**NN & Deep learning Project 2022**

**STEM:**

For making images to patches we have used Einops and attained non-overlapping patches then the patches are transformed into a feature vector and stored in a matrix.



Now coming to backbone part of our model we have built two multilayer perceptron’s one input is the output of another layer.

And also added a non-linear activation function which is Relu.

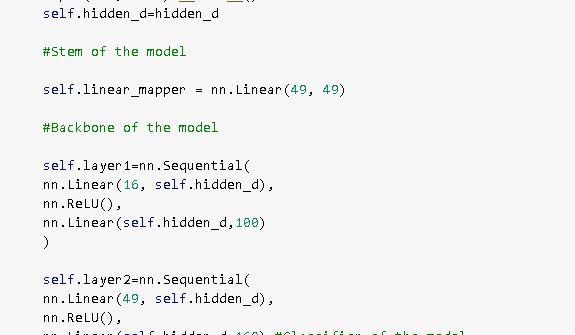
**ADVANTAGES OF ReLU :**

Relu does not activate all the neurons at the same time.

Other benefit is the reduced likelihood of the gradient to vanish.

**CLASSIFIER:**

We have used Linear classifier here. Which will constrict any value to a certain range, generally (0,1) or (-1,1). This makes it far easier for us to generate decision boundaries to determine classes.



To calculate loss, I have used cross entropy loss which is mainly used as it can describe how likely a model is and its error function of every data point.

**OPTIMIZER’S:**

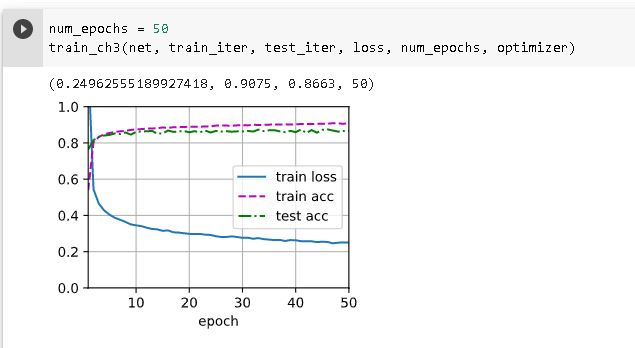
For optimizer I have used ADAM optimizer over SGD optimizer. Cause Adam is an algorithm for gradient-based optimization of stochastic objective functions. It combines the advantages of two SGD extensions — Root Mean Square Propagation (RMSProp) and Adaptive Gradient Algorithm (AdaGrad) — and computes individual adaptive learning rates for different parameters

For SGD optimizer I got an accuracy of 83.4 but using ADAM optimizer the accuracy slightly increased to 84 and when using SGD optimizer, the computing time increased compared to the computation time while using ADAM optimizer.

ADAM uses estimations of first and second moments of gradient to adapt the learning rate for each weight of the neural network.

**CURVES FOR THE EVOLUTION OF LOSS:**

The curves of the training loss decrease gradually as the epochs increase. This indicates we are approaching towards the global minima.



**CURVES FOR THE EVOLUTION OF TRAINING AND TEST ACCURACIES:**

As we can see from the graph above the training accuracy is slightly higher than the test accuracy this indicates its slightly overfitting.

**HYPERMATERS USED:**

I have used 50 epochs to train our model which got an accuracy of 86.6.

The accuracy of the model increased slightly as we increased our epochs.

In Adam optimizer I have used the learning rate as 0.01 cause with low learning rates the improvements will be linear. With high learning rates they will start to look more exponential. Higher learning rates will decay the loss faster, but they get stuck at worse values of loss.

And the weight decay I have set it to 0.

**ACCURACY OF THE FINAL MODEL:**

**The final model accuracy is 86.663**

.