13. What are the features of TensorFlow?

**1. Open Source**

* TensorFlow is free and open source, backed by Google.
* Large community support and extensive documentation.

**2. Cross-Platform**

* Runs on **CPUs, GPUs, and TPUs**.
* Supports **desktop, server, mobile, and embedded devices**.

**3. Flexible Architecture**

* Can run computations on **single devices** or **distributed systems**.
* Allows building **complex neural networks** and custom models.

**4. High-Level APIs**

* Provides **Keras** for easy model building.
* Simplifies tasks like **training, evaluation, and prediction**.

**5. Computational Graphs**

* Uses **dataflow graphs** for computations, optimizing execution.
* Supports both **eager execution** (imperative programming) and **graph mode**.

**6. Scalable**

* Can scale from **small experiments on a laptop** to **large distributed training** in the cloud.

**7. Visualization Support**

* Integrated with **TensorBoard** to visualize metrics, graphs, and embeddings.

**8. Pre-trained Models & Libraries**

* Offers **TensorFlow Hub**, **TensorFlow Lite**, and **TensorFlow Extended (TFX)** for production pipelines.
* Includes pre-trained models for **image, text, and speech tasks**.

**9. Production Ready**

* Can deploy models to **servers, web (TensorFlow.js), mobile devices (TensorFlow Lite), and edge devi**

Top of Form

Bottom of Form

**14. List a few limitations of TensorFlow.**

**1. Steep Learning Curve**

* **TensorFlow can be complex for beginners, especially when using low-level APIs.**
* **Debugging computational graphs can be challenging.**

**2. Verbose Syntax**

* **Compared to some libraries like PyTorch, TensorFlow code can be more verbose and harder to read.**

**3. Dynamic Computation Limitations**

* **Early versions relied on static computation graphs, making it less intuitive for dynamic models.**
* **Although Eager Execution improves this, some flexibility is still limited compared to PyTorch.**

**4. Large Binary Size**

* **TensorFlow libraries can be large, increasing memory usage, which may be an issue for mobile or embedded deployment.**

**5. Performance Overhead**

* **For smaller tasks, TensorFlow may have slower execution due to graph compilation overhead.**

**6. Limited Debugging Tools**

* **While TensorBoard is great for visualization, debugging complex models can still be difficult.**

**7. Rapid Changes**

* **Frequent updates and API changes can cause compatibility issues between ve**

**15. What do you know about supervised and unsupervised machine learning?**

**1. Supervised Learning**

* **Definition: The model is trained on a dataset that includes input features (X) and corresponding output labels (Y).**
* **Goal: Learn the mapping from input to output so it can predict new unseen data.**
* **Key Idea: “Learning with a teacher” – the correct answers are provided.**

**Examples:**

* **Predicting house prices based on features like size, location → Regression**
* **Classifying emails as spam or not → Classification**

**Common Algorithms:**

* **Linear Regression**
* **Logistic Regression**
* **Decision Trees / Random Forests**
* **Support Vector Machines (SVM)**
* **Neural Networks**

**When to Use:**

* **When you have labeled data.**

**2. Unsupervised Learning**

* **Definition: The model is trained on a dataset that includes only input features (X), with no labels.**
* **Goal: Identify patterns, structures, or clusters in the data.**
* **Key Idea: “Learning without a teacher” – the model tries to make sense of the data on its own.**

**Examples:**

* **Customer segmentation for marketing → Clustering**
* **Reducing the number of features → Dimensionality Reduction (PCA, t-SNE)**
* **Anomaly detection (detecting fraud in transactions)**

**Common Algorithms:**

* **K-Means Clustering**
* **Hierarchical Clustering**
* **DBSCAN**
* **Principal Component Analysis (PCA)**

**When to Use:**

* **When you have unlabeled data and want to explore patterns or grouping**

| **Feature** |  |
| --- | --- |
|  |  | |  |