Introduction to Python and Deep Learning

Workshop 1: Introduction to Python, Deep Learning and Openvino

HACKER DOJO

Session 1: Introduction to Python, Deep Learning and OpenVINO

- About this series of workshops.
- Introduction to Machine Learning and Deep Learning.
- Explaining the different types of tasks that deep learning can be used for (classification, detection, segmentation, etc.).
- Why Python is a popular language for machine learning and deep learning, and an overview of popular libraries and frameworks, including TensorFlow, PyTorch, and OpenCV.
- Setting up your development environment. Overview of Docker and its advantages for development environments.
- An overview of OpenVINO and how it can help optimize deep learning models for different hardware architectures.

Intro

Workshop 1: Introduction to Python, Deep Learning and OpenVINO

- Introduction to deep learning and its applications.
- Why Python is a popular language for machine learning and deep learning.
- Overview of Docker and its advantages for development environments.
- Explaining the different types of tasks that deep learning can be used for (classification, detection, segmentation, etc.).
- An overview of OpenVINO and how it can help optimize deep learning models for different hardware architectures.

Data

Workshop 2: Data Preparation and Model Training

- The importance of data preparation and cleaning.
- How to load and preprocess data for deep learning tasks.
- Data visualization and exploration.
- Exploring the basics of model training, including loss functions and optimization algorithms.
- Building a simple neural network for image classification.
- Training the model on a dataset and evaluate its performance.

Classification

Workshop 3: Building Classification Models with OpenVINO

- Introduction to image classification and its applications.
- Building a simple image classification model using a pre-trained model.
- Converting the model to the OpenVINO format and optimize it for different hardware architectures.
- Deploying the optimized model and measure its performance.

Detection

Workshop 4: Building Object Detection Models with OpenVINO

- Introduction to object detection and its applications.
- Building a simple object detection model using a pre-trained model.
- Converting the model to the OpenVINO format and optimize it for different hardware architectures.
- Deploying the optimized model and measuring its performance.

Segmentation

Workshop 5: Building Image Segmentation Models with OpenVINO

- Introduction to image segmentation and its applications.
- Build a simple image segmentation model using a pre-trained model.
- Convert the model to the OpenVINO format and optimize it for different hardware architectures.
- Deploy the optimized model and measure its performance.

Deployment

Workshop 6: Model Conversion and Deployment with OpenVINO

- Introduce the concept of model conversion and deployment
- Discuss the different formats of deep learning models and how to convert between them
- Show how to deploy models on different hardware architectures using OpenVINO
- Introduce techniques such as quantization to optimize models for low-power devices

Generative Models

Workshop 7: Generative Models with OpenVINO

- Introduce the concept of generative models
- Discuss the different types of generative models, including generative adversarial networks (GANs) and variational autoencoders (VAEs)
- Build a simple generative model using TensorFlow or PyTorch
- Convert the model to the OpenVINO format and optimize it for different hardware architectures
- Deploy the optimized model and measure its performance

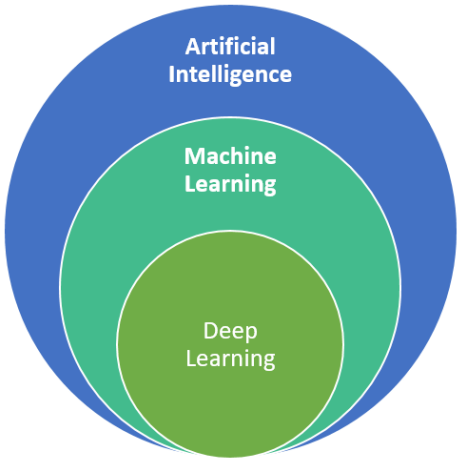
Advanced Topics

Workshop 8: Advanced Topics and Model Optimization with OpenVINO

- Discuss advanced topics such as neural architecture search, pruning, and knowledge distillation.
- Provide an overview of the available tools for model optimization with OpenVINO.

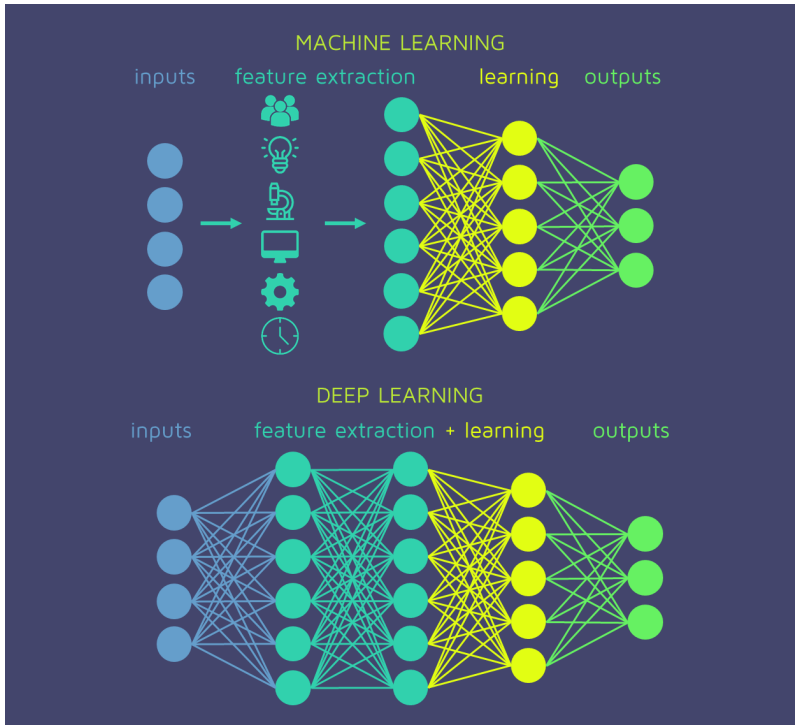
Workshop 1: Introduction to Python, Deep Learning and OpenVINO

What is Machine Learning?



- Machine learning is a subset of artificial intelligence that involves training algorithms to recognize patterns and make predictions based on data.
- The key idea behind machine learning is to enable computers to learn from data rather than being explicitly programmed.
- In other words, we provide the machine with examples of what we want it to do and it figures out how to do it on its own.
- Machine learning is a broad field with many different types of algorithms and approaches, including supervised learning, unsupervised learning, and reinforcement learning.
- Some common applications of machine learning include image and speech recognition, natural language processing, and predictive modeling.

What is Deep Learning?



- Deep Learning is a subfield of Machine Learning that involves the use of neural networks with many layers.
- Neural networks are designed to mimic the way the human brain processes information, by building layers of interconnected nodes that can recognize patterns and make decisions based on those patterns.
- Deep Learning algorithms are capable of handling very large and complex datasets, and can learn to identify patterns in data that are not easily recognized by humans or traditional machine learning algorithms.
- Deep Learning has many applications, including computer vision, speech recognition, natural language processing, and more.

Traditional Programming Vs Machine Learning

- In traditional programming, the programmer writes rules that specify the input-output relationship of the program.
- In machine learning, the program learns from the data to make predictions or decisions.
- Traditional programming is suitable for tasks with well-defined rules and limited complexity.
- Machine learning is suitable for tasks where the rules are difficult to define, or the problem has high complexity or variability.
- Machine learning involves data preprocessing, model selection and training, and evaluation.
- Deep learning requires a lot of data, computation, and time for training, but can achieve high accuracy and generalize well to new data.

Challenges in Deep Learning

- Limited availability of data.
- Model overfitting and underfitting.
- Challenges in model optimization and hyperparameter tuning.
- Ethical and social considerations in deep learning applications.

Real-world applications

- Computer vision applications (image and video recognition, segmentation, etc.)
- Natural Language Processing (NLP) applications (speech recognition, language translation, etc.)
- Robotics applications (autonomous vehicles, drones, etc.)

Image Classification

- Image classification is a task of assigning a label or a class to an image.
- In deep learning, image classification is achieved using a neural network that has been trained on a large dataset of images.
- The neural network learns to recognize patterns and features in the images that are associated with different labels or classes.
- The trained model can then be used to classify new images by comparing the patterns and features in the new image to those learned by the model during training.

Object Detection

- Object detection is a computer vision task that involves identifying objects in an image or video and locating them with a bounding box.
- It is a more complex task than image classification, as it requires not only identifying the objects but also their locations within the image.
- Object detection can be used in a variety of applications, such as autonomous driving, surveillance, and robotics.
- Deep learning models like Faster R-CNN and YOLO (You Only Look Once) are commonly used for object detection tasks, as they can provide accurate and efficient results.
- OpenVINO provides a variety of pre-trained models for object detection, which can be optimized for different hardware architectures.

Semantic and Instance Segmentation

- In image analysis, segmentation refers to the process of dividing an image into multiple segments, where each segment represents a meaningful part of the image.
- In semantic segmentation, each pixel in an image is assigned a label that corresponds to a particular object or class. This means that every pixel in the same object is given the same label.
- For instance segmentation, each pixel in an image is assigned a unique label that corresponds to a particular object instance. This means that every pixel within the same object is given a different label than pixels in other objects.
- For example, in a street scene image, a car may be labeled as one object, and each pixel within the car's boundaries may be labeled as belonging to that car instance.
- Semantic segmentation is useful for understanding the objects and their locations in an image, while instance segmentation is useful for understanding the objects' exact boundaries and separating them from each other.

Generative Models

- Generative models are a type of deep learning model that can generate new data that is similar to the original data it was trained on.
- They learn to model the probability distribution of the original data, and then generate new samples from that distribution.
- Examples of generative models include Variational Autoencoders (VAEs), Generative Adversarial Networks (GANs), and Autoregressive models.
- Applications of generative models include image synthesis, data augmentation, and text generation.

Python

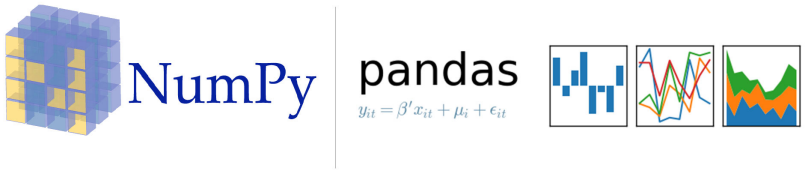
Why Python for Deep Learning?

- Python is a high-level programming language that is easy to learn and use.
- Python has a large and active community of developers who have built a rich ecosystem of libraries and tools for scientific computing, including machine learning and deep learning.
- Some of the most popular libraries and frameworks for data science and deep learning in Python include:
  - TensorFlow
  - PyTorch
  - Keras
  - Pandas
  - Numpy
  - OpenCV
- Python's simple and intuitive syntax allows developers to quickly prototype and experiment with different models and techniques.
- Python's flexibility and ease of use make it a popular choice for both academic research and industry applications.

Libraries and Frameworks for Deep Learning

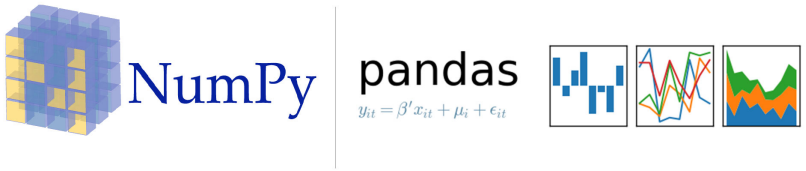
- TensorFlow: A popular open-source platform for building and deploying machine learning models.
- PyTorch: Another open-source machine learning framework that is widely used in research and industry.
- Keras: A user-friendly deep learning library that is built on top of TensorFlow and allows for rapid prototyping.

Libraries and Frameworks for Data Manipulation



- NumPy: A library for numerical computing in Python, providing support for arrays and matrices as well as mathematical functions.
- Pandas: A library for data manipulation and analysis, providing data structures for working with tabular data.

Jupyter Notebooks

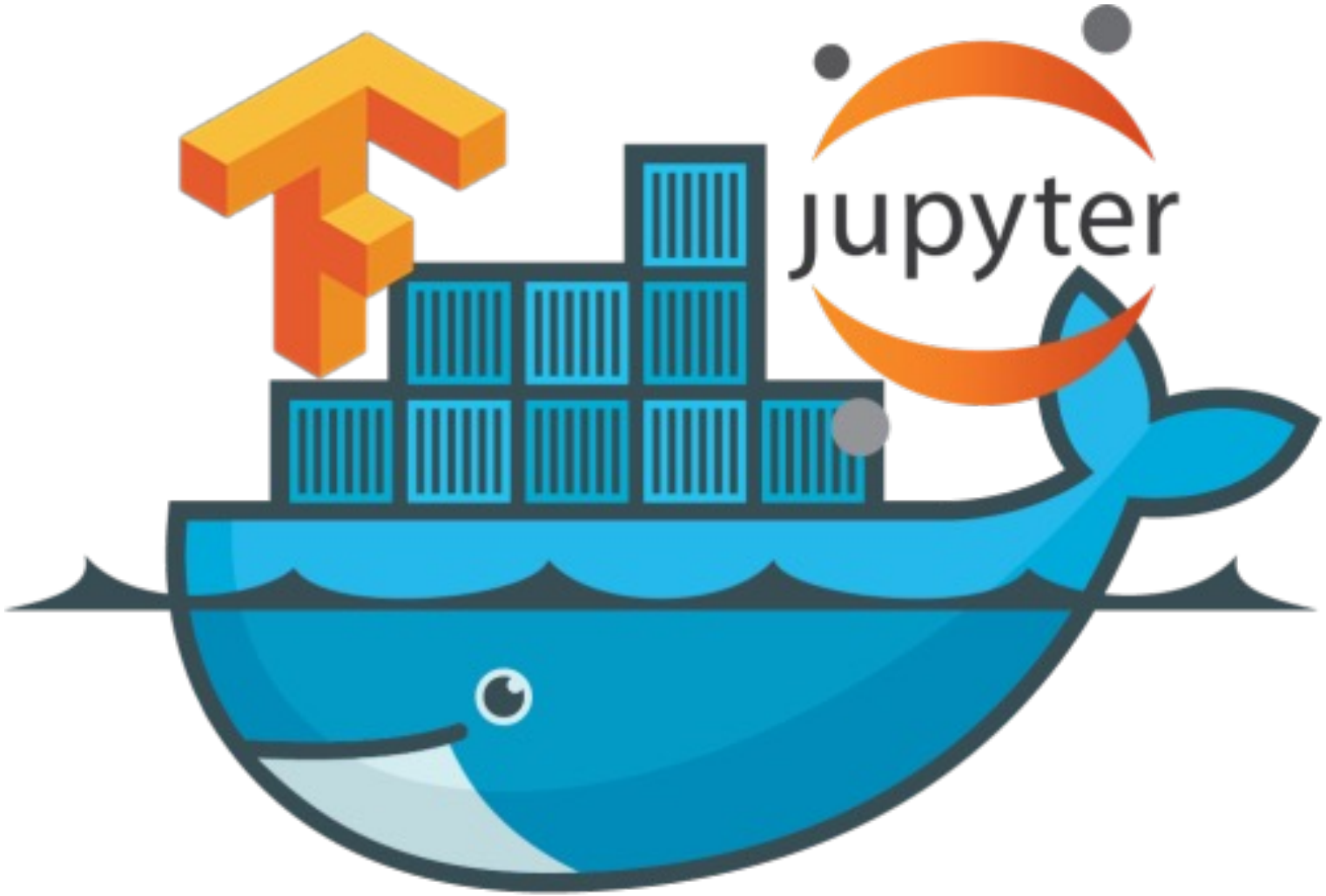


- Jupyter Notebooks are web-based interactive computational environments for creating and sharing documents that contain live code, equations, visualizations, and narrative text.
- It allows for easy experimentation, prototyping, and collaboration in Python and other languages.
- Features include code execution, inline documentation, visualizations, and collaboration tools.
- Jupyter Notebooks are commonly used for data analysis, scientific computing, and machine learning.
- Popular machine learning libraries such as TensorFlow, PyTorch and OpenVINO provide pre-built Jupyter Notebooks for users to get started with.

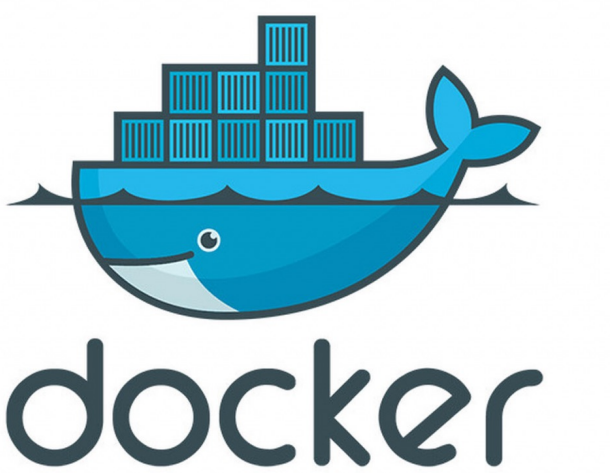
Getting Started with Python

- Python documentation: The official Python documentation provides detailed explanations of the language syntax and built-in functions.
- Codecademy: Codecademy offers a free interactive Python course for beginners.
- Learn Python the Hard Way: This online book provides a comprehensive introduction to Python with a focus on practical exercises.
- Coursera: Coursera offers many Python courses, including some from top universities such as the University of Michigan and Rice University.

Development Environment Setup

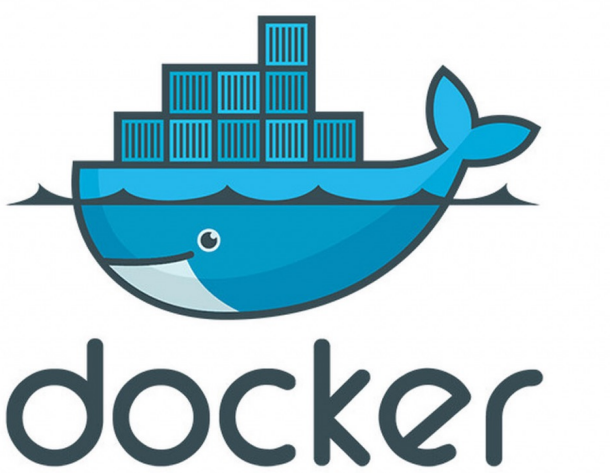


Local Setup using Docker



- Docker is a platform that allows you to run applications in isolated containers
- Containers are lightweight and portable, making them an ideal solution for managing development environments
- We will be providing a Docker image that contains all the necessary dependencies for this workshop
- This will ensure that everyone is working with the same environment, eliminating any potential setup issues

Installing Docker



- Go to the Docker website and download Docker Desktop for your operating system.
- Once the download is complete, run the installer and follow the installation wizard to install Docker Desktop.
- After installation, open Docker Desktop and verify that it's running correctly by checking the status in the taskbar or system tray.
- Test Docker by running the command `docker run hello-world` in a terminal or command prompt. If it runs successfully, Docker is installed and running correctly.

Remote Setup

- In addition to a local setup using Docker, there are also remote setups available that support Jupyter notebooks.
- Two popular examples are Google Colab and Kaggle.
- Google Colab is a free service that provides a Jupyter notebook environment with access to GPUs and TPUs for machine learning tasks.
- Kaggle is a data science community platform that also provides a Jupyter notebook environment for machine learning, as well as access to datasets, competitions, and tutorials.
- Both services allow you to run Python code and visualize results directly in the browser, making it easy to get started with machine learning and deep learning without having to worry about local hardware requirements.

Running the example (locally)

- Local setup:

```
docker run -it --rm -p 8888:8888 tensorflow/tensorflow:latest-jupyter
```