BEGINNER TO INTERMEDIATE

PYTHON

AND MACHINE LEARNING

WORKSHOP

WORKSHOP 2: DATA PREPARATION & MODEL TRAINING

BY JOSE RUIZ

intel HACKER DOJO

SPONSORED BY

BEFORE WE START...

AGENDA
Data preparation and preprocessing.
Data exploration and visualization.
Python toolset for analyzing data.
Deep learning model training and evaluation.
Overview of deep learning frameworks.
Training an Image Classification model.

IMPORTANCE OF DATA PREPARATION

Why & How?

- The quality of the training data can significantly impact the performance of the model.
- It involves cleaning, transforming and organizing the data.
- It also helps identify errors and inconsistencies in the dataset.

What happens if I don't do it?

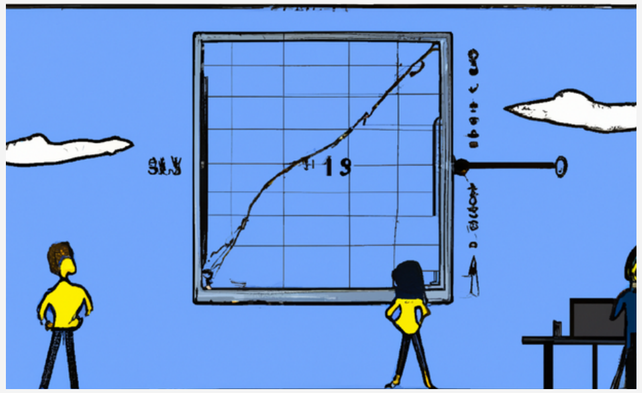
Poor data preparation can result in inaccurate or biased models.

CLEANING AND PREPROCESSING DATA

What to do?

- Remove duplicates and missing values.
- Identify and deal with outliers and anomalies (depends on each scenario)
- Handle categorical data, you may need to change the format or data encoding.
- Feature scaling and normalization.
- Data augmentation techniques.

DATA NORMALIZATION AND SCALING



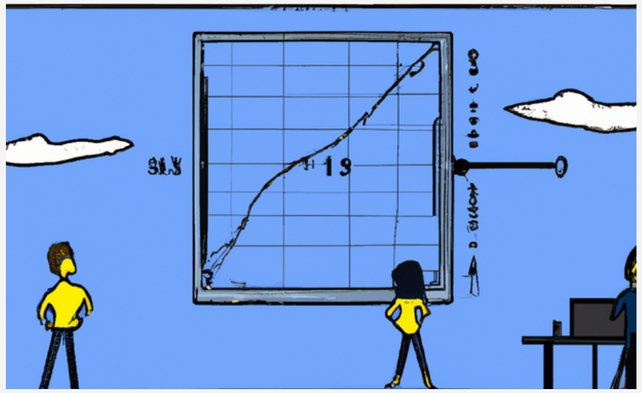
What is it?

- Normalization means scaling numerical data between 0 and 1.
- Scaling involves transforming data so that it has a mean of 0 and a standard deviation of 1.

Why should I do it?

- Helps to prevent any one feature from having a disproportionate impact on the model.
- Can help with convergence and training time for certain models.
- Some models require data normalization.

DATA EXPLORATION AND VISUALIZATION



What is it?

- Exploration: Understanding the data, identifying patterns and relationships and finding anomalies.
- Visualization: Visually represent complex information in an intuitive way.

Why should I do it?

- Understanding the distribution and range of the data.
- Identifying relationships and correlations between features.
- Detect outliers and anomalies.
- Discover patterns and trends.

TOOLS FOR DATA EXPLORATION AND VISUALIZATION

Pandas

- Data manipulation and analysis.
- Provides tools for cleaning, merging, filtering and visualizing.
- Supports various formats including CSV, Excel and SQL.
- Offers functions for data exploration, descriptive statistics and data aggregation.

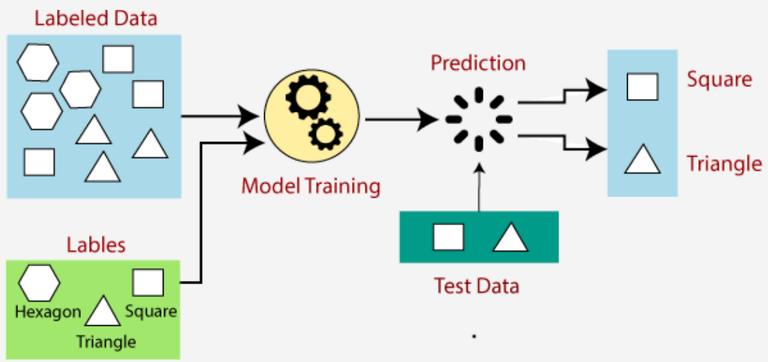
NumPy

- A fundamental package for scientific computing in Python.
- Provides support for multi-dimensional arrays and matrices.
- Offers a large library of mathematical functions for array operations.
- Efficiently handles numerical computations and data processing tasks.

Matplotlib

- A 2D plotting library for Python.
- Provides a wide range of visualization options including line plots, scatter plots, histograms, bar charts, and more.
- Offers customizable options for labels, titles, axes, and legends.

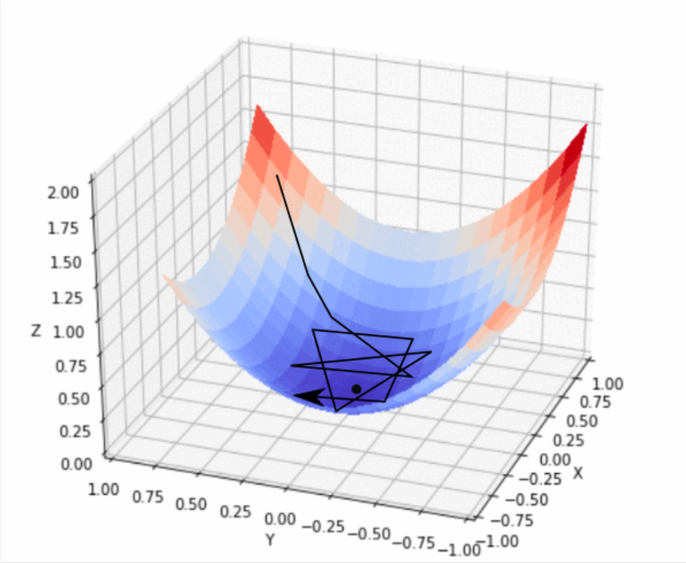
MODEL TRAINING



What is it?

- Model training is the process of teaching a machine learning model to make predictions based on input data.
- It involves feeding the model with labeled training data, allowing it to learn patterns and relationships between the input features and the corresponding output labels.
- The model uses an optimization algorithm to adjust the weights and biases of its internal parameters in order to minimize the difference between its predicted outputs and the true labels in the training data.
- The goal of training is to create a model that can generalize well to new, unseen data, and accurately predict the correct outputs for new input examples.

OPTIMIZER



What is it?

- Iteratively updates the model parameters to minimize the loss function.
- Calculates the gradient of the loss function with respect to the model parameters.
- Utilizes a learning rate that controls the step size taken in the direction of the negative gradient.
- Can be first-order (such as stochastic gradient descent) or second-order (such as Newton's method).
- May include additional features such as momentum, adaptive learning rates, or weight decay to improve performance.

MECHANICS OF TRAINING A MODEL

- **Data Splitting**
  - Split into training, validation and test sets.
- **Model Architecture**
  - Define the neural network architecture.
  - Choose the number of layers and neurons.
  - Select the activation functions.
- **Model Compilation**
  - Compile the model with a loss function and an optimizer.
  - Loss function measures the difference between predicted and ground truth.
  - Optimizer updates the weights in the model.
- **Model Training**
  - Feeding the training data to the model
  - Backpropagating the error through the neural network
  - Updating the weights in the neural network
- **Model Evaluation**
  - Evaluating the performance of the model on the validation set
  - Fine-tuning the hyperparameters based on the evaluation results
- **Model Testing**
  - Evaluating the final performance of the model on the test set
  - Making predictions on new, unseen data.

EXAMPLES

docker run --rm --gpus=all -it -p 8888:8888 -p 6006:6006 --name workshop2 hdpythonista/workshop2

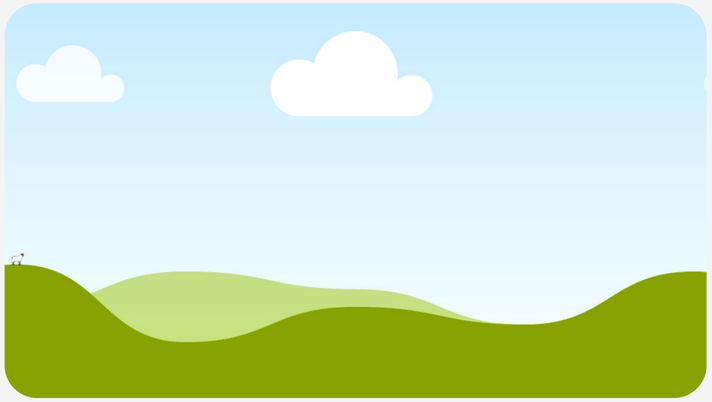
THANKYOU!

JOSE RUIZ

intel HACKER DOJO

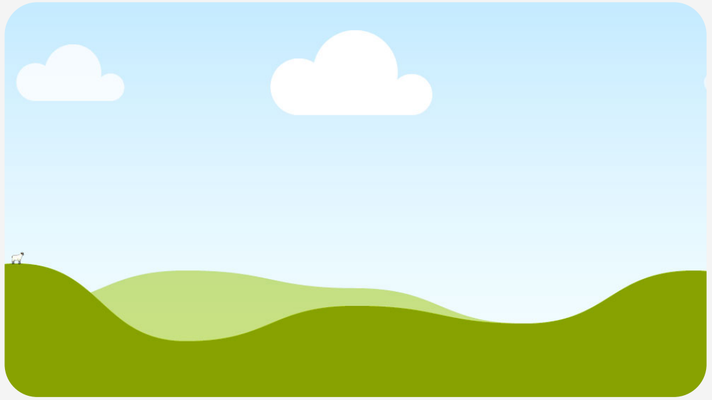
SPONSORED BY

HEADER



HEADER

text text text



HEADER

text text text