INVOCA

Machine Learning Engineer Interview

Model Training - Take Home Assignment

Last edit: April 2025

Interview Time: 1 hour (Estimated Prep Time: ~6-8 hours, including research, Training Time: ~2-4 hours)

Overview

At Invoca, we process vast amounts of data, including conversational data like call transcripts. Efficient and accurate machine learning models are crucial. This exercise simulates a common challenge: optimizing a model for deployment by balancing performance and efficiency. We need you to build and evaluate an improved training pipeline using Knowledge Distillation to create a smaller, faster classification model suitable for production environments. While we use the AG News dataset for this exercise, the techniques are applicable to tasks like classifying call transcript topics or sentiment.

Dataset

The AG News Articles dataset is an open source collection of news article headlines from over 2000 sources. This dataset contains 120,000 examples in the training set and 7,600 in the test set. This dataset contains labels based on the type of the article.

- **World**
- **Sports**
- **Business**
- **Sci/Tech**

```
# %pip install datasets transformers torch # or tensorflow

from datasets import load_dataset
ag_news_dataset = load_dataset("ag_news")
# You can access splits like: train_data = ag_news_dataset['train']
```

Note: Ensure your environment has necessary libraries like `datasets`, `transformers`, `torch` or `tensorflow`)

Assignment: Building an Efficient Classifier

Your goal is to improve the performance of a relatively small text classifier (`distilbert-base-uncased`, the "student") on the AG News task, going beyond its standard fine-tuning performance. To achieve this, you should leverage a larger, more powerful model (`bert-base-uncased`, the "teacher") that has already been fine-tuned for this task. Techniques related to **Knowledge Distillation** or **Knowledge Transfer** are often employed for such scenarios.

1. **Models & Setup:**
    a. **Teacher Model:** Use the provided `bert-base-uncased` model fine-tuned on AG News available on HuggingFace: `fabriceyhc/bert-base-uncased-ag_news`. This model's weights should remain frozen.
    b. **Student Model:** `distilbert-base-uncased`.
    c. **Tokenizer:** Use the tokenizer associated with `bert-base-uncased`.
2. **Baseline Performance:**
    a. First, establish the baseline performance. Fine-tune the `distilbert-base-uncased` (Student) model directly on the AG News training set using standard supervised learning methods and evaluate it on the test set. Train for a few epochs (e.g., 2-3). This baseline is crucial for comparison.
3. **Enhanced Training Pipeline:**
    a. **Task Definition**
        i. Develop and implement an enhanced training pipeline for the student model (`distilbert-base-uncased`) that utilizes the frozen teacher model (`bert-base-uncased`) as an additional source of information during training. Your objective is to significantly improve the student model's performance compared to its baseline.
        ii. **Strategy: Research and choose an appropriate strategy** based on Knowledge Distillation or Knowledge Transfer principles to achieve this.
            1. Feature based
            2. Similarity based
            3. Response based (soft logits with temperature scaling)
            4. Discreet output based
        iii. **Optional:** Use a different architecture with teacher and student model (e.g. BERT based teacher and decoder only student, or vice-versa, etc.) to utilize white-box distillation techniques. You can choose underlying baseline model/s of your choice.

   b. Train the student model using your implemented enhanced pipeline for a similar number of epochs as the baseline.

4. **Evaluation & Comparison:**
   a. Evaluate the performance (Accuracy, F1-Score, and potentially other relevant metrics) of your **Enhanced Student** model on the AG News test set.
   b. Compare the parameter counts of the Teacher and Student models.
   c. Prepare a comparison:
      i. Teacher Performance (obtain from its Hugging Face page or run inference)
      ii. Your Enhanced Student Performance
      iii. Your Baseline Student Performance

Deliverables and Discussion

During the interview please be prepared to share your screen, walk through your code and results, and discuss:

1. **Training Strategy:** Describe the methodology you researched and implemented to improve the student model using the teacher. What principles or specific techniques of Knowledge Distillation) did you apply? Explain the core mechanics of your implementation and justify your design choices.
2. **Implementation Details:** Guide us through the key code sections related to your knowledge transfer mechanism and training loop. What were the most significant implementation challenges, and how did you overcome them?
3. **Model Tuning:** Discuss the key hyperparameters relevant to *your chosen method* and how you selected their values (including standard ones like learning rate, epochs).
4. **Results Interpretation:** Analyze your comparison table. How successful was your strategy in improving the student model beyond its baseline? How did it compare to the teacher? Discuss the resulting trade-offs between performance and model efficiency.
5. **Pipeline & Code Quality:** Discuss the structure, clarity, and potential reusability of your training pipeline code.
6. **Conceptual Alternatives:** What other approaches exist for transferring knowledge between models or improving model efficiency? What did you learn from selecting and implementing your strategy? What are its limitations? What further experiments would you run?

INVOCA

Important Notes:

- **Focus:** We are most interested in your ability to research, design, implement, and evaluate a non-trivial training strategy to solve the stated problem. Your justification for your chosen method and your analysis are key.
- **Time Estimate:** We estimate **6-8 hours of focused effort**, plus variable model training time (**approximately 2-4 hours**). Focus on a sound implementation and analysis over exhaustive tuning. Training for just 2-3 epochs per model is sufficient.
- **Hardware Resources:** This interview process should be achievable using resources like **Google Colab** or a personal computer with a modern CPU or modest GPU.
- **Submission:** Please share a link to your code in a Github repo or notebook prior to the interview, Please include any necessary setup instructions.