Autoregressive Models: Multimodal Causal Transformer (Text and Image Generation)

Name: Frederik Stihler

Objective

This project focuses on training an autoregressive (AR) model capable of handling multimodal data, specifically text and image tokens. The model is designed to generate sequences in both modalities, conditioned on either modality or unconditionally. For example, the model can generate an image corresponding to the text prompt "dark green eight on bright cyan" or, conversely, produce descriptive text for a given image.

Transformer Architecture

Custom Implementation of Transformers

While prebuilt transformer implementations exist in popular frameworks such as PyTorch, this project focuses on building a transformer from the ground up. By constructing the model from primitives (e.g., Linear/Dense layers, LayerNorm, GeLU activation, and Embedding layers), we gain a deeper understanding of the architecture and its underlying mechanisms.

Key Components

- **Attention Mechanism:** Scaled dot-product attention forms the core of the transformer's ability to focus on specific parts of the input sequence.
- **Positional Encodings:** These are added to token embeddings to incorporate sequence information.
- **Layer Normalization:** Ensures stability in training by normalizing activations within layers.
- **GeLU Activation:** Nonlinearities introduced via the GeLU activation enhance model expressiveness.

This hands-on approach enables fine-grained control over the architecture and allows us to experiment with and customize various aspects of the model design.

Data Processing and Tokenization

Dataset

We use the text labeled colored MNIST dataset, which has a text description of the MNIST image.

Figure 1: Training Data

Text Tokens

- Each word in the text data is mapped to a unique token.
- All text descriptions are standardized to contain the same number of words, simplifying sequence processing.

Image Tokens

- Images are quantized into tokens using a VQVAE tokenizer.

Multimodal Batch Formulation

- Sequences are prepared in two orders for training:
  1. `<end of image>, text_tokens, <end of text>, image_tokens`
  2. `<end of text>, image_tokens, <end of image>, text_tokens`
- A 50/50 split is maintained between the two orderings within batches.

Special Tokens

- `<end of text>` and `<end of image>` tokens signal a modality switch during generation.
- A `<bos>` token is used as the initial conditioning token during sampling.

Model Architecture

Design Parameters

- **Layers:** 4
- **Hidden Dimension (`d_model`):** 128
- **Heads:** 4
- **Activation:** GeLU nonlinearities

Training Details

- **Epochs:** 30
- **Batch Size:** 32 (adjustable based on GPU memory)
- **Learning Rate:** 10^{-3}
- **Optimizer:** Adam

Inference and Sampling

Modality-Specific Sampling Rules

- After `<end of image>`: Only text tokens (including `<end of text>`) are allowed.
- After `<end of text>`: Only image tokens (including `<end of image>`) are allowed.
- At the start: Sampling is restricted to either `<end of image>` or `<end of text>`.

Error Mitigation

- The model is forced to generate a fixed number of image tokens (49) to ensure correct image generation length.

Results

Training and Evaluation

The average negative log-likelihood (nats/dim) is recorded for both training and test datasets.
Final test loss: 2.6772 nats / dim

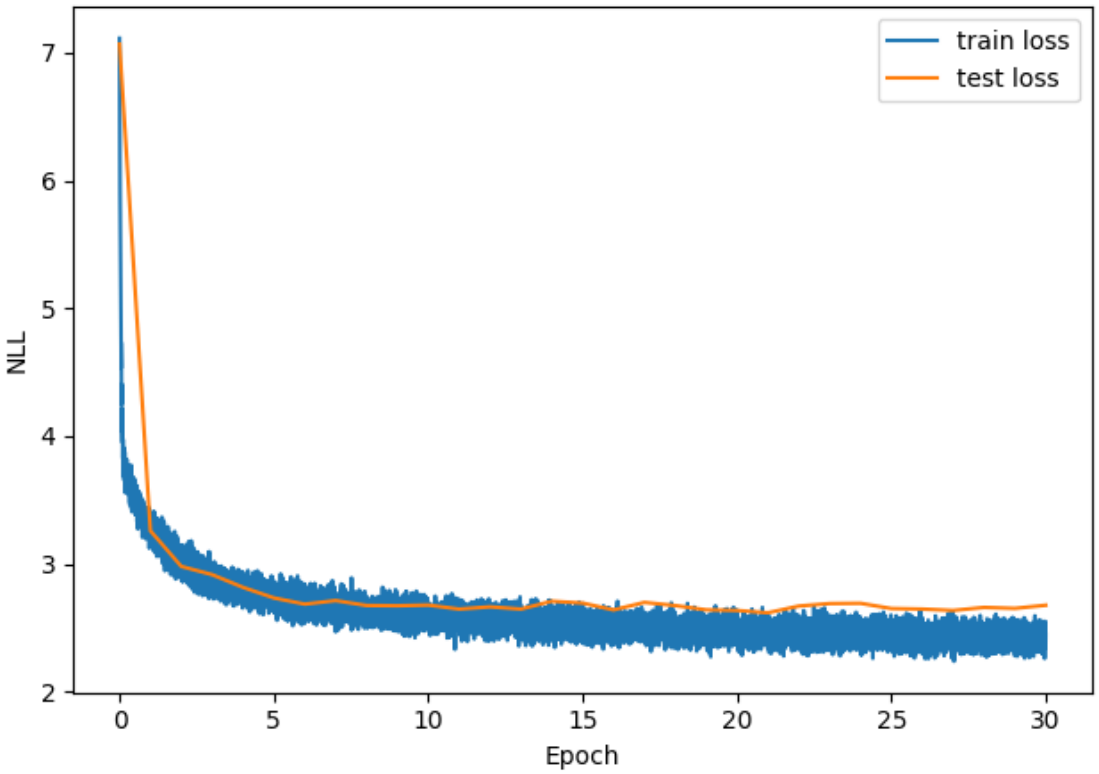

Figure 2: Training curve

Sample generation

We generate the following samples:

- *Text-Conditioned Generation:* 9 samples.
- *Image-Conditioned Generation:* 9 samples.
- *Unconditional Generation:* 9 samples, showcasing standalone text and image capabilities.

Figure 3: Text Conditioned Samples (= Images generated)

Figure 4: Image Conditioned Samples (= Text generated)

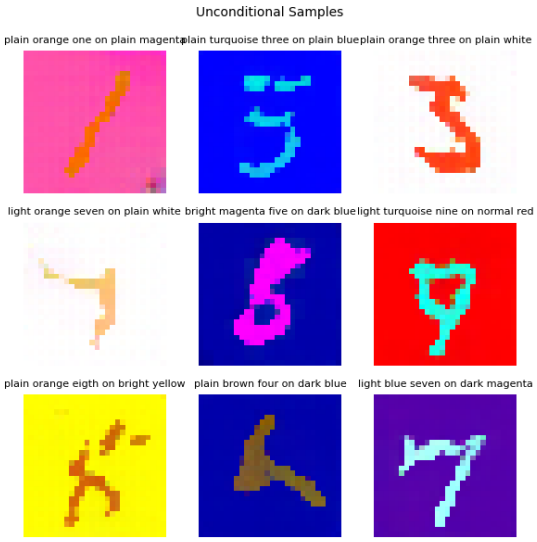

Figure 5: Unconditioned Samples (= Images and Text generated)

Conclusion

This comprehensive training pipeline demonstrates the model's ability to generate cohesive multimodal outputs, reflecting its robustness and versatility in handling text and image data.

We observe high-quality results in text-conditioned image generation, with generated digits matching the textual descriptions and being clear and readable. Similarly, image-conditioned text generation performs well, producing coherent and descriptive text outputs.

However, the quality of samples generated from unconditional sampling appears slightly lower. This could be attributed to the absence of explicit conditioning, making it more challenging for the model to anchor its generation process. Another possible explanation is that the model's training emphasizes conditioned tasks more heavily, potentially leading to less focus on the unconditional generation capability.