In the name of GOD

Machine Learning Final Project

Title:
Classification of satellite dataset

Professor:
Dr. Shoaran

Producers:
Paniz Hossein Pour Ardeshiri
Reza Farhanghi

January 26, 2021

Introduction

The aim of this project is to classify satellite images. Classification of satellite images is one of the critical methods in interpreting satellite images that have many applications in studying Earth's changes. The main issue in this research is to find the optimal method in order to classify satellite images with high spatial resolution. Satellite dataset has been used for classification.

Data Set Information:

The Landsat satellite data is one of the many sources of information available for a scene. The interpretation of a scene by integrating spatial data of diverse types and resolutions, including multispectral and radar data, maps indicating topography, land use, etc. It is expected to assume significant importance with the onset of an era characterized by integrative approaches to remote sensing (for example, NASA's Earth Observing System commencing this decade). Existing statistical methods are ill-equipped for handling such diverse data types. Note that this is not true for Landsat MSS data combined in isolation (as in this sample database). This data satisfies the important requirements of numerical and at a single resolution, and standard maximum-likelihood classification performs very well. Consequently, for this data, it should be interesting to compare the performance of other methods against the statistical approach.

Frame of Landsat MSS imagery consists of four digital images of the same scene in different spectral bands. Two of these are in the visible region (corresponding approximately to green and red regions of the visible spectrum), and two are in the (near) infra-red. Each pixel is an 8-bit binary word, with 0 corresponding to black and 255 to white. The spatial resolution of a pixel is about 80m x 80m. Each image contains 2340 x 3380 such pixels.

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 neighborhood 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 neighborhood and a number indicating the classification labels of the central pixel. The number is a code for the following classes:

Number Class:

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, NB: There are no examples with class 6 in this dataset.

The data is given in random order, and certain lines of data have been removed, so you cannot reconstruct the original image from this dataset.

In each line of data, the four spectral values for the top-left pixel are given first followed by the four spectral values for the top-middle pixel and then those for the to-right pixel, and so on, with the pixels read out in sequence left-to-right and top-to-bottom. Thus, the four spectral values for the central pixel are given by attributes 17,18,19, and 20. If you like, you can use only these four attributes, while ignoring the others. This avoids the problem which arises when a 3x3 neighborhood straddles a boundary.

Attribute Information:

The attributes are numerical, in the range 0 to 255.

The original Statlog (Landsat Satellite) dataset from UCI machine learning repository is a multi-class classification dataset

Description: X = Multi-dimensional point data, y = labels (1 = outliers, 0 = inliers)

Due to the development of machine learning and deep learning methods and the remarkable results of these methods in classifying on different datasets, in this research, these methods have been used to classify satellite images.

In the field of machine learning, Logistic Classifier, SVM Classifier, and Bagging Classifier have been used, and in the field of deep learning, ANN Classifier has been used.

Problem definition

The purpose of this project is to classify satellite images. To this classification, it is needed to have a cognition of satellite images. The dataset used in this project is Statlog (Landsat Satellite) Data Set that has been obtained from UCI Machine Learning Repository.

The original Statlog (Landsat Satellite) dataset from UCI Machine Learning Repository is a multiclass classification dataset. Here, the training and test data are combined. This dataset was multiclass, the smallest three classes, 2, 4, 5 are combined to form the outliers class, while all the other classes are combined to form an inlier class, so this dataset was changed to a two-class dataset. The combined classes of 2,4,5, have been named as class ,1 and the combination of other classes have been named as class 0. This dataset contains 6435 samples that each one of these samples has 36 attributes. The format of this dataset is (.mat). The process of this project has been done on the two-class dataset that have been obtained from: http://odds.cs.stonybrook.edu/satellite-dataset/.

The programming language used for this project in Python. Due to the powerful ready libraries of this programming language, this programming language has been used. According to the selected algorithms, Pre-processing must be done on the dataset.

Data Pre-processing is an integral step in Machine Learning and Deep Learning as the quality of data and the useful information that can be derived from it, directly affects the ability of these models to learn; therefore, it is essential that data must preprocessed before feeding it to these models.

The data Pre-processing process is done according to the equation (1):

$$z = \frac{x_i - \mu}{\sigma} \tag{1}$$

In this formula:

z is the standardized data, x_i is the attribute i data, μ is the mean of attribute i data, σ is a deviation from the standard of attribute i data.

In standardization data, values are transformed such that the mean of the value is 0 and the standard deviation is 1.

Proposed Methods

1) Logistic Classifier:

The Logistic classifier algorithm is one of the machine learning algorithms that has been chosen for this classification task. The reason for choosing this method is probabilistic results that indicate each sample of data belongs to which class with what probability.

The method of operation of this algorithm is that it first considers a zero initial weight vector (W=0) and uses a reduction gradient algorithm to update weights at each stage.

The reduction gradient of each step is obtained from the equation (2):

$$\nabla E_{in}(w_t) = -\frac{1}{n}\sum_{i=1}^{n} \frac{y_i x_i}{1+e^{y_i w_t^T x_i}} \tag{2}$$

In this formula:

$\nabla E_{in}(w_t)$ is reduction gradient in step t, n is the number of samples of data, y_i is the class of sample i, x_i is the attribute of sample i, w_t is the weight vector in step t

After calculating the reduction gradient in each step, the next step is to update weights by the equation (3):

$$w_{t+1} = w_t - \eta \nabla Ein(W_t) \tag{3}$$

In this formula:

w_{t+1} is the weight vector in step t+1, w_t is the weight vector in step t, η is the learning rate,

$\nabla E_{in}(w_t)$ is the reduction gradient in step t
According to this knowledge, the class of samples is computed by the f equation (4):

$$h(x) = \frac{1}{1+e^{-wTx}} \quad (4)$$

In this formula: $h(x)$ is the hypothesis function found for sample x

There are two conditions for $h(x)$:

a) If $h(x) > 0 \cdot 5$, then y = 1
b) If $h(x) \leq 0 \cdot 5$, then y = -1

Implementation of this algorithm is done by Python programming language, and for convenience, ready-made libraries of this programming language have been used.

For this purpose, hyperparameter C should be tuned. The interval between 0.1 to 2 have been divided to 50 part, and in each step, one of these parts have been replaced to C and Logistic Classifier algorithm fitted on training data. In each step of use of values for C and fitting training data for this algorithm, the accuracy of this algorithm is computed on the validation data.

By considering the model accuracy on validation data with different values for C, the value of C that causes the best accuracy of the model on validation data, is chosen as best value for hyper parameter C. Using the best value for C, this model is fitted on the mixture of training and validation data. At last, the accuracy of the trained model is computed on the test data. For more information, the accuracy of this model is calculated for on the combination of training and validation data.

The accuracy of the model on data is computed as the following formula:

acc = number of correctly classified samples/number of all samples

2) SVM Classifier:

Another method used for classification is SVM, which has remarkable accuracy for this classification task. This algorithm uses a linear model that finds the closest sample data of different classes, and according to these sample points, it draws support vectors. Considering the dimensions of data attributes, this method draws the classifier line or plane or hyperplanes. The samples in the sides of the dividing line or plane or hyper planes, shows the different classes.

This algorithm uses the formula : $wx + b = 0$, that w is weight vector and x is sample attribute, and b is bias. Lagrange and Dual functions are used to find the best weight vector that causes better classifier.

Implementation of this algorithm is done by ready libraries. In this algorithm, there are three hyperparameters; C, gamma, kernel. To find the optimal values of these hyper parameters, a grid search ready library is used. Grid search algorithm is fitted on the training data to find the best hyperparameters, which cause the best accuracy.

Values for C: [0.01, 0.1, 1, 10, 100], Values for gamma: [1, 0.1, 0.01, 0.001, 0.0001]

Values for the kernel: linear, RBF but according to data and task RBF is chosen.

By examining this method, optimal values of these hyperparameters are found such that the model results in the best accuracy. After finding the best hyperparameters, this model is fitted on the mixture of training and validation data, and the accuracy of the model is computed on the test data. For more investigation, the accuracy of this model is considered on the combination of training and validation data. Finally, the test data classes are predicted by this model.

3) Bagging (Bootstrap Aggregation) Classifier:

There is no algorithm that is the best and most accurate algorithm in all conditions. For this reason, a group of base models is created to work together to create the final model. Combining the output of each of the base models will increase the accuracy of the model. This approach is called the ensemble methods. The main purpose of the ensemble is to increase the accuracy of the model. There are several approaches to combining models. One of these approaches is combining by consensus. In this project, the bagging algorithm, which is in the mentioned approach, has been used. Bagging increase, the accuracy and stability of the model.

The bagging algorithm works as follow:

First, it generates M datasets from one training dataset by bootstrap sampling. Then, using the single algorithm, the training is performed on M generated datasets and generates M base models. Finally, the outputs of these M base models are combined by voting.

In this project, bagging is done for the SVM algorithm with the best hyperparameters. For this model, bace_estimator must be specified which in this model is SVM. Then n_estimator must be defined, which is the number of estimators. One more thing that could be defined is max_samples, which is the number of samples that choose from each dataset to train each base learner. Finally, the accuracy of the model is measured on the test data. Test set classes are also predicted by this model.

4) Artificial Neural Network (ANN) Classifier:

An artificial neuron network (ANN) is a computational model based on the structure and functions of biological neural networks.

ANNs are deep learning models capable of pattern recognition and machine learning.

Elements of a Neural Network:

- **Input Layer:** This layer accepts input features. It provides information from the outside world to the network; no computation is performed at this layer; nodes here just pass on the information(features) to the hidden layer.
- **Hidden Layer:** Nodes of this layer are not exposed to the outer world. The hidden layer performs all sorts of computation on the features entered through the input layer and transfer the result to the output layer.
- **Output Layer: This** layer brings up the information learned by the network to the outer world.

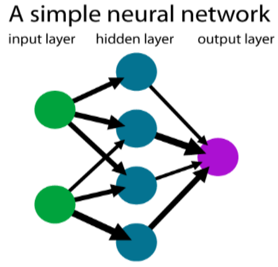

Figure 1- Diagram of a simple feed-forward artificial neural network.

The Artificial Neural Network receives input from the external world. Weights are placed on the connections between neurons in different layers. There is also a weighted connection for bias in the hidden and output layers. First, the weighted sum of the input data enters the hidden layer. And then, the output of each neuron in hidden and output layers is computed by the following equation (5):

$$y = f\left(\sum w_i x_i + b\right) \quad (5)$$

In the above equation, the weighted sum of the data in the previous layer of the considered neuron is computed. The bias of the considered neuron is then added to it. Finally, it passes through an activation function and results in the output of the desired neuron.

For the neural network, layers and activation functions are considered as follows:

- Hidden layer 1: 32 neurons, ReLU activation
- Hidden layer 2: 32 neurons, ReLU activation
- Dropout layer: Dropout(0.5)
- Output Layer: 1 neuron, Sigmoid activation

RELU: Stands for the Rectified linear unit. It is the most widely used activation function. It is chiefly implemented in hidden layers of Neural network.

- Equation: A(x) = max(0,x). It gives an output x if x is positive and 0 otherwise.
- Value Range**:** [0, inf)

Sigmoid Function:

Because the output is for binary classification then, the sigmoid function is a very natural choice for the output layer.

- It is a function which is plotted as **'S'** shaped graph.
- Equation: $A = 1/(1 + e^{-x})$
- Value Range: 0 to 1
- Uses: Usually used in the output layer of binary classification, where the result is either 0 or 1, as value for sigmoid function lies between 0 and 1 only so, result can be predicted easily to be **1** if the value is greater than **0.5** and **0** otherwise.

In this model, the Dropout layer is applied between the last hidden layer and the output layer. It is a technique that is designed to reduce the possibility of overfitting in the model. "Binary_crossentropy" is used for loss because there are two class zero and one. "Adam" is used as an optimizer for this model.

Now the model is retrained on the combination of train and validation sets. Then the accuracy of the model is measured on the test set. The accuracy of the model is also measured on the combination of train and validation sets. Test set classes are predicted by this model.

Experiments

List of questions that these experiments are designed to answer:

1.What is the accuracy of the models on the train set?
2.What is the accuracy of the models on the test set?
3.What are the predicted classes of the model on the test set?

Details: observation

In this project, Logistic Classifier, SVM Classifier, Bagging Classifier, and ANN Classifier are used for classification. First, the data has been splitted into three random sets, 60% as train, 20% as validation, and 20% as the test.

I) Logistic Classifier Model:

In this issue, the classes of data are changed from **0** and **1** to **-1** and **1.** Also, the attributes of the data are standardized.

The Logistic classifier model is defined. This model has one hyperparameter that should be tuned. The hyperparameter is C. In order to tune C, the range 0.1 to 2 is divided into 50 equal parts. Each time, the model is trained on the train set by selecting any of the C values. And the accuracy of the model is measured on the validation set. The value of C that leads the most accuracy on the validation set, is considered as the best hyperparameter.

The best value for C is: 0.1

Using this value, the best validation accuracy is: 87.95%

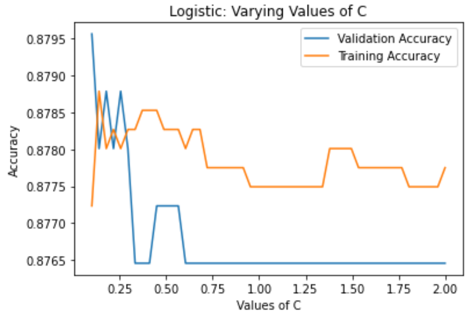

Figure 2- validation and training data accuracy by varying values for C in Logistic Classifier

The model with the best hyperparameter is now retrained on the combination of train and validation sets, and the results reviewed as follow:
Accuracy of model, on train and test sets:
train_acc_Logistic_Classifier = 87.91763791763792 %
test_acc_Logistic_Classifier = 86.7132867132867 %
Predicted classes of this model on test set:
y_predicted_ test _Logistic_Classifier: [-1 -1 -1 ... 1 -1 1]

II) SVM Classifier Model:

In this model, the classes of data are also changed from 0 and 1 to -1 and 1. The three hyperparameters of this classifier are C, Kernel, and gamma. The grid Search method is used in order to tune these hyperparameters.
The values which are considered to tune hyperparameters are as follow:
C: [0.01, 0.1, 1, 10, 100], gamma: [1, 0.1, 0.01, 0.001, 0.0001], kernel: ' RBF '

The grid search algorithm is implemented on the combination of train and validation set with the above values as hyperparameters. This algorithm results in the best hyperparameters for the most accurate SVM model.
The best hyperparameters are as follow:
C = 10, gamma = 0.1, kernel: ' RBF '
The SVM Classifier with the best hyperparameters is now retrained on the combination of train and validation sets, and the results reviewed as follow:
Accuracy of model on train and test sets:
train_acc_SVC_Classifier = 97.24164724164724 %
test_acc_SVC_Classifier = 94.32789432789433 %
Predicted classes of this model on test set:
y_predicted_ test _SVC_Classifier: [-1 -1 -1 ... 1 -1 1]

III) Bagging Classifier Model:

In this algorithm, the base estimator is SVM Classifier that has been achieved in the previous step. Hyperparameters of this model is the number of estimators and max samples. Using different values for these hyperparameters, this model is fitted on the training data, and the accuracy of the model is measured on the validation data.

Best values for hyperparameters of this model that causes the best accuracy of model on the validation data is found as:

n_estimator = 63, max_sample = 1.00, Using these values, max validation accuracy is: 93.62 %

This algorithm is retrained on the combination of training and validation data, and results reviewed as follow:

Accuracy of model on the training and test data:

train_acc_Bagging_SVC_Classifier = 97.3970473970474 %

test_acc_Bagging_SVC_Classifier = 94.4055944055944 %

Predicted classes of this model on the test data:

y_predicted_test_Bagging_SVC_Classifier: [-1 -1 -1 ... 1 -1 1]

IV) ANN Classifier Model:

Hyperparameters of ANN are as follow:

Number of layers, number of neurons of each layer, activation function of each layer, loss, optimizer, batch size, number of epochs.

In this algorithm, classes are used as 1 and 0. The structure of this model is constructed as follow:

In order to achieve the best accuracy of this model, hyperparameters of this model are tuned as:

Two hidden layers with 36 neurons for each one of these layers and activation function relu are implemented. After these layers, one dropout layer with rate = 0.5 is implemented to avoid overfitting. Finally, the output layer is constructed with one neuron and activation function sigmoid.

Compilation of model is done with loss = binary cross entropy and optimizer = Adam and metrics = accuracy.

The model is fitted on the training data with epochs = 500 and batch size = 32. The accuracy of the model on the validation data is considered in each step of epochs.

Model's loss and accuracy on data in terms of epochs are drawn as fig (3):

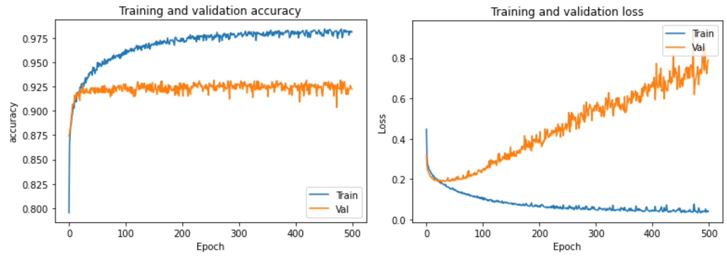

Figure 3- accuracy and loss of ANN model on the training and validation data

Considering these charts, the best number for epochs to have the best accuracy is obtained 200. Epoch = 200 is chosen in order to avoid overfitting.

After doing this, the previous model using suitable hyperparameters is constructed, and then has been fitted on the combination of training and validation data, and results are as follow:

Accuracy of the model on the training and test data:

train_acc_ANN_Classifier = 98.15462231636047 %

test_acc_ANN_Classifier = 93.16239356994629 %

Predicted classes of this model on the test data (0 classes are changed to -1):

y_predicted_test_ANN_Classifier: [-1 -1 -1 ... 1 -1 1]

The results:

A few examples of the classes predicted by these models on test data:

Actual class	Logistic Classifier	SVM Classifier	Bagging SVM Classifier	ANN Classifier
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
1	1	1	1	1
1	-1	1	1	1
…	…	…	…	…
-1	-1	-1	-1	-1
-1	-1	-1	-1	-1
1	1	1	1	1
-1	-1	-1	-1	-1
1	1	1	1	1

Table 1- samples of predicted classes by these models

The accuracies of these models for this classification are as follow:

Accuracy of models on the mixture of training and validation data (train accuracy):

Logistic Classifier	SVM Classifier	Bagging SVM Classifier	ANN Classifier
87.917638	97.241647	97.397047	98.154622

Table 2- accuracy of models on the training data

Accuracy of models on the test data:

Logistic Classifier	SVM Classifier	Bagging SVM Classifier	ANN Classifier
86.713287	94.327894	94.405594	93.162394

Table 3- accuracy of models on the testing data

Accuracy values for ANN and Bagging SVM are different in each run of model.

ANN Classifier test accuracy = 92.5 ± 1

Bagging SVM Classifier test accuracy = 94.2 ± 0.3

Conclusion:

In this research, satellite image classification has been performed for satellite data. Logistic and SVM and Bagging SVM and ANN Classifiers are the algorithms that have been used for this classification. The most powerful methods for this classification are SVM and Bagging SVM Classifiers, and the weakest method is Logistic Classifier.

The performance speed of these algorithms is as follows:

Run Speed: Logistic Classifier > SVM Classifier > Bagging SVM Classifier > ANN Classifier

To classify this data in cases where the highest classification accuracy is considered, SVM Classifier or Bagging SVM Classifier algorithm should be chosen, on the other hand, on cases where high execution speed is considered, the Logistic Classifier algorithm can be preferred.