Google Colab
A Beginner's Guide

What is Google Colab?

- Definition: "Google Colab is a free, cloud-based platform that allows you to write and execute Python code in a web browser.“

- Features: GPU support, no setup required, easy sharing, etc.

Setting Up Google Colab

1. Open your browser.

2. Go to Google Colab. https://colab.research.google.com/

3. Sign in with your Google account.

Starting a New Notebook

1. Click on "NEW NOTEBOOK".

2. Your new notebook will open in a new tab.

Renaming Your Notebook

1.Click on the notebook name (e.g., "Untitled0.ipynb") at the top.

2. Rename it to something meaningful related to your project.

Writing and Running Code

1. Click on a code cell.

2. Write some Python code (e.g., print("Hello, Google Colab!")).

3. Click on the Play button or press Shift + Enter to run the code.

4. See the output displayed below the cell.

Adding Text/Comments

1. Click on + TEXT at the top.

2. Add notes, explanations, or any text.

Saving Your Work

1. Google Colab automatically saves your notebook to Google Drive.

2. You can also click on "File" then "Save", or press Ctrl + S.

Sharing Your Notebook

1. Click on the "Share" button (top-right corner).

2. Choose sharing options: Shareable link or specific people.

3. Adjust permissions: Viewer, Commenter, or Editor.

Using GPUs and TPUs

1. Click on "Runtime" in the top menu.

2. Choose "Change runtime type".

3. Select either GPU or TPU under "Hardware accelerator".

Importing Libraries

1.Use the !pip install command for libraries not pre-installed (e.g., !pip install pandas).

2.Showcase importing a library: import tensorflow as tf.

Uploading and Using Data

1. Use the Files tab on the left panel.

2. Click on "Upload".

3. Navigate and select files from your device.

Downloading Your Notebook

1. Click on "File" in the top menu.

2. Hover over "Download".

3. Select your preferred format (e.g., .ipynb, .pdf).

Important Tips

1. List some general tips and best practices (e.g., "Clear the output before sharing if your notebook has a lot of visualizations to reduce the file size").

Top 5 Tips for Deep Learning in Google Colab

1.Maximize GPU Usage 🚀

Ensure you're using a GPU runtime for training deep learning models.

Navigate to Runtime > Change runtime type and select GPU.

2. Persistent Storage with Google Drive ☁️

Colab VMs are temporary. Avoid data loss by using Google Drive as persistent storage. Mount with:

```
from google.colab import drive
drive.mount('/content/drive')
```

Top 5 Tips for Deep Learning in Google Colab

3. Libraries & Dependencies 📚

Ensure compatibility when using specific versions of deep learning libraries. Double-check if importing both TensorFlow and PyTorch.

4. Regularly Save Work 💾

Manual saving can prevent loss of important results, even though Colab autosaves. Press Ctrl+S frequently.

5. Enable Tooltips ℹ️

While typing or when you hover over a function, press Ctrl + Shift + Space to display its signature and docstring as a tooltip.

Persistent Storage with Google Drive ☁️

- Google Colab provides Virtual Machines (VMs) to run the notebooks. Once you close your browser or remain inactive for a while, these VMs can be recycled. This means that any data stored directly on the VM will be lost.

- Google Drive is a cloud storage solution that offers persistent storage. By integrating Google Drive with Colab, you can ensure that your data, models, or any other files remain accessible even after your Colab session ends.

Use code snippet below to mount Google Drive within your Colab environment.

- from google.colab import drive
- drive.mount('/content/drive’)

After successful authorization, your Google Drive will appear as a folder named 'drive' in the file browser on the left side of your Colab interface.

Conceptually, what are artificial neural networks?

A *neuron* receives a signal, processes it,

and propagates the signal (or not)

• The brain is comprised of around 100

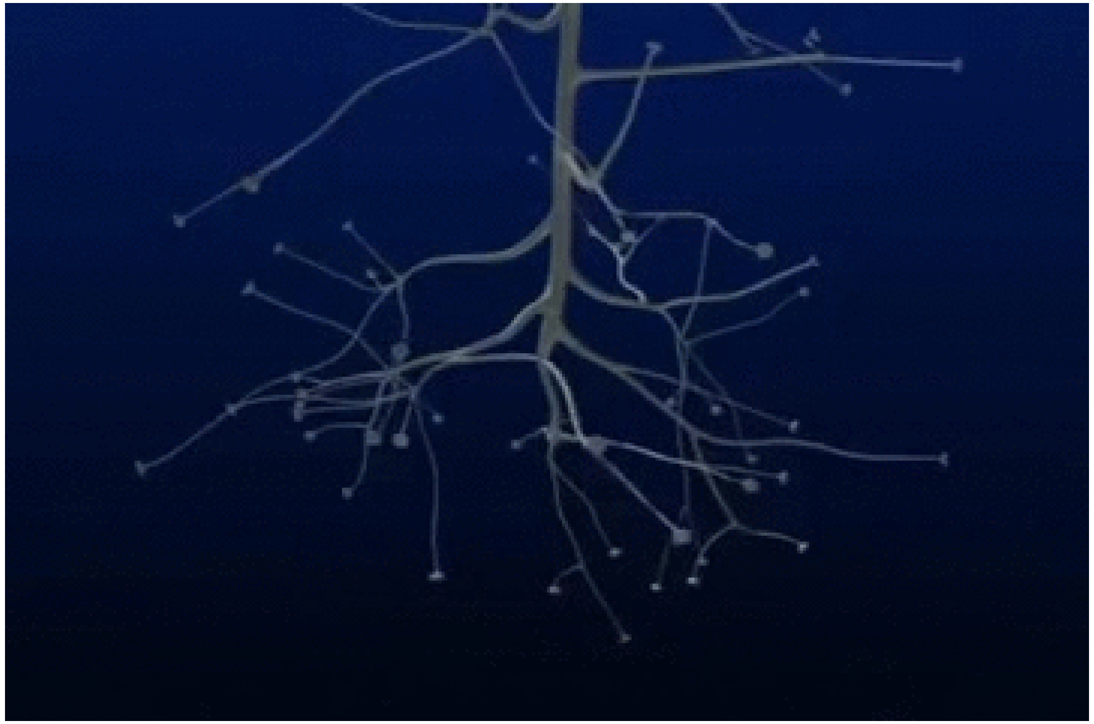
billion neurons, each connected to ~10k

other neurons: 1015 synaptic
connections

• *ANNs* are a *simplistic* imitation of a
brain

comprised of dense net of simple

structures



Source: databricks: Deep Learning Fundamentals

AI, ML and DL Defined

- AI refers to the broad concept of machines being able to carry out tasks in a way that we would consider "smart". It encompasses any technique that enables computers to mimic human behavior and intelligence. This includes rule-based systems, logic, and decision trees, among other methodologies.

- ML is a subset of AI that use computational methods to "learn" information directly from data without relying on a predetermined equation as a model.

- DL is a subset of machine learning that uses multi-layered neural networks to simulate human decision-making.

Traditional Programming Approach

1.Rules-based: In traditional programming, developers write explicit rules to process data and produce results. They essentially instruct the computer on exactly how to solve a specific problem.

2.Human-driven logic: Developers require domain knowledge to write effective rules. For complex problems, encoding rules can become increasingly intricate and error-prone.

3.Limitations: It's excellent for problems with well-defined and unchanging rules. But for problems with too many variables and complexities, like speech recognition, traditional programming can become unfeasible.

Machine Learning Approach

1. **Data-driven**: Instead of writing explicit rules, developers feed algorithms with data and let the algorithms figure out the rules or patterns on their own.

2. **Automated learning**: The logic (or model) is derived from the data itself. The algorithm identifies patterns or relationships within the data without being explicitly programmed to do so.

3. **Limitations**: ML requires a substantial amount of data, especially for complex problems. Additionally, the learned models can sometimes be "black boxes", making it hard to interpret their decision-making process.

Types of Machine Learning

Supervised Learning

Email spam classification
Handwriting recognition

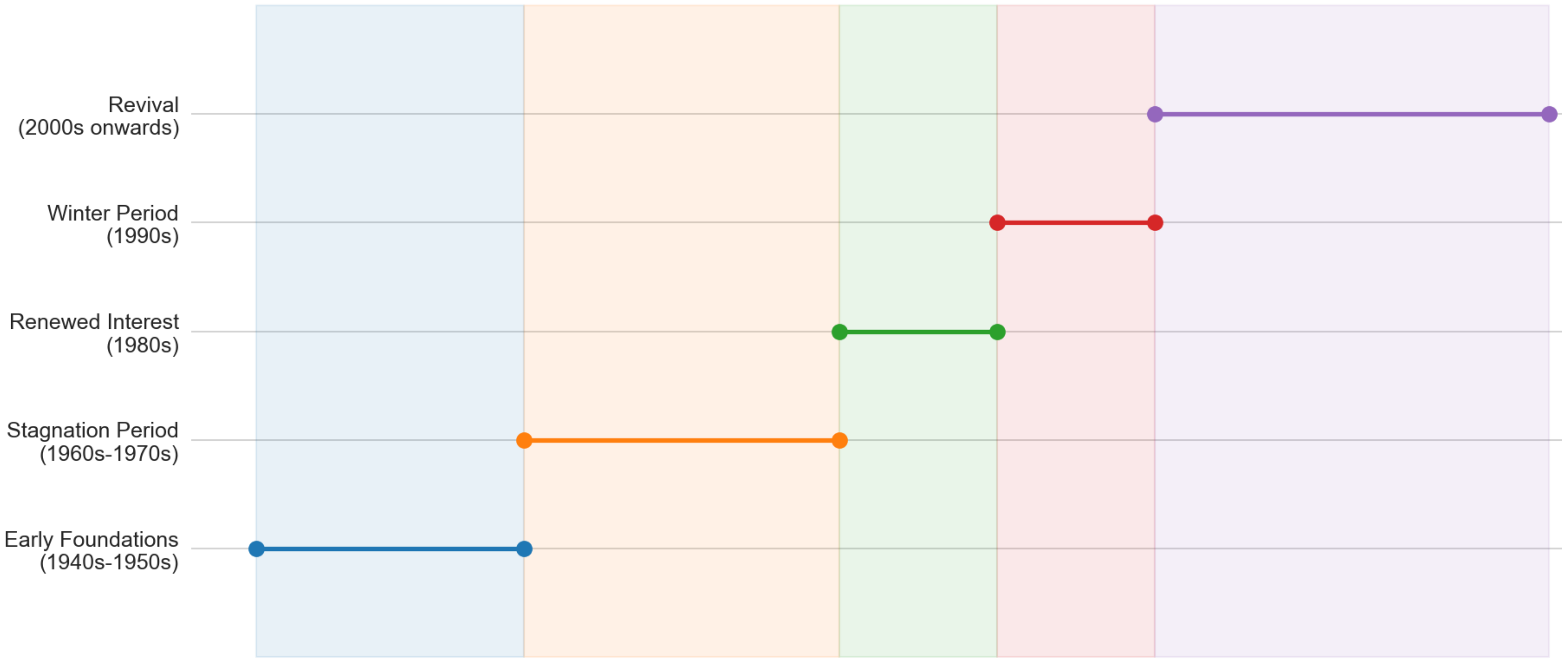
Unsupervised Learning

Market basket analysis
Anomaly detection in network traffic

Reinforcement Learning

Autonomous vehicle navigation
Go Games

History of Deep Learning



The Beginning(1940s)

- Neural networks are computational models inspired by how neurons in the brain function.

- McCulloch-Pitts neuron model (1943) simplified neurons as binary switches, simulating intelligent behavior.

- Donald Hebb's "The Organization of Behaviour" (1949): Introduced the idea of neurons strengthening their connections based on activity. This concept is encapsulated in the phrase "Cells that fire together, wire together." Meaning, if two neurons often activate at the same time, the connection between them strengthens. This is an essential principle for associative learning.

Hebbian Learning and Perceptron(1950s)

- 'Threshold Logic': A neuron activates (fires) if input surpasses a specific threshold.
- The first Hebbian network was built at MIT (1954) demonstrating associative memory.
- Frank Rosenblatt's Perceptron (1958): A single-layer neural network that adjusted weights using an algorithm.

Limitations and Skepticism(1960s)

- **The Perceptron was found to be limited in its abilities, notably it couldn't learn the XOR function.**

- Marvin Minsky and Seymour Papert’s “Perceptrons” (1969) highlighted these limitations.

- The skepticism led to reduced funding and interest, ushering in the 'AI winter.'

Renewed Interest(1980s)

- The Hopfield network (1982) offered solutions for optimization problems using energy functions.

- **Discovery of Backpropagation: An algorithm that adjusts neural network weights to minimize output errors. Made multi-layer neural networks feasible.**

Deep Learning Revolution(90s to 2010s)

- Advancements in computational power and data availability empowered more complex neural network architectures.

- "Deep" neural networks, with many hidden layers, began to achieve remarkable results in various tasks.

- Major events:
  - **Geoffrey Hinton's team (2012)**: Utilized deep learning to drastically reduce error rates in image classification in the ImageNet competition.
  - **Availability of Big Data**: Vast datasets became accessible, enabling training of larger neural networks.
  - **GPUs**: Graphics processing units accelerated computations, making training deep networks feasible.

Current Landscape(2010s onwards)

- Explosion of deep learning applications: Image & speech recognition, natural language processing, and more.

- Development of advanced architectures like CNNs (for image processing) and RNNs/LSTM/Transformer models (for sequential data).

- Open-source tools (e.g., TensorFlow, PyTorch) democratized access to deep learning.

- Continued improvement of model capabilities, leading to models like GPT-3 and BERT, which have pushed the boundaries of natural language understanding.

Current Landscape(2010s onwards)

- Explosion of deep learning applications: Image & speech recognition, natural language processing, and more.
- Development of advanced architectures like CNNs (for image processing) and RNNs/LSTM/Transformer models (for sequential data).
- Open-source tools (e.g., TensorFlow, PyTorch) democratized access to deep learning.
- Continued improvement of model capabilities, leading to models like GPT-3 and BERT, which have pushed the boundaries of natural language understanding.

BREAK

Please come back @

10:30 AM EST

9:30 AM CST