Projektaufgabe: Fine-Tuning eines Sprachmodells für die Sentiment-Analyse von Finanznachrichten

Zielsetzung

In diesem Projekt soll ein kompaktes Sprachmodell von Hugging Face für die Sentiment-Analyse von Texten aus dem Finanzbereich trainiert werden. Das Ziel ist es, ein Modell zu entwickeln, das Sätze aus Finanznachrichten als positiv, negativ oder neutral klassifizieren kann.

Modell und Datensatz

- **Modell:** Es wird **DistilRoBERTa** (`distilroberta-base`) verwendet, eine leichtere und schnellere Version des RoBERTa-Modells, die sich gut für das Training auf einer CPU eignet.
- **Datensatz:** Das Training erfolgt auf dem `financial_phrasebank`-Datensatz. Dieser enthält Sätze aus Finanznachrichten, die bereits entsprechend ihres Sentiments klassifiziert wurden.
- **Literatur:** https://huggingface.co/learn/llm-course/en/chapter3/1

Schritt 1: Projekteinrichtung und Installation

Zuerst müssen die notwendigen Python-Bibliotheken installiert werden. Für dieses Projekt sind `transformers`, `datasets` und `torch` erforderlich.

Schritt 2: Laden des Datensatzes

Laden Sie den `financial_phrasebank`-Datensatz mithilfe der `datasets`-Bibliothek. Teilen Sie den Datensatz anschließend in ein Trainings- und ein Test-Set auf.

Schritt 3: Auswahl von Modell und Tokenizer

Laden Sie das `distilroberta-base`-Modell und den dazugehörigen Tokenizer. Das Modell wird für eine Klassifizierungsaufgabe mit drei Labels (positiv, negativ, neutral) initialisiert.

Schritt 4: Datenvorverarbeitung (Tokenisierung)

Die Textdaten müssen in ein numerisches Format umgewandelt werden, das das Modell versteht. Erstellen Sie eine Funktion zur Tokenisierung und wenden Sie diese auf die Datensätze an.

Schritt 5: Fine-Tuning des Modells

Dies ist der zentrale Schritt, in dem das Modelltraining stattfindet.

Schritt 6: Evaluierung des Modells

Nach dem Training bewerten Sie die Leistung des Modells auf dem Test-Set. Hierzu wird die Genauigkeit (Accuracy) als Metrik verwendet.

Abgabe:
Reichen Sie Ihren vollständigen und lauffähigen Python-Code (als Jupyter-Notebook oder .py-Datei) ein. Fügen Sie eine kurze Dokumentation hinzu, die die Ergebnisse der Evaluierung (Schritt 6) enthält und beschreibt, welche Herausforderungen während des Projekts aufgetreten sind.