

PAULINE ONYANGO

Part 1: Theoretical Understanding (40%)

1. Short Answer Questions

Q1: Explain the primary differences between TensorFlow and PyTorch. When would you choose one over the other?

TensorFlow and PyTorch are open-source deep learning frameworks that are also popular with the community; however, they have a number of differences:

Feature	TensorFlow	PyTorch
Execution Style	Static graph (TensorFlow 1.x), eager mode with TF 2.x	Dynamic computation graph (eager execution)
Ease of Debugging	More complex due to static graph	Easier due to Pythonic, step-by-step debugging
Deployment	Better support via TensorFlow Lite, TF Serving	Deployment options exist but require more setup
Ecosystem	Larger ecosystem (TFLite, TensorBoard, TF Hub)	Simpler ecosystem focused on research

When to choose:

- TensorFlow: Basically the same advantages as Pytorch (production-scale, model deployment (particularly mobile/edge), etc), but has better support for TensorBoard visualizations.
- PyTorch: Libraries favored in academic research and prototyping because of being flexible as well as Python-like in nature.

Q2: Describe two use cases for Jupyter Notebooks in AI development.

1. Interactive Model Prototyping: Jupyter enables developers to develop, test, experiment with architecture or parameters, cell by cell, providing quick iteration on models.
2. Data Exploration and Visualization: Its application is mainly to visualize data with libraries such as Matplotlib or Seaborn and enable practitioners to dialog with the data distributions, patterns, and outliers before training a model.

Q3: How does spaCy enhance NLP tasks compared to basic Python string operations?

The basic string operations lack sophisticated linguistic features and machine-learning-based models of text analysis which are offered by spaCy. Specifically:

- Named Entity Recognition (NER): Identifies people, brands, locations in text, which is not feasible with `str.split()` or `re`.
- Part-of-Speech (POS) Tagging: spaCy annotates tokens with grammar roles, aiding in syntactic parsing.
- Efficiency and Accuracy: spaCy is optimized in Cython for speed and is more accurate than hand-coded regex or simple Python logic.

2. Comparative Analysis

Scikit-learn vs. TensorFlow

Category	Scikit-learn	TensorFlow
Target Applications	Classical ML (e.g., SVM, Random Forest, KNN)	Deep Learning (CNNs, RNNs, transformers)
Model Complexity	Shallow models, quick to train	Suitable for complex neural network models
Ease of Use	Very beginner-friendly; few lines of code needed	More complex API, requires deeper ML knowledge
Deployment	Limited built-in deployment tools	Rich deployment options (TFLite, TF Serving)
Community Support	Strong and mature community	Massive community and enterprise backing
Integration	Works well with pandas, NumPy, matplotlib	Integrates with Keras, TensorBoard, TFLite

Summary:

- Select Scikit-learn to use it with speedy experimentations and traditional ML applications.
- Select TensorFlow when you need the neural network and high production scale in deep learning applications.