**DLCV**

**Assignment-2 (Report)**

**Name : Ankit Kumar Sinha**

**SR. No : 21289**

**Ques 2:**

**(a)**

The architecture of the basic version of the vision transformer implemented is :

* The images of CIFAR 10 datasets have the dimension 32 X 32 and the number of images are 6000 (5000 train set and 1000 test set).
* Tokenize the images with the help of convolution operation, taking stride same as filter size. The number of filter dimensions is same as taking the number of embedding dimensions in the patches.
* Add the class token (dim : same as embed dimension of the token ) to the images patches. So the number of tokens would be increased by 1. Then add positional embedding to all the tokens (dim : same as embed dimension of the token ). These positional embedding and class token are the learnable parameters.
* Now these tokens are fed to the self-attention module one by one. The number of heads in self-attention modules are considered 8. The first step is to extract query, key and values pairs. This is done with the help of linear module (where bias is set to false).
* Then the residual and output of self attention module ( where Q is the query value, K is key value and is the scaling factor and d is dimension of the head.
* The output of the self-attention and the residual is added and passed input to the multi layer perceptron with one hidden layer. The activation function is GELU() is used.
* The dimension of the output of above encoder layer is same as token, which is passed sequentially into other encoder layers. The number of encoder layers are 4
* Then finally the class token is taken from final layer and is passed into head of the transformer which gives output as the probabilities for the number of the classes.

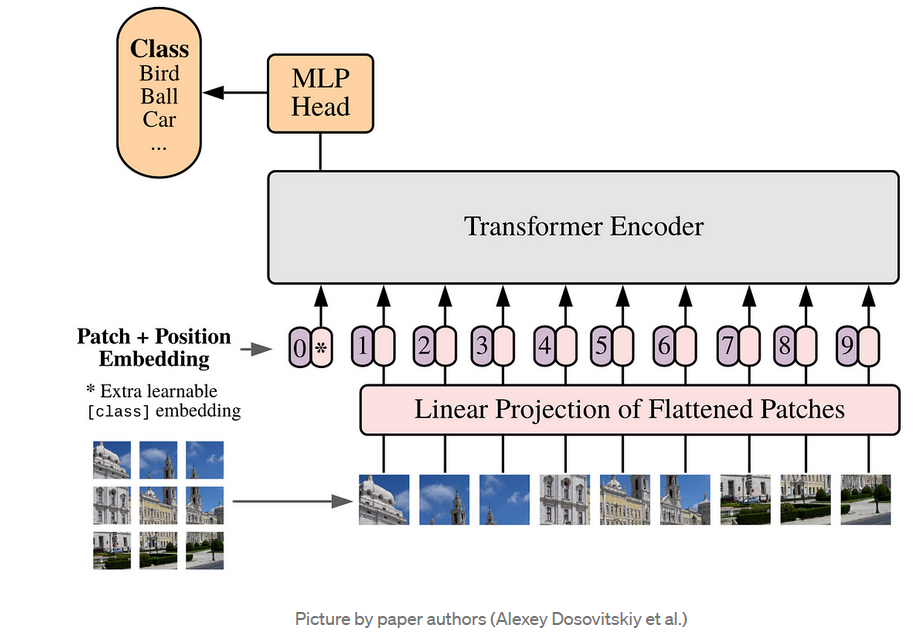


Fig 1: The basic version for vision transformer.

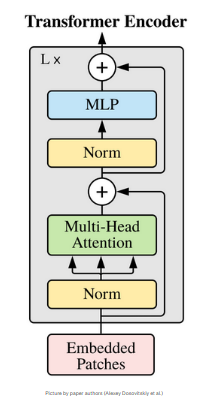


Fig 2: The encoder layer of the vision transformer.

(c)

The number of epochs used in all different patch sizes is 10.

The size of the input image (CIFAR 10) : 32X32X3

The validation set is 0.2 the size of train set.

**Non-Overlapping case**

**Model performance for patch size (4X4):**

Token dimension: (4X4X3) = 48

Number of tokens: 64

Head dimension: 48

Number of head : 8

Number of hidden layer in multi layer perceptron: 512

Dropout probability: 0.2

Chart, line chart

Description automatically generated

The model starts overfitting as the training accuracy crosses 85 % while the validation accuracy is saturated around 60 %.

Accuracy of the model on the test images is 59.09 %

**Model performance for patch size (8X8):**

The size of the input image (CIFAR 10) : 32X32X3

Token dimension: (8X8X3) = 192

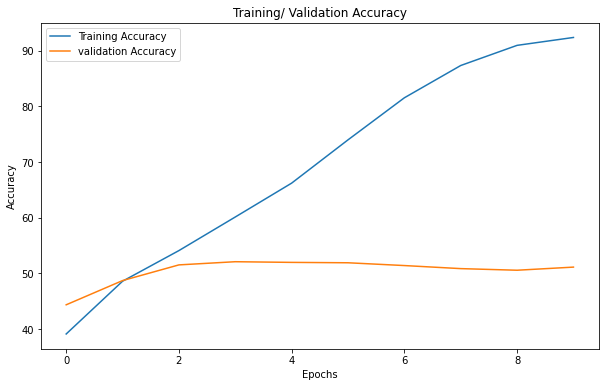
Number of tokens: 16

Head dimension: 192

Number of head : 8

Number of hidden layer in multi layer perceptron: 512

Dropout probability: 0.2



The model starts overfitting here also as the training accuracy crosses 92 % while the validation accuracy is saturated around 51.11 %.

The accuracy of the model around test sets is around 49 %

**Model performance for patch size (16X16):**

The size of the input image (CIFAR 10) : 32X32X3

Token dimension: (16X16X3) = 768

Number of tokens: 4

Head dimension: 768

Number of head : 8

Number of hidden layer in multi layer perceptron: 512

Dropout probability: 0.2

Chart, line chart

Description automatically generated

The model is not overfitting till the 10 epochs and its training accuracy is around 47 % at the end of the 10 epochs .

The validation accuracy is also performing similar to the train accuracy is around 45.35 % at the end of the 10 epochs

Accuracy of the model on the test images is around 45 %

**Overlapping case**

Stride used is half the patch size in all the cases.

**Model performance for patch size (4X4):**

Token dimension: (4X4X3) = 48

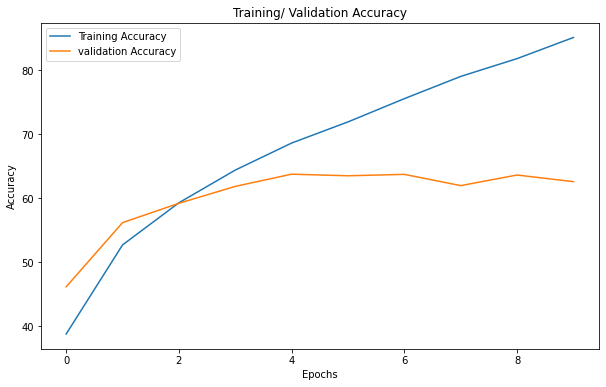
Number of tokens: 64

Head dimension: 48

Number of head : 8

Number of hidden layer in multi layer perceptron: 512

Dropout probability: 0.2



The model is overfitting.

The training loss in the final epoch is around 85 % and the validation loss is around 62 %

The test loss is also around 62 %

**Model performance for patch size (8X8):**

Token dimension: (8X8X3) = 192

Number of tokens: 16

Head dimension: 192

Number of head : 8

Number of hidden layer in multi layer perceptron: 512

Dropout probability: 0.1

Chart, line chart

Description automatically generated

The training loss in the final epoch is around 92 % and the validation loss is around 55 %

The test loss is also around 54 %

**Model performance for patch size (16X16):**

Token dimension: (16X16X3) = 768

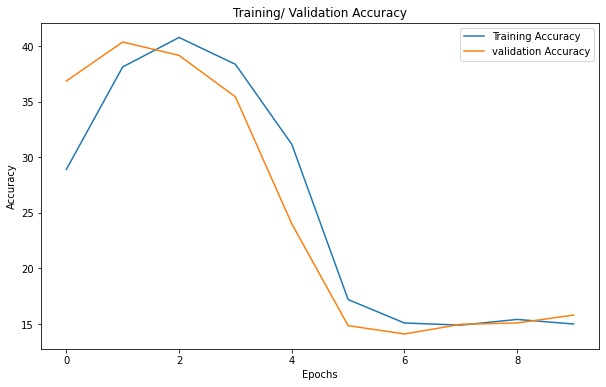
Number of tokens: 4

Head dimension: 768

Number of head : 8

Number of hidden layer in multi layer perceptron: 1024

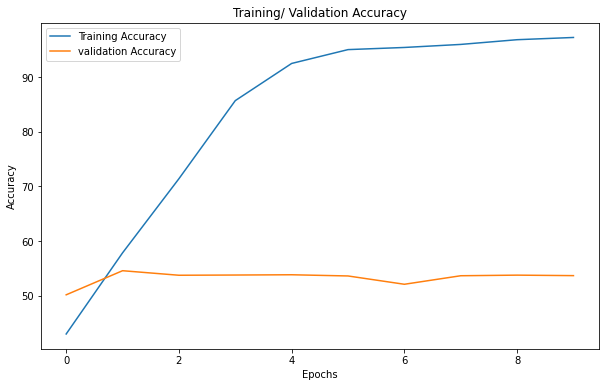
Dropout probability: 0.1



The training loss in the final epoch is around 15 % and the validation loss is around 15 %

The test loss is also around 15.58 %

The result and plot is strange so I changed the learning rate to 10^(-4)



The training loss in the final epoch is around 97 % and the validation loss is around 53 %

The test loss is also around 53 %

Observation :

* As the patch size is decreased the model accuracy is reaching around 60 percent at most.
* The transformer model is overfitting in most of the cases
* Overlapping patch shows little improvement than non overlapping patches.

(d)

Model performance on varying number of heads:

The model parameters I am using

Patch size = 4X4 and overlapping with stride half the patch size

Token dimension: (16X16X3) = 48

Number of tokens: 64

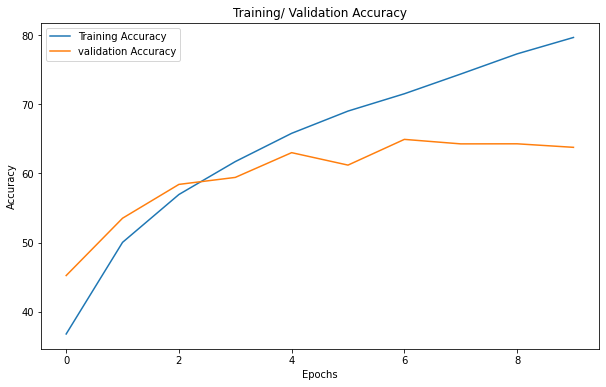
Head dimension: 48

Number of hidden layer in multi layer perceptron: 512

Dropout probability: 0.1

Learning rate is 0.001

For heads : 4



* The training accuracy in the final epoch is 80 %.
* The validation accuracy in the final epoch is around 64 %
* The test accuracy is around 63 %

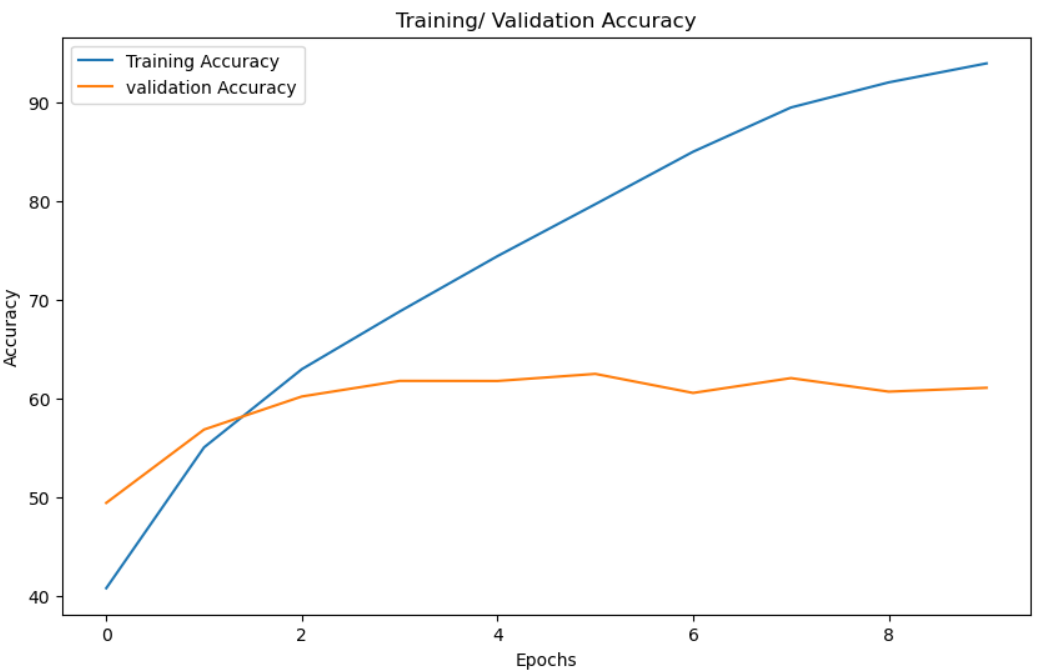
For heads : 8

Chart, line chart

Description automatically generated

* The training accuracy in the final epoch is 87 %
* The validation accuracy in the final epoch is 63 %
* The test accuracy is 62.7 %

For heads : 16



* The training accuracy for the last epoch is 94 %
* The validation accuracy for the last epoch is 61 %
* The test accuracy is around 61 %

(e)

The performance of the CLS token from different layers:

The model parameters I am using

Patch size = 4X4 and overlapping with stride half the patch size

Token dimension: (16X16X3) = 48

Number of tokens: 64

Head dimension: 48

Number of head : 8

Number of hidden layer in multi layer perceptron: 512

Dropout probability: 0.1

Learning rate is 0.001

Layer 0

* The model accuracy for the first layer cls token used for classification are 30.92 %

Layer 1:

* The model accuracy for the second layer cls token used for classification are 42.18 %

Layer 2:

* The model accuracy for the second layer cls token used for classification are 53.57 %

Layer 3:

* The model accuracy for the second layer cls token used for classification are 62.22 %