NATURAL LANGUAGE PROCESSING
INDIVIDUAL PROJECT REPORT

Bhanu Sai Praneeth Sarva

G46159306

INTRODUCTION

This project focuses on Image Caption Generation using CNN+GRU, an exciting interdisciplinary task that combines computer vision and natural language processing to generate meaningful descriptions for images. The project aims to build a robust pipeline capable of transforming image data into descriptive captions, leveraging state-of-the-art techniques like pretrained CNNs, encoder-decoder architectures, and attention mechanisms. The application of such systems spans diverse domains, including accessibility tools, automated image tagging, and content-based image retrieval.

PROJECT OVERVIEW

The central goal of this project is to develop a model that effectively integrates visual and textual data to produce accurate and contextually appropriate captions. This is achieved through a structured process involving data preprocessing, model building, training, and evaluation. Key components include:

1. **Data Understanding and Preparation:**

    a. **Dataset:** The Flickr8k dataset, containing 8,091 images, each paired with five descriptive captions, was used as the foundation. This provided a total of 40,455 image-caption pairs for training and evaluation.

b. **Preprocessing:** Images were resized to (299, 299) and normalized, while captions were tokenized, converted to lowercase, and padded. Rare words were replaced with "UNK" to ensure consistent vocabulary.

2. **Model Architecture:**

    a. **Encoder:** A pretrained **Inception-V3** CNN extracts features from images, providing a compressed representation of visual data.

    b. **Decoder:** A GRU-based decoder processes these features alongside textual embeddings, generating captions word by word.

    c. **Attention Mechanism:** This dynamically focuses on specific image regions, enabling the decoder to incorporate relevant visual context at each step.

3. **Model Training:**

    a. Hyperparameter optimization, loss function monitoring, and the use of teacher forcing stabilized and improved the training process.

    b. Techniques like masking and batching enhanced computational efficiency and model generalization.

4. **Evaluation:**

    a. **Greedy Search** was employed to generate captions iteratively, selecting the highest probability words at each step.
    b. The **BLEU Score** metric quantitatively assessed caption quality, with scores ranging between 50% and 70%, showcasing reasonable accuracy.

SHARED CONTRIBUTIONS

The project involved collaborative efforts distributed among team members, as outlined below:

- **Data Handling and Preprocessing:**
  - Tasks like cleaning captions, tokenization, and image normalization were shared to ensure high-quality inputs for model training.
- **Model Development:**
  - Encoder and decoder architectures were implemented collaboratively, with a focus on integrating attention mechanisms.
- **Model Training:**
  - The team collectively tuned hyperparameters, monitored loss, and applied optimization strategies to enhance model performance.
- **Evaluation and Visualization:**
  - BLEU score computation, loss curve plotting, and visualization of attention weights were integral to validating the model's effectiveness.

DESCRIPTION OF MY INDIVIDUAL WORK AND IMPLEMENTATIONS

My primary responsibilities in this project were focused on the implementation of the decoder mechanism and overseeing the model training process. These components formed the backbone of the encoder-decoder architecture, enabling the model to generate descriptive captions for images. While my teammate worked on designing the attention mechanism, I concentrated on the decoder implementation, training optimization, and ensuring the model's ability to effectively learn and generate captions.

BACKGROUND ON THE DEVELOPMENT OF THE ALGORITHM

Encoder-Decoder Framework

The encoder-decoder framework is a widely used sequence-to-sequence architecture designed for tasks like image captioning, machine translation, and text summarization. The encoder processes input data (in this case, images) to extract meaningful features, while the decoder utilizes these features to generate an output sequence (captions).

1. **Encoder:**

    - Utilized a pretrained Inception-V3 CNN to extract high-level feature embeddings from images.
    - These embeddings encode the image's semantic and spatial information into a fixed-size feature vector, serving as the input for the decoder.

2. **Decoder:**

    - The decoder, a GRU-based recurrent neural network, processes the encoded image features and generates captions word by word.
    - It models temporal dependencies, ensuring that each word in the generated caption is contextually relevant to the image and the preceding words.

MY CONTRIBUTIONS

1. Decoder Implementation

- The decoder leverages GRU (Gated Recurrent Unit) cells, which are known for their efficiency in capturing long-term dependencies while reducing computational overhead compared to LSTMs. The GRU computes the hidden state using the following equations:

$$z_t = \sigma(W_z \cdot [h_{t-1}, y_{t-1}])$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, y_{t-1}])$$

$$\tilde{h}_t = \tanh(W_h \cdot [r_t \odot h_{t-1}, y_{t-1}])$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$$

where:

  - z_t : Update gate.
  - r_t : Reset gate.
  - ~h_t : Candidate hidden state.
  - h_t : Final hidden state at time t.

- At each timestep, the decoder predicts the next word by passing the current hidden state through a dense layer with a softmax activation, generating a probability distribution over the vocabulary:

$$y_t = \text{softmax}(W_h \cdot h_t)$$

- The decoder continues generating words until it predicts the <end> token.

2. Teacher Forcing Mechanism

- A critical part of my work involved implementing the teacher forcing mechanism to stabilize and improve the training process.

- Teacher forcing is a technique where, during training, the actual ground truth word from the dataset is used as input for the next timestep instead of the word predicted by the model. This helps the model converge faster by learning from the correct sequences.

- Mathematically:

$$y_{t+1} = \begin{cases} y_t^{\text{true}} & \text{(with probability } p\text{)} \\ y_t^{\text{predicted}} & \text{(with probability } 1 - p\text{)} \end{cases}$$

Where $y^{\text{true}}{}_{t}$ is the actual word at time step t and $y^{\text{predicted}}{}_{t}$ is the predicted word at time step t.

The ratio p was gradually reduced during training to allow the model to handle real-world scenarios where it must rely on its own predictions.

- This mechanism helps the model avoid compounding errors, which occur when incorrect predictions are fed back into the network during training. It significantly enhances learning stability and allows the model to better handle long sequences.

3. Model Training

- I designed and managed the training pipeline, including hyperparameter tuning, batching, and monitoring loss metrics.

- The loss function used was **Sparse Categorical Crossentropy**, defined as:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij} \log \hat{y}_{ij}$$

  Where :

  - N: Number of samples.
  - C: Vocabulary size.
  - y_{ij} : True label for word j in sample i.
  - $y^{\wedge}{}_{ij}$: Predicted probability for word j in sample i.

- Training strategies included:

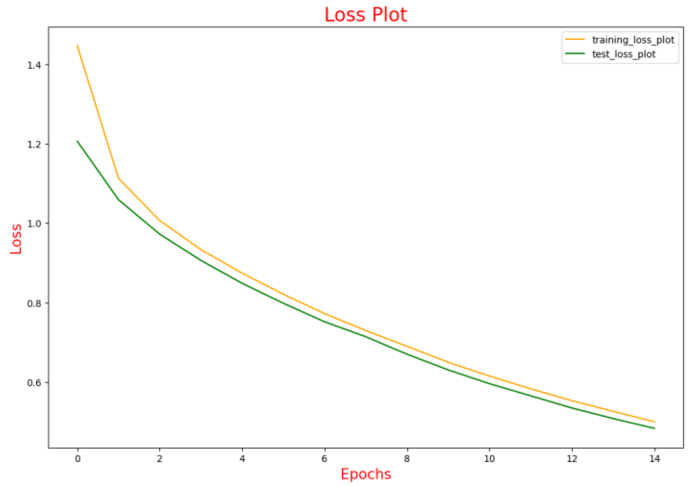
  - **Early stopping** to prevent overfitting by halting training when validation loss stopped improving.
  - **Batching and masking** to handle padded sequences efficiently, ensuring that padding tokens did not contribute to the loss.

RESULTS

The results of the image captioning project highlight the model's ability to generate accurate and meaningful captions for a wide range of input images. These findings are supported by quantitative metrics, qualitative observations, and visualizations that demonstrate the effectiveness of the model's implementation. Below is a detailed description of the results, including relevant figures and tables.

1. Loss curve

This plot shows the training and test loss over epochs, illustrating the model's learning progression and generalization capability.



LOSS PLOT

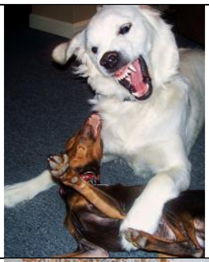
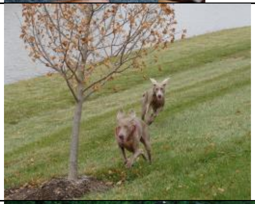
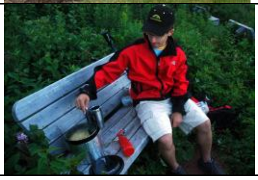
Explanation :

- The orange curve represents the **training loss**, and the green curve represents the **test loss**.
- Both curves exhibit a steady decline, indicating effective optimization of the model during training.
- The convergence of training and test loss values suggests that the model generalizes well to unseen data without overfitting.
- The stability in loss reduction reflects the robustness of the training pipeline, including the teacher forcing mechanism and proper hyperparameter tuning.

2. Qualitative results

The generated captions were evaluated qualitatively to assess their alignment with the content of the input images. Below are some examples of input images alongside their generated captions.

INPUT IMAGE	**GROUND TRUTH**	**GENERATED CAPTION**

	white dog opens its mouth near smaller dog	white dog opens its mouth
	two dogs run along the green grass near the water	two dogs run along the green grass
	man wearing red jacket sitting on bench next to various camping items	man wearing red jacket sitting on bench

Ground truth Vs Generated Captions

Explanation :

- In each example, the generated captions accurately describe the objects and actions depicted in the images.
- The phrasing of the captions may differ slightly from the reference captions, but the semantic meaning is well preserved.
- The results demonstrate the decoder's ability to process visual information and generate text that aligns with the image context.

3. Training Efficiency

The training pipeline was designed to ensure computational efficiency and effective optimization.

METRIC	VALUE
Total Training Time	~2 Hours
Batch size	64
Learning Rate	0.001
Optimizer	Adam
Test final loss	0.483

Explanation :

- The Adam optimizer enabled fast convergence, while the learning rate of 0.001 provided a balance between speed and stability.
- Early stopping helped prevent overfitting by halting training once validation loss stopped improving.
- The Test loss of 0.483 reflects the model's strong generalization ability, correlating with the high-quality captions generated during testing.

4. **Observations and Challenges**

Successes:

- The model performed well on simple and moderately complex images, accurately identifying objects and actions.
- Generated captions were contextually relevant and grammatically correct in most cases.

Challenges:

- For images with highly complex scenes or multiple objects, the model occasionally omitted details or incorrectly described certain elements. For example, in an image with a group of two dogs, the caption mentioned "white dog opens its mouth" but omitted the other dog.

CONLCUSION AND SUMMARY

The image captioning project successfully developed a model capable of generating meaningful and contextually relevant captions for images, achieving the primary goal of bridging visual and textual understanding. Utilizing a powerful encoder-decoder framework, with a pretrained Inception-V3 CNN as the encoder and a GRU-based decoder to process extracted image features, the project demonstrated the feasibility of applying deep learning techniques to generate descriptive captions. Thoughtful preprocessing and training strategies contributed to the model's strong performance on image-caption alignment tasks. The lessons learned provide a solid foundation for further exploration, while suggested improvements offer avenues to enhance accuracy, efficiency, and real-world applicability. These advancements promise to make image captioning systems even more versatile and impactful across various domains.

Code Utilisation :

Copied codes from internet : 78

Modified Lines : 23

Added lines : 24

Calculated percentage : 55/102 = 53.9%

References :

1. GRU-Based Decoder in Image Captioning :

   [1] A Deep Neural Framework for Image Caption Generation Using GRU-Based Attention Mechanism

   [2] A Comparative Study of Pre-trained CNNs and GRU-Based Attention for Image Caption Generation

2. Teacher Forcing Mechanism in Sequence-to-Sequence Models:

   [1] Attention Forcing for Sequence-to-Sequence Model Training

[2] Teacher Forcing in Recurrent Neural Networks: An Advanced Concept Made Simple