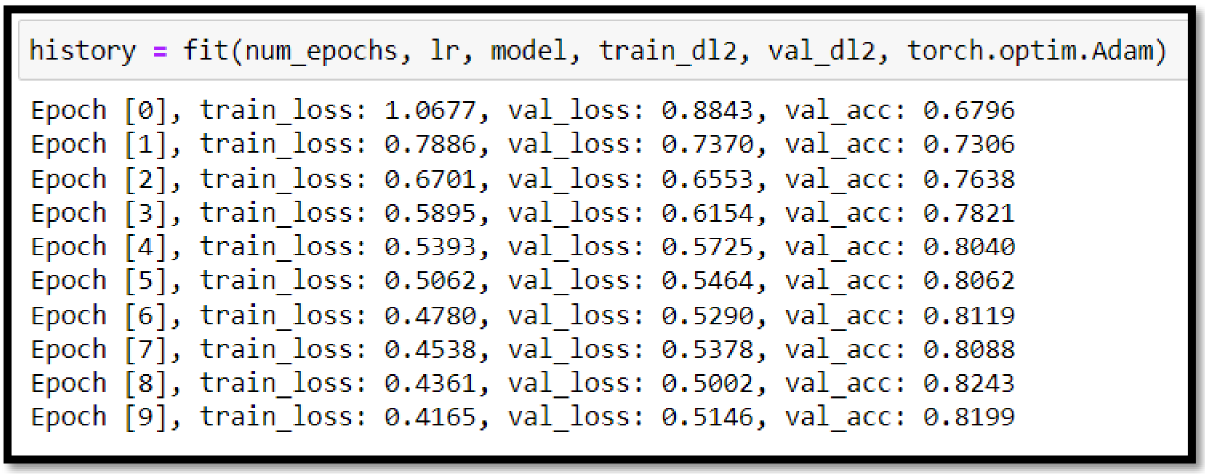
Assignment 3

Ans-a]

The training data's mean and standard deviation are used to normalize the tensor image. We scale the tensor images to an equal range via normalization, such that all images contribute equally to the overall loss. Normalization also ensures that all tensor images have the similar learning rate.

Ans-b]



```
history = fit(num_epochs, lr, model, train_dl2, val_dl2, torch.optim.Adam)

Epoch [0], train_loss: 1.0677, val_loss: 0.8843, val_acc: 0.6796
Epoch [1], train_loss: 0.7886, val_loss: 0.7370, val_acc: 0.7306
Epoch [2], train_loss: 0.6701, val_loss: 0.6553, val_acc: 0.7638
Epoch [3], train_loss: 0.5895, val_loss: 0.6154, val_acc: 0.7821
Epoch [4], train_loss: 0.5393, val_loss: 0.5725, val_acc: 0.8040
Epoch [5], train_loss: 0.5062, val_loss: 0.5464, val_acc: 0.8062
Epoch [6], train_loss: 0.4780, val_loss: 0.5290, val_acc: 0.8119
Epoch [7], train_loss: 0.4538, val_loss: 0.5378, val_acc: 0.8088
Epoch [8], train_loss: 0.4361, val_loss: 0.5002, val_acc: 0.8243
Epoch [9], train_loss: 0.4165, val_loss: 0.5146, val_acc: 0.8199
```

Figure 1: Training the model

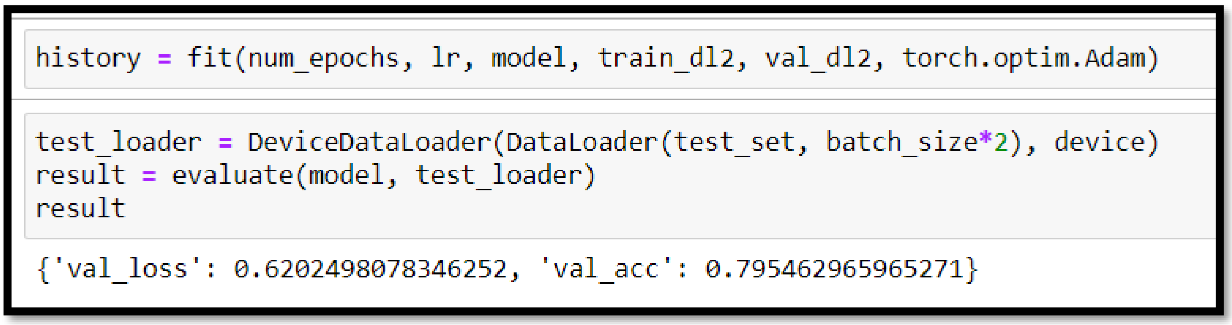
We use convolutional neural network as a classifier to predict the tissue type.

Through the use of appropriate filters, a convolutional neural network can properly capture the spatial and temporal relationships in an image. Due to the reduced number of parameters and reusability of weights, the architecture performs superior fitting to the image dataset. The network can be trained to better understand the image's complexity.

By calling the fit( ) method it is going to pass the batches of images from the dataset and pass them through the CNN model, get the output, calculate gradients, perform gradient descent and change the weights and biases that will reduce the loss and repeat the process for all the batches over the entire epoch. At the end of the epoch it is going to evaluate the model on the validation dataset.

So, from the results we see that after the 10th epoch we get a validation accuracy of 82%. Thus, we provide the evidence that the model has trained correctly.

Ans-c]

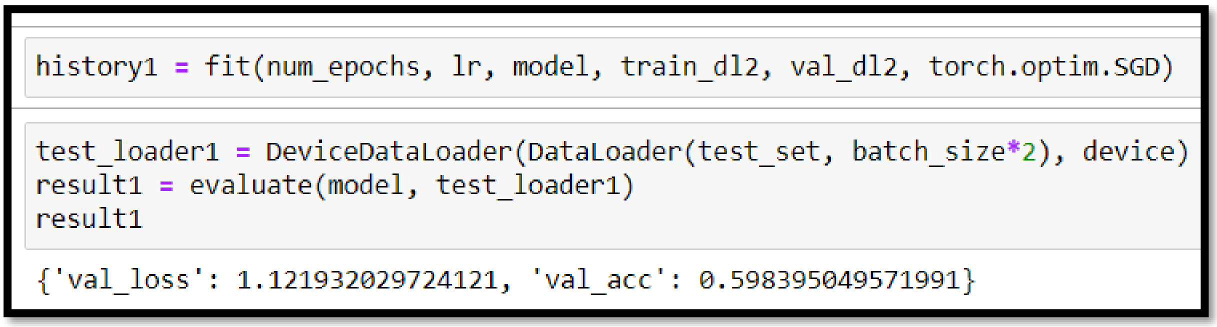


```
history = fit(num_epochs, lr, model, train_dl2, val_dl2, torch.optim.Adam)
```

```
test_loader = DeviceDataLoader(DataLoader(test_set, batch_size*2), device)
result = evaluate(model, test_loader)
result
```

```
{'val_loss': 0.6202498078346252, 'val_acc': 0.795462965965271}
```

Figure 2: Using Adam optimizer



```
history1 = fit(num_epochs, lr, model, train_dl2, val_dl2, torch.optim.SGD)
```

```
test_loader1 = DeviceDataLoader(DataLoader(test_set, batch_size*2), device)
result1 = evaluate(model, test_loader1)
result1
```

```
{'val_loss': 1.121932029724121, 'val_acc': 0.598395049571991}
```

Figure 3: Using SGD optimizer

We here optimize the hyperparameter 'optimization function' by choosing the best optimizer between Adam and SGD.

By using Adam optimizer to train the model the overall accuracy on the testing dataset is almost 80% [from fig: 2] whereas by using SGD as the optimizer we get an accuracy of only 60% [from fig: 3].

Thus, we choose Adam optimizer and it helps to train the model better such that the model performs well in classifying the tissue types better in the unknown dataset.

The evaluate( ) function average out the loss and accuracy across the batches of data from test_set and return a single output. 'val_acc' and 'val_loss' actually represent 'test_accuracy' and 'test_loss' respectively.

Ans-d]

Using 'Accuracy' as the metric for the validation and test set.

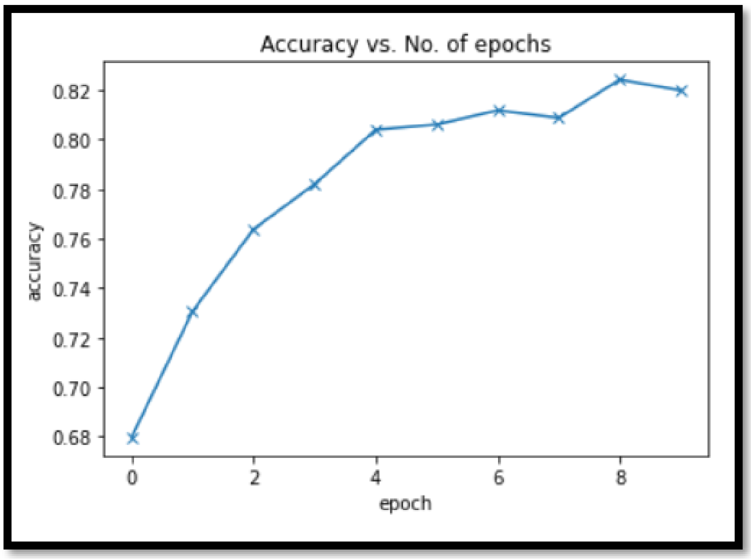


Figure 4

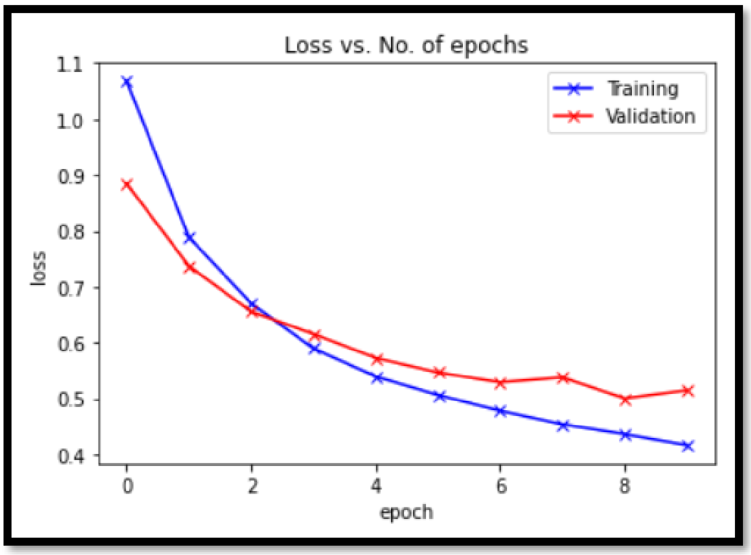
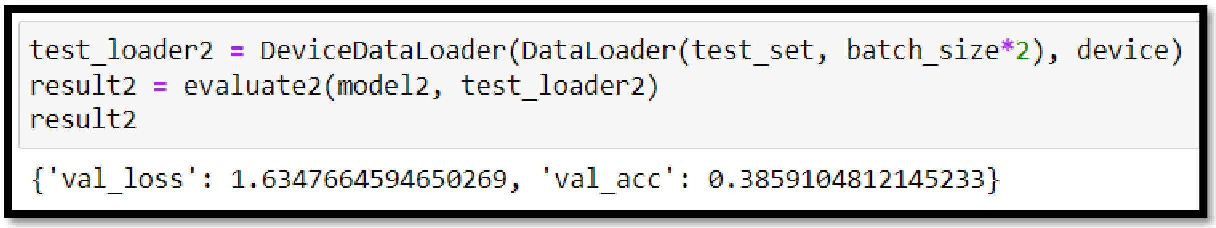


Figure 5

Our model reaches as accuracy of around 82% [from fig: 4] on the validation dataset and on the testing data the model has an accuracy of around 80% [from fig: 2]. This indicates our model is good and decent in predicting the tissue type.

Initially, both the training and validation losses seem to decrease over time. However, the validation loss stops decreasing after 'epoch 8' and even starts to increase [from fig: 5], this may be due to overfitting. It happens because the model tries to memorize certain patterns in training data.

Ans-e]



```
test_loader2 = DeviceDataLoader(DataLoader(test_set, batch_size*2), device)
result2 = evaluate2(model2, test_loader2)
result2
```

```
{'val_loss': 1.6347664594650269, 'val_acc': 0.3859104812145233}
```

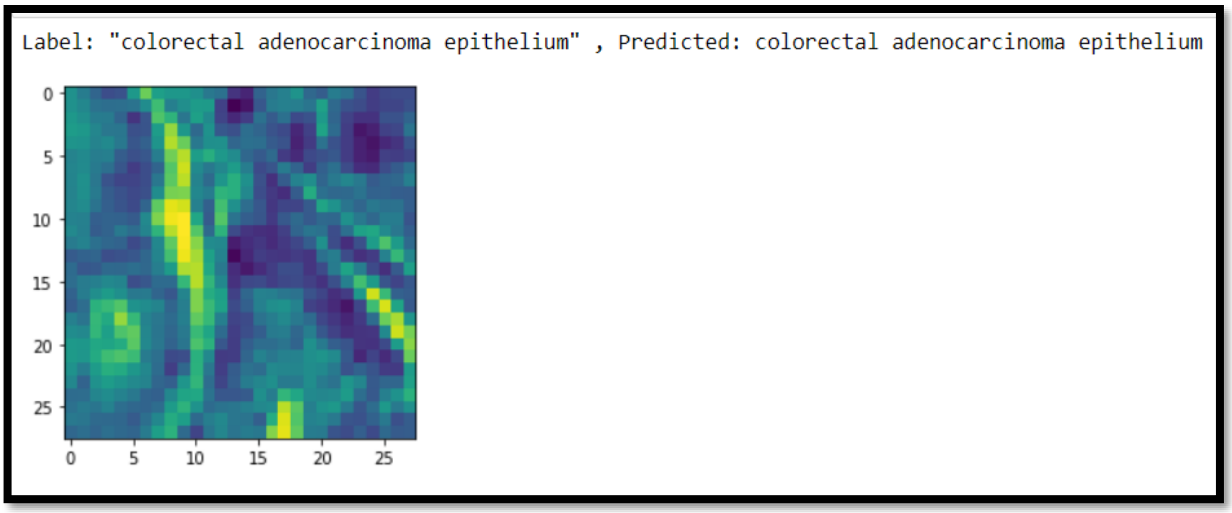
Figure 6

The accuracy in predicting the tissue type is only around 39% [from fig: 6] by using the logistic regression. This is because the model assumes linear relationship between pixel intensities and image labels and the model is also not able to capture non-linear relationships between inputs and targets.

Our CNN model by using two convolutional layers is able to extract high-level features giving a wholesome understanding of the images in the dataset. The non-linearity is also captured by passing the feature maps to the Relu activation function. Thus, CNN model is able to predict the tissue type with an accuracy of 80% [from fig: 2].

Hence, CNN model is much better than logistic regression model.

Prediction Process:



```
Label: "colorectal adenocarcinoma epithelium" , Predicted: colorectal adenocarcinoma epithelium
```

Figure 7: Image Prediction

From the above image we see that our CNN model is able to predict the tissue type accurately.

The input image is first converted to dimension 1x28x28 and passed to the model. We get the predictions from the model and these are nine outputs (9 tissue types) which are the scores given by the model for each tissue type. The highest score is selected and the label corresponding to the highest score is then our predicted output.

Questions:

You will be using the pathMNIST dataset, which contains 100,000 28x28 pixel images (x3 RGB colour channels). It contains image patches from hematoxylin & eosin stained histological images. There is a separate test dataset consisting of 7180 images (recorded from a different geographical location) The task for this assignment is:

Use a deep learning classifier to predict the tissue type. You should:

- Undertake appropriate pre-processing
- Train a deep neural network classifier of your choice and show evidence that the model has trained correctly.
- Show that you have considered hyperparameters (e.g. architecture of the model) and devise and implement a strategy to optimize at least one hyperparameter.
- Report suitable validation and test set metrics – it is up to you to decide what is 'suitable'
- Comment on your results and compare them to a simple baseline model (e.g. logistic regression)