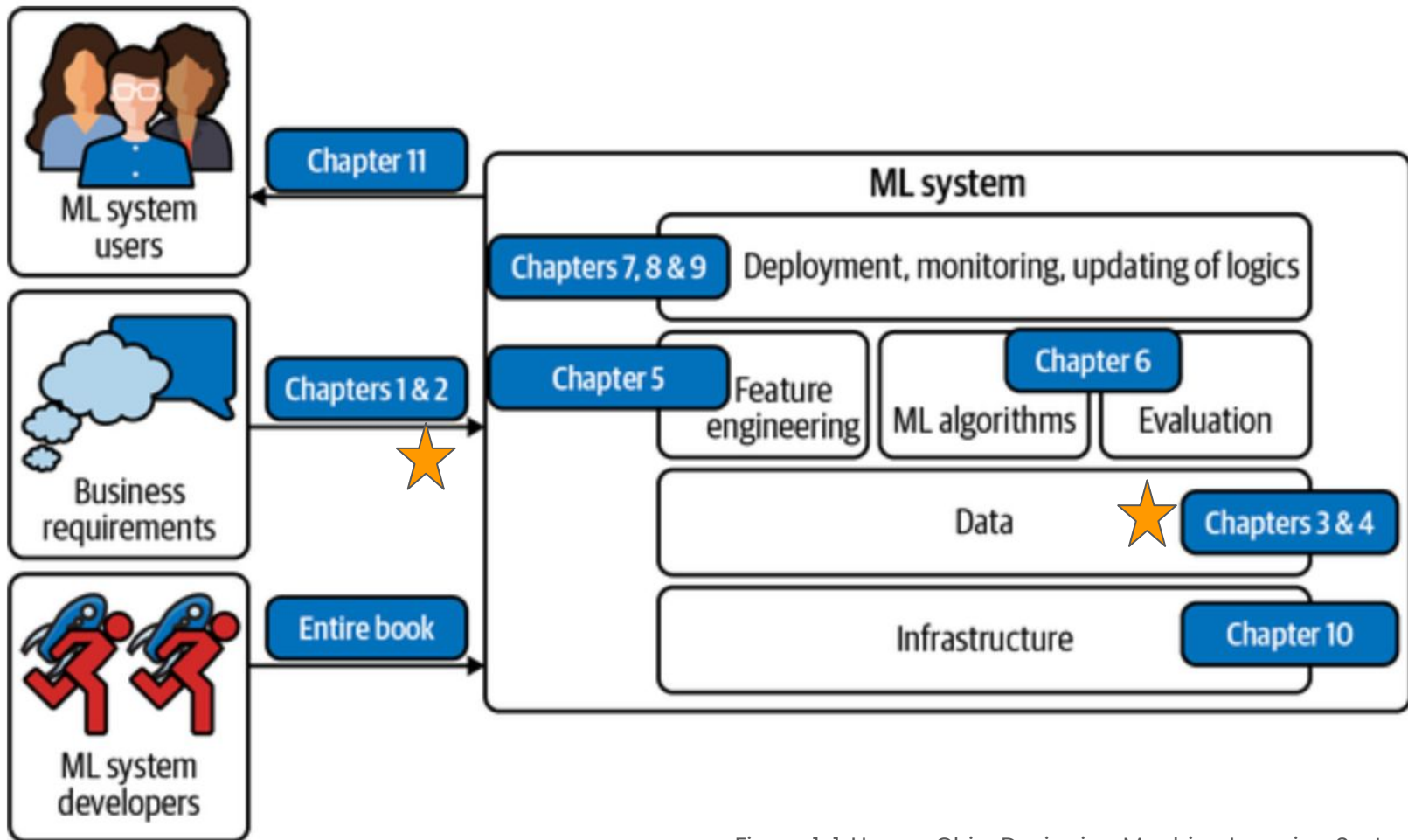


Figure 1-1, Huyen, Chip. Designing Machine Learning Systems

What is the primary role of MLOps in the context of machine learning systems?

- A. Developing new ML algorithms
- B. Ensuring ethical use of data
- C. Bringing ML models into production
- D. Conducting academic research

In the context of Machine Learning Systems, which of the following is not a general requirement for a good system?

- A. Reliability
- B. Scalability
- C. Complexity
- D. Maintainability

According to the chapters, what is a key difference between Machine Learning in research and production regarding computational priority?

- A. Research prioritizes fast inference, production prioritizes fast training
- B. Both prioritize fast training and high throughput
- C. Research prioritizes fast training, production prioritizes fast inference
- D. Both prioritize fast inference and low latency

Which of the following best describes the role of data in the success of Machine Learning systems as mentioned in the chapters?

- A. Data quality is more important than the quantity
- B. The success of ML systems relies heavily on access to a large amount of data
- C. Data is less important than the choice of algorithm
- D. Historical data is generally irrelevant in modern ML systems

Machine Learning Design Questions

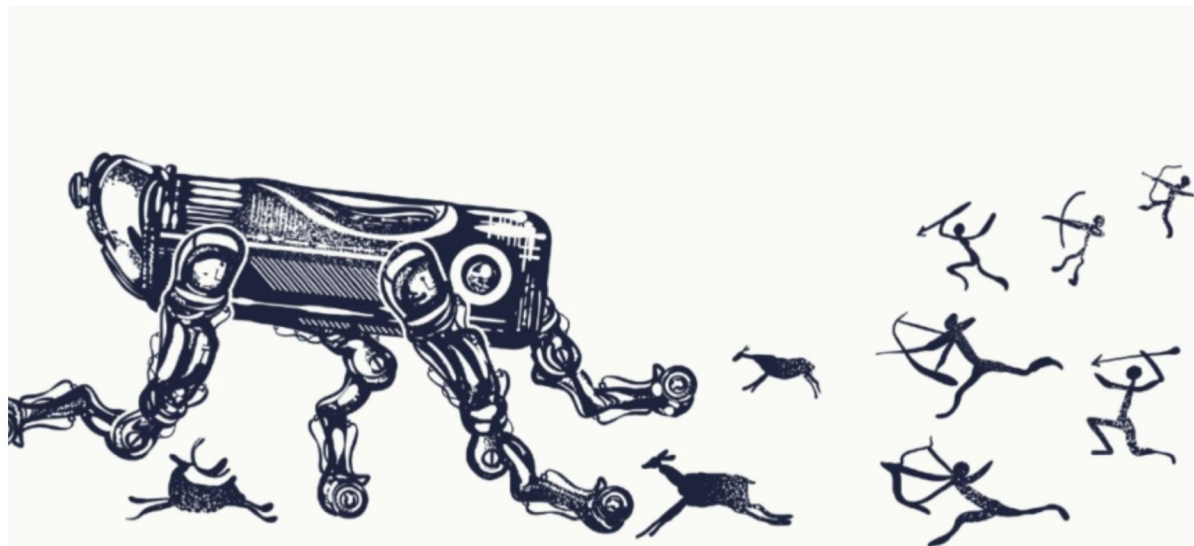
1. Design a machine learning platform
2. Design Facebook photo tagging
3. Design Amazon Alexa
4. Design a fake news detector
5. Design Youtube's recommendation system

24

Previously on IA world ...

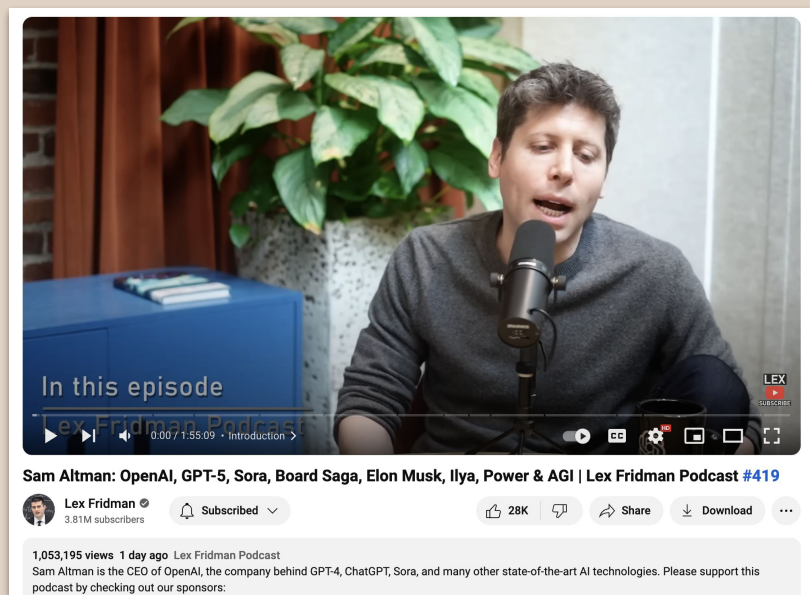
Silvio Meira: “Estamos na era da pedra lascada da IA, mas o futuro chega em 800 dias”

16 de março de 2024



"As empresas, de todos os portes e todos os mercados, que vão ter mais sucesso com inteligência artificial são aquelas que vão entender que inteligências artificiais não vieram para substituir pessoas, mas para trabalhar junto com pessoas e grupos de pessoas em prol de modelos de negócios de resolução de problemas.

Se eu posso fazer com que cada pessoa trabalhe por dez, eu tenho que ir atrás de 10 vezes mais mercado – e não demitir. Se eu demitir as pessoas do call center, tenho que contratar TI para operar a inteligência artificial que vai conversar com os clientes."



When asked about GPT-4, Altman said that the leap in capabilities for GPT-5 will be similar to GPT-3's jump to GPT-4.



Yann Lecun: Meta AI, Open Source, Limits of LLMs, AGI & the Future of AI | Lex Fridman Podcast #416



Lex Fridman ✓
3.81M subscribers



Subscribed ▼



10K



Share



Download



621K views 12 days ago Lex Fridman Podcast

Yann LeCun is the Chief AI Scientist at Meta, professor at NYU, Turing Award winner, and one of the most influential researchers in the history of AI.

Please support this podcast by checking out our sponsors:

On the Societal Impact of Open Foundation Models

Sayash Kapoor^{*1} Rishi Bommasani^{*2}

Kevin Klyman² Shayne Longpre³ Ashwin Ramaswami⁴ Peter Cihon⁵ Aspen Hopkins³
Kevin Bankston^{6,4} Stella Biderman⁷ Miranda Bogen^{6,1} Rumman Chowdhury⁸ Alex Engler⁹
Peter Henderson¹ Yacine Jernite¹⁰ Seth Lazar¹¹ Stefano Maffulli¹² Alondra Nelson¹³
Joelle Pineau¹⁴ Aviya Skowron⁷ Dawn Song¹⁵ Victor Storchan¹⁶ Daniel Zhang²
Daniel E. Ho² Percy Liang² Arvind Narayanan¹

February 27, 2024

Abstract

Foundation models are powerful technologies: how they are released publicly directly shapes their societal impact. In this position paper, we focus on *open* foundation models, defined here as those with broadly available model weights (e.g. Llama 2, Stable Diffusion XL). We identify five distinctive properties (e.g. greater customizability, poor monitoring) of open foundation models that lead to both their benefits and risks. Open foundation models present significant benefits, with some caveats, that span innovation, competition, the distribution of decision-making power, and transparency. To understand their risks of misuse, we design a risk assessment framework for analyzing their *marginal risk*. Across several misuse vectors (e.g. cyberattacks, bioweapons), we find that current research is insufficient to effectively

1. Introduction

Foundation models (Bommasani et al., 2021) are the centerpiece of the modern AI ecosystem, catalyzing a frenetic pace of technological development, deployment, and adoption that brings with it controversy, scrutiny, and public attention. *Open foundation models*¹ like BERT, CLIP, Whisper, BLOOM, Pythia, Llama 2, Falcon, Stable Diffusion, Mistral, OLMo, Aya, and Gemma play an important role in this ecosystem. These models allow greater customization and deeper inspection of how they operate, giving developers greater choice in selecting foundation models. However, they may also increase risk, especially given broader adoption, which has prompted pushback, especially around risks relating to biosecurity, cybersecurity, and disinformation. How to release foundation models is a central debate today, often described as open vs. closed.

Simultaneously, policymakers are confronting how to gov-



facebookresearch / faiss



facebookresearch / segment-anything



facebookresearch / jepa

Nature of Human Intelligence versus Artificial Intelligence



Moravec's paradox

18 languages

Contents

(Top)

[The biological basis of human skills](#)

[Historical influence on artificial intelligence](#)

[Reception](#)

[See also](#)

Article [Talk](#)

[Read](#) [Edit](#) [View history](#) [Tools](#)

From Wikipedia, the free encyclopedia

Moravec's paradox is the observation in [artificial intelligence](#) and [robotics](#) that, contrary to traditional assumptions, [reasoning](#) requires very little [computation](#), but [sensorimotor](#) and perception skills require enormous computational resources. The principle was articulated by [Hans Moravec](#), [Rodney Brooks](#), [Marvin Minsky](#) and others in the 1980s. Moravec wrote in 1988, "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".^[1]

Moravec's Paradox questions why computers can perform complex tasks like playing chess or solving integrals, but struggle with tasks that humans take for granted, like driving a car or clearing the dinner table.

Limitations of Large Language Models in Achieving Superhuman Intelligence

- **Understanding of the Physical World:**
 - Lack the innate ability to comprehend real-world physics and dynamics.
- **Persistent Memory:**
 - Difficulty in maintaining and utilizing long-term memory effectively.
- **Reasoning Capabilities:**
 - Struggle with logical deduction and complex problem-solving.
- **Planning Skills:**
 - Limited proficiency in strategizing and anticipating future consequences.

THE NEW YORK TIMES BESTSELLER

THINKING, FAST AND SLOW



DANIEL
KAHNEMAN

WINNER OF THE NOBEL PRIZE IN ECONOMICS

"[A] masterpiece . . . This is one of the greatest and most engaging collections of insights into the human mind I have read." —WILLIAM EASTERLY, *Financial Times*

System 1 (automatic reasoning)

Versus

System 2 (reflective reasoning)

THE NEW YORK TIMES BESTSELLER

THINKING, FAST AND SLOW



DANIEL
KAHNEMAN

WINNER OF THE NOBEL PRIZE IN ECONOMICS

"[A] masterpiece . . . This is one of the greatest and most engaging collections of insights into the human mind I have read." —WILLIAM EASTERLY, *Financial Times*

System 1 (Automatic Reasoning):

- **Automatic:** Operates automatically and quickly, with little or no effort and no sense of voluntary control.
- **Based on Impressions:** Uses quick associations and emotions to reach conclusions. It is responsible for intuitive and rapid judgments.
- **Habits and Implicit Learning:** Often based on habits and knowledge acquired through experience, without the person being consciously aware of how they know something.
- **Parallel Processing:** Capable of processing multiple pieces of information simultaneously, which is useful for tasks such as pattern recognition and familiar situations.

THE NEW YORK TIMES BESTSELLER

THINKING, FAST AND SLOW



DANIEL
KAHNEMAN

WINNER OF THE NOBEL PRIZE IN ECONOMICS

"[A] masterpiece . . . This is one of the greatest and most engaging collections of insights into the human mind I have read." —WILLIAM EASTERLY, *Financial Times*

System 2 (Reflective Reasoning):

- **Deliberate and Slow:** Requires effort and is used in complex mental operations, such as complicated mathematical calculations, critical assessments, and considered decisions.
- **Logical and Sequential:** Functions in a more logical and sequential manner, dealing with abstractions, analysis, and planning.
- **Conscious and Controllable:** The operations of System 2 are usually conscious, and the person feels as if they are doing "mental work" when employing it.
- **Limited Capacity:** Has limited capacity, quickly becoming overloaded, and can be slow and laborious.

And about Large Language Models (LLM)?

- Machine Learning sucks!! Compared to humans and animals.
- Most current ML-based AI systems
 - Make stupid mistakes, do not reason nor plan
- Animals and humans
 - Can learn new tasks very quickly
 - Understand how the world works
 - Can reason and plan
- Humans and animals have common sense
- Current machines, not so much (it is very superficial)

And about Large Language Models (LLM)?

- You want systems that can reason, and certainty that can plan!!!
- The type of reasoning that takes into account in LLM is very, very primitive.
- Because the amount of computation that is spent per token produced is constant.
 - If you ask a question and that question has an answer in a given number of token, the amount of computation devoted to computing that answer can be exactly estimated.
 - Essentially, it doesn't matter if the question being asked is simple to answer, complicated to answer, impossible to answer. The amount of computation the system will be able to devote to answer is constant (proportional to the number of tokens in the answer)
 - This is not the way we work, the way we reason is that when we are faced with a complex problem, we spend more time trying to solve it and answer it, right?

The blueprint of the future data systems

- They will think about their answer
- Plan their answer by optimization before turning it into text.
- You have an abstract representation inside the system.
- You have a prompt. The prompt goes through an encoder, produces a representation (or predict ones). But that representation may not be a good answer because there might be some complicated reasoning. So then you have another process that takes the representation of the answers and modifies it so as to minimize a cost function that measures to what extent the answer is a good answer for the question.

Language is low bandwidth

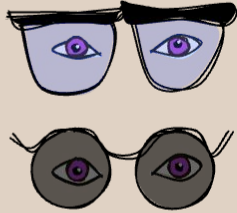


A person can read 270 words/minutes
or 4.5 words/second, which is 12 bytes/s
(assuming 2 bytes per token and
0.75 words per token)



A modern LLM is typically trained
with 1×10^{13} two-byte tokens,
which is 2×10^{13} bytes.
This would take about
100,000 years for a person
to read (at 12 hours a day).

Vision is much higher bandwidth: about 20MB/s



Each of the two optical nerves
has 1 million nerve fibers,
each carrying about
10 bytes per second

A 4 year-old child has been
awake a total 16,000 hours,
which translates into 1×10^{15} bytes.

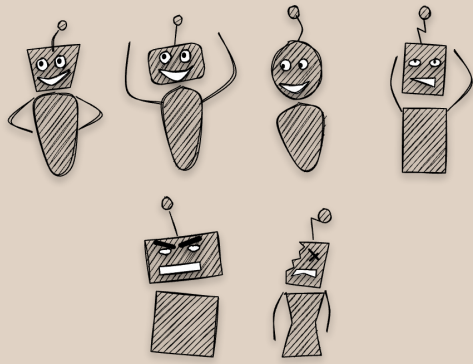
In a mere 4 years, a child has
seen 50 times more data than
the biggest LLMs trained on all
the text publicly available on the internet.



bandwidth: 20MB/s

bandwidth: 12 bytes/s

The data bandwidth of visual perception is roughly 1.6 million times higher than the data bandwidth of written (or spoken) language.



Most of human knowledge (and almost all of animal knowledge) comes from our sensory experience of the physical world.

Language is the icing on the cake.
We need the cake to support the icing.

There is absolutely no way in hell we will ever reach human-level AI without getting machines to learn from high-bandwidth sensory inputs, such as vision.

Note: Yes, humans can get smart without vision, even pretty smart without vision and audition. But not without touch. Touch is pretty high bandwidth, too.

Fundamentals of Machine Learning

Fundamental
Concepts in
ML

Data Preparation:
cleaning, feature
selection, data
transforms

Fundamental
Concepts in
Statistics

Linear
Regression

Gradient
Descent

Logistic
Regression

Naive
Bayes

Assessing
Model
Performance

Preventing
Overfitting with
Regularization

Unbalancing
Data
Methods

Support
Vector
Machines

Decision
Tree

Random
Forest

Boosting

Ensemble

Neural
Networks

Dimensionality
Reduction


Clustering

A classical ML course

*Undergrad

Weeks 02, 03, 04 Machine Learning Fundamentals and Decision Trees [Jupyter PDF](#)

- Outline [Video](#)
- What is Machine Learning (ML)? [Video](#)
- ML types [Video](#)
- Main challenges of ML
 - Variables, pipeline, and controlling chaos [Video](#)
 - Train, dev and test sets [Video](#)
 - Bias vs Variance [Video](#)
- Decision Trees
 - Introduction [Video](#)
 - Mathematical foundations [Video](#)
- Evaluation metrics
 - How to choose an evaluation metric? [Video](#)
 - Threshold metrics [Video](#)
 - Ranking metrics [Video](#)
- Case Study [Notebook](#)
 - Google Colaboratory [Video](#) [Video](#)
 - Setup of the environment [Video](#)
 - Extract, Transform and Load (ETL)
 - Exploratory Data Analysis (EDA) [Video](#)
 - Fetch Data [Video](#)
 - EDA using Pandas-Profiling [Video](#)
 - Manual EDA [Video](#)
 - Preprocessing [Video](#)
 - Data Check [Video](#)
 - Data Segregation [Video](#)
 - Train
 - Train and validation component [Video](#)
 - Data preparation and outlier removal [Video](#)
 - Encoding the target variable [Video](#)
 - Encoding the independent variables manually [Video](#)
 - Using a full-pipeline to prepare categorical features [Video](#)
 - Using a full-pipeline to prepare numerical features [Video](#)
 - Creating a full-preprocessing pipeline [Video](#)
 - Holdout training [Video](#)
 - Evaluation metrics [Video](#)
 - Hyperparameter tuning using Wandb [Video](#)
 - Configure, train and export the best model [Video](#)
 - Test [Video](#)



ivanovitchm / ppgeecmachinelearning

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

ppgeecmachinelearning Public


main 1 Branch 0 Tags

Go to file Add file Code

ivanovitchm add week 14 ec32e11 · 2 years ago 82 Commits

images	added a model card for week 02	2 years ago
lessons	add week 14	2 years ago
.gitignore	configure .gitignore	2 years ago
README.md	add week 14	2 years ago

README



About

Repository for EEC1509, a graduate course on PPgEEC about Machine Learning

Readme Activity 34 stars 7 watching 17 forks

Releases

No releases published [Create a new release](#)

Packages

No packages published [Publish your first package](#)

Languages

Jupyter Notebook 99.1% TeX 0.9%

Machine Learning Types

Supervised Learning

Supervised learning requires large numbers of labeled samples.

Unsupervised Learning

Unsupervised learning requires large numbers of labeled or unlabeled samples.

Semi-Supervised Learning

Semi-Supervised learning requires small numbers of labeled samples and large numbers of unlabeled samples.

Reinforcement Learning

Reinforcement learning requires insane amounts of trials.

Self-Supervised Learning

Self-Supervised learning requires large numbers of unlabeled samples.

Active Learning

Active learning requires small numbers of labeled samples and large numbers of unlabeled samples.

Weak Supervision

Weak supervision requires small numbers of labeled samples and large numbers of unlabeled samples. Leverage heuristics to generate labels.

Semi-Supervised Learning

Context: You're developing an email classification system to label emails as "spam" or "non-spam." You have a large number of emails, but only a small portion is labeled.

1. **Initial Training:** First, you train your model on the available labeled emails.
2. **Applying to Unlabeled Data:** Next, you use the trained model to make predictions on the unlabeled emails.
3. **Model Refinement:** Based on the predictions, you might assume some of these classifications is correct (especially those where the model is most confident) and use these new "pseudo-labels" to retrain the model.

Key Point: The model attempts to learn from the entirety of available data (both labeled and unlabeled), and the process is largely automated.

Active Learning

Context: You're developing an email classification system to label emails as "spam" or "non-spam." You have a large number of emails, but only a small portion is labeled.

1. **Initial Training:** You start by training your model with a small set of labeled data.
2. **Active Selection of Examples:** The model identifies and selects the unlabeled emails about which it's most uncertain.
3. **Human Intervention:** You, or another expert, manually label these selected emails.
4. **Model Update:** The model is re-trained with the newly labeled data.

Key Point: The model actively seeks human intervention to obtain the most informative labels, maximizing learning efficiency with minimal labeled data.

Weak Supervision

Context: You're developing an email classification system to label emails as "spam" or "non-spam." You have a large number of emails, but only a small portion is labeled.

1. **Generating Weak Labels:** Use various heuristics, external knowledge sources, or simpler models to label the large dataset. For example, emails containing certain keywords like "offer" might be weakly labeled as "spam."
2. **Training with Weak Labels:** Train your model on this weakly labeled data. The labels aren't perfectly accurate, but they provide a broader context and learning opportunity.
3. **Refining the Model:** Optionally, use the model's predictions, in combination with the small set of accurately labeled data, for further refinement.

Key Point: The model relies on noisily or weakly labeled data, often generated through heuristics or auxiliary information, to provide a broad base for initial training.

Semi-Supervised versus Active Learning versus Weak Supervision

Each approach tackles the challenge of limited labeled data in a different way: **semi-supervised learning** leverages unlabeled data through model predictions, **active learning** focuses on human labeling of the most informative samples, and **weak supervision** uses readily available but less accurate labeling methods to quickly annotate large datasets.

Self-Supervised Learning

Context: You are developing a model to understand and process natural language, specifically to improve the performance of a chatbot.

1. **Data Preparation:** Assume you have a vast collection of text data (like articles, books, or internet comments) but none of it is labeled for any specific NLP task like sentiment analysis or topic classification.
2. **Creating a Pretext Task:** To train your model, you first create a '*pretext*' task. This task should be something that forces the model to understand the structure and semantics of language. A common example is the 'masked word prediction' task, where some words in a sentence are randomly masked (hidden), and the model's job is to predict these masked words based only on the context provided by the surrounding words. For instance, in the sentence "*I love eating ___; it's my favorite fruit,*" the model needs to predict the masked word (e.g., 'umbu').

Cont.

Self-Supervised Learning (continue)

Context: You are developing a model to understand and process natural language, specifically to improve the performance of a chatbot.

3. **Training on the Pretext Task:** You train your model on this task using your large dataset of text. The model learns valuable information about the language, like word associations, grammar, and common phrases, even though it isn't being trained on a specific downstream task like classification or translation yet.
4. **Applying Learned Representations:** After training, the model has learned rich language representations. These can now be fine-tuned or transferred to specific NLP tasks, such as sentiment analysis, named entity recognition, or language translation, using smaller sets of labeled data specific to these tasks.

Key Point: The model learns from data that hasn't been explicitly labeled for the task at hand (learning from unlabeled data). A task is created (predicting masked words) that is related to the skills needed for the eventual target tasks. The representations learned through the pretext task are broadly useful for a range of NLP tasks, not just the one it was trained on (transferable knowledge).