Image Captioning using Attention Mechanism

Week 2: Introduction to Convolutional Neural Networks (CNNs)

Welcome to the **second week** of our project course! This week, we will dive into Convolutional Neural Networks (CNNs), a powerful class of deep learning models that have revolutionized the field of computer vision. By the end of this week, you will have a solid understanding of CNNs and will be able to implement a basic CNN for image classification.

Topics to be Covered:

1. **Introduction to CNNs**
    - Understanding the basic structure and functionality of CNNs.
    - Differences between CNNs and traditional neural networks.
2. **Convolutional Layers and Filters**
    - Learning about convolution operations and how filters (kernels) work.
    - Exploring how convolutional layers extract features from images.
3. **Pooling Layers and Feature Extraction**
    - Understanding the purpose of pooling layers (e.g., max pooling, average pooling).
    - How pooling helps in downsampling and retaining essential features.
4. **Implementing a Basic CNN for Image Classification**
    - A step-by-step guide to building a simple CNN using TensorFlow and Keras.
    - Training the model on a dataset and evaluating its performance.

Resources

To help you understand and implement the concepts covered this week, here are some recommended resources:

- Day 5-Understanding CNN &Impementation| Live Deep Learning Community S…
- **Understanding Convolutional Neural Networks for NLP · Denny's Blog**
- Convolutional Neural Network Tutorial In TensorFlow / Keras

Assignment-2: Basic Classification Task using CNN (Deadline: 10th June)

Problem Statement:

Develop a CNN model for object detection using TensorFlow Keras. Utilize images from the COCO dataset and construct a CNN architecture suitable for object detection. You may choose to start with a pre-trained model such as ResNet or VGG. The objective is to build a model that can accurately identify objects within images and evaluate its performance using standard metrics.

Tasks:

1. **Data Preparation:**
    - Download a subset of the COCO dataset containing a variety of object classes.
    - Preprocess the images and annotations to a suitable format for training a CNN.
2. **Model Development:**
    - Construct a CNN architecture from scratch or fine-tune a pre-trained model (e.g., ResNet, VGG).
    - Ensure your model includes essential components such as convolutional layers, pooling layers, and fully connected layers for classification.
3. **Training the Model:**
    - Train your CNN on the prepared dataset.
    - Use techniques like data augmentation and regularization to improve model performance.
4. **Evaluation:**
    - Evaluate your model on a separate test set not used during training.
    - Report metrics such as precision, recall, F1 score, and any other relevant performance metrics.
5. **Reporting:**
    - Document your model architecture, training process, and results.
    - Provide a detailed analysis of the model's performance and potential improvements.

By the end of this checkpoint, you should have a functional CNN model capable of performing basic object detection. Good luck!

Submission Instructions

Submit the code and analysis on your respective GitHub and Google Drive link as stated in the Submission Form.

Extra for Interested Learners

For those of you who are particularly interested in the inner workings of CNNs, a fascinating exercise is to visualize how kernels evolve during training. By printing the images of kernels over time, you can see what features these kernels are learning to detect.