Project Proposal - Feature generation using transformers

I want to work on this project for the following reasons:

- Real-world applications: Sequence decision making is an essential task in many real-world applications, such as recommendation systems, chatbots, and autonomous vehicles. By generating useful features using Transformers, we can improve the performance of these systems and provide better user experiences.
- Efficiency: Transformers are designed to handle massive amounts of data efficiently, which makes them an excellent choice for feature generation in large-scale applications.
- Interpretability: Transformers can generate features that are not only effective but also interpretable. This can help in understanding the decision-making process and identifying the key factors that influence the outcome.
- Innovation: Feature engineering has been a critical task in machine learning, but traditional methods are often labour-intensive and domain-specific. Using Transformers for feature generation represents a new approach that has the potential to revolutionize the field.

Outline

Month 1

Week 1:

- Research and study feature engineering and Transformer architecture
- Read literature on feature generation using Transformers and identify potential applications and challenges
- Identify suitable datasets for testing the pipeline
- Develop a plan for the Python-based pipeline for feature generation using Transformers

Week 2:

- Develop a Python-based pipeline for feature generation using Transformers
- Implement the pipeline on the selected datasets
- Analyse the generated features and compare them with the raw features in terms of their effectiveness for sequence decision making
- Identify potential areas for improvement in the pipeline

Week 3:

- Learn how to use Vowpal Wabbit (VW) for online decision making
- Implement online decision-making using VW with the raw features and the Transformer-generated features

- Evaluate the performance of the Transformer-generated features in VW and compare them with the raw features
- Analyse the results and identify potential limitations and areas for improvement

Week 4:

- Explore the use of Transformer-generated features for off-policy evaluation
- Develop a simulation environment for off-policy evaluation and test the Transformer-generated features
- Analyse the results and compare them with the raw features
- Identify potential areas for improvement and future research directions

Month 2

Week 5:

- Based on the results from the previous month, improve the pipeline for feature generation using Transformers
- Test the improved pipeline on the same and/or new datasets
- Analyse the generated features and compare them with the previous results

Week 6:

- Learn about reinforcement learning and contextual bandits
- Implement reinforcement learning and contextual bandits using the raw and Transformer-generated features
- Analyse the results and identify potential areas for improvement

Week 7:

- Explore the use of adversarial training to improve the performance of the Transformer-generated features
- Implement adversarial training and test the performance of the improved features in VW and/or the simulation environment

Week 8:

- Investigate the interpretability of the Transformer-generated features and develop methods for visualizing and understanding them
- Analyze the interpretability of the features and compare them with the raw features

Month 3

Week 9:

- Test the performance of the Transformer-generated features on new datasets and in different applications

- Compare the results with the previous experiments and identify potential areas for improvement

Week 10:

- Explore the use of other deep learning architectures for feature generation and compare their performance with the Transformer-generated features
- Analyse the results and identify the strengths and weaknesses of each approach

Week 11:

- Write a report summarizing the project, including a discussion of the results, limitations, and potential future directions
- Create a presentation of the findings and prepare for a potential demo or presentation

Week 12:

- Finalize the report and presentation
- Practice presenting the findings and prepare for potential questions and feedback

Challenges:

- **Data pre-processing**: One of the major challenges is to pre-process the raw data to make it suitable for feature generation using Transformers. The data may have missing values, outliers, and inconsistencies that need to be addressed before feeding the data into the Transformer.
- **Hyperparameter tuning**: The performance of the Transformer-based feature generation pipeline may be sensitive to hyperparameters such as the number of layers, hidden size, and attention mechanism. Hyperparameter tuning may require extensive experimentation and can be computationally expensive.
- **Evaluating feature effectiveness**: Evaluating the effectiveness of the generated features for sequence decision making may be challenging. It may be difficult to quantify the improvement in performance due to the generated features, especially in real-world applications where the outcomes are not always clear.
- **Integrating with online decision-making frameworks**: Integrating the Transformer-generated features with online decision-making frameworks such as Vowpal Wabbit (VW) or reinforcement learning algorithms may require expertise in these frameworks. It may also require optimizing the pipeline for speed and scalability to handle large datasets in real-time.
- **Interpreting the generated features**: Transformers are known to generate highly complex and abstract features, which may be difficult to interpret. Developing methods to visualize and understand the generated features can be challenging, and there may be a trade-off between interpretability and performance.
- **Hardware limitations**: Transformers are computationally expensive and require specialized hardware such as GPUs or TPUs to train and generate features efficiently. Limited access to such hardware can be a bottleneck in the project and may require optimization strategies such as distributed computing or cloud computing.