Signal Processing in Practice

Assignment 2

Due Date: 17/03/2025

General Instructions:

- You can use Google Colab to run the codes. You can use any deep learning libraries like PyTorch, Tensorflow, etc. for the experiments.
- Please submit a brief one page report for the observations.
- Submit the codes and the report in a single zip file. Name this zip file in the format *SPP_< YourName>_Mixup.zip*.

MixUp: Interpolating Data for Robust Representations. (20 Marks)

Load the CIFAR10 dataset using torchvision.datasets.CIFAR10 in PyTorch. Normalize the images to the range [-1,1] using torchvision.transforms. Create a simple CNN with 3 convolutional layers, batch normalization, and ReLU activations. Use cross-entropy loss and the Adam optimizer for the training part. Train the CNN Under Three Different Conditions:

1. (Marks = 4) Train for 20 epochs using only raw images.
2. (Marks = 4) Traditional Augmentations: Apply random horizontal flips, random crops, and color jittering on the images, then train for 20 epochs.
3. (Marks = 12) Mixup: Mixup is used as a data augmentation technique where the images and labels in each minibatch are linearly interpolated to create new examples as follows:

$$\tilde{x} = \lambda x_i + (1-\lambda)x_j \; ; \qquad \tilde{y} = \lambda y_i + (1-\lambda)y_j \tag{1}$$

where, (x_i, y_i) and (x_j, y_j) are two different datapoints, and $\lambda \sim Beta(\alpha, \alpha)$. You can use torch.distributions for the Beta distribution.

Report the accuracies for all the three cases, and plot the training and validation loss curves. What do you observe? Does Mixup perform better than traditional augmentations? Why do you think this happens?

Experiment with different values of α, and report your observations on the strength of the mixing coefficient. Also, visualize some image augmentation results from Mixup with these different coefficients.

Note: You can use a subset of the dataset for faster training.

Manifold Mixup: Mixup in the feature space (10 Marks)

This is an extension of the Mixup concept, where the interpolation occurs in the feature space instead of the input space.

Consider the same CNN architecture as the previous question, and perform manifold mixup as follows:

$$\tilde{g}_k = \lambda g_k(x_i) + (1-\lambda)g_k(x_j) \; ; \qquad \tilde{y} = \lambda y_i + (1-\lambda)y_j \tag{2}$$

where, g_k denotes a randomly selected kth layer of the CNN model.

Calculate the accuracy for the test set and plot the training curves. Report your observations. You can experiment with different values of k and see the effect.