DL part short summary:

The code works with the UCI HAR Dataset, which contains accelerometer and gyroscope readings (nine channels: three for total acceleration, three for body acceleration, and three for angular velocity).
The readings are divided into fixed-size windows (128 timesteps per window).
StandardScaler is used to normalize these signals, and LabelEncoder encodes the activity labels (e.g., WALKING, SITTING, etc.) into numeric classes.
(I have kept the architecture very simple, which is described below)
LSTM Model:

- Uses a two-layer LSTM to capture temporal dependencies in the sensor data (with a hidden dimension of 64).
- After processing the sequence, the final hidden state is passed to a fully connected layer to output class predictions.

1D CNN Model:

- Applies two 1D convolutional layers (with ReLU activations and max pooling) across the time dimension.
- Flattens the final feature maps and passes them through a fully connected layer to produce class predictions.

Both models are trained with the same procedure:

- CrossEntropyLoss is the loss function.
- Adam optimizer is used with a learning rate of 0.001.
- Each model is trained for 200 epochs.

The training loss for both models decreases significantly over epochs, indicating effective learning.

Performance & Observations

- **Final Accuracies**:
  - LSTM: ~91.45%
  - 1D CNN: ~92.13%
  - The 1D CNN shows a slightly higher final accuracy than the LSTM.
- **Confusion Matrices**:
  - Both models generally classify most activities correctly, but some off-diagonal elements indicate confusion between certain activities.
  - Common confusions often appear between SITTING vs. STANDING or STANDING vs. LAYING, which can be similar in low motion.

Overall, the LSTM and 1D CNN approaches handle the temporal sensor data well, with the CNN achieving a slightly higher accuracy in this particular setup.

ML part short summary:

The code loads the raw inertial signals (accelerometer and gyroscope) for each axis (X, Y, Z) from the train and test sets. Each sample is 128 readings (2.56 seconds at 50 Hz) and has 9 channels (3 total accelerations, 3 body accelerations, 3 gyroscope signals).
Corresponding activity labels (1–6) are also loaded.

Feature Extraction

- **TSFEL-Extracted Features**: The raw signals are fed into the *TSFEL* library (Time Series Feature Extraction Library) to automatically compute various time and frequency domain features for each sample.
- **Provided Features**: The dataset also comes with 561 pre-computed features (in `X_train.txt` and `X_test.txt`) derived by the original dataset authors. These are used for comparison.

Three classifiers are used: **Random Forest**, **Support Vector Machine (SVM)**, and **Logistic Regression**.
Each classifier is wrapped in a pipeline with a **StandardScaler** to normalize the features.
The models are trained on the training portion of the data, and accuracy is measured on the test set. A classification report is also displayed (precision, recall, f1-score).
Performance is compared between the TSFEL-extracted features and the author-provided features.
The code outputs the accuracy and a detailed classification report for each model. A confusion matrix (for the Random Forest with provided features) is plotted to visualize class-by-class performance.

Key Observations

- The TSFEL-extracted and provided feature sets achieved substantial results across all three models, with **SVM** and **Logistic Regression** performing best (over 95% accuracy).
- The TSFEL features slightly outperform the provided SVM and Logistic Regression features in this particular setup.
- Random Forest performs well with both feature sets, though its accuracy is slightly lower than the other two models.
- The confusion matrix (for Random Forest with provided features) confirms that certain activities (e.g., classes 1 and 6) are classified almost perfectly. In contrast, classes involving similar body postures or transitions (like 3 vs. 4) can be slightly more challenging to distinguish.

Overall, the code demonstrates how automated feature extraction (TSFEL) can yield results comparable to carefully engineered features and how standard machine learning models can be used to classify human activities from smartphone sensor data with high accuracy.