ML A2 (Neural Network) Report

Part b

Batch Size	Hidden Layers	Initial Learning Rate	Max Iterations	Accuracy	Loss Function
128	50	2	40 epochs	35.42	Cross-entropy
128	100	2	40 epochs	48.812	Cross-entropy
128	200	2	40 epochs	54.681	Cross-entropy
128	100,50	2	40 epochs (early break)	2.17	Cross-entropy
128	75,50	2	40 epochs (early break)	2.17	Cross-entropy
128	50	2	40 epochs	77.42	MSE
128	100	2	40 epochs	54.957	MSE
128	200	2	40 epochs	49.594	MSE
128	100,50	2	40 epochs (early break)	2.17	MSE
128	75,50	2	40 epochs (early break)	2.17	MSE
128	**100,50**	**5**	**40 epochs**	**80.725**	**MSE**

Some of the observations on running various architectures with different parameters are as written in the table above **(Best Performing Model in Bold)**. The effects of varying various parameters are listed below:-

- **The number of hidden layers**:- The variation in the number of hidden layers was linked to several other hyper-parameters. Increasing the number of hidden layers

required extra training (increased number of iterations) or a higher learning rate or both.

- **Learning Rate**:- Low learning rate caused the model to converge very slowly while higher learning rates caused the model to overshoot the minima of the loss function. Whenever the average loss over an epoch increased the learning rate was decreased for the model to be able to converge.
- **Max Iterations:**- In the cases of learning rate where loss generally decreased, an increased number of iterations allowed the model to gain higher accuracies, while too high a number of iterations caused the model to overfit the data. With learning rates where loss did not decrease, the model usually oscillated or caused the gradients in earlier rates to vanish which resulted in no convergence of model even for a high number of iterations.
- **Loss Functions**:- It was observed that in general Mean squared error loss function (MSE) allowed the model to converge faster due to higher gradients whereas the cross-entropy loss function converged very slowly and also required smaller learning rates to be trained correctly.
- **Batch Size**:- On decreasing the batch size to too low values, the model took a long time to train because the low batch size caused the neural net to generalize over local trends in data rather than global trends.
- **Activation Function**:- This was kept fixed as asked in the Note.

Part c

Batch Size	Hidden Layers	Initial Learning Rate	Max Iterations	Accuracy	Activation Function	Loss Function	Extra Feature
128	**200**	**0.01**	**40 epochs**	**92.928**	**relu**	**Cross-entropy**	**-**
128	50	0.01	40 epochs	86.174	relu	Cross-entropy	-
128	200	0.01	40 epochs	76.275	relu	MSE	-
128	100,50	0.01	40 epochs	67.682	relu	Cross-entropy	-
128	200	0.01	40 epochs	88.585	tanh	Cross-entropy	-
128	150,75	5	40 epochs	77.652	tanh	MSE	-
128	200	0.01	40 epochs	82.739	relu	Cross-entropy	FFT
128	200	0.01	40 epochs	82.681	relu	Cross-entropy	DCT

The results of experimenting with other non-linearities are given in the above table **(Best Performing Model in Bold)**. The observations are as follows:-

- It was observed that tanh and relu perform better with cross-entropy loss function because of their higher derivatives. The lower derivative of the cross-entropy loss function is balanced out by the higher derivatives.
- Lower learning rates give a much faster convergence than in sigmoid where higher learning rates were required.
- Multi-layered networks are easier and faster to train than the sigmoid activation function.

- Using FFT and DCT on the best performing models reduced the accuracy. This can be due to the fact that the new values (concentrated towards higher frequencies) require more time to train than the normal pixel values which are more spread out.

Other features researched:-

- Gabor filter:- These filters are used in 2D image processing to detect textures. They isolate components of fixed frequency. They are basically gaussian distributions modulated with a sine wave of the required frequency. This is useful for text identification as text images are high-frequency images.
- DCT:- DCT is a form of Fourier transform where any number series is represented as a sum of weighted cosines of different values. This is helpful in image compression as lower value frequencies can be ignored to represent images without changing human perception quality of that image.
- FFT:- FFT or fast Fourier transform is used to convert a series of values/signal into its components for different frequency waves. This can be used in text image classification as text images have more high-frequency components than normal images.
- HOG:- HOG or histogram of oriented gradients is usually used for object detection in images. It looks for gradients within local portions of images.