Advanced Machine Learning Assignment 1

Utpalraj Kemprai(MDS202352), Aman (MDS202305), Ankan Kar (MCS202303)

September 2024

Task 1 : Classifier for Fashion MNIST

Data

The dataset contains 70,000 grayscale images, each 28x28 pixels, of 10 types of clothing, shoes and other material. The dataset is already split into train and test data. The train data has 60000 images and the test data has 10000 images.

Data Preprocessing

We performed the following data preprocessing steps:

1. Converted the images into tensors
2. The tensors are normalized to have mean 0.1307 and standard deviation 0.3081
3. We relabeled the images in the training and the test data into 3 classes (clothes, shoes and other)

CNN classifier

We created a CNN classifier with the following structure:

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 12, 26, 26]	120
Conv2d-2	[-1, 24, 21, 21]	10,392
Conv2d-3	[-1, 32, 5, 5]	27,680
Linear-4	[-1, 200]	25,800
Linear-5	[-1, 10]	2,010

Figure 1: CNN Architecture

Training the CNN

For training our CNN we choose cross entropy as our loss function and learn the optimal weights using Stochastic Gradient Descent (SGD) with learning rate 0.001 and momentum 0.9. We train our model for 10 epochs with a batch size of 64.

Performance on Fashion MNIST dataset

We achieved the following results on the test data after training the model.
Average loss: 0.0375, Accuracy: 9887/10000 (98.87%)

Effect of permuting pixel images

We apply a permutation to all images in the Fashion MNIST dataset. We then trained our CNN on the scrambled training data with exact same hyperparameters as before. We achieved the following results on the scrambled test data.

Average loss : 0.0312, Accuracy : 9908/10000 (99.08%)

Task 2: Resnet-18 for Emotion Detection

Data

The dataset contains 35887 examples of 48x48 gray scale images of faces divided into train and test dataset. Images are categorized based on the emotion showed in the facial expressions (happiness, neutral, sadness, anger, surprise, disgust, fear). 28709 images for training data and 7178 examples for test data.

Data Preprocessing

We performed the following data preprocessing steps:

1. Resize the images to 224x224
2. Converted grayscale image to 3-channel RGB by duplicating the grayscale channel
3. Converted the images to tensors
4. Normalize the image using mean and std for ImageNet-trained ResNet. ImageNet mean and standard deviation for RGB: mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]

Training Resnet-18 model from scratch

For training the Resnet-18 model we choose cross entropy as our loss function and learn the optimal weights using Stochastic Gradient Descent with learning rate 0.001 and momentum 0.9. We train the model for 10 epochs with a batch size of 64.

Performance of the Resnet-18 model trained from scratch

We achieved the following results,
Test set: Average loss: 1.5938, Accuracy: 3311/7178 (46.13%)

Finetuning a pretrained Resnet-18 model for the dataset

1. We used the pretrained Resnet-18 model from PyTorch
2. We modified the final layer to match the number of classes (7 in our case)
3. We freeze all layers except layer4 and the final layer in the pretrained model.
4. We trained the pretrained model with the same hyperparameters as the model that was trained from scratch

Performance of the pretrained Resnet-18 model on test data

We achieved the following results,
Test set: Average loss: 1.6123, Accuracy: 4363/7178 (60.78%)

Comparing the results

The from-scratch approach achieved 46.13% accuracy, while fine-tuning boosted it to 60.78%. This highlights the value of pretrained models in capturing features from large datasets for specific tasks. Although average loss slightly increased during fine-tuning, the significant accuracy gain demonstrates its effectiveness for emotion detection applications.