Nikhil Kapila

nkapila.me | github | nkapila6@gatech.edu

Project Proposal

Gemma Model Fine-tuning UI

For the project, I would like to help develop a user-friendly web interface using Gradio/Streamlit to fine tune Gemma models.

Previous relevant experience

I previously developed and deployed a **Gradio** application to an **HuggingFace Space** to compare electricity usage data across various buildings. The goal was to conduct a qualitative analysis of how energy consumption patterns change in response to different building upgrades. This analysis was complemented by predictive modeling using LSTMs for forecasting energy usage, and anomaly detection techniques using KNNs.

You can view a presentation of this work here: https://youtu.be/rGKhB1wkei4

I have good experience with **PyTorch**. I wrote a pre-print [1] where we augmented a deep ResNet with attention layers. I worked on the Pipeline and used **Skorch** alongside **MLFlow** to be able to track experiments. Wrote most of the attention augment model layers as well.

GitHub repo: https://github.com/AttentionSeekers/CNNtention

Approach

In the below paragraph, I outline my approach and thinking on how to solve the problem statement at hand.

- **UI**:
  - **Framework**: User interface creation using **Gradio/Streamlit**. If something more advanced is desirable then there are always projects like fastHTML that allow you to build great user interfaces in Python: https://www.fastht.ml/
  - **Interactive plots**: Similar to my previous project, I am thinking to use **Plotly** to have interactive graphs.
  - **Hyperparam selection**: Different sliders to choose different hyperparams. Mostly batch_size, epochs, learning rate, possibly Optimizer?
  - **MLflow**: Possible use of MLFlow so user can see progress across runs?
- **Dataset Uploading**: This should be fairly straightforward. Only problem I foresee is memory usage. Maybe we set a limit? Or use a library like **Polars** that can handle memory well.
  - **Augmentation**: Different options if modality image. Possibly the user can specify different transforms to choose from. Text augmentation could be looked at but I haven't seen it to be common.
- **Training progress Visualization**: Should be easy to do using Plotly. Covered in UI section.
- **Model download**: Spit out a .pth file once user is satisfied with training progress.
- **Train at scale**: This would need time to explore. How do I execute code on a VM? Write a bash script of .py files?

- **Documentation**: Not a problem, it’s my forte! My personal website and even mlrose-ky docs are written by me. It’s basically a static site generator that uses markdown to make web pages. Very simple! Even this project seems promising: https://pdoc.dev/

References

[1] Nikhil Kapila, Julian Glattki, and Tejas Rathi, “CNNtention: Can CNNs learn better with Attention?,” Dec. 2024. [Online]. Available: https://arxiv.org/abs/2412.11657v3