Open4Tech Summer School 2025

How to Train Your Personal AI Assistant

Razvan Tudosie
Software Developer, Syncro Soft

Alexandru Smarandache
Software Developer, Syncro Soft

Agenda

- What is AI?
- AI Core Components
- The AI Model Training Process
- Fine-Tuning
- Demo
- Conclusions

What is AI

- Artificial Intelligence (AI) is a broad collection of techniques and algorithms developed to solve specific problems.
- The history of AI is long and tumultuous, marked by periods of rapid advancement as well as times when progress stalled almost completely.

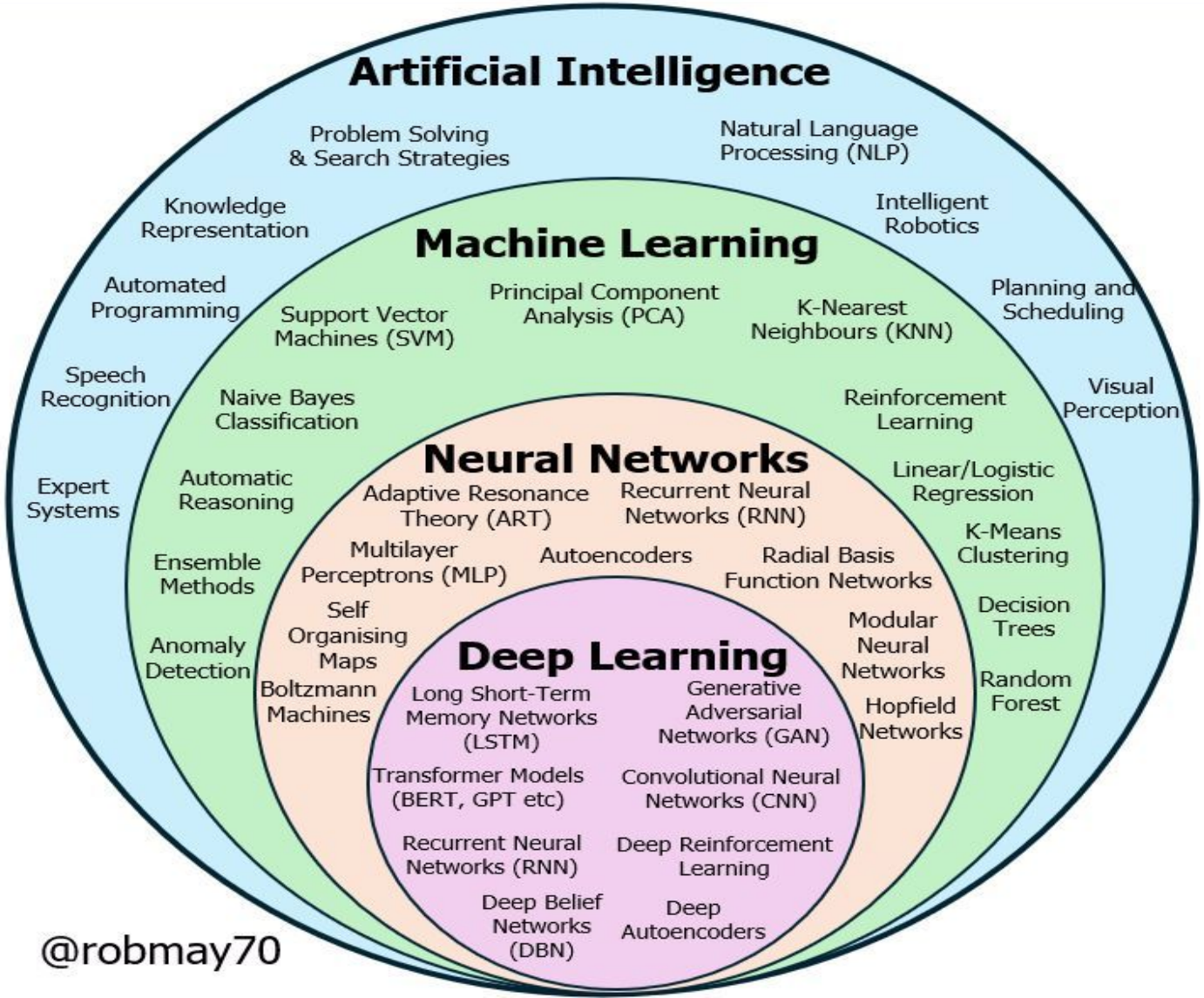
- Most important “AI” related events (credits digitalwellbeing.org):

What is AI(now)

- Now, AI is capable of performing a wide range of tasks—from simple ones like predicting weather patterns, to more complex activities such as image recognition and powering personal assistants. The field of AI is evolving rapidly, growing at a pace no one could have anticipated.

AI Core Components

- “AI” represented as an onion of different algorithms and techniques ->

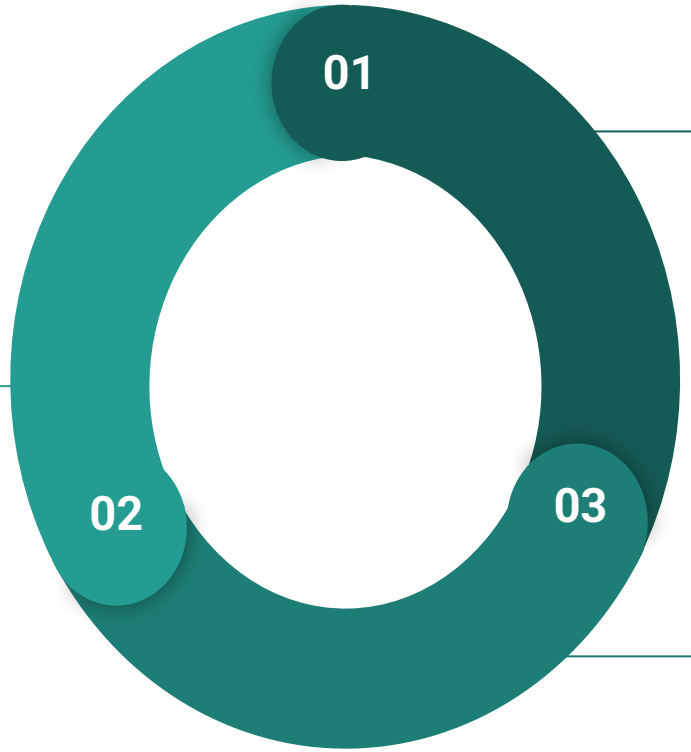


Goals

- The goal of this course is to familiarize participants with the AI field and guide them through the process of training a personal AI model to solve a specific task.
- We will also investigate some practical examples.

How to train Your Personal AI Assistant

- The process of “training” an AI model is usually done in 3 steps.



Data preprocessing

Collecting, processing and other data related operations.

Training

The training process that will generate the model.

Testing and validation

Testing the capabilities of the model for flaws.

Step I:
Data preprocessing

How to train Your Personal AI Assistant(I)

- The “Data preprocessing” step is the first stage in the machine learning workflow. It involves creating or collecting relevant data that the AI model will train on.

How to train Your Personal AI Assistant(I)

- During this phase, specific data processing tasks are essential, such as **cleaning the data**, **handling missing values**, **normalizing or standardizing features**, and sometimes **transforming data** into formats suitable for the model.
- Proper data preprocessing is **critical**, as the quality and structure of the data directly impact the performance and reliability of the AI model.
- This is the step where **most mistakes** are made!

How to train Your Personal AI Assistant(I)

When searching for data to train your AI model, two main concerns often arise: privacy and cost.

Many of the best sources of data are hidden behind paywalls, or are simply proprietary assets of other companies.

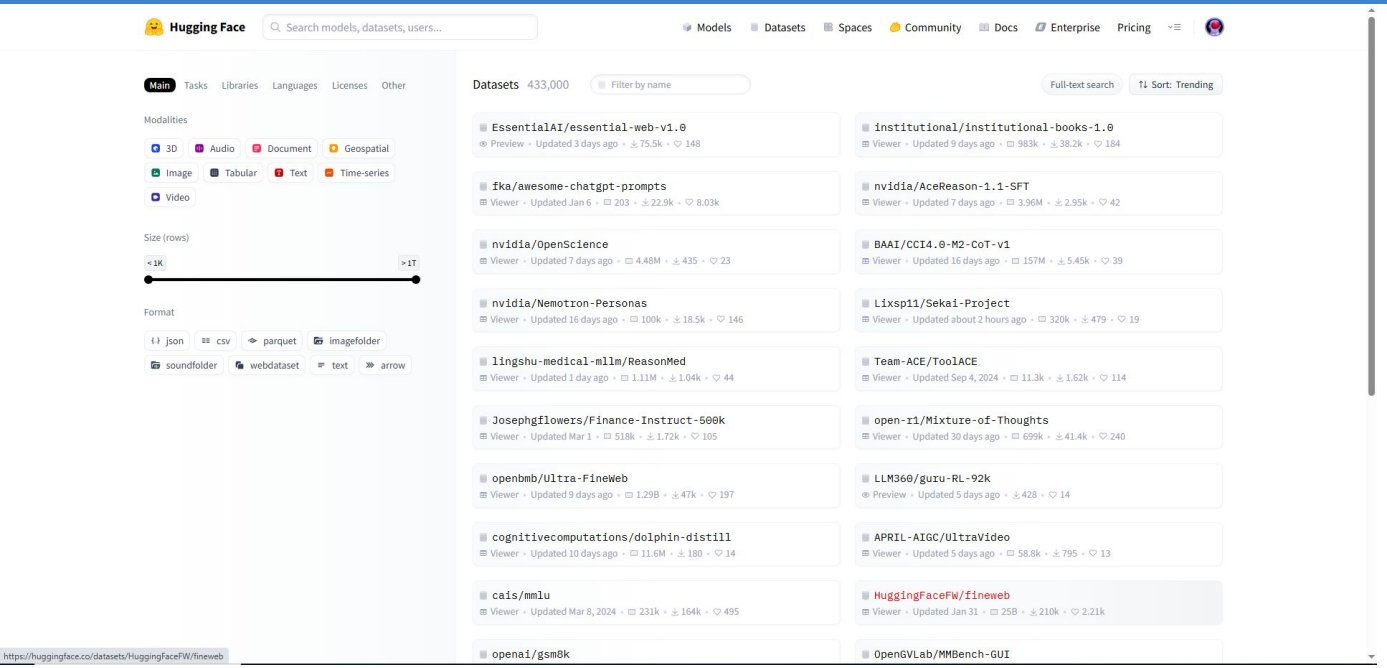
But… what about open source datasets?

How to train Your Personal AI Assistant(I)

- There are a number of open-source datasets hosted on sites like ***HuggingFace*** or ***Kaggle***.
- These datasets range from useless, to mildly interesting (such as a *"Zara sales analysis"* or *"English Premier League - Player Stats - 24/25"* :)), to even game-changing, depending on what we want our AI to accomplish.

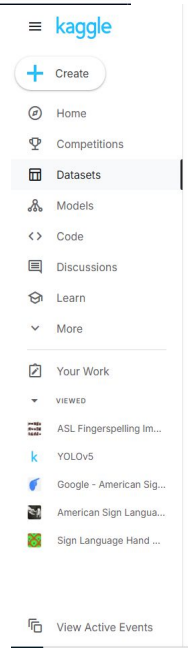
Hugging Face

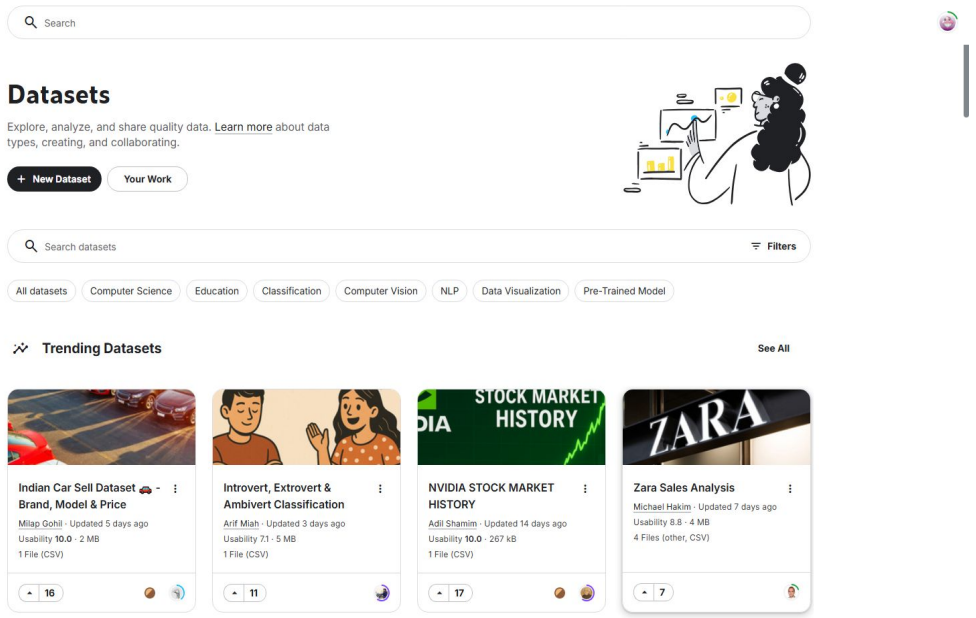
kaggle



- A typical search for a specific dataset involves looking for relevant keywords and/or intended uses.

- However, you might also need to look for specific scientific papers. In many cases, the data published in peer-reviewed journals is made openly available after the article is published.





How to train Your Personal AI Assistant(I)

- A common way to improve an AI model is to create a training dataset by merging two or more open-source datasets.

- By combining different sources, you can expose your model to a wider variety of data, increasing its robustness and ability to perform well even in unexpected edge cases *.

*There might be some drawbacks if the merging isn't done carefully, which we will discuss in the "Testing and Validation" phase.

For example, improper merging can lead to issues such as overfitting, where the model learns patterns that do not generalize well to new, unseen data.

How to train Your Personal AI Assistant(I)

← premier league

Notebooks 1,477 | Comments 839 | Datasets 743 ✕ | Topics 215 | Hugging Face Models 10 | People 5 | Models 4 | Competitions 3

Filter by

DATE

- Last 90 days 49
- This week 5
- Today 3

VIEWED BY YOU

- Viewed 1
- Not Viewed 742

CREATOR

- You 0
- Others 743

743 Results Relevance

Premier League 313
Dataset · 7y ago · by Zaeem Nalla
Content The data was acquired from the **Premier League** website and is representative of seasons 2006/
17,139 downloads

Indian Premier League (Cricket) 894
Dataset · 8y ago · by Manas
Source: Content All Indian **Premier League** Cricket matches between 2008 and 2016.
84,320 downloads

English Premier League 172
Dataset · 3y ago · by SAIF UDDIN
Barclays **premier league**
25,648 downloads

Premier League 48
Dataset · 24d ago · by IvanRamosDataTech
The English **Premier League** is the most popular and highest level football **league** in the world.
3,605 downloads

Indian Premier League 2008-2019 417
Dataset · 6y ago · by Navaneesh Kumar
Context Indian **Premier League** (IPL) is a Twenty20 cricket format **league** in India.
34,938 downloads

English Premier League (EPL) Results 152
Dataset · 3y ago · by Alvin
Premier League Results from 1993-94 to 2021-22
8,935 downloads

How to train Your Personal AI Assistant(I)

But what if there are no open source datasets? Then what?

How to train Your Personal AI Assistant(I)

If you can't find any open-source datasets, there are generally two alternatives:

1. You can ***purchase data*** from a data broker, which can be expensive but provides instant access;
2. You can ***collect and curate*** your own dataset, which is often time-consuming but gives you full control over data quality and relevance.

Step II: Training

How to train Your Personal AI Assistant(II)

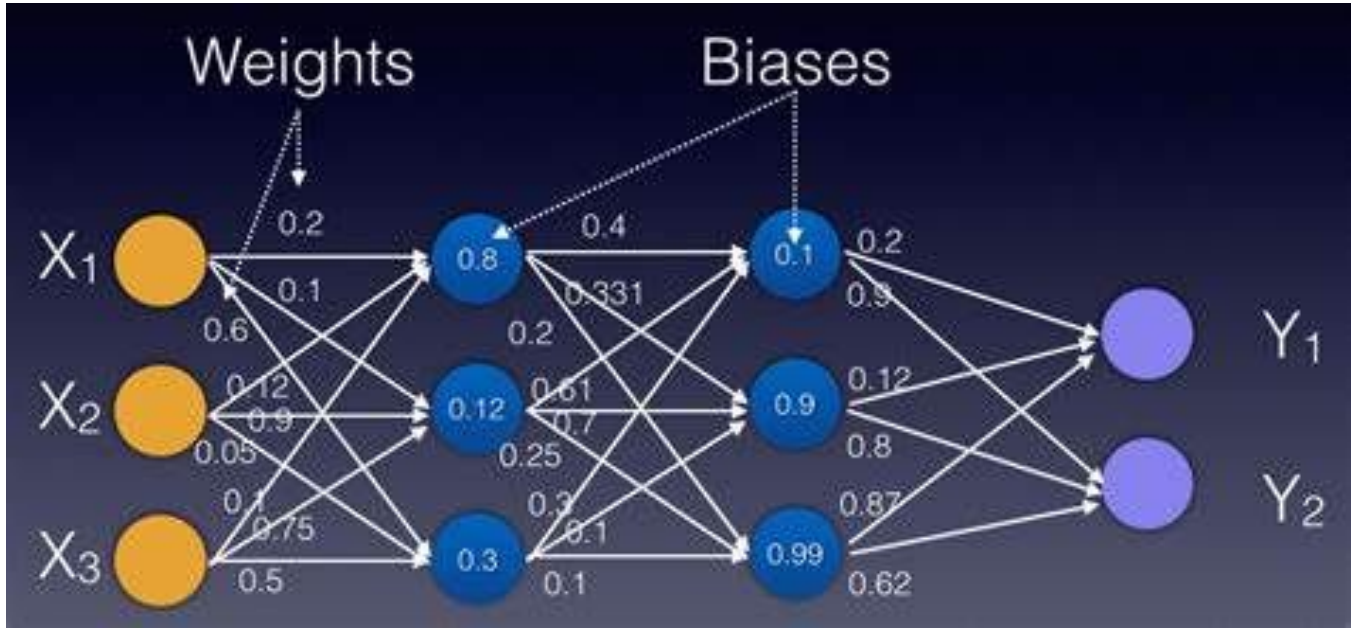
- The “Training” step is often the most straightforward one. At this stage, we use the dataset we have collected to feed a training algorithm, which adjusts the model’s parameters in order to learn useful patterns from the data.

How to train Your Personal AI Assistant(II)

- This step involves writing some code (usually in Python) and running it either on our local machine or in the cloud.
- Without going into much detail about how AI works, this is the stage where we train the “intelligent” part of the artificial “brain” (similar to training the “neurons” in a human brain).

How to train Your Personal AI Assistant(II)

- The “training” step focuses on adjusting the weights and biases between the nodes in the network.
- An AI model (specifically, a machine learning or neural network model) consists of interconnected nodes, each with associated values that are updated during training.



Introduction to Deep Learning

How to train Your Personal AI Assistant(II)

- Usually, the dataset is split into three parts: “training”, “validation”, and “testing”.
- This allows us to properly evaluate our model by measuring how well it performs on data it hasn’t seen during training.

```
[ ] data = gesture_recognizer.Dataset.from_folder(
        dirname=dataset_path,
        hparams=gesture_recognizer.HandDataPreprocessingParams()
    )
    train_data, rest_data = data.split(0.8)
    validation_data, test_data = rest_data.split(0.5)
```

How to train Your Personal AI Assistant(II)

- For a lone developer, a common way to train a model is by using ***Google Colab***.
- You need to create a ***.ipynb*** file (which is a combination of Python code and documentation, and can be executed step by step).
- The code will then run in the cloud.

How to train Your Personal AI Assistant(II)

- This is an example of a running training process. Depending on the size of the dataset, training can take anywhere from a few minutes to several hours.

gesture_recognizer.ipynb Save in GitHub to keep changes

File Edit View Insert Runtime Tools Help

Commands + Code + Text Run all Copy to Drive

from the dataset. The resulting dataset will contain the extracted hand landmark positions from each image, rather than images themselves.

The `HandDataPreprocessingParams` class contains two configurable options for the data loading process:

- `shuffle`: A boolean controlling whether to shuffle the dataset. Defaults to true.
- `min_detection_confidence`: A float between 0 and 1 controlling the confidence threshold for hand detection.

Split the dataset: 80% for training, 10% for validation, and 10% for testing.

```
data = gesture_recognizer.Dataset.from_folder(
    dirname=dataset_path,
    hparams=gesture_recognizer.HandDataPreprocessingParams()
)
train_data, rest_data = data.split(0.8)
validation_data, test_data = rest_data.split(0.5)
```

Train the model

Train the custom gesture recognizer by using the create method and passing in the training data, validation data, model options, and hyperparameters. For more information on model options and hyperparameters, see the Hyperparameters section below.

```
hparams = gesture_recognizer.HParams(export_dir="exported_model")
options = gesture_recognizer.GestureRecognizerOptions(hparams=hparams)
model = gesture_recognizer.GestureRecognizer.create(
    train_data=train_data,
    validation_data=validation_data,
    options=options
)
```

Evaluate the model performance

After training the model, evaluate it on a test dataset and print the loss and accuracy metrics.

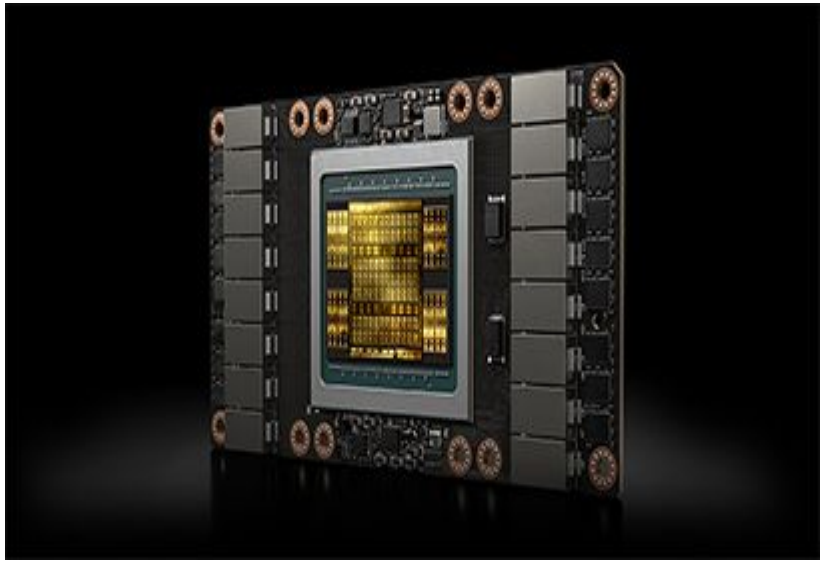
```
loss, acc = model.evaluate(test_data, batch_size=1)
print(f"Test loss:{loss}, Test accuracy:{acc}")
```

Variables Terminal Executing (13s) Python 3

<- Gesture recognition task guide example from ai.google.dev

How to train Your Personal AI Assistant(II)

- For large datasets—such as those used for enterprise or production-ready models—cloud-based training can be quite **expensive**, and **training locally** may be a more cost-effective solution.

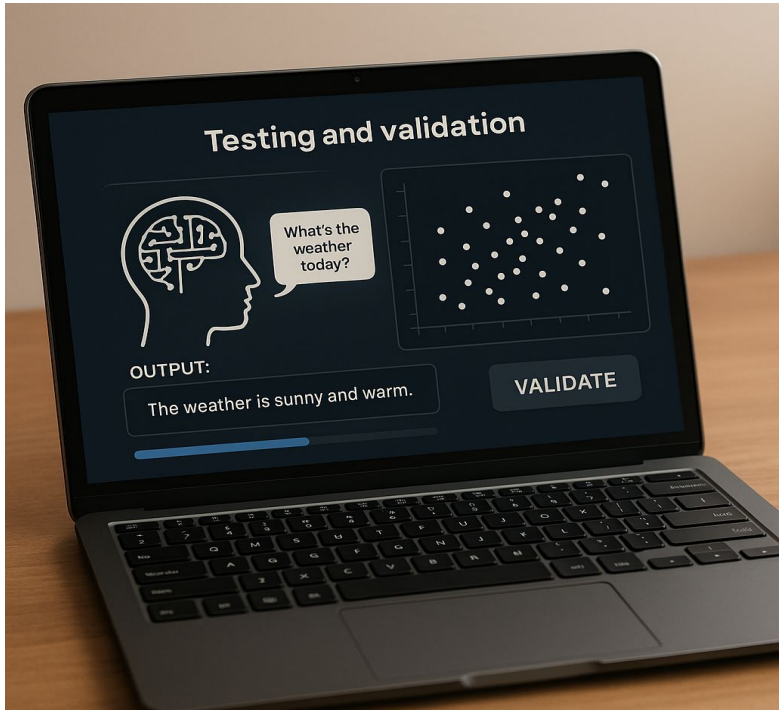


NVIDIA Volta, a special architecture for training deep learning models.

Step III:
Testing and validation

How to train Your Personal AI Assistant(III)

- The “Testing and Validation” step is where we analyze how well our trained model performs.
- There are several ways to measure its performance, but the most important metrics are accuracy, precision, recall, and F1-score.



How to train Your Personal AI Assistant(III)

- **Accuracy**: The percentage of all predictions that the model got correct.
- **Precision**: Of all the items the model identified as positive, how many actually are positive. (It measures how many selected items are relevant.)
- **Recall**: Of all the actual positive items, how many did the model identify as positive. (It measures how many relevant items are selected.)
- **F1-score**: The harmonic mean of precision and recall. It provides a balance between precision and recall, especially useful when you need to balance both.

How to train Your Personal AI Assistant(III)

- After analyzing these metrics, the next step is the manual testing phase.
- At this stage, we might be pleased with the results and consider the training process a success. 🙂
- But more often than not, we find that the model is… lacking. The most common problems are:
  1) ***Underfitting***
  2) ***Overfitting***

How to train Your Personal AI Assistant(III)

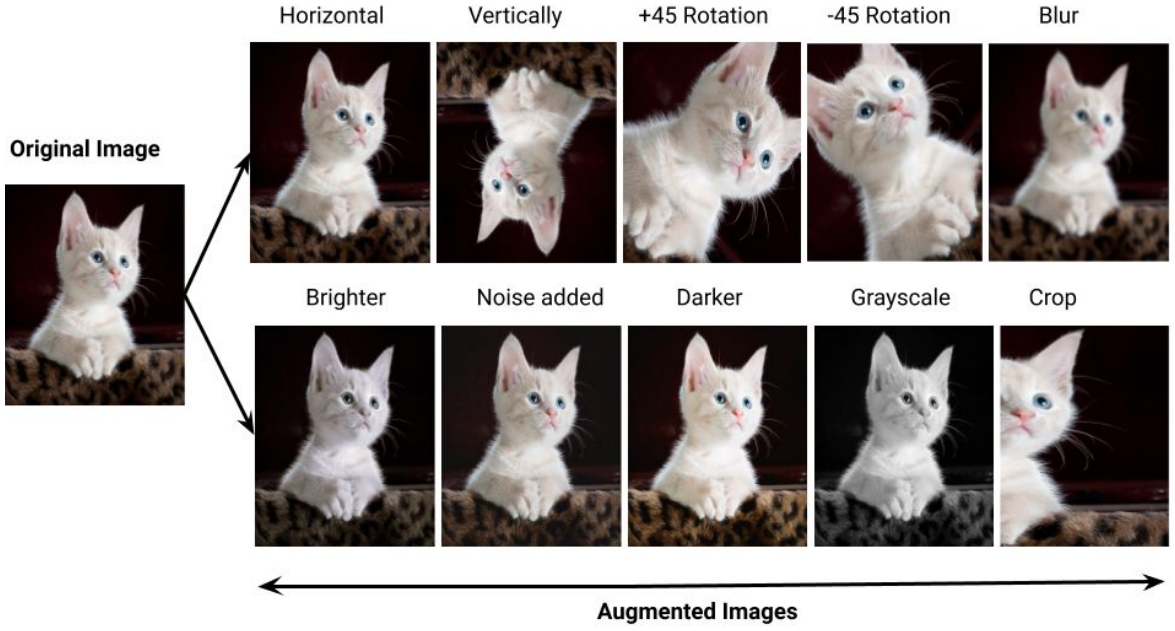
1) Underfitting: The model is too simple and fails to capture the underlying patterns in the data.
Possible solutions:

- Use a **more complex model** (e.g., add more layers or nodes to a neural network).
- **Increase** the duration or number of **training epochs**(how many times the model “trained” on the data).
- Provide **more or better quality data**. <-usually the culprit :(
- **Reduce regularization** or **avoid** techniques that **artificially reduce** the amount of training data.

How to train Your Personal AI Assistant(III)

What is Data Augmentation? A technique used to artificially expand your dataset by transforming existing examples — improving generalization and reducing overfitting.

- 📝 For Text:
  - Synonym replacement
  - Back-translation
  - Prompt rewording
  - Sentence shuffling
- 🖼️ For Images:
  - Rotation & flipping
  - Cropping & zoom
  - Brightness/contrast changes
  - Noise or blur injection



How to train Your Personal AI Assistant(III)

2) Overfitting: The model **performs well on training** data but **poorly on new, unseen** data because it has “memorized” the training examples, including noise.
Possible solutions:

- Use a **simpler model** with fewer **parameters**.
- Use **regularization techniques** (like dropout, L1/L2 regularization).
- **Gather more data** to help the model generalize better.
- **Perform early stopping** during training (stop training when validation performance stops improving).

How to train Your Personal AI Assistant(III)

- Once you have identified the problem, apply the appropriate solution and repeat steps I and II.
- After a few iterations, you will have an efficient AI model tailored to your task.

Well done! 🙂

How to train Your Personal AI Assistant

But…what about the costs?

- To fully train a ***good*** working model from scratch it could cost "**tens of thousands to hundreds of millions of dollars**" (based on estimates made by *epoch.ai*)
- Think about the data licensing costs, computing costs and other such expenses…

Solution: Fine-tuning

What is fine-tuning?

Fine-tuning is the process of adapting a pre-trained AI model to a specific task.

Instead of training a model from scratch, we use an existing one as a foundation.

This drastically reduces:

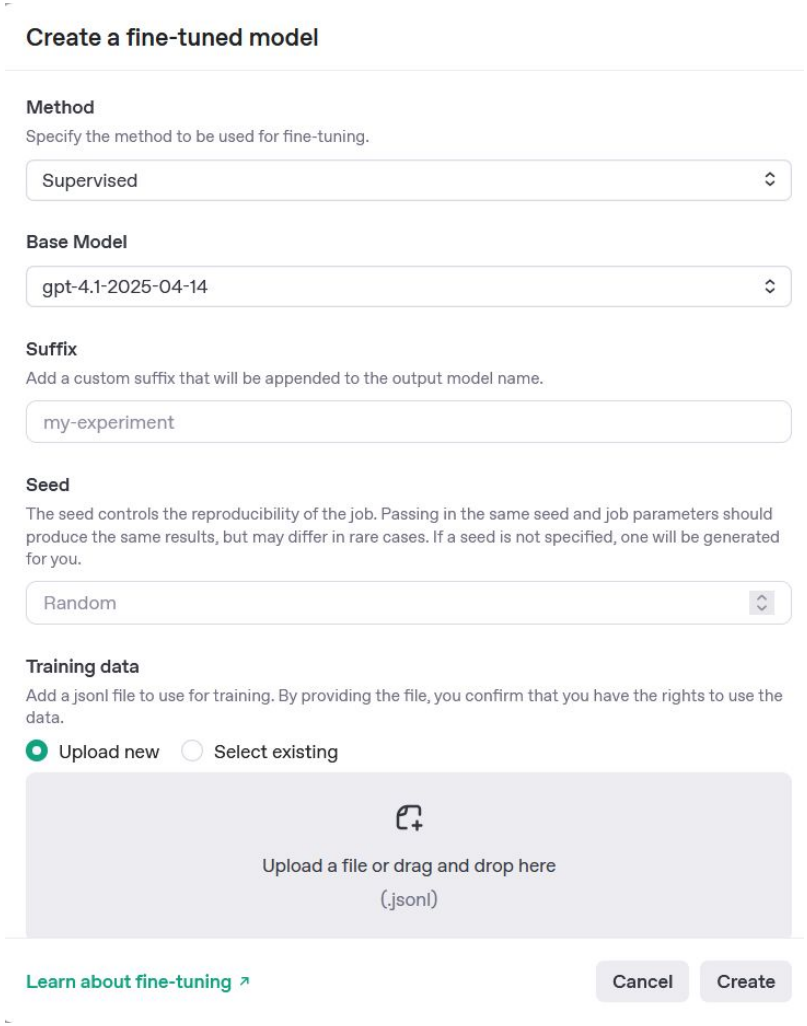
- 🕒 Time
- 💻 Compute power
- 💸 Cost

Why Fine-Tuning Instead of Full Training?

- Pre-trained models already understand language patterns and general knowledge.
- You only need a small amount of task-specific data to guide them.
- Ideal when:
  - You have limited data or resources.
  - You want to focus on a niche domain (e.g. legal, medical, banking).
  - You need fast iteration and deployment.

"Pretraining costs millions. Fine-tuning costs hundreds."

OpenAI Fine-Tuning



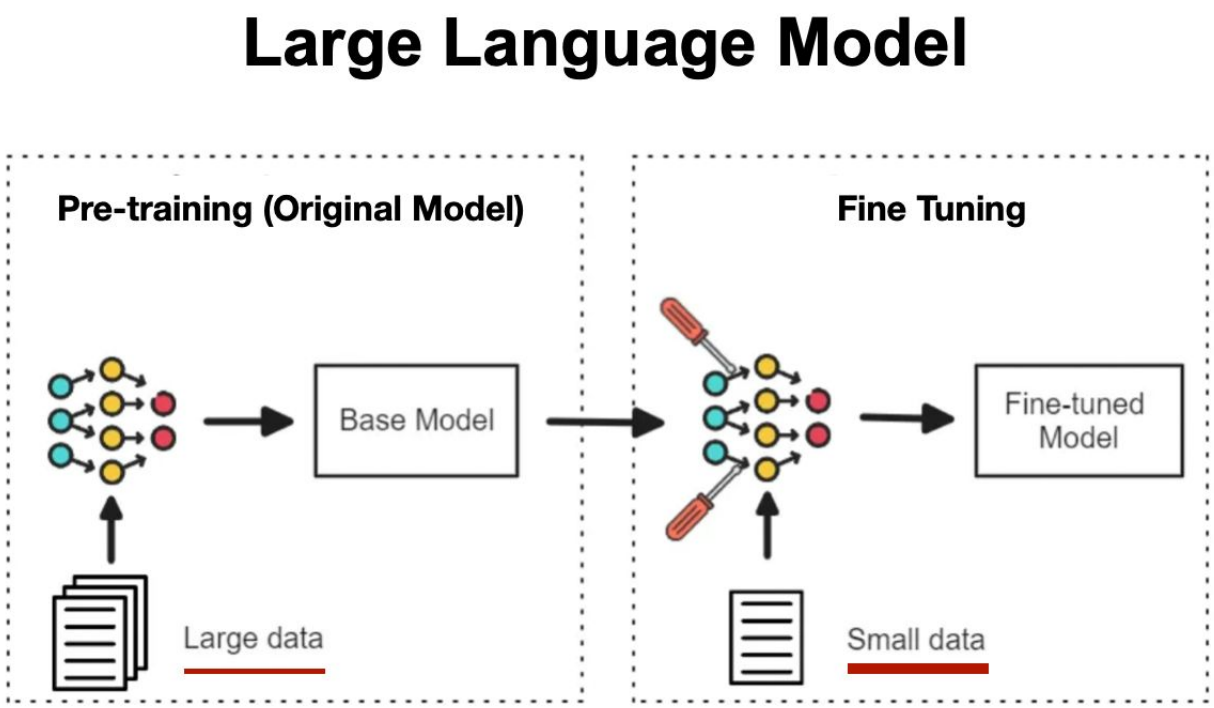
- Example for Fine-Tuning using OpenAI API

Fine-Tuning on LLMs (Large Language Models)

Fine-tuning an LLM means adapting a general-purpose language model (e.g., GPT, LLaMA, Falcon) to a specific domain or task using a smaller, specialized dataset.

Why do it?

- Makes the model more accurate and relevant to your use case (e.g., banking, legal, medical).
- Reduces the need for prompt engineering.
- Achieves better performance with fewer inputs.



Fine-Tuning on LLMs (Large Language Models)

Resources:

https://cookbook.openai.com/examples/how_to_fine-tune_chat_models

https://www.datacamp.com/tutorial/fine-tuning-large-language-models

Recap...

- **The three main steps** in training an AI model are: collecting and preprocessing data, selecting and configuring the model architecture, and iteratively training and evaluating the model for optimization.
- **Common pitfalls** include using insufficient or low-quality data, overfitting the model, and neglecting proper validation and testing processes.
- **Fine-tuning** involves adapting a pre-trained model to a specific task with less data and resources, while training from scratch requires building and training a model entirely with new data.
- **Useful resources for training** AI models include open-source datasets, tutorials, online courses, and extensive documentation found on platforms like TensorFlow, PyTorch, Hugging Face, and academic repositories.

THANK YOU!

Any questions?

Razvan Tudosie
razvan_tudosie@sync.ro

Alexandru Smarandache
alex_smarandache@sync.ro