TRANSFORMERS NETWORK
(Report)

OVERALL OBSERVATIONS AND ANALYSIS:

Observations:

(1) Patch_size (4, 8, 16) vs. test accuracy with constant heads = 8:
For non-overlapping patches:

Patch size 4: Test accuracy = 0.8022
Patch size 8: Test accuracy = 0.7748
Patch size 16: Test accuracy = 0.6801

For overlapping patches:

Patch size 4: Test accuracy = 0.8089
Patch size 8: Test accuracy = 0.8252
Patch size 16: Test accuracy = 0.7564

(2) Heads (4, 8, 16) vs. test accuracy with constant patch_size = 4x4:
4 heads: Test accuracy = 0.8020 (non-overlapping)
8 heads: Test accuracy = 0.8022 (non-overlapping)
16 heads: Test accuracy = 0.7876 (non-overlapping)

(3) CLS tokens from different layers for classification with patch_size = 4x4, heads = 8:
Non-overlapping patches : Test accuracy = 0.7992
Over-lapping patches: Test accuracy = 0.8031

Note: Train accuracy for all was between 90-95 % .

Analysis:

1. The test accuracy decreased with the increase in patch size for non-overlapping patches because larger patch sizes reduce the spatial resolution of the image representation, leading to a loss of fine-grained details. Smaller patches allow the model to capture finer details, which might be important for classification tasks.

2. The test accuracy increased with an increase in patch size from 4 to 8 but decreased for 16 patches for overlapping patches because overlapping patches provide a more robust representation of the image, allowing the model to capture both local and global features. For patch size 8, this benefit is maximized, but when the patch size increases to 16, the loss of spatial resolution begins to outweigh the benefits of overlapping patches, leading to a decrease in accuracy.

3. The test accuracy decreased with an increase in heads because having more heads allows the model to attend to different aspects of the input simultaneously, which can be beneficial for complex tasks. However, it also increases the number of parameters in the model, which can lead to overfitting or make the model harder to train. In this case, the additional heads may not have provided significant benefits to the task, and the increased complexity may have resulted in reduced accuracy.

4. For all patches (4, 8, 16), overlapping patches had better accuracy compared to non-overlapping patches because overlapping patches provide a richer and more robust representation of the image. Overlapping patches capture local features and contextual information more effectively, which can be particularly important for tasks like image classification, where both local and global features contribute to the decision-making process. This leads to better performance for overlapping patches compared to their non-overlapping counterparts.

5. With overlapping patches, the model receives more contextual information about the image as adjacent patches share some regions of the image. This enables the model to better understand the relationships between neighboring regions and learn more discriminative features for classification. In the case of using patch_size = 4x4 and heads = 8, the overlapping patches allow the model to capture fine-grained details while maintaining the benefits of multi-head attention. The combination of a smaller patch size and an appropriate number of attention heads enables the model to attend to different aspects of the input simultaneously, which is beneficial for complex tasks like image classification.
   By utilizing CLS tokens from different layers for classification, the model can also take advantage of hierarchical feature representations. This further helps the model to make better decisions based on the information gathered from various stages of the network, leading to improved test accuracy for overlapping patches compared to non-overlapping patches.

(a) **Implementing ViT**

The code implements a Vision Transformer (ViT) architecture for image classification. The ViT architecture is an adaptation of the Transformer architecture, originally designed for natural language processing, to the computer vision domain. The key components of this architecture include:

1. ***PatchEmbedding***: This class is responsible for breaking the input image into non-overlapping patches and converting them into a linear embedding. It uses a 2D convolutional layer (nn.Conv2d) with a kernel size and stride equal to the patch size to achieve this.
2. ***MultiHeadAttention***: This class implements the multi-head self-attention mechanism, a key component of the Transformer architecture. It consists of linear transformations (nn.Linear) for query, key, and value (q, k, v), an attention dropout layer (nn.Dropout), and a linear projection layer (nn.Linear) followed by another dropout layer.
3. ***MLP***: This class implements a two-layer feedforward neural network with GELU activation (nn.GELU) and dropout (nn.Dropout).
4. ***TransformerEncoder***: This class combines the multi-head self-attention mechanism and the feedforward neural network (MLP) to create a single Transformer encoder layer. It uses layer normalization (nn.LayerNorm) and residual connections.
5. ***VisionTransformer***: This is the main class that assembles the overall architecture. It starts by applying the PatchEmbedding to the input image, then adds the class (CLS) token and positional embeddings. The encoded image is passed through a series of TransformerEncoder layers. Finally, the output of the final layer's CLS token is normalized and passed through a linear layer (nn.Linear) for classification.

(b) **Training this model on the CIFAR-10 dataset for 10-class classification**

The architecture processes the input image in a hierarchical manner, capturing local and global patterns at different levels of abstraction. It has been shown to achieve state-of-the-art performance on image classification tasks.
The Vision Transformer (ViT) model is trained and evaluated on the CIFAR-10 dataset for image classification. Throughout all the experiments, the *hyperparameters* used are:

Image size: 64x64
Patch size: 4x4
Number of classes: 10
Input channels: 3 (RGB)
Hidden dimension: 64
Depth: 4
Number of heads: 8
MLP ratio: 4
Batch size: 128
Number of epochs: 200
Learning rate: 1e-3

Data augmentation is applied to the training set using random horizontal flips and color jitter. Images in both training and test sets are resized to 64x64 and normalized.

The CIFAR-10 dataset is split into train, validation, and test sets. The model is trained using the AdamW optimizer with weight decay for regularization and the CosineAnnealingLR learning rate scheduler. The CrossEntropyLoss function is used as the loss function. The model is trained on a GPU (NVIDIA A100), and the number of workers(=4) and GPUs used (=1, =8 for few tasks which required more memory).

Training and validation loss and accuracy are recorded for each epoch, and the total training time is reported. The model is then evaluated on the train and test sets, and the final train and test accuracies are reported.

(c) **Trying out different patch sizes (like 4x4, 8x8, 16x16) for both Overlapping and Non-overlapping patches.**

***Non-Overlapping patches*:**

(i) 4x4:
Observations:
The training loss and validation loss decreased over the course of 200 epochs.
The training accuracy increased consistently, reaching 94.51% at the end of training.
The validation accuracy improved but seemed to plateau around 80.22% towards the end of training

Analysis:
Patch size: The 4x4 patch size used in this ViT model leads to more patches, resulting in increased complexity of the model. While this can help the model capture more fine-grained features, it can also increase the risk of overfitting.

The ViT model employed here is relatively shallow, with only 4 depth levels. A deeper model might potentially learn more complex representations and achieve better results. However, increasing the depth would also increase the risk of overfitting.
Hyperparameters: The chosen hyperparameters seem reasonable for this training process. A learning rate of 1e-3 is common, and the model has an appropriate number of heads (8), depth (4), and hidden dimensions (64). However, different combinations of these hyperparameters could potentially yield better results.

Addressing overfitting: There is a noticeable gap between the training accuracy (94.51%) and the validation accuracy (80.22%). This indicates that the model might be overfitting to the training data, meaning it has learned the training data too well, at the expense of generalizing to unseen data.
To address the overfitting issue, you could consider the following:
Hyperparameter tuning:
Experiment with different hyperparameters, such as the number of layers, patch size, and number of attention heads, to find a better balance between model complexity and generalization.
Cross-validation: Use k-fold cross-validation to make better use of the available data and reduce the risk of overfitting.

(ii) **8x8:**
Observations:
In the training process of the Vision Transformer (ViT) model with a patch size of 8x8 (non-overlapping), we can see that the training and validation losses decrease over time, and the accuracies improve. The model's performance on the test set reaches 77.48% after 200 epochs, which is a decent accuracy for a relatively simple ViT model.

Analysis:
Patch size: The patch size of 8x8 may result in some loss of local information, as the patches are non-overlapping, and the image size is relatively small (64x64). A smaller patch size could potentially result in better performance, as it would capture more local information.

The ViT architecture is relatively simple, with 4 layers and 8 attention heads. Increasing the depth or number of heads might improve the model's performance.

Regarding overfitting:
There is a slight indication of overfitting, as the train accuracy reaches 93.9%, which is significantly higher than the test accuracy of 77.48%. This gap indicates that the model is learning to fit the training data very well, but it does not generalize as effectively to unseen data. However, the degree of overfitting is not severe, as the model still achieves a reasonable test accuracy.

(iii) **16x16**
Observations:
As the number of epochs increases, both training and validation accuracy improve up to a certain point.
The training loss decreases, and the training accuracy increases consistently throughout the epochs, reaching a train accuracy of 0.8913.
The validation accuracy increases but at a slower rate than the training accuracy, reaching a test accuracy of 0.6801.
The training time per epoch remains relatively constant.

Analysis:
patch size of 16x16 used in the ViT model is a compromise between model complexity and the ability to capture local features. With a larger patch size, the model would not be able to capture finer details. A smaller patch size would make the model more complex and increase the risk of overfitting. The choice of 16x16 patches allows the model to learn useful features while keeping the model complexity under control.
The ViT architecture itself is designed to handle large-scale image classification tasks. It has the advantage of being able to learn hierarchical features in an image without the need for a large number of layers or weights. This helps to reduce overfitting and maintain a reasonable training time per epoch.

Regarding overfitting:
The model seems to have a relatively high train accuracy (0.8913) compared to the test accuracy (0.6801). This difference indicates that the model is fitting the training data well but not generalizing as effectively to the unseen validation data.
While there is a gap between training and validation accuracy, the validation accuracy still increases over the epochs, suggesting that the model is learning useful features. However, the increase in validation accuracy is not as rapid as the increase in training accuracy

***Overlapping patches*:**

ViT architecture components:

1. **PatchEmbedding:** This class is responsible for breaking the input image into non-overlapping or overlapping patches and converting them into flat embeddings using a convolutional layer (nn.Conv2d). The default stride is half of the patch size, which makes the patches overlapping.
2. **MultiHeadAttention:** This class implements the multi-head self-attention mechanism, which is a key component of the Transformer architecture. It takes the input embeddings, splits them into multiple heads, computes the self-attention for each head, and then concatenates the results.
3. **MLP**: This class implements a simple Multi-Layer Perceptron with a GELU activation function and dropout. This is used as a feed-forward network in the Transformer architecture.
4. **TransformerEncoder**: This class combines the multi-head self-attention and MLP components with residual connections and layer normalization. The input embeddings are passed through the attention mechanism and MLP in sequence, with layer normalization applied before each component.
5. **VisionTransformer**: This is the main class that combines all the previously mentioned components. It initializes the PatchEmbedding layer, positional embeddings, and a sequence of TransformerEncoder layers. The forward pass of this class processes the input image through the patch embedding, adds the class token and positional embeddings, and passes the result through the sequence of Transformer layers. Finally, the output from the class token is passed through a layer normalization and a linear layer to produce the classification logits.

(i) 4x4:
Observation:
As the number of epochs increases, the train loss decreases, and train accuracy increases, which is expected in any training process.
The validation accuracy increases up to a certain point, after which it stabilizes and does not improve significantly.

Analysis:
The chosen architecture, Vision Transformer (ViT) with a patch size of 4x4 and 8 attention heads, seems to perform well on the training set, reaching a train accuracy of 0.9535 after 200 epochs. However, there is a gap between the train accuracy and test accuracy (0.8089), which suggests overfitting. The model has learned the training set very well, but its performance on the validation set is not as strong.

The overfitting might be due to the following factors:
Patch size: A smaller patch size, like 4x4, might lead to a higher capacity model that captures more local information, leading to overfitting. A larger patch size could be considered to make the model learn more global features and reduce overfitting.
ViT is known to require a larger amount of data to avoid overfitting. If the dataset used for training is relatively small, the model might not generalize well to unseen data.

To address the overfitting issue, you could consider the following:
Hyperparameter tuning:
Experiment with different hyperparameters, such as the number of layers, patch size, and number of attention heads, to find a better balance between model complexity and generalization.
Cross-validation: Use k-fold cross-validation to make better use of the available data and reduce the risk of overfitting.

(ii) 8x8:
Observation:
From the given data, it is observed that as the number of epochs increases, the training loss decreases, and the training accuracy increases. Similarly, the validation loss decreases, and the validation accuracy increases for most of the epochs. This suggests that the model is learning to generalize well on the given dataset.

Regarding overfitting:
As we progress towards epoch 200, the gap between the training and validation accuracies widens. At epoch 200, we have a training accuracy of 0.9524 and a test accuracy of 0.8252.
This indicates that the model might be overfitting, i.e., it is performing very well on the training dataset but not generalizing as well on the validation dataset.

Analysis:
Patch Size: Using a patch size of 8x8 allows the model to capture more local information in each patch, which can be beneficial for the learning process. However, it also increases the number of patches and the overall model complexity, which could contribute to overfitting.
Number of Heads (8): The number of heads in a multi-head self-attention mechanism influences the ability of the model to capture different types of relationships within the data. Having more heads allows the model to learn diverse feature representations, which can improve the model's performance. However, a higher number of heads might also increase the risk of overfitting as the model capacity increases.
The Vision Transformer (ViT) is designed to handle image data effectively by dividing images into patches and treating them as sequences. This allows the model to leverage the power of transformers for vision tasks. However, the architecture can be prone to overfitting, especially when training on smaller datasets, as transformers have a large number

(iii) 16x16:
Observation:
The training process for the Vision Transformer (ViT) model using a patch size of 16x16, 8 attention heads, and the given hyperparameters show a gradual increase in both training and validation accuracy over the course of 200 epochs.
The training accuracy reaches 91.20%, while the test accuracy is 75.64% after 200 epochs. This indicates that the model is learning the underlying patterns in the training data, but there is still a gap between the training and test accuracies, which suggests some overfitting.
The validation loss shows a decreasing trend in general but fluctuates during the training process, indicating that the model may be sensitive to some extent to the choice of learning rate, weight decay, or other hyperparameters.

Analysis:
The patch size of 16x16 allows the model to capture local information within each patch and then combine this information across patches using the self-attention mechanism. This approach has been shown to be effective for image classification tasks.
Using 8 attention heads enables the model to focus on different aspects of the input data and learn multiple features simultaneously. This can help improve the model's expressiveness and ability to capture complex patterns in the data.
The ViT architecture itself has been designed to handle image classification tasks efficiently, leveraging the power of transformers and self-attention mechanisms.

Regarding overfitting:
The difference between the training and test accuracies (91.20% vs. 75.64%) suggests that the model may be overfitting to some extent. This is further supported by the fluctuations observed in the validation loss over time.

(d) **Trying out different number of attention heads (4,8,16) for Non-Overlapping patches (constant patch_size = 4x4).**

(i) heads = 4

Observation:
The Vision Transformer (ViT) model with 4 heads and non-overlapping 4x4 patches has shown consistent improvement in training and validation accuracy over 200 epochs. The train and test accuracies after the final evaluation are 0.9251 and 0.8020, respectively. The training loss and accuracy show a steady improvement throughout the training process, indicating that the model is learning well.

Analysis:
Patch size: The 4x4 patch size could be an appropriate choice for the given dataset. It seems to be large enough to capture the relevant features in the images and small enough not to lose too much spatial information.
Heads: Using 4 heads in the multi-head self-attention mechanism allows the model to capture different aspects of the data. This can contribute to the learning process, as the model can attend to different features simultaneously.

Regarding overfitting:
There are some signs of overfitting in the later epochs, as the training accuracy reaches 0.9251, while the test accuracy is 0.8020. The gap between the training and validation accuracies grows larger in the final epochs, suggesting that the model might be overfitting to the training data. However, it is important to note that the test accuracy is still relatively high at 0.8020.

(ii) heads = 8

Observation:

With the given Vision Transformer (ViT) model with 8 heads and a patch size of 4x4 (non-overlapping), we can see that the training process gradually improves the train and validation accuracies. The model training and validation losses also decrease over time. The training time per epoch is relatively stable, with some minor fluctuations.

Analysis:

The patch size of 4x4 (non-overlapping) allows the model to focus on smaller regions within the image, extracting local features.

The 8-head attention mechanism enables the model to learn and attend to multiple relevant features simultaneously. These settings contribute to the model's capability to improve its performance over time.

Regarding overfitting:

When looking at the final results after 200 epochs, we have a train accuracy of 0.9451 and a test accuracy of 0.8022. There is a noticeable gap between the training and test accuracies, suggesting that the model might be overfitting to the training data. This overfitting is also visible throughout the training process, as the difference between the training and validation accuracies consistently increases.

(iii) heads = 16

Observation:

The model's performance improved consistently during the training process, with train accuracy reaching 0.9506 and test accuracy 0.7876.

The training loss decreased consistently, while the validation loss stabilized after some epochs, showing that the model is learning from the training data.

The training time per epoch remained relatively stable throughout the training process.

Analysis:

The given ViT architecture has 16 heads and a patch size of 4x4 with non-overlapping patches.

A patch size of 4x4 and non-overlapping patches indicate that the model has a relatively coarse view of the input images, which can help in extracting features from the images efficiently.

The usage of 16 heads enables the model to attend to different parts of the input simultaneously, allowing for a better representation of the features and relationships in the input data.

Regarding overfitting:

The train accuracy (0.9506) is significantly higher than the test accuracy (0.7876), which indicates that the model is overfitting the training data. Overfitting can be attributed to the limited capacity of the model to generalize well due to the given architecture and hyperparameters.

(e) Perform classification by using the CLS token from different layers of the model.

***(i) Non-Overlapping patches* with patch_size = 4x4 , heads = 8**

The main components of the ViT architecture in the code:

1. **PatchEmbedding:** This class is responsible for dividing the input image into non-overlapping patches and embedding them into a specified dimension using a 2D convolution operation.
2. **MultiHeadAttention**: This class implements the multi-head self-attention mechanism, a core component of the transformer architecture. It computes query, key, and value matrices, and then uses these to compute the attention weights and output.
3. **MLP:** This class defines a simple feed-forward neural network with one hidden layer, using the GELU activation function and dropout for regularization.
4. **TransformerEncoder:** This class combines the MultiHeadAttention and MLP components, along with layer normalization and residual connections, to form a single transformer encoder block.
5. **VisionTransformer:** This is the main class that brings all the components together. It initializes the PatchEmbedding, TransformerEncoder layers, and the final classification layer. During the forward pass, it processes the input image through the PatchEmbedding, adds positional embeddings, and applies each TransformerEncoder layer sequentially. It computes the *average of the Class (CLS) tokens across the depth dimension*, applies Layer Normalization, and passes the result through the classification layer.

Observations:
In this experiment, we trained the ViT model with 8 heads and a patch size of 4x4 (non-overlapping). As the training progresses, the training loss and validation loss generally decrease, and the training accuracy and validation accuracy improve. The model reaches a train accuracy of 93.83% and a test accuracy of 79.92% after 200 epochs.

Analysis:
The choice of 4x4 non-overlapping patch size allows the model to capture local features within each patch, and the 8 heads provide multiple attention mechanisms for the model to focus on different aspects of the input. The training process shows improvement in accuracy and reduction in loss as the model learns from the data, generalizes the features, and refines its understanding of the problem.

Regarding Overfitting:
There is some overfitting observed as the training accuracy (93.83%) is considerably higher than the test accuracy (79.92%). This could be due to the model learning specific details of the training set, which does not generalize well to the test set.

(ii) Overlapping patches **with patch_size = 4x4 , heads = 8**

Abrief overview of each component in the code:

1. **PatchEmbedding:** A class that divides the input image into overlapping patches and projects them into a specified embedding dimension using a convolutional layer. The stride determines the degree of overlap between patches.
2. **MultiHeadAttention:** This class implements the multi-head self-attention mechanism, which is an essential component of the transformer architecture. It calculates query, key, and value matrices and computes the scaled dot-product attention for each head. The outputs are then concatenated and projected back to the original embedding dimension.
3. **MLP**: A two-layer feed-forward neural network with GELU activation and dropout layers. This MLP is used in the transformer encoder layers.
4. **TransformerEncoder:** A single transformer encoder layer that includes layer normalization, multi-head self-attention, and an MLP.
5. **VisionTransformer:** The main class that brings all the components together to form the Vision Transformer architecture. It starts by creating patch embeddings, adding a learnable CLS token, and applying positional embeddings. Then, it processes the input through multiple TransformerEncoder layers, extracting the CLS token from each layer. Finally, it computes the mean of the extracted CLS tokens, applies layer normalization, and uses a linear layer to produce the class logits.

Observations

The training loss and accuracy show a consistent improvement throughout the training process. By the end of training, the model achieves a train accuracy of 94.96%. This indicates that the model is learning from the data and generalizing well on the training set.
The validation loss and accuracy also show an improvement over time, but they do not follow the same smooth trajectory as the training loss and accuracy.
At epoch 200, the model achieves a test accuracy of 80.31%. This difference between training and test accuracies is indicative of some overfitting. The training time per epoch seems to be consistent, with minor fluctuations throughout the training process.

Analysis:

A smaller patch size (4x4) captures more fine-grained information from the input image, allowing the model to learn local features more effectively. This can lead to a better representation of the input data and improve the model's performance.
Using 8 heads in the multi-head attention mechanism allows the model to capture different aspects of the input data and learn more diverse feature representations. This can contribute to a more expressive model and can lead to better performance.

Regarding Overfitting:
The observed overfitting might be due to the following reasons:
A smaller patch size and a higher number of heads could make the model more complex and increase its capacity to memorize the training data. This can lead to overfitting, as seen in the difference between the training and test accuracies.
Insufficient regularization techniques, such as dropout or weight decay, might be contributing to overfitting. Applying more regularization might help to mitigate this issue.