Fazioli 1604197

Report project- Sequencer: Deep LSTM for Image Classification

Overview:

This code is dedicated to training a deep learning model using PyTorch. The model is a custom architecture, and the training is performed on the CIFAR-10 dataset. The code includes several components, including the model architecture definition, data loading, preprocessing, training loop, and optimizer settings.

Vision Transformer (ViT) with BiLSTM: A Variant with Sequential Memory Integration

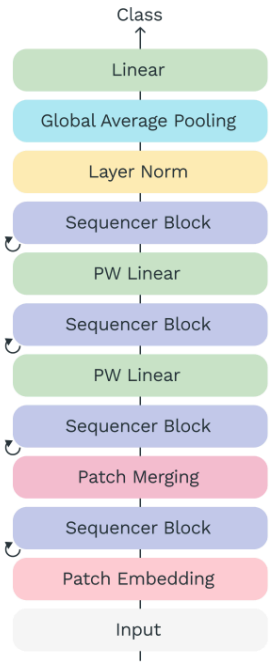
Vision Transformer (ViT) has emerged as a groundbreaking architecture for image analysis and processing. Conventionally, ViT relies on the powerful self-attention mechanism to capture contextual information from images. However, a variant of ViT introduces an innovative twist by substituting the traditional self-attention mechanism with Bidirectional Long Short-Term Memory (BiLSTM) layers. This adaptation aims to harness the strengths of recurrent neural networks for sequential memory integration in vision tasks.

Understanding Vision Transformer (ViT):

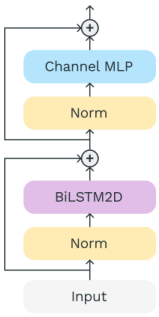
The Vision Transformer (ViT) is a deep learning model designed to process and understand images in a manner inspired by the Transformer architecture originally developed for natural language processing. In its typical form, ViT decomposes an image into non-overlapping patches, which are then embedded and processed through a series of self-attention layers. This self-attention mechanism allows the model to capture dependencies and relationships between different image patches.

The ViT Variant with BiLSTM2D:

The variant of ViT with BiLSTM2d integrates the BiLSTM2d layer, which, unlike self-attention, operates sequentially. A BiLSTM2D, short for Bidirectional Long Short-Term Memory, is a type of recurrent neural network capable of processing sequential data in both forward and backward directions. By introducing BiLSTM2D layers into the ViT architecture, this model leverages the sequential memory integration capabilities of BiLSTM2D. This allows the model to understand images in a more context-aware manner, considering both the local and global information while processing image patches.(in the figure there is the original architecture that I've tried to follow in structures and params in my code)



(a) Sequencer



(e) Sequencer2D block

The sequencer 2D is made by Normal layer followed by Bilstm2D (BiLSTM means that it got 2 bidirectional layer one for vertical data and other one for horizontal data), other normal layer and a fully connected layer.

(In the figure there is the original structure of Sequencer2D).

There is other variant made by using BiLSTM instead of BiLSTM whose name is **VanillaSequencer** , I've developed it too, but as in the original work the accuracy was lower than sequencer2D.

Model Architecture (Sequencer2DModel):

- The model is a custom architecture designed for image classification.
- It consists of multiple layers, including patch embedding, sequencer blocks, point wise linear (PW) layers, layer normalization, global average pooling, and fully connected layers.
- The model is designed to handle images with 3 input channels and predict one of 5 classes.

Data Loading and Preprocessing:

- The code loads the CIFAR-10 dataset using a custom dataset class (CustomCIFAR2).
- Data augmentation techniques are applied, including random horizontal flip, rotation, color jittering, resizing, and affine transformations.
- Data is normalized using specified mean and standard deviation values.

Training Loop:

- The training loop iterates through multiple epochs and batches.
- Each batch of data is passed through the model, and the predictions are compared to the true labels to calculate the loss.
- The loss is backpropagated, and the model's weights are updated using the Adam optimizer.

I*ssues Encountere*

Fluctuating Accuracy: The model's accuracy is fluctuating, and it's not learning effectively. This issue may be due to various factors:

- Learning Rate: The learning rate might not be appropriate for the model and problem.
- Weight Initialization: The weights may not be initialized optimally for training with SGD.
- Local Minima: The model could be trapped in local minima.
- Data Preprocessing: Data preprocessing may not be consistent with the optimizer used.
- Complexity: The model's architecture could be too complex, making training challenging.
- Optimizer Change: Changing from Adam to SGD did not yield significant accuracy improvements. Possible reasons include inappropriate hyperparameters for SGD, loss landscape challenges, and weight initialization issues.

In summary, the code defines a custom deep learning model for image classification using PyTorch. However, it encounters issues related to accuracy fluctuations and optimization.

Poor Initial Learning

- Initially, the model struggled to learn from the data, with low accuracy.
- Xavier (Glorot) initialization was introduced to mitigate the problem of poor weight initialization, but it did not significantly improve the situation.
-

Data-Related Problems

- Data preprocessing and augmentation were scrutinized to ensure proper alignment of labels and data.but it turns out cifar10 has already preprocessed images

Optimization Algorithm and Loss Function

- The model initially used the Adam optimizer, but experimenting with other optimization algorithms and loss functions was done

Hyperparameter Tuning

- The model has several hyperparameters, such as batch size, number of epochs, and network architecture, that maybe have not been extensively tuned.