GenAI: Phase 1

Objective:

The goal of this project is to introduce mentees to generative AI, focusing on text generation with models like transformers. Through this, mentees will gain practical experience in building a basic transformer model, and fine-tuning the pre-existing models like gpt-2 for creative writing tasks.

Project Structure:

The project will be divided into several stages that gradually build upon each other. Each stage will involve both theoretical and practical components, allowing mentees to understand both the underlying principles of generative AI and the hands-on experience of implementing them.

Project Breakdown:

1. Dataset Preparation (Week 1):

- **Objective:** Prepare a creative dataset for training the transformer model.
- **Dataset Choice:** mentees will select a creative category like poetry, short stories, recipes, etc. A dataset can be created by scraping publicly available resources, such as *poets.org* for poems or using other literary datasets.
- **Tasks:**
    - Scraping text data (e.g., poems, stories) from the chosen website.
    - Preprocessing the data (removing unnecessary punctuation, normalizing the text, etc.).
    - Tokenizing the dataset to convert text into numerical representations (word-level or subword-level tokenization).

2. Theory of Transformers (Week 1 - 2):

- **Objective:** Introduce mentees to the foundational concepts of transformers.
- **Resources:**
  The Illustrated Transformer – Jay Alammar – Visualizing machine learning one concept at a time.
  All you need to know about 'Attention' and 'Transformers' — In-depth Understanding — Part 1
  All you need to know about 'Attention' and 'Transformers' — In-depth Understanding — Part 2
  Transformer Neural Networks, ChatGPT's foundation, Clearly Explained!!! ***
  3B1B's series on LLMs

//skipped the 3rd point

3. Building a Basic Transformer (Week 2):

- **Objective:** mentees will implement a very basic version of a transformer model from scratch.
- **Tasks:**
    - **Breakdown of Components:**
        - **Self-Attention:** Code for the self-attention mechanism, explaining how the model weighs different parts of the input.
        - **Positional Encoding:** Implement the positional encoding to account for the order of words in the sequence.
        - **Encoder-Decoder Layers:** Implement the encoder and decoder layers, showing how information flows between them in the transformer architecture.
    - **Expected Output:**
        - A model that can generate the next word (or token) given a sequence of previous words, for tasks like text completion (with detailed comments on what each code block is doing)
        - **Voiceover:** Create a 10-minute voiceover explaining your code work, including concepts like self-attention, positional encoding, and multi-head attention, in groups of 3 or 4.
    - **Evaluation Metric:** Basic loss (cross-entropy loss) during training. The model's output can be evaluated qualitatively by generating a few text samples and seeing if they make sense contextually.
- **Resources:**
    Build your own Transformer from scratch using Pytorch
    Transformer Model from Scratch using TensorFlow - GeeksforGeeks

4. Fine-Tuning Pre-Trained GPT-2 Model (Week 3):

- **Objective:** Explore state-of-art pre-trained models, pipeline functions and fine-tune a pre-trained GPT-2 model on the creative dataset prepared earlier.
- **Tasks:**
    - **Pre-trained Model:** Use an existing GPT-2 model as a base.
    - **Fine-Tuning:** Train the model on the dataset to adapt it to the specific genre or type of creative writing chosen by the mentees.
    - **Model Evaluation:** Use metrics like BLEU scores or human evaluation to assess the quality of the generated text. The output should be coherent and contextually relevant to the dataset (e.g., a poem, story, or recipe).
    - **Deployment:** Upload the fine-tuned model to a repository like Hugging Face, experimenting with different prompts and generating text.
- **Resources:**
    First 4 chapters from the HuggingFace NLP course
    Fine tuning a gpt2 model for poetry generation

Assessment Criteria:

The project will be assessed based on:

1. **Dataset Quality:** How well the dataset was curated and prepared.

2. **Theoretical Understanding and model:** The depth of understanding of transformer models, as demonstrated in the voiceover video, and the code.
3. **Fine-Tuning Performance:** The improvement of the GPT-2 model after fine-tuning and the quality of the generated text.

Deliverables:

1. **Cleaned Dataset:** A prepared and tokenized dataset for training.
   //skipped the 2nd part
2. **Transformer Model Code with a voiceover explaining it:** Code for the basic transformer model including the self-attention mechanism, positional encoding, and encoder-decoder layers.
3. **Fine-Tuned GPT-2 Model:** The fine-tuned model, uploaded to Hugging Face or a similar platform.

Expected Outcomes:

- **Learning Outcomes:** mentees will gain hands-on experience with modern generative models like transformers and GPT-2. They will understand the underlying mechanisms of attention, encoding, and decoding in transformers.
- **Creativity and Innovation:** mentees will also get the opportunity to explore the creative potential of AI in generating text, experimenting with prompts and seeing how AI can assist in creative writing tasks.
- **Saving and uploading the trained model:** Uploading the fine-tuned models to platforms like Hugging Face encourages mentees to learn how to share their models with the broader AI community.

---

*by the end of this project, you should be able to create a chat-gpt of your own, and overtake OpenAI
Materials covered for track 1:

1) Dataset creation and cleansing
2) Tokenization and an intro to generative models
3) Data loading and pipeline functions
4) RNN, LSTM and GRU
5) word embeddings
6) Transformers
7) Track 1: assignment 1
8) Fine tuning pre-trained transformers

GenAI: Phase 2

Reference Youtube Playlist : Youtube

Project Breakdown:x

1. GANs Intro (Week 1):

- **Objective:** Learn the basics of the Generator and Discriminator part.
- **Tasks:**
  - Understand what the Generator and Discriminator does.
  - Understanding Binary Cross Entropy loss function and Minimax game with its loss function.
  - Understanding the training process of GANs in tensorflow.
- **Notebooks and Docs**
  - Session 1 on GANs : **Info on GANs**
  - Vanilla Old TensorFlow : Kaggle
  - Vanilla GANs New Tensorflow : Main_GANs.ipynb

2. Building FCGAN and DCGAN (Week 1 - 2):

- **Objective:** Assignment for deeper understanding of GANs with comparing performance of FCGAN and DCGAN.
- **Assignment** : GANs Assignment.ipynb

3. Learning Wasserstein GANs (Week 2):

- **Objective:** Understanding better loss function for GANs
- **Tasks:**
  - Implement wGANs
  - Docs - Additional Type of GANs
  - wGANs : wGANs.ipynb

4. Learning Conditional GANs (Week 3):

- **Objective:** Understanding way to control the output of GANs
- **Tasks:**
  - Implement cGANs.
  - Assignment on GANs.
  - Docs - Additional Type of GANs x
  - cGANs : cGANs.ipynb
  - cGANs : Kaggle

Assessment Criteria:

The project will be assessed based on:

4. **Assignments:** How well the assignments were submitted.
5. **Theoretical Understanding of GANs**

---

*by the end of this project, you should be able to understand and create GANs of your own and understand research papers related to GANs