Machine Learning (CS60050)

Assignment-3

Parth Jindal, Pranav Rajput

Group : 59

Abstract

This report contains an analysis for building an Artificial neural network for classification of satellite imaging based on the Statlog (Landsat Satellite) Data set
The following tasks are covered in the report:

- Analysis of the dataset to find appropriate hyperparameters for the ANN.
- Vary the number of hidden layers and number of nodes in each hidden layer.
- Plot the results for learning rate vs accuracy for each model and learning rate
- Reduce the input space dimension using PCA and plot the dataset onto a 2d-plane.
- Apply MLP on the reduced feature space.

Index

Data analysis

The database consists of the multi-spectral values of pixels in 3x3 neighbourhoods in a satellite image, and the classification associated with the central pixel in each neighbourhood. The aim is to predict this classification, given the multi-spectral values. In the sample database, the class of a pixel is coded as a number.

The database is a (tiny) sub-area of a scene, consisting of 82 x 100 pixels. Each line of data corresponds to a 3x3 square neighbourhood of pixels completely contained within the 82x100 sub-area. Each line contains the pixel values in the four spectral bands (converted to ASCII) of each of the 9 pixels in the 3x3 neighbourhood and a number indicating the classification label of the central pixel.

The number is a code for the following classes:

1. red soil
2. cotton crop
3. grey soil
4. damp grey soil
5. soil with vegetation stubble
6. mixture class (all types present)
7. very damp grey soil

Note: There are no examples for class 6.

Data preprocessing

Standard normalization is done for the pixel values by recentering them by their mean and scaling by their standard deviation. The mean and standard deviation are calculated differently for all 4 spectral bands

Hyperparameters

The input dimension is 36 hence 36 nodes are required in the input layer.
The output dimension is 6, hence 6 nodes are required in the output layer.

Other hyperparameters of concern:

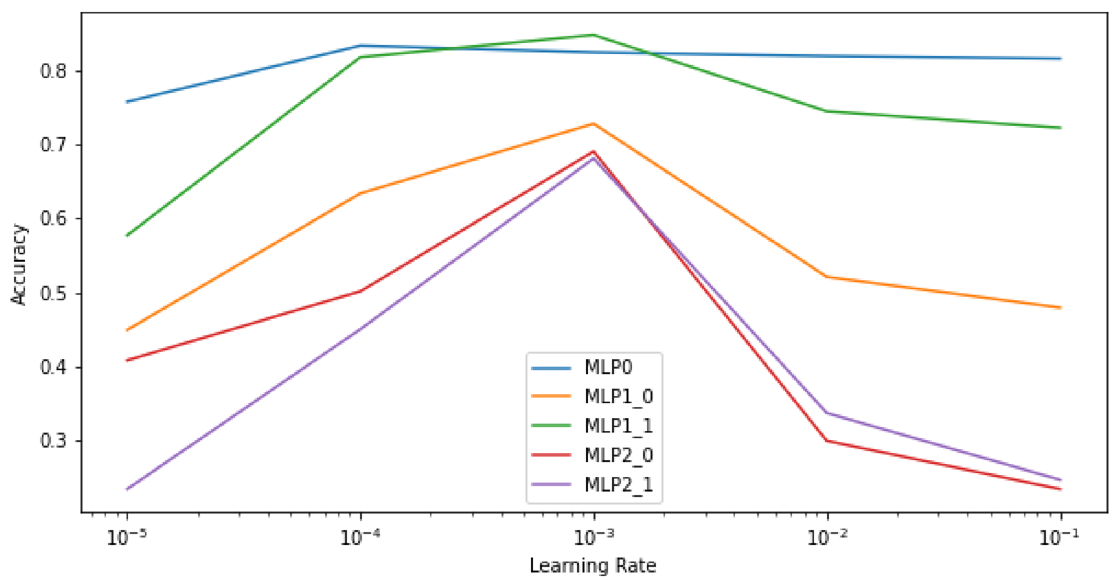
- Learning rate
- Non-linearity function
- Epochs
- Model under consideration

Varying learning rate for different model architectures

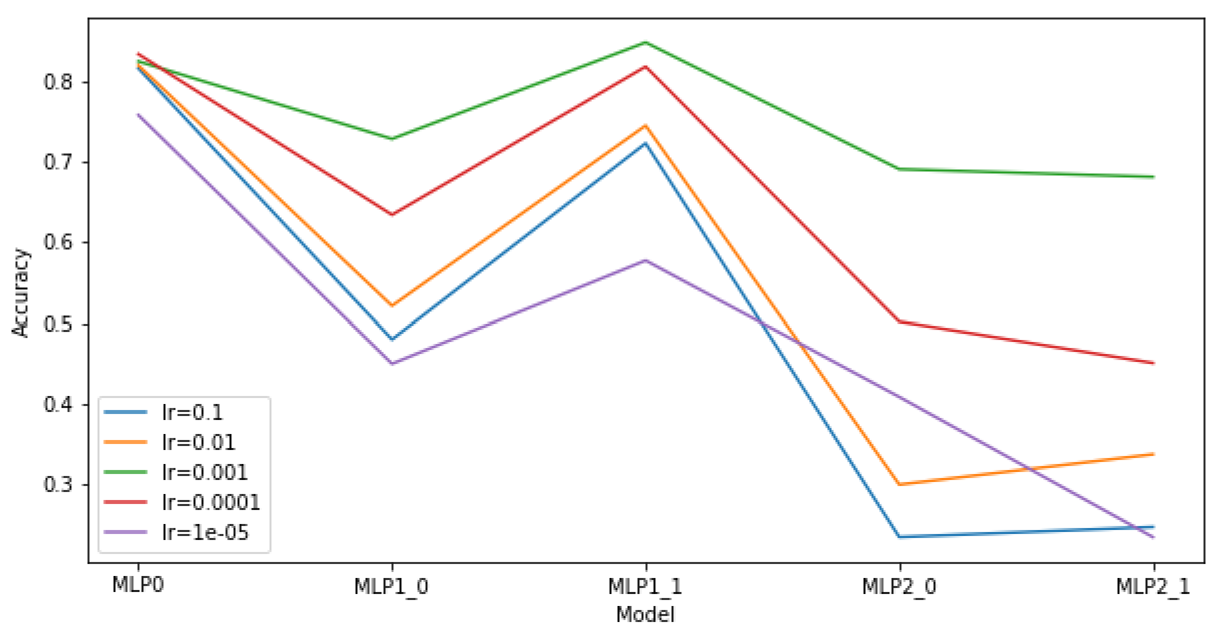
Note: The model legend description is as follows:

- MLP0 : MLP with no hidden layer
- MLP1_0: MLP with 1 hidden layer of 2 nodes
- MLP1_1: MLP with 1 hidden layer of 6 nodes
- MLP2_0: MLP with 2 hidden layers of 2 and 3 nodes
- MLP2_1: MLP with 2 hidden layers of 3 and 2 nodes

1. Learning rate vs Accuracy for each model



2. Model vs Accuracy for all learning rates



Model parameters for Best model

The best found model is a single hidden layer Multi layer perceptron with 6 nodes in the hidden layer. Other important hyperparameters in consideration are:

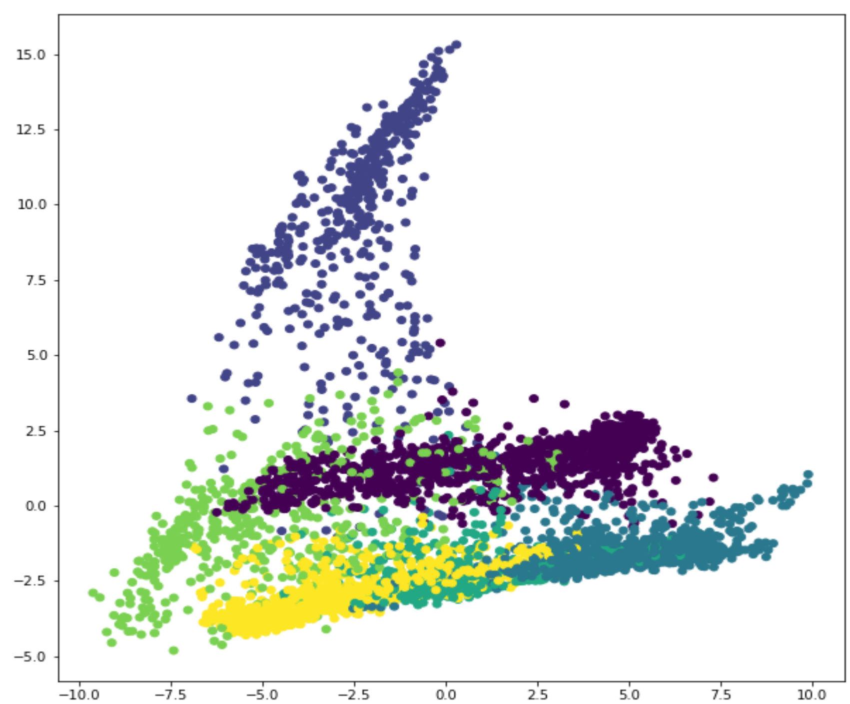
- Learning rate: *0.001*
- Optimizer: *Adam*
- Gradient descent algorithm*: SGD*
- Non-linearity: Sigmoid

Note: Sigmoid activation has been used instead of the usual ReLU activation because the number of nodes in the hidden layers are very less.
There are perceivably many reasons for this being the best model under consideration.

- This model has the highest number of learnable parameters hence the solution space is the biggest amongst all.
- The hidden layer unlike other models (except linear model) has equal no. of nodes as the output layer hence is not a bottleneck shaped model which acts like a compression model with as many independent variables as in the bottleneck layers. The data under consideration might not be so correlated as suggested by such architectures.
- The learning rate of 0.001 is sufficiently small for convergence under the lipschitz continuity bounds and large enough to achieve convergence fast unlike very small gradients such as 0.00001.

PCA transformed Dataset representation

After applying PCA based on the training dataset the following representation is learned as represented in 2d space with different colors representing different classes



MLP for PCA reduced dataset

The following results were obtained for all the models based on the learning rate obtained from step 3 (0.001):

Model	Accuracy
0 hidden layers	78.13 %
1 hidden layer with 2 nodes	67.93 %
1 hidden layer with 6 nodes	78.48 %
2 hidden layers with 2 and 3 nodes resp.	66.83 %
2 hidden layers with 3 and 2 nodes resp.	69.73 %

Steps to run the code:

1. Run the following command to install all dependencies:
   `pip install -r requirements.txt`
2. To run the experiments use main.ipynb either from vscode or jupyter-notebook server.

Code description:

Major classes:

- *SatelliteDataset:* Pytorch compatible dataset class
- *DatasetTransform*: Callable to modify training data on the fly from the dataloader itself
- *MultiTransform*: Callable extension to apply multiple callables one by one
- *TransformPCA*: PCA wrapper for doing PCA reduction on dataset
- *MLP0*: torch model class for 0 hidden layers
- *MLP1*: torch model class for 1 hidden layers
- *MLP2*: torch model class for 2 hidden layers

Major functions:

- seed_everything: seed the experiment to have repeatable results
- create_dataloaders: return pytorch data loaders
- preprocess_dataset: normalize the dataset with mean and std_dev
- train: trains the given model for a single epoch with Cross entropy loss function and returns the final acc and loss
- tests: test the given model on test dataset and return accuracy
- fit: wrapper function to train and test a given model and return lots of metrics and plots