LESSON 1

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING - INTRODUCTION

Hi! It's me, your instructor for the course

Rohit Singh Chauhan

Current Location - Noida

I'm an AI enthusiast, currently heading data division at Calance

02

I love walking, sketching, reading and teaching mathematical concepts behind AI/ML algorithms and solving design problems.

Past Work/Education Experiences

A quick run down of my education and companies I've worked with

2013

IIT - Varanasi

B.Tech + M.Tech

2016

Columbia University (Ivy League)

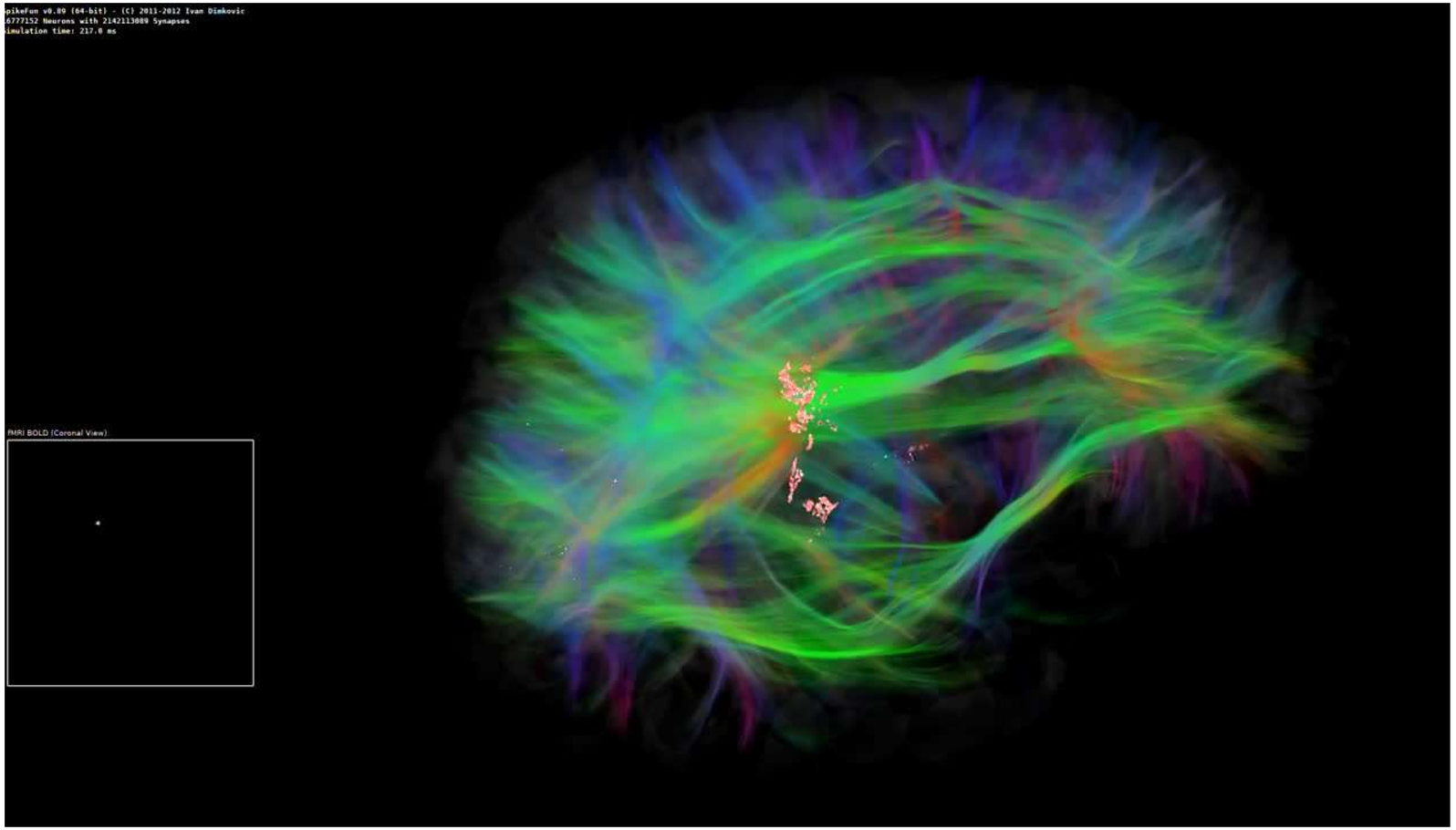
M.S - BioStatistics

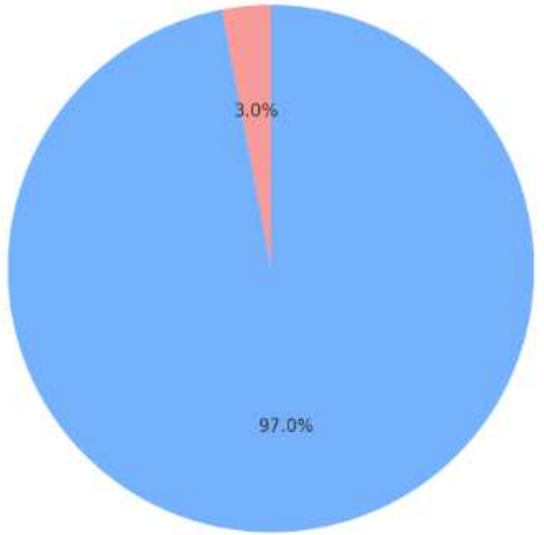
2017-2023

Data Scientist/Head

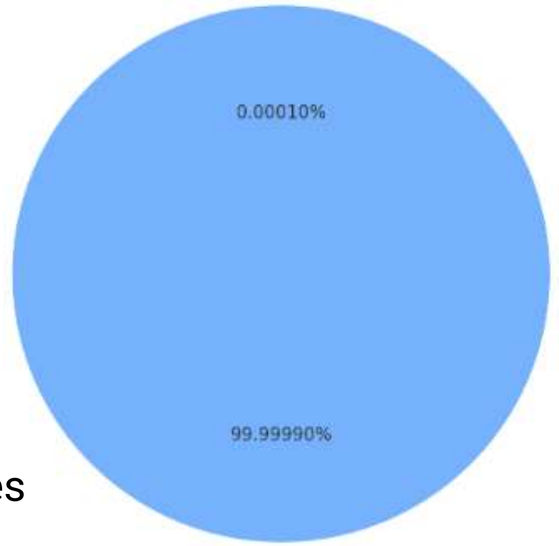
Capital One, Calance

Scale of the brain





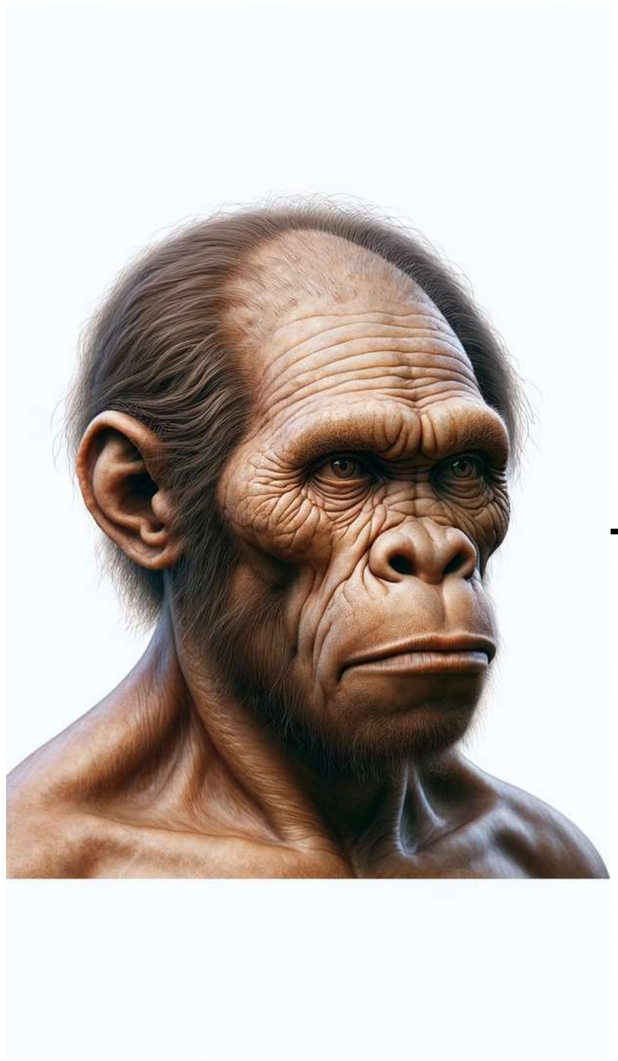
3% of brain neurons



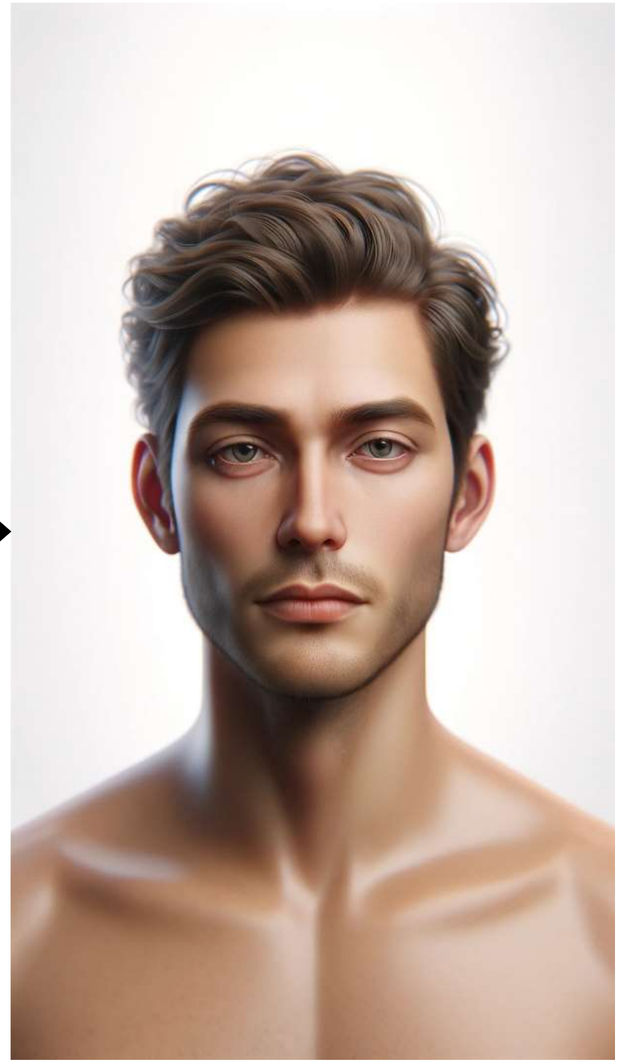
0.0001% of brain synapses

Artificial Brain Simulation - Thalamocortical System, 16.7 Million Neurons - 2.1 Billion Synapses

Time to evolve - Humans



300,000
years



Where does evolution stand now : Both of these are AI (DALLE) generated images, instructed by me

Time to evolve - AI

1943

Neural Networks Paper - Warren McCulloch and Walter Pitts

1950s

Alan Turing - Turing Test
John McCarthy coins term : Artificial Intelligence

1960s

First Robot Assembly Line - Unimate
Early NLP Progrm - ELIZA by Joseph Weizenbaum

1970s - 1980s

AI Winter : Realization that AI was difficult with limited compute and limited data
Expert System : Rule Based

1990s

Machine Learning Rise : IBM's Deep Blue defeat Gary Kasparov in Chess
More data and compute

2000s

Reinvestment by tech companies like Google, IBM
Rebranded as Deep Learning in 2006

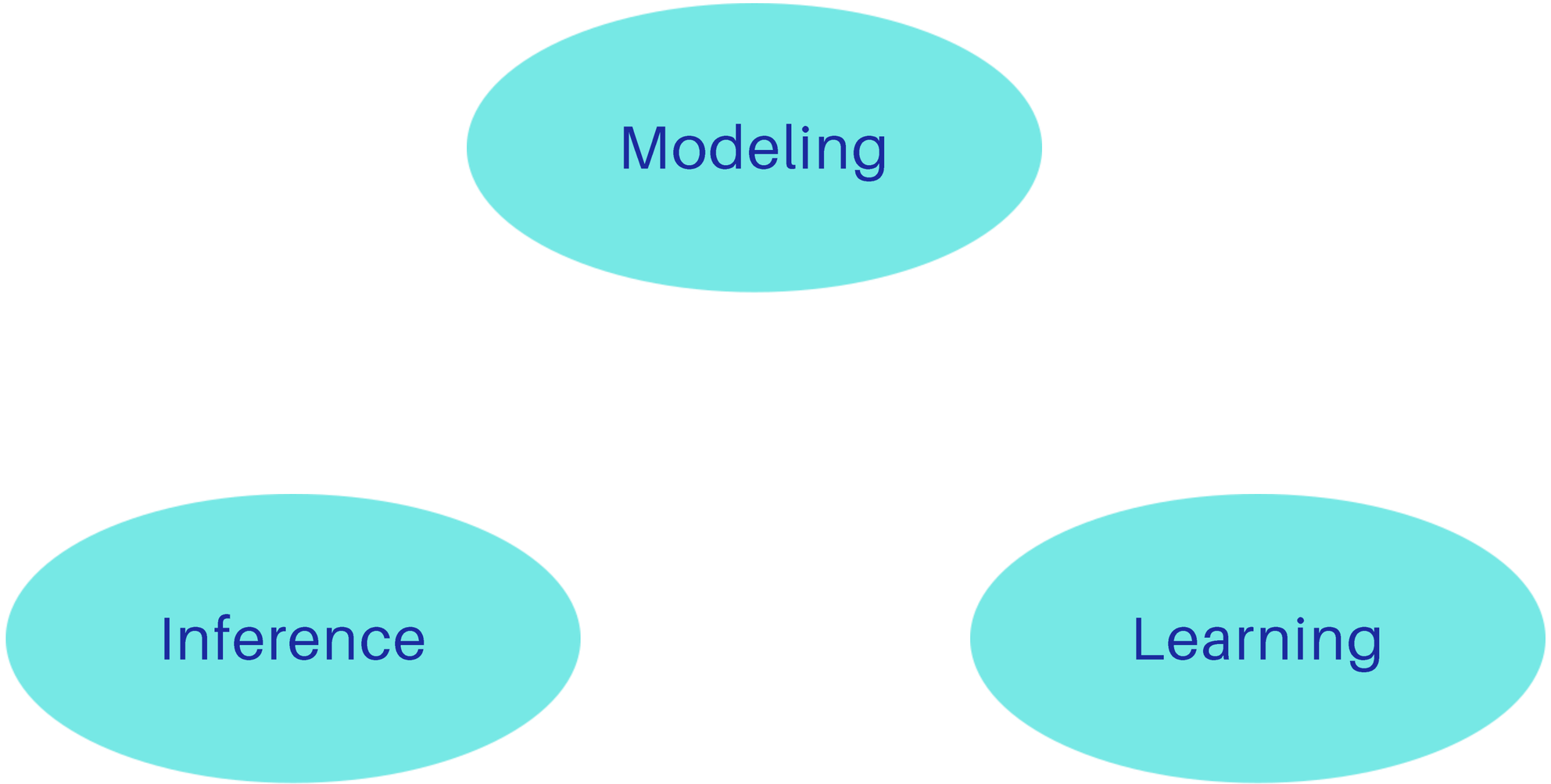
2010s

AlexNet : Significant outperform previous architecture in ImageNet Object recognition
Google AlphaGo : defeats world champion

2020s

GenAI boom : OpenAI's ChatGpt for text (GPT 3 - 2020), DALLLE for image generation (2021)

Generic paradigm of AI

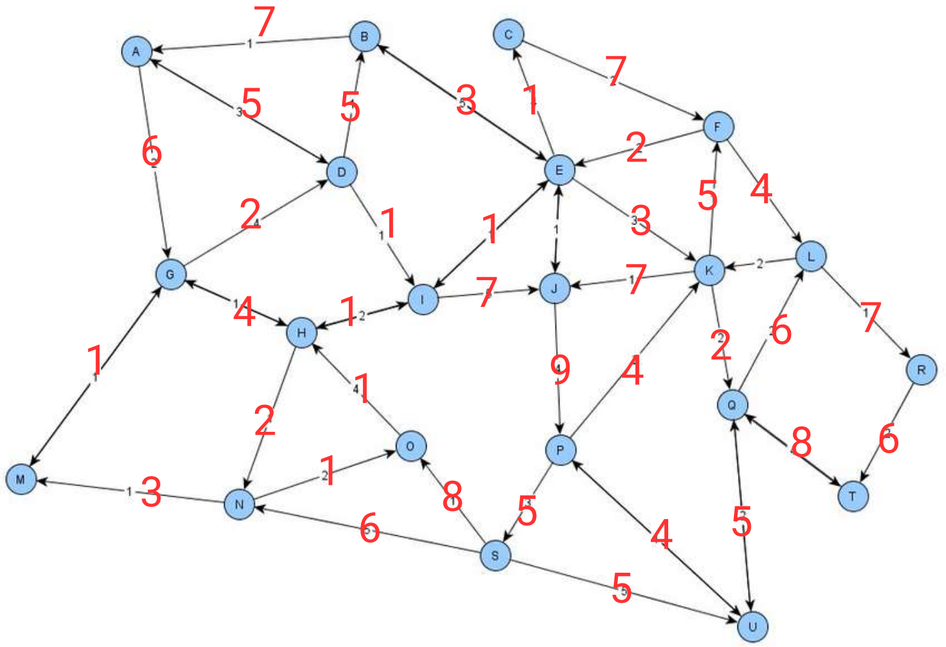


* Percy Liang lecture - Stanford

Paradigm : Modelling

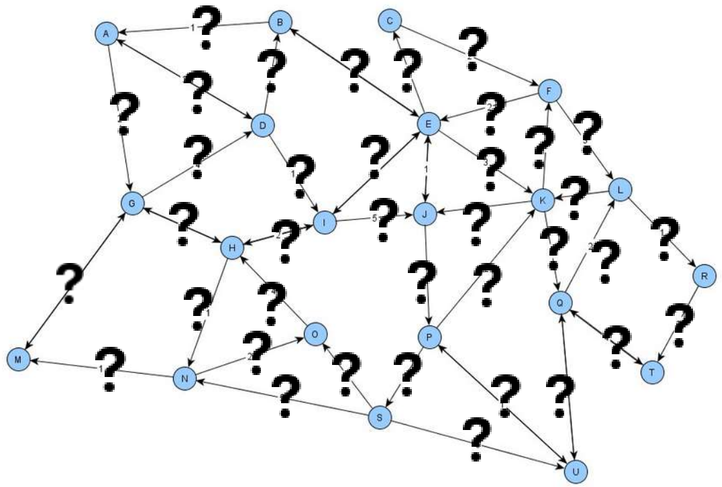
Real world road network

Model

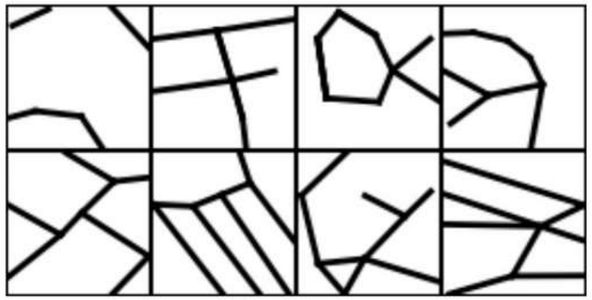


Graph model of road network

Paradigm : Learning

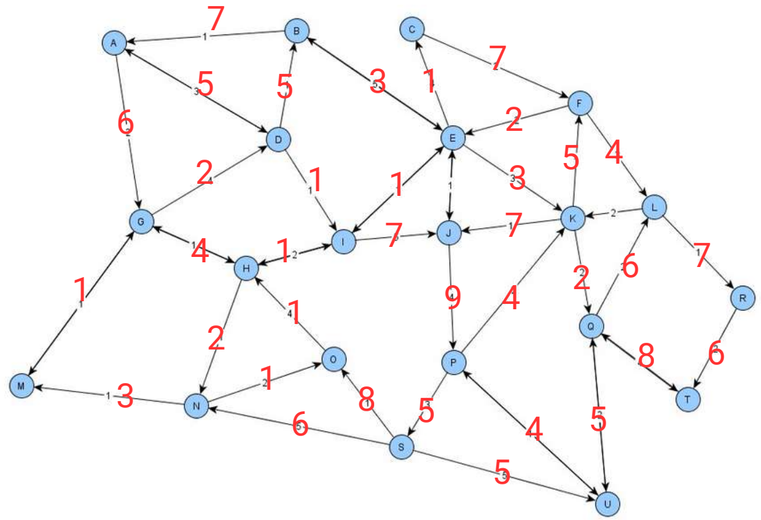


Graph model of road network



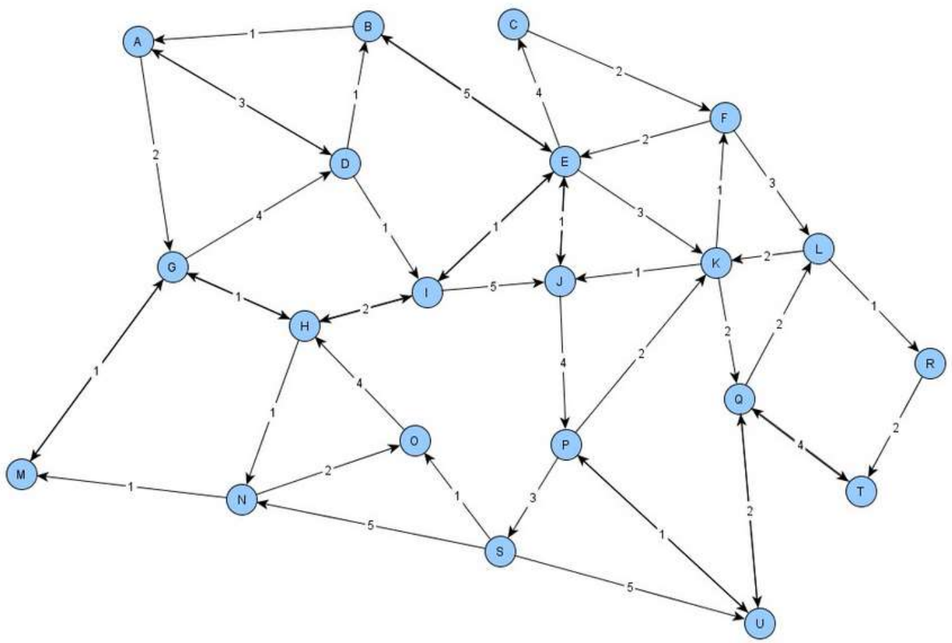
Road network data

Learning



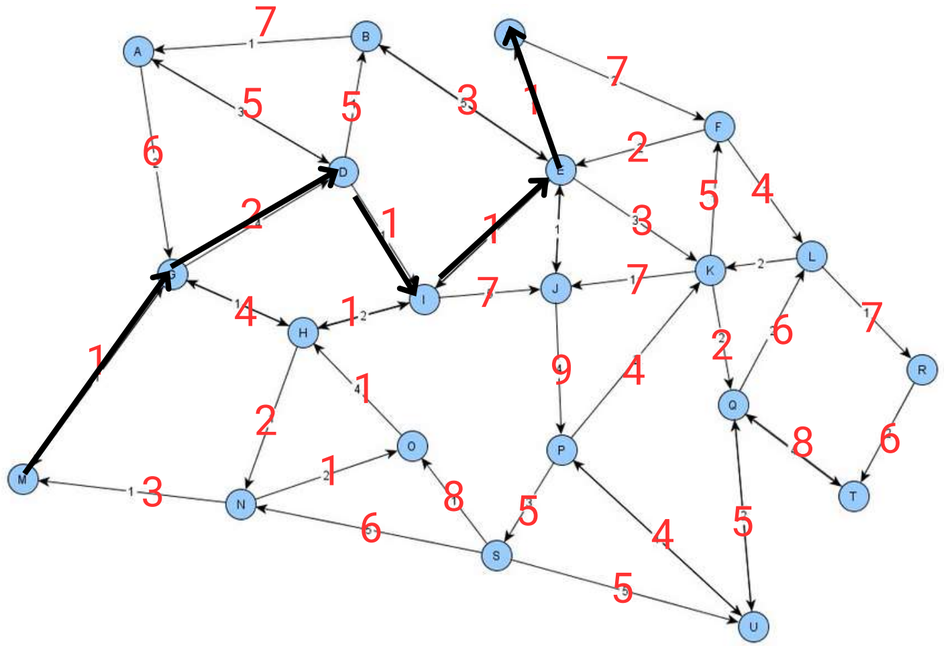
Graph edge weights learned from data

Paradigm : Inference



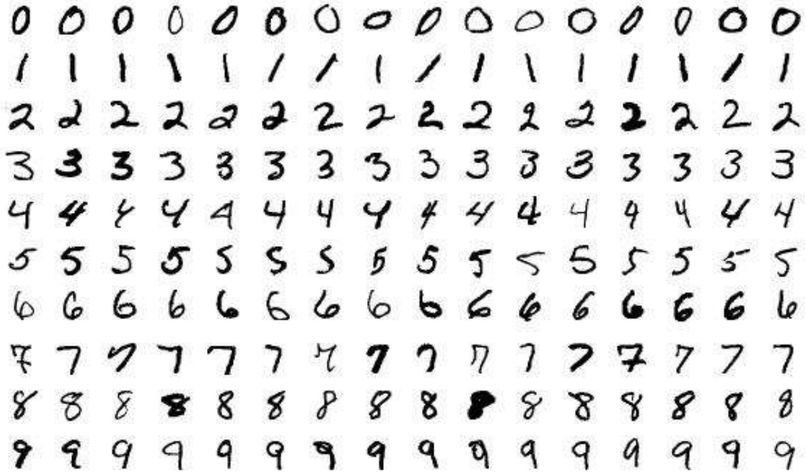
Graph model of road network

Inference



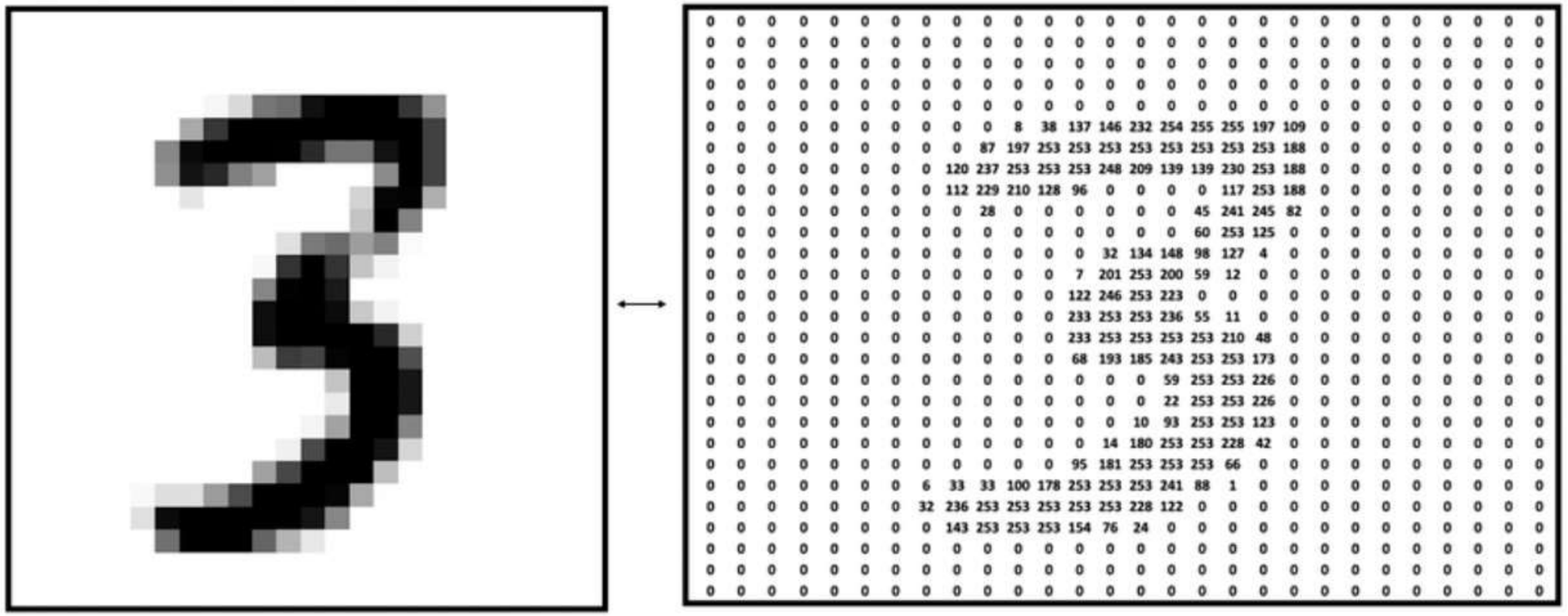
Shortest path distance

Example of all paradigms using MNIST dataset - Data



Handwritten digits :

- 60,000 training set
- 10,000 testing set

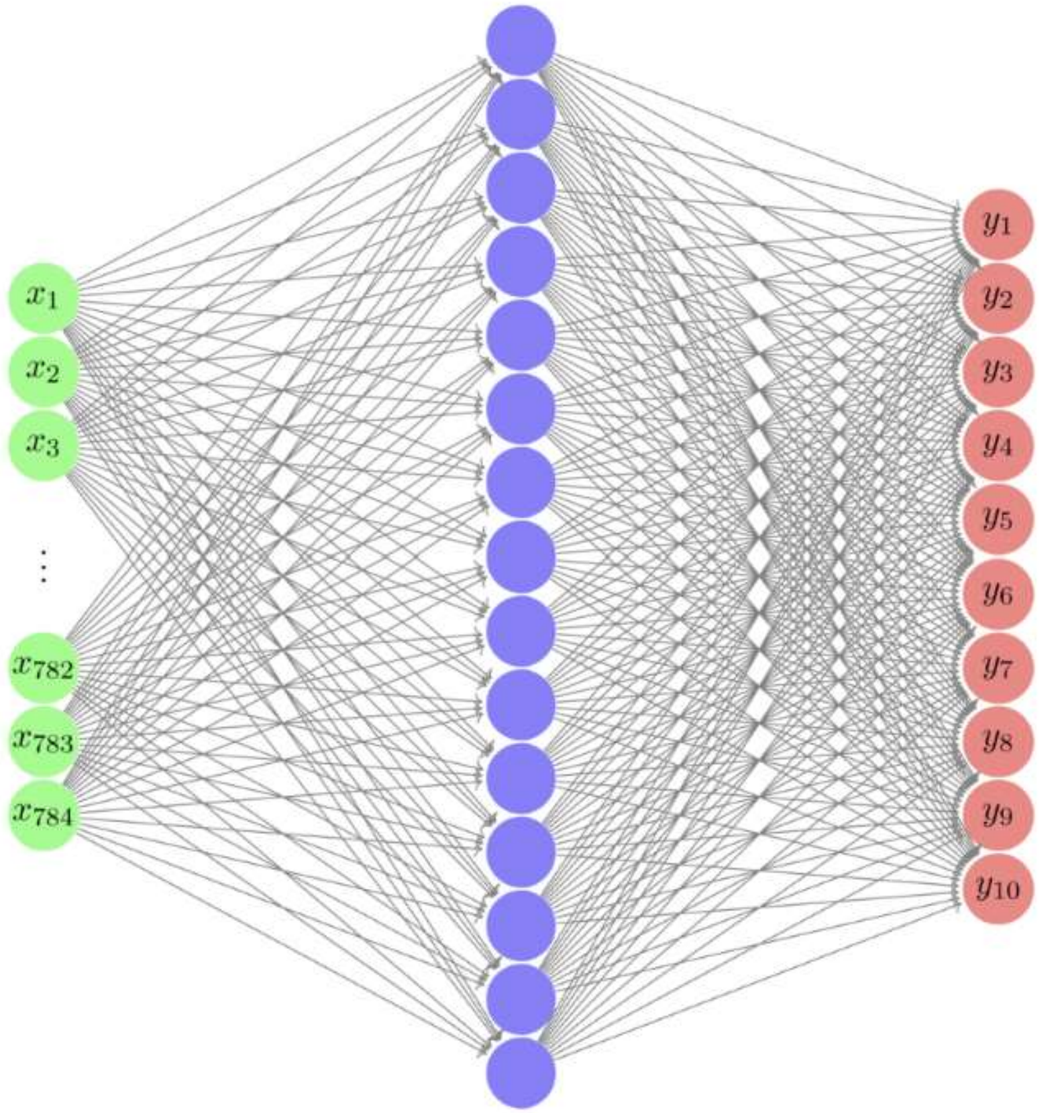


Each digit :

- Grayscale images - Intensity of grey on a scale of 0 to 255
- 28 X 28 pixels

Example of all paradigms using MNIST dataset - Model

Our Model choice - Neural Network



1st Layer - Input layer of flattened 784 nodes (28 X 28 pixels flattened)

2nd Layer - Hidden dense layer of 128 nodes

3rd Layer - Output layer of 10 nodes, each node representing a digit (0-9)

Mathematical aspects:

2nd Layer activation function - Rectified Linear Unit (ReLU)

3rd Layer activation function - Softmax function

Optimizer - Adam Optimizer

Loss function - Sparse Categorical CrossEntropy

Example of all paradigms using MNIST dataset - Learning (Code example)

Step 1 : Import prebuilt APIs (python libraries)
Step 2 : Download dataset
Step 3 : Build Neural Network Model
Step 4 : Configure Neural Network model for training
Step 5 : Train the model on training images
Step 6 : Model evaluation/getting test accuracy
Step 7 : Visualize the results

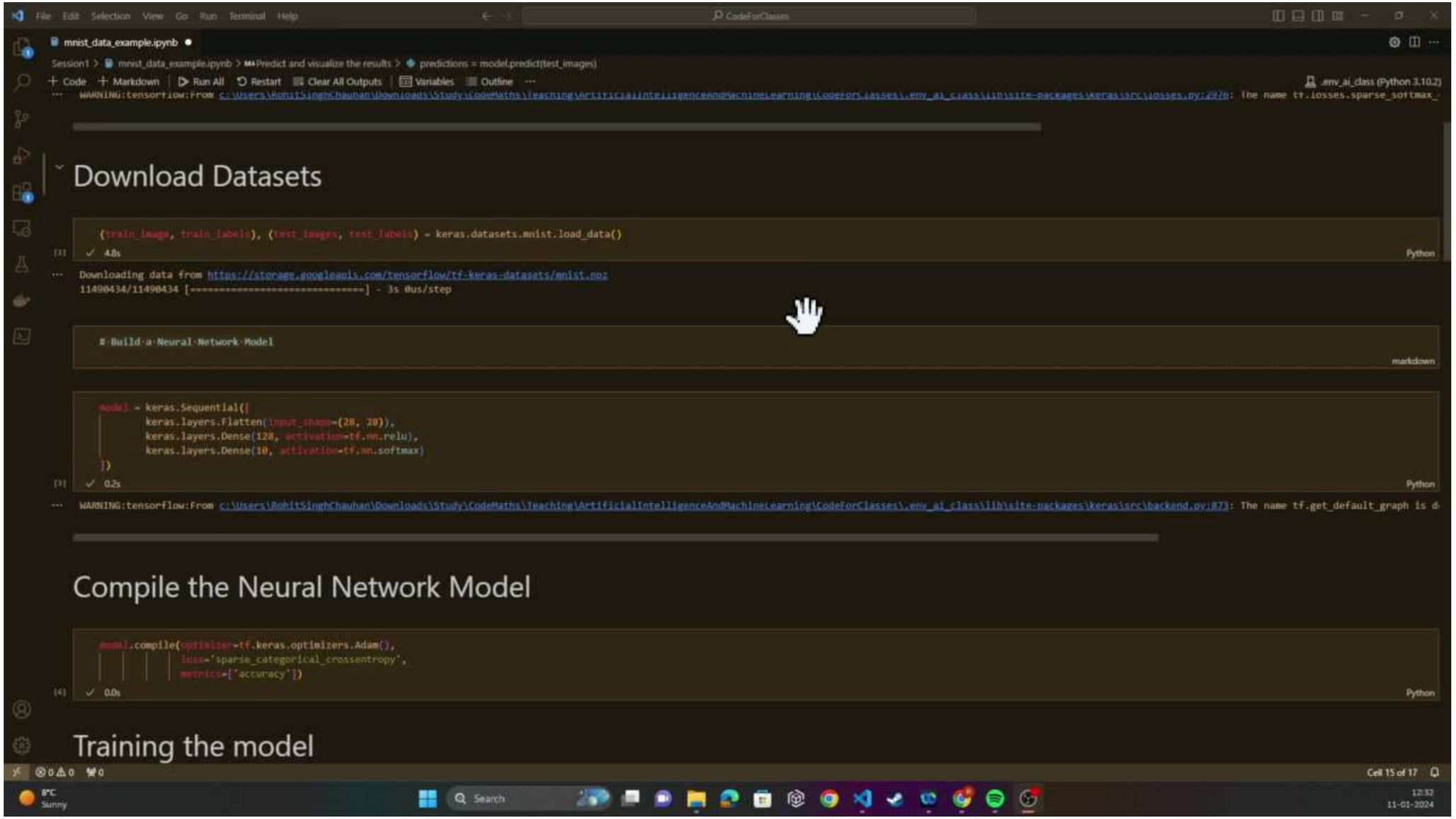
Step 1
Step 2
Step 3
Step 4
Step 5
Step 6
Step 7

Session1 > mnist_data_example.py > ...

```python
# Import libraries
import tensorflow as tf
from tensorflow import keras

# Download Datasets
(train_image, train_labels), (test_images, test_labels) = keras.datasets.mnist.load_data()

# Build a Neural Network Model
model = keras.Sequential([
        keras.layers.Flatten(input_shape=(28, 28)),
        keras.layers.Dense(128, activation=tf.nn.relu),
        keras.layers.Dense(10, activation=tf.nn.softmax)
])

# Compile the Neural Network Model
model.compile(optimizer=tf.keras.optimizers.Adam(),
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

# Training the model
model.fit(train_image, train_labels, epochs=5)

# Evaluate model performance on test dataset
test_loss, test_acc = model.evaluate(test_images, test_labels)

print('test_accuracy', test_acc)

# Predict and visualize the results
predictions = model.predict(test_images)


import matplotlib.pyplot as plt
import numpy as np

# Function to plot images, their predicted label, and the true label
def plot_image(i, predictions_array, true_label, img):...

# Displaying the first 15 test images, their predicted label, and the true label
num_rows = 3
num_cols = 5
num_images = num_rows * num_cols
plt.figure(figsize=(2*2*num_cols, 2*num_rows))
for i in range(num_images):
    plt.subplot(num_rows, 2*num_cols, 2*i+1)
    plot_image(i, predictions, test_labels, test_images)
plt.tight_layout()
plt.show()
```

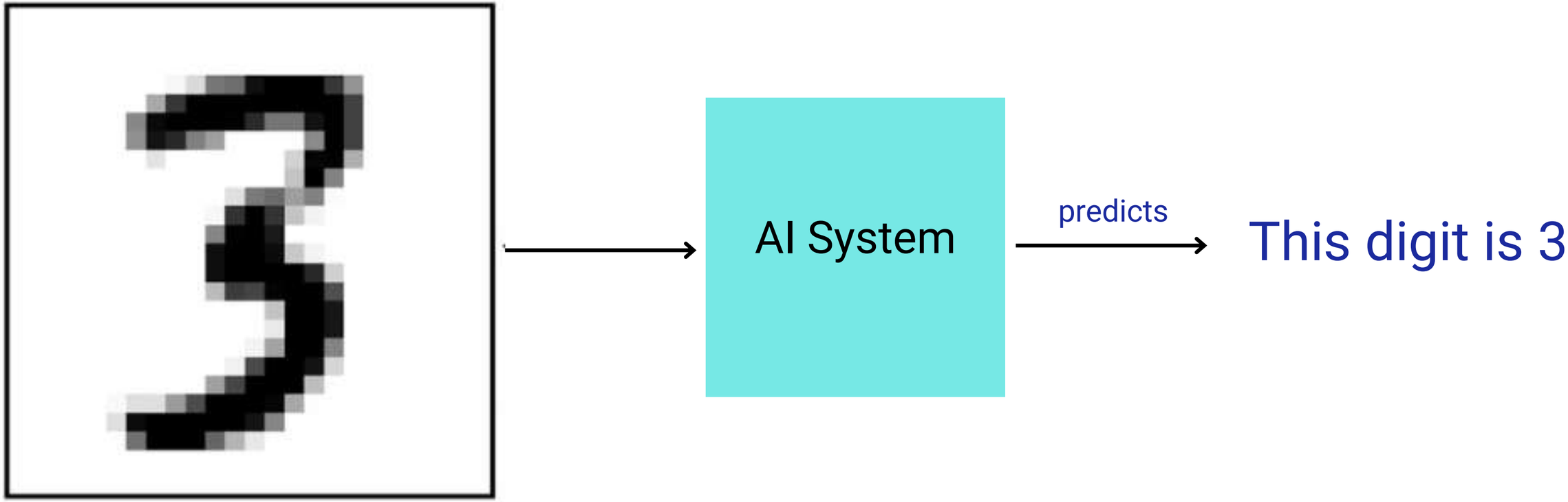
* Inspired from Lex Friedman lecture

Example of all paradigms using MNIST dataset - Learning (Code example)

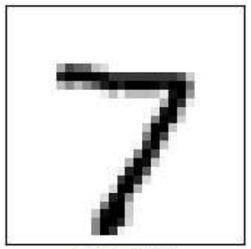


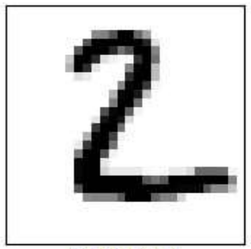
Example of all paradigms using MNIST dataset - Inference

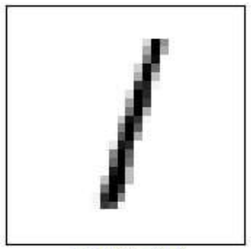
Digit goes to AI system (not human), tell which digit it is

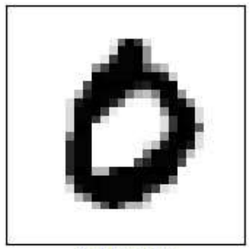


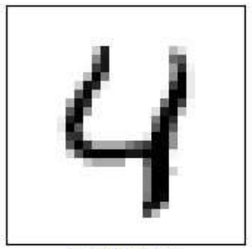
Example of all paradigms using MNIST dataset - Inference

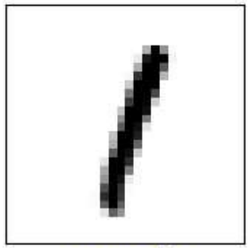

7 99% (7)

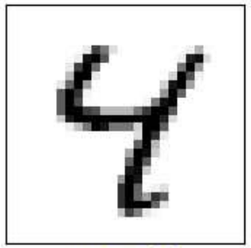

2 100% (2)

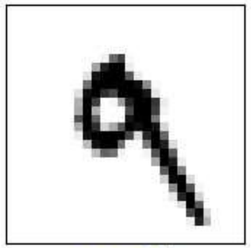

1 100% (1)

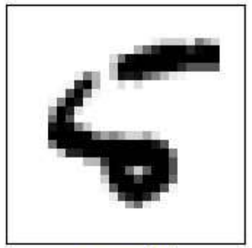

0 97% (0)


4 100% (4)

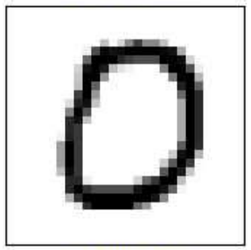

1 100% (1)

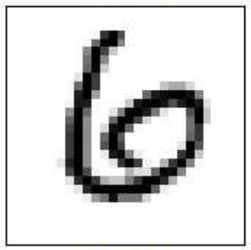

4 100% (4)


9 91% (9)

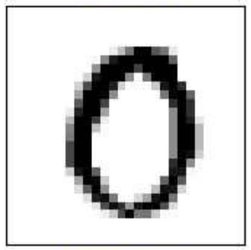

5 82% (5)

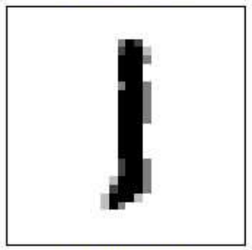

9 100% (9)


0 100% (0)

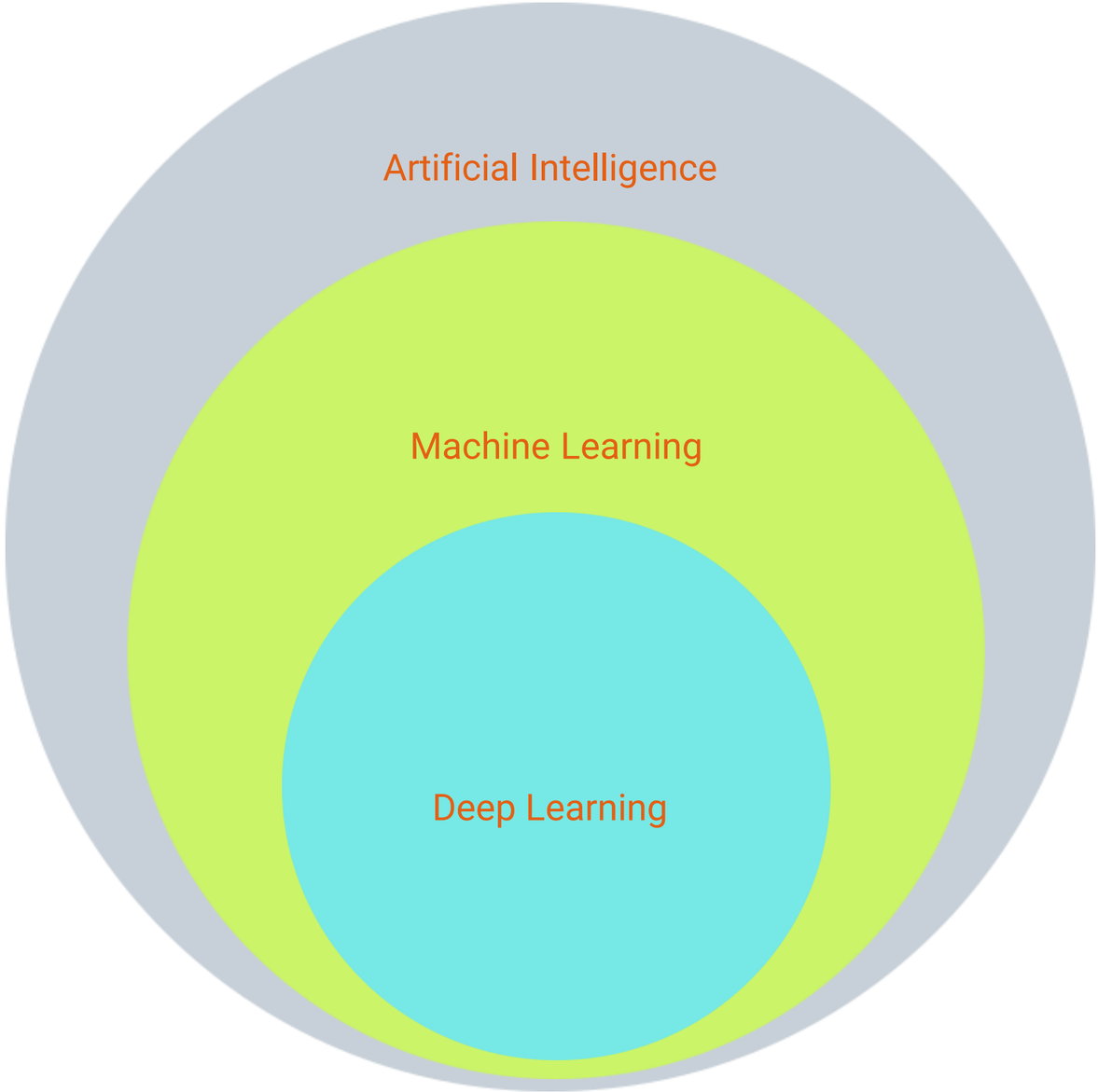

6 100% (6)


9 99% (9)


0 100% (0)


1 100% (1)

Models - Artificial Intelligence/Machine Learning/Deep Learning



Types of Artificial Intelligence

Artificial Intelligence : Simulation of human intelligence in machines

Based on Capabilities of the AI system

Still only here

Weak/Narrow AI

amazon alexa

ChatGPT

General AI

Super AI

Types of Machine Learning

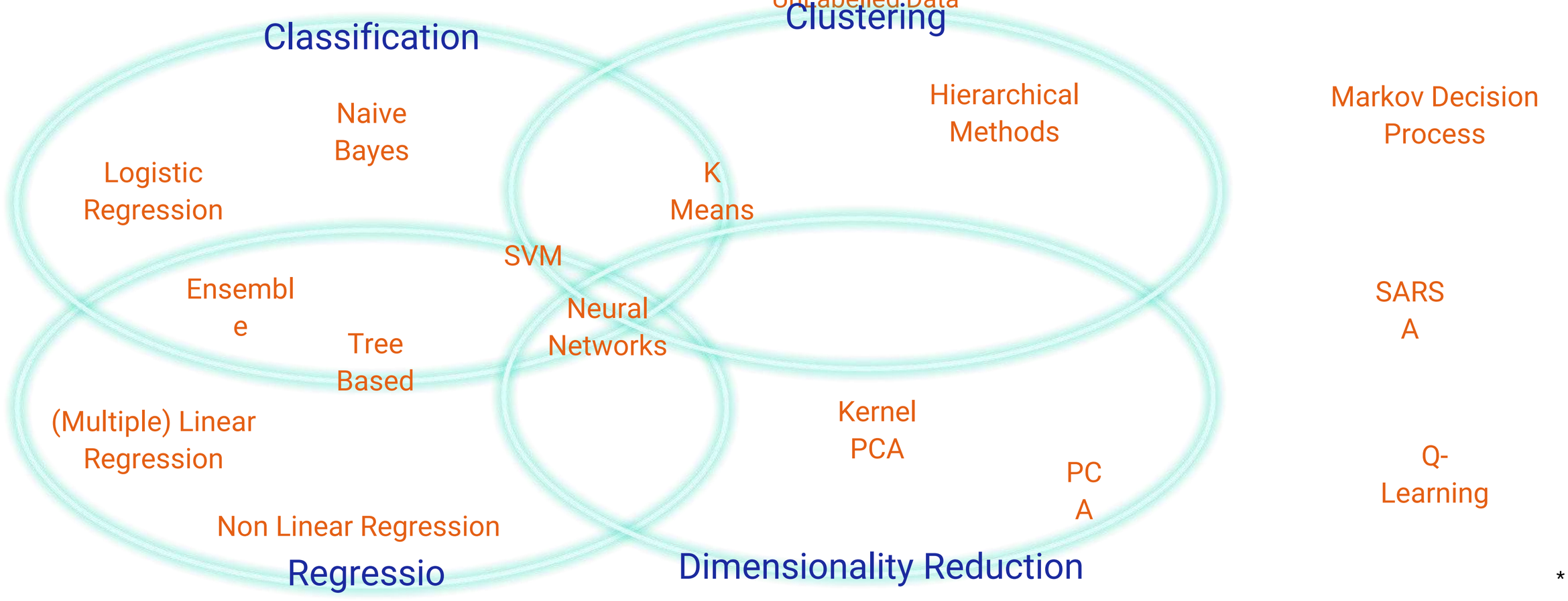
Machine Learning : Build systems learnt from data

Based on Capabilities of the AI system

Supervised Learning : Labelled data

Unsupervised Learning : UnLabelled Data

Reinforcement Learning : Reward Information



* MIT Course

Types of Deep Learning

Deep Learning : Based on artificial neural networks with feature learning

Based on types of neural networks

Feedforward Neural Networks

- Connections don't form a cycle
- Data with no temporal dependency

Recurrent Neural Networks

Connections don't form a cycle

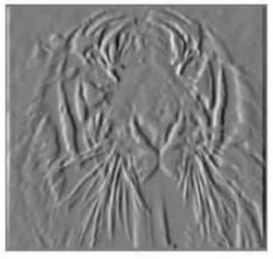
- Has loops allowing information to persist
- Time series data or language data

Autoencoders

- For unsupervised learning tasks, especially dimenionality reduction and feature learning
- For complex, high dimensional data

Convolutional Neural Networks

- Identify inherent structure in data using convolutions
- Data with spatial hierarchy



Long Short Term Memory Networks (LSTM)

- Special RNN, long term dependencies
- Sequential data with long term temporal dependency

Generative Adversarial Networks

- Competing networks, generator and discriminator
- For generating highly realistic synthetic data, like images from DALLE

Applications of AI/ML/DL

Lets brainstorm use cases

AI/ML Series Structure

Topic 2 - Regression

- Simple Linear Regression
- Multiple Linear Regression
- Extensions of Linear model
- Case study - Maketing plan
- Case study - Advertising
- Lab

Topic 3 - Classification

- Logistic Regression
- Linear Discriminant Analysis
- Application of Classification
- Case study - Stock Market
- Lab

Topic 3 - Classification

- Logistic Regression
- Linear Discriminant Analysis
- Application of Classification
- Case study - Stock Market
- Lab

Topic 4 - Resampling

- Cross Validation
- Bootstrapping
- Applications of resampling
- Case study - Stock Market
- Lab

Topic 5 - Linear Model Selection

- Subset selection
- Shrinkage methods
- Dimension Reduction Methods
- Case - Hitters baseball data

Topic 6 - Tree based methods

- Decision Tree basics
- Bagging
- Random Forests
- Lab - Classification Trees (Carseats data)
- Lab : Regression Trees (Boston data)

Topic 7 - Deep Learning

- Single Layer NN
- MultiLayer NN
- CNN
- RNN
- LSTM
- Applications of Deep Learning
- Lab : Hitters, IMDB data

Topic 8 - Large Language Model

- Cross Validation
- Bootstrapping
- Applications of resampling
- Case study - Stock Market
- Lab : Prompt Engineering
- Lab : Instruction Fine Tuning
- Lab : RAG approach

Books

Primary Reference Book :
Introduction to Statistical
Learning with Python/R