Utilizing G-DIG For Model Training Improvement

Xingyuan Pan, Hangkai Qian, Yi Huang, Tianning Zhu

[Functions and Users]

We propose to develop a software tool leveraging the G-DIG algorithm, designed to diagnose poorly performing model predictions, particularly for new, knowledge-based data (model never seen before). The primary functionality will include tracing bad cases back to specific training data points that have significantly influenced these outcomes. Users of our tool will be machine learning engineers and AI researchers aiming to enhance the performance of the model on unseen data.

[Significance]

AI model training mostly deals with issues like misdiagnosing unpredicted forecasting results (bad cases). Conventional techniques are not up to the standard when it comes to identifying the exact training samples that lead to misclassifications or a drop in performance. Our tool G-DIG handles the issue systematically by finding the training samples that are influential, which the user has no clue about.

[Approach]

Python will be the primary language for this tool that we will develop and it will be a standalone application. This program will use the PyTorch library for deep learning functionalities and the main algorithm implemented in it will be the G-DIG algorithm. The features of this algorithm will include the user's ability to feed it with bad-case samples as input and get a ranked list of influential training samples in return. We are planning to make use of current open-source implementations and libraries like NumPy and PyTorch for data manipulation and preprocessing.

[Evaluation]

We will focus on a case study as the main evaluation method. This case study will be based on a real machine learning project and cover these steps:

1. Case selection: Select an LLM failure setting with typical errors (such as bias, noise, or non-stopping).
2. Problem diagnosis: Use the tool we developed to input the bad cases in the model to track and identify the top 10 influential training samples.
3. Manual verification: Our team manually reviews these influential samples to determine whether they have label errors, abnormal distributions, or information noise.

[Timeline]

- tentative

[Task Division]

- tentative