SE5110 - Fundamentals of Cognitive Computing

Final Assignment

MS24901376

H.G.B.S.T Geeganage

Contents

Introduction

This report describes the development and experimentation of two deep learning models, which were a part of the final assignment for the FCC (SE5110) module. The assignment was split into two phases, with each phase working in a different way within the field of multimodal learning using the same dataset, Flickr8k.

In Phase 1 the aim was to create an image captioning model from scratch - with no use of any pretrained components. This phase was focussed on designing a custom convolutional neural network (CNN) for feature extraction, and a recurrent neural network (RNN) using LSTM units, from which a descriptive caption can be produced from an image. The purpose of Phase 1 was to build a solid foundational understanding of how encoder-decoder architectures function together, and how they can relate both vision and language.

In Phase 2, the reverse order was attempted, using pretrained text-to-image generation model and creating synthetic images from text prompts (i.e., captions), from the same dataset. The phase focused on fine-tuning and experimentation with a large-scale generative model to give one experience with advanced techniques in image synthesis.

The two phases demonstrated the continuum that exists between computer vision and natural language processing, and, more importantly, how we can use neural networks in both types of tasks, generative and descriptive.

Data Preparation

For this phase, the dataset used was the Flickr8k Dataset, which consists of 8000 real images, and each image has 5 human-written captions. The objective of data preparation was to prepare the dataset to train a deep learning model that would take images and map the image features to natural language captions.

Phase 1: Image Captioning

Pre-processing Steps

Downloaded required the dataset from the project repository using "*wget*" and loaded the caption data from "*Flickr8k.token.txt"* and mapped each image to its 5 captions.

Then pre-processed the captions

1. Converted the text to lowercase
2. Removed all punctuation
3. Removed extra whitespaces
4. Added startseq and endseq tokens so that could denote the beginning and end of a sequence.

Tokenization

All captions were tokenized using Keras' Tokenizer.
The vocab size (vocab size) and max caption length (max length) were calculated and stored.

Image Preprocessing

All images were resized to 128x128 pixels to be consistent with the custom CNN used for the task.
Normalized the image from 0-255 scale to the [0,1] range.

Feature Extraction

A custom CNN was used to convert images to 256-dimensional feature vectors.
This was stored in a dictionary for training using the features.

Phase 2: Image Generation

In this phase an image was generated by a natural language prompt. This model uses "Stable Diffusion 2.1" provided by Stability AI and accessed via the Hugging Face diffusers library.

Pre-processing Steps

In this phase there were no model was crated but instead of training a model prompts were created manually and passed directly to the model.

Each prompt was developed in natural language describing a desired visual scene (e.g. "a house on top of a mountain", "a sunset over the ocean")

The prompts were passed directly to the Stable Diffusion pipeline, which internally executed: Tokenization (via CLIP text encoder) Latent space transformation Image decoding There was no additional preprocessing, augmented, or captions to map. The intent of this phase was simply to explore the generative capacity of a large pretrained model, through the application of descriptive prompts, and not training or tuning with a dataset.

Model Training

Phase 1: Image Captioning

This image caption model was trained using more than 8000 images and 5 human written captions from “Flickr8k” for each image. This model employees a model that followed an encoder–decoder architecture, where a custom CNN was used as the encoder to extract image details, and an LSTM-based decoder for generating captions word by word.

Training Configuration

- Number of images used for training: 8000
- Number of epochs: 20
- Batch size: 32
- Optimizer: Adam
- Loss Function: Categorical Crossentropy
- Callbacks Used:
    - *ModelCheckpoint* for saving best model during training.
    - EarlyStopping for halt training when loss stopped improving

Training process

A custom data generator was used to allow the model to receive image features as well as partial caption sequences through training. We created multiple training samples for each image-caption pair, training the model to predict the next word based on the given caption and the context of the image.

The model was trained using TensorFlow in Google Colab with GPU acceleration. A total of 1,000 steps per epoch were utilized to provide maximum variety in samples per epoch.

Performance Metrics

Final Training Loss: Decreased consistently over epochs

Evaluation: Caption quality was evaluated manually based on coherence and relevance. Some outputs demonstrated proper object-action recognition, while others showed repetitive phrasing, suggesting overfitting or limited vocabulary generalization.

Validation: Formal validation set was not formalized for training but the generalizations capability of the model was reviewed in part by examining captions output for images never seen by the model.

Phase 2: Image Generation

In Phase 2, no model training was conducted. Rather we used the pretrained text-to-image generation model, Stable Diffusion 2.1, to synthesize high-resolution images based on user-defined natural language prompts. The model was only used in its frozen state through the Hugging Face diffusers pipeline, and we did not fine-tune or adapt it.

Results

Phase 1: Image Captioning

The model was trained for 20 epochs with a batch size of 32 on the full Flickr8k dataset using the Adam optimizer. The training process converged steadily and consistently the loss decreased from 5.52 in epoch 1 to 3.47 in epoch 20.

Final Training Loss:

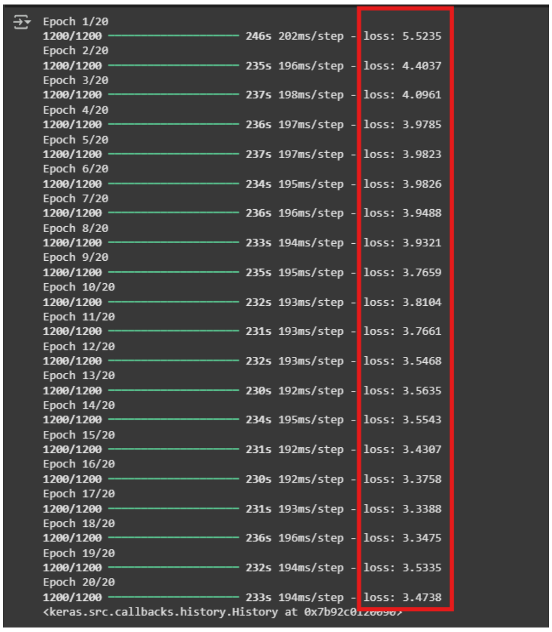
- **Epoch 1**: 5.5235
- **Epoch 10**: 3.8104
- **Epoch 20**: 3.4738
- **Total Epochs**: 20
- **Loss Decrease**: ~2.05 (39% improvement over 20 epochs)

No formal test accuracy was assessed since this was generative, but caption quality was qualitatively checked. Captions were rated on:

- Object identification
- Fluency of the sentences
- Action--object mapping

Many of the outputs had reasonable noun--verb construction (e.g., "a man is playing with a dog") and others had slight repetition or stereotypical description which may have been from limited vocabulary or a short training schedule.

**The training logs printed in Colab were used to examine the training progress. For each epoch, loss value was recorded and observed to monitor a learning trend.*
The terminal output showing loss per epoch is included to illustrate the model learning curve.



```
Epoch 1/20
1200/1200 ━━━━━━━━━━━━━━━━━━━━ 246s 202ms/step - loss: 5.5235
Epoch 2/20
1200/1200 ━━━━━━━━━━━━━━━━━━━━ 235s 196ms/step - loss: 4.4037
Epoch 3/20
1200/1200 ━━━━━━━━━━━━━━━━━━━━ 237s 198ms/step - loss: 4.0961
Epoch 4/20
1200/1200 ━━━━━━━━━━━━━━━━━━━━ 236s 197ms/step - loss: 3.9785
Epoch 5/20
1200/1200 ━━━━━━━━━━━━━━━━━━━━ 237s 197ms/step - loss: 3.9823
Epoch 6/20
1200/1200 ━━━━━━━━━━━━━━━━━━━━ 234s 195ms/step - loss: 3.9826
Epoch 7/20
1200/1200 ━━━━━━━━━━━━━━━━━━━━ 236s 196ms/step - loss: 3.9488
Epoch 8/20
1200/1200 ━━━━━━━━━━━━━━━━━━━━ 233s 194ms/step - loss: 3.9321
Epoch 9/20
1200/1200 ━━━━━━━━━━━━━━━━━━━━ 235s 195ms/step - loss: 3.7659
Epoch 10/20
1200/1200 ━━━━━━━━━━━━━━━━━━━━ 232s 193ms/step - loss: 3.8104
Epoch 11/20
1200/1200 ━━━━━━━━━━━━━━━━━━━━ 231s 193ms/step - loss: 3.7661
Epoch 12/20
1200/1200 ━━━━━━━━━━━━━━━━━━━━ 232s 193ms/step - loss: 3.5468
Epoch 13/20
1200/1200 ━━━━━━━━━━━━━━━━━━━━ 230s 192ms/step - loss: 3.5635
Epoch 14/20
1200/1200 ━━━━━━━━━━━━━━━━━━━━ 234s 195ms/step - loss: 3.5543
Epoch 15/20
1200/1200 ━━━━━━━━━━━━━━━━━━━━ 231s 192ms/step - loss: 3.4307
Epoch 16/20
1200/1200 ━━━━━━━━━━━━━━━━━━━━ 230s 192ms/step - loss: 3.3758
Epoch 17/20
1200/1200 ━━━━━━━━━━━━━━━━━━━━ 231s 193ms/step - loss: 3.3388
Epoch 18/20
1200/1200 ━━━━━━━━━━━━━━━━━━━━ 236s 196ms/step - loss: 3.3475
Epoch 19/20
1200/1200 ━━━━━━━━━━━━━━━━━━━━ 232s 194ms/step - loss: 3.5335
Epoch 20/20
1200/1200 ━━━━━━━━━━━━━━━━━━━━ 233s 194ms/step - loss: 3.4738
<keras.src.callbacks.history.History at 0x7b92c0120090>
```

Figure 1: Loss of each epoch