AI Hardware Ultimate Notes - SHUBHAM + YASH

TinyML Notes

ML Models – very common in day to day life today

Training and running tests on ML Models – computationally expensive

Need of computing systems that are fast enough to handle it. Thus most models are run on huge data centres with clusters of CPUs, GPUs and even TPUs in some case.

CPU – Central Processing Unit

GPU – Graphics Processing Unit - specifically designed to accelerate computer graphics workloads - used for gaming, video edition/creation, ML models

TPU – Tensor Processing Unit - an AI accelerator application-specific integrated circuit (ASIC) developed by Google used to accelerate machine learning workloads.

TinyML is a field of study in Machine Learning and Embedded Systems that explores the types of models you can run on small, low-powered devices like microcontrollers.

It enables low-latency, low power and low bandwidth model inference at edge devices.

Standard CPUs consume between 65 watts and 85 watts

Standard GPU consumes anywhere between 200 watts to 500 watts

A typical microcontroller consumes power in the order of milliwatts or microwatts.

Thousand times less power consumption!

This low power consumption enables the TinyML devices to run unplugged on batteries for weeks, months, and in some cases, even years, while running ML applications on edge.

Advantages of TinyML

1. **Low Latency:** Since the model runs on the edge, the data doesn't have to be sent to a server to run inference. This reduces the latency of the output.
2. **Low Power Consumption:** As we discussed before, microcontrollers consume very little power. This enables them to run without being charged for a really long time.
3. **Low Bandwidth:** As the data doesn't have to be sent to the server constantly, less internet bandwidth is used.
4. **Privacy:** Since the model is running on the edge, your data is not stored in any servers.

Hardware:

1. The **Arduino Nano 33 BLE Sense** is the suggested hardware for deploying Machine Learning models on edge. It contains a **32-bit ARM Cortex-M4F** microcontroller running at 64MHz with 1MB of program memory and 256KB RAM. This microcontroller provides enough horsepower to run TinyML models. The Arduino Nano 33 BLE Sense also contains colour, brightness, proximity, gesture, motion, vibration, orientation,

temperature, humidity, and pressure sensors. It also contains a digital microphone and a Bluetooth low energy(BLE) module. This sensor suite will be more than enough for most applications.

2. **Raspberry Pi 3 or 4**: 1-2 GB Memory, RaspberryPi OS, camera access

https://www.javatpoint.com/microprocessor-vs-microcontroller

2. **Machine Learning Framework: TensorFlow Lite** is the most popular and has the most community support. Using TensorFlow Lite Micro, we can deploy models on microcontrollers.

TFLite Quantization

https://medium.com/sclable/model-quantization-using-tensorflow-lite-2fe6a171a90d

TFLite Inferencing

https://medium.com/analytics-vidhya/tensorflow-lite-model-inferencing-fast-and-lean-c59a19d02daa

https://www.geeksforgeeks.org/difference-between-von-neumann-and-harvard-architecture/

https://www.tutorialspoint.com/difference-between-microprocessor-and-microcontroller

R Notes

https://www.javatpoint.com/r-advantages-and-disadvantages

https://www.javatpoint.com/r-data-visualization

https://www.geeksforgeeks.org/data-visualization-in-r/#:~:text=Data%20visualization%20is%20the%20technique,make%20better%20decisions%20regarding%20it.

https://www.tutorialspoint.com/r/r_data_types.htm

Data visualization is the technique used to deliver insights in data using visual cues such as graphs, charts, maps, and many others. This is useful as it helps in intuitive and easy understanding of the large quantities of data and thereby make better decisions regarding it.

Datatypes in R:

Unlike other programming languages, variables are assigned to objects rather than data types in R. There are many types of R-objects. The frequently used ones are −

- Vectors
- Lists
- Matrices
- Arrays
- Factors
- Data Frames

Libraries we have used:

1. Dplyr - For data frame manipulation
2. Lubridate - For Date manipulation
3. Ggplot2
4. Scales - for easily manipulation the labels and scales of different graphs
5. Tidyr - For cleaning up of data
6. Pacman - Package Manager for easily downloading packages

Plots we have used:

1. # Trips vs Hour
2. # Trips vs Hours (stacked Months)
3. # Trips vs Date
4. # Trips vs Day of week (Dodge)
5. # Trips vs Month
6. Heat Map of Hours and Day
7. Heat Map of Month and Day
8. Heat Map of Month and Day of Week
9. Scatter plot on NY's Map using Longitude and Latitude

Inference:

1. Number of booking increase substantially after 6pm

2. Number of booking in NY is significantly in Septempter because the weather in NY is pleasant for tourists in Sepetember (No garmi no thand)
3. Morning mei bookings ki frequency kam h

Power BI Notes

https://towardsdatascience.com/how-to-visualize-data-using-power-bi-9ec1413e976e

https://data-flair.training/blogs/power-bi-advantages-and-disadvantages/

https://www.guru99.com/tableau-vs-power-bi-difference.html

Business Intelligence: Business intelligence (BI) refers to **the procedural and technical infrastructure that collects, stores, and analyzes the data produced by a company's activities**.

Data analytics: Analytics is a technique of converting raw facts and figures into some particular actions by analyzing those raw data evaluations and perceptions in the context of organizational problem-solving and also with the decision making.

Data Visulization: Data visualization is the practice of translating information into a visual context, such as a map or graph, to make data easier for the human brain to understand and pull insights from.

https://www.geeksforgeeks.org/difference-between-data-analytics-and-data-analysis/

1. Slicer for Year (single select off)
2. Slicer for selecting crypto
3. Card for Key Performance Indicators like Market Cap, volume, high, low, average
4. Line graph for marketcap, high and low
5. Prediction on the marketcap of a crypto using PowerBI Forcasting (exponential smoothing)

// Can leave

BI Reports that we can make in power BI:

1. Sales Analytics
2. Financial Analytics
3. HR
4. Website
5. Digital Marketing
6. Customer Profitability

//

Visualization types in Power BI:

1. Slicer
2. Area chart
3. Bar graph
4. Cards: Multi row cards display one or more data points, one per row.
5. Pie Chart
6. Donut Chart

Apache Spark Notes

https://medium.datadriveninvestor.com/distributed-data-processing-with-apache-spark-2a5e473b0cb1

Apache Spark is an open-source unified analytics engine for large-scale distributed data processing.

5 components of Spark:

1. Core Spark: Responsible for RDDs and core processing

2. SparkSQL: For higher-level tools and structured data processing
3. MLlib: For machine learning
4. Spark Streaming: For real time data streaming
5. GraphX: For graph processing

RDD stands for Resilient Distribution Datasets. It is a fault-tolerant collection of parallel running operational elements. The partitioned data of RDD is distributed and immutable.

Apache Hadoop is a collection of open-source software utilities that facilitates using a network of many computers to solve problems involving massive amounts of data and computation.

*Spark is generally **faster than Hadoop**. This is because Hadoop writes intermediate results to disk (that is, lots of **I/O operations**) whereas Spark tries to keep intermediate results in memory (that is, **in-memory computation**) whenever possible.*

*Spark offers **lazy evaluation** of operations and **optimizes them just before the final result**; Sparks maintains a series of transformations that are to be performed without actually performing those operations unless we try to obtain the results. This way, Spark is able to find the best path looking at overall transformations required*

*The **Hadoop ecosystem** includes a distributed file storage system called **HDFS** (Hadoop Distributed File System). **Spark**, on the other hand, does not include a file storage system. You can use Spark on top of HDFS but you do not have to.*

Features of Spark:

1. Fast Processing:Spark uses RDDs which saves a lot of time in reading and writing of data and hence they are 10x or 100x faster than other solutions

2. In memory computing: Data being loaded into memory only when we want to read the data.Otherwise it is kept in RAM.
3. Flexible: Supports multiple languages SCALA JAVA R Python
4. Fault tolerant: Spark supports fault tolerance using RDD. Spark RDDs are the abstractions designed to handle failures of worker nodes which ensures zero data loss.
5. BetterAnalytics: Spark has a rich set of SQL queries, ML Libraries and other analytic tools

Devops Notes

DevOps defines an agile relationship between operations and Development. It is a process that is practiced by the development team and operational engineers together from beginning to the final stage of the product.

OR

Development and operations both play essential roles in order to deliver applications. The deployment comprises analyzing the requirements, designing, developing, and testing of the software components or frameworks.

The operation consists of the administrative processes, services, and support for the software. When both the development and operations are combined with collaborating, then the DevOps architecture is the solution to fix the gap between deployment and operation terms; therefore, delivery can be faster.

Components of DevOp architecture:

1. Plan - Plan the work to be done between the dev and op teams so that productivity increase
2. Code
3. Build
4. Test
5. Deploy: Host the app on cloud
6. Operate: Work on the app over cloud and find and fix errors

7. Monitor: To identify any risk of failure
8. Release:

https://www.javatpoint.com/devops-architecture

Devops Lifecycles (DIT-DO-MF)

1. Continuous Development: This phase involves the planning and coding of the software.
2. Continuous Integration: This stage is the heart of the entire DevOps lifecycle. It is a software development practice in which the developers require to commit changes to the source code more frequently. This may be on a daily or weekly basis. Then every commit is built, and this allows early detection of problems if they are present.
3. Continuous Testing: This phase, where the developed software is continuously testing for bugs.
4. Continuous Monitoring: Monitoring is a phase that involves watching all the operational factors of the entire DevOps process to find out trends and identify problem areas.
5. Continuous Feedback: The application development is consistently improved by analyzing the results from the operations of the software. This is carried out by placing the critical phase of constant feedback between the operations and the development of the next version of the current software application.
6. Continuous Deployment: In this phase, the code is deployed to the production servers. Also, it is essential to ensure that the code is correctly used on all the servers.
7. Continuous Operations: All DevOps operations are based on the continuity with complete automation of the release process and allow the organization to accelerate the overall time to market continuingly.

Ansible: Ansible is a leading DevOps tool. Ansible is an open-source IT engine that automates application deployment, cloud provisioning, intra service orchestration, and other IT tools.

Docker: Docker is a high-end DevOps tool that allows building, ship, and run distributed applications on multiple systems. It also helps to assemble the apps quickly from the components, and it is typically suitable for container management.

Nagios: Nagios is one of the more useful tools for DevOps. It can determine the errors and rectify them with the help of network, infrastructure, server, and log monitoring systems.

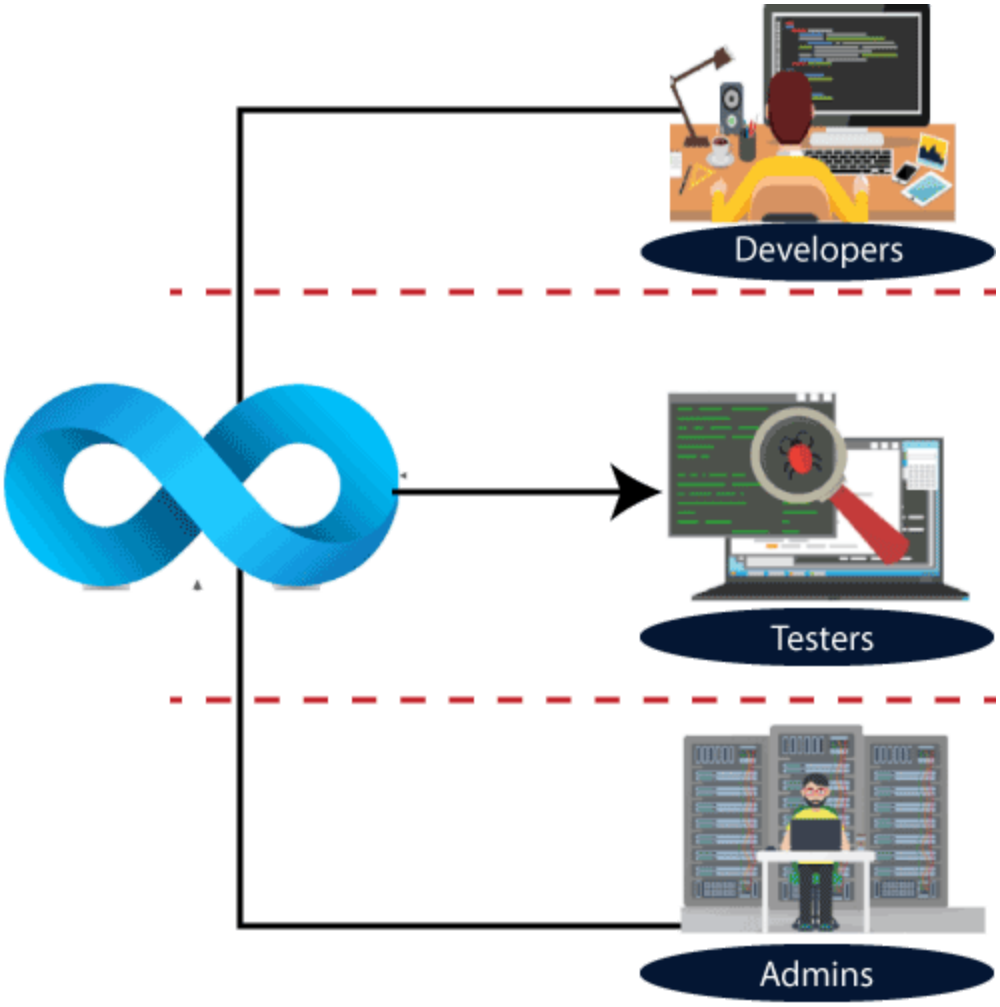
Jenkins: Jenkins is a DevOps tool for monitoring the execution of repeated tasks. Jenkins is a software that allows continuous integration. Jenkins will be installed on a server where the central build will take place. It helps to integrate project changes more efficiently by finding the issues quickly.

Git: Git is an open-source distributed version control system that is freely available for everyone. It is designed to handle minor to major projects with speed and efficiency.

DevOps Engineers - DevOps Engineer is an IT professional who works with system operators, software developers, and other production IT staff to administer code releases.

DevOps engineer understands the software development lifecycle and various automation tools for developing digital pipelines.

Collaborates with development, testing, and operations teams.



DevOps Engineers occasionally code from scratch, and they must have the basics of software development languages.

DevOps Engineer Roles and Responsibilities

1. Manage projects effectively through an open standard based platform.
2. Increases project visibility through traceability.
3. Improve quality and reduce the development cost with collaboration.
4. DevOps should have the soft skill of problem solver and a quick learner.
5. Analyze, design, and evaluate automation scripts and systems.

6. Able to perform system troubleshooting and problem-solving across the platform and application domains.
7. Ensuring the critical resolution of system issues by using the best cloud security solution services.

https://www.javatpoint.com/devops-pipeline-and-methodology
(methodology not important)

Difference between Devops and Agile methodology -

The DevOps is a combination of two words, one is software Development, and second is Operations. This allows a single team to handle the entire application lifecycle, from **development** to **testing, deployment**, and **operations**. DevOps helps you to reduce the disconnection between software developers, quality assurance (QA) engineers, and system administrators.

The Agile involves continuous iteration of development and testing in the **SDLC** process. Both development and testing activities are concurrent, unlike the waterfall model. This software development method emphasizes on incremental, iterative, and evolutionary development.

This is the environment that's on your computer. Here is where you'll do all of your code updates. It's where all of your commits and branches live along with those of your co-workers. The development environment is usually configured differently from the environment that users work in.

The stage environment is as similar to the production environment as it can be. You'll have all of the code on a server this time instead of a local machine. It'll connect to as many services as it can without touching the production environment.
All of the hard core testing happens here.

Every time you talk about making your project live, this is the environment you are talking about. The production environment is where users access the final code after all of the updates and testing.

GUIDELINES FOR EXAM

1. Prepare difference between technologies mentioned in the course:
    a. Tableau vs Power BI
    b. Apache vs Hadoop
    c. R vs Python (https://towardsdatascience.com/python-vs-r-for-data-science-cf2699dfff4b)