<https://github.com/Shahidmscs/Deep-Learning/tree/MLLP--CNN-CIFAR-10>

Abstract

Artificial Neural Networks (ANN) are becoming core domain of Artificial Intelligence. Generally, Machine learning and specifically, deep learning gain popularity in problem solving by virtue of Multi-Layer Perceptron (MLP), Convolutional Neural Network (CNN) and its back propagation. CNN is becoming a powerful and successful technique for a variety of computer vision and image analysis applications. Understanding and tweaking hyper parameters of CNNs is one of the challengeable tasks to achieve optimum accuracy in various image analysis tasks. This paper analyzes accuracy of CNN comparative to MLP for image classification on CIFAR-10 dataset.

MLP has shown about 55 percent accuracy in 300 epochs. Later on the same problem is tried to solve using CNN and optimizing various hyper parameters the accuracy is achieved 75 percent. It is concluded that image classification problems could be efficiently solved with CNN.

Introduction

Artificial intelligence (AI) is influencing almost every field of life through implementation of information technologies. In current era, AI generally refers to machine learning (ML) and more precisely ML refers to Deep learning (DL) which is Artificial Neural Networks (ANN) with Multi-Layer Perceptron (MLP).

Single layer perceptron are mostly used for solving linear problem however, additional layers may be placed among the input layer and the output layer to solve non-linear problems. These networks architectures with multiple hidden layers are called Multi-Layer Perceptron (MLP). MLP neural networks are remained successful and accurate for classification problems when input are ambiguous and variable. In this paper accuracy of CNN is analyzed for image classification by employing CNN on CIFAR-10 dataset. The dataset CIFAR-10 is considered as benchmark standard dataset in research for evaluation of image classification algorithms. This dataset contains 60,000 images and 10 classes, each a section containing 6000 images of each class. Accuracy of the MLP is analyzed by varying number of Convolutional Filters (32, 64 and 128 of size 3x3), activation functions on convolutional Layer and , number of neurons, varying batch size of training data. Padding parameter is not changed and kept the value ‘same’. It means that system automatically pad as much as needed to preserve the spatial dimensions. It means that after applying convolution the output size of the image will remain same as input size of the image

The Model has shown about 55% accuracy in 300 epochs. Therefore, it can be concluded that MLP can show this much optimum accuracy for image classification.

In this study experiments are performed by tweaking with hyper parameters to achieve optimum accuracy for image classification on CIFAR-10 dataset. The CIFAR-10 dataset containing 60,000 images, out of these 50000 are used in the training and 10,000 in the test set. Implementation started with input, middle and output layers and one output layer. Literature survey revealed that number of hidden layers can be determined by trial and error. There are total four types of experiments are performed as following:

1. Changing of **activation functions**
2. Changing Filter size (30 epochs, 3x3 and 5x5)
3. Increasing number of Convolutional Layers (100 epochs 2 x Conv 3 x Conv (3rd layer 128))
4. Increasing Batch Size
5. changing number of filters

Initially, experiments start with changing activation function and 300 number of epochs but accuracy was not increased more than 55 percent. There are two types of activation functions are used at input and hidden layer i.e., relu and sigmoid. The activation function at output layer is used is softmax because there are ten output classes are required to be predicted. In other words there are ten types of images containing in Cifar-10 Dataset therefore activation function relu and sigmoid cannot be used because both are used for binary classification problem. In this initial model accuracy was not improved after 300 epochs for both relu and sigmoid as shown in table-I.

After running 300 epochs with each activation function (relu and sigmoid) it was determined that optimum accuracy is achieved after 30 epochs. Therefore, number of hidden layer are increased to Two, Four, Six and Eight but accuracy still not achieved. More precisely, the accuracy increased very slightly till six number of layers, even gone down when number of layers increased to eight as shown in Table-II.

It was established that by increasing number of hidden layers accuracy cannot be improved. At this stage model reverted back to initial stage with input, muddle and output layer and increased number of neurons at middle layer. In initial model there were 512 neurons at middle layer but increased to 1024 and 2048 number of neurons. This option also could not worked out in terms of improving accuracy as shown in Table-III.

At this stage when accuracy was not improved even after increasing number of neurons. Batch size of training data was changed from 128, 256, 512 and 1024. This also has not increased noteworthy accuracy as shown in Table-IV.

**Methods**

This is an experimental and analytical study, in which performance of MLP is tried to investigate and analyzed by changing various parameters. The experiments started with one input layer, middle layers and one output layer with activation functions relu, sigmoid and softmax. Later on increased the number of hidden layers to four, six and eight. Input layer contains 3072 (32x32x3) number of neurons because each image is 32x32 and multiplied with 3 for color images. The middle layer contains 512 number of neurons, which is revealed from literature review for solving same type of problem \cite {ajala}. Output layer contains ten number of neurons because our dataset contains ten number of classes. This is how initial structure of MLP is formulated as shown in Figure below.

**Experiments for impact of activation functions**

There are two experiments performed for observing impact of activation function on accuracy of solving image classification problem on the dataset. In the first experiment on the MLP model training and testing performed for 300 epochs with the activation functions (relu, relu and softmax) at input, middle and output layer. In the second experiment all other parameter remained same with only change in activation functions i.e., (sigmoid, sigmoid and softmax) and same experiment is run for 300 epochs.

**Experiments for impact of Hidden Layers (Two, Four, Six and Eight)**

There are four experiments performed for observing impact of increasing hidden layer for accuracy of solving image classification problem on the dataset. The initial setup remained same and experiments performed with one input layer, middle layers and one output layer with activation functions relu, sigmoid and softmax. All the additional fully connected hidden layers of 512 neurons are increased with activation function sigmoid. As mentioned earlier that started with two Layers and output layer and then gradually increased by two additional layer in each iteration. During these experiments impact of time taken also observed.

**Experiments for impact of number of neurons at hidden Layer (512, 1024 and 2048)**

There are three experiments performed for observing impact of increasing number of neurons at hidden Layer for accuracy of solving image classification problem on the dataset. The initial setup remained same and experiments performed with one input layer, middle layers and one output layer with activation functions relu, sigmoid and softmax. Number of Neurons increased at middle layers only from 512, 1024 and 2048.

**Experiments for impact of Batch Size (128, 256, 512 and 1024)**

There are four experiments performed for observing impact of increasing the batch size of training dataset for accuracy of solving image classification problem on the dataset. By increasing the batch size number of steps are decreased in each iteration. The objective of this experiment was to increase accuracy of image detection and fast convergence to optimum accuracy values.

**Results and Analysis**

Results are analyzed for all four types of experiments in terms of improved accuracy and reduction of loss. The accuracy with all parameters changing and observed results is discussed with respect to changing parameters.

**Impact and analysis of activation functions**

Activation function of input and middle layer is changed which has slightly improved accuracy but not significant. Objective of changing activation function was to improve accuracy and reduce loss but significant change was not observed major accuracy around 50 percent was achieved at twenty to thirty epochs. Later on 270 epochs has only achieved four to five percent. The experiment was performed for 300 epochs as shown in figure below.

**Impact and analysis hidden Layers (Two, Four, Six and Eight)**

Hidden layers were increased because accuracy was not achieved after 300 epochs. Therefore, keeping in view complexity of the problem hidden layers were increased to four hidden layers, six hidden layers and eight hidden layer. This experiment has increased time required for each epoch from 15 seconds, 20 seconds and 25 seconds approximately. This variation also not worked significantly as shown in figure below.

**Impact and analysis number of neurons at hidden Layer (512, 1024 and 2048)**

Initial model was remained same and experiments performed by increasing number of Neurons increased at middle layers only from 512, 1024 and 2048. There is no significant change in accuracy is observed even accuracy reduced from 53 percent to 51 percent. Similarly loss is also increased from 1.29 to 1.34 as shown in figure below.

**Impact and analysis batch Size (128, 256, 512 and 1024)**

In vew of above, when accuracy was not much improved, the batch size of training data was increased. Initially, when batch size increased from 128 to 256 and 512 it has improved accuracy from 53 percent to 54 percent but later on when increased batch size to 1024 the accuracy again decreased to 53 percent and similarly no major change in loss is also observed as shown in figure below.

**Conclusion.**

It is concluded that MLP can classify images in given dataset with accuracy of almost 55 percent. Optimum accuracy about 50 percent achieved in initial twenty to thirty epochs. Later on, other epochs does not affect accuracy significantly. Similarly, changing activation function, increasing number of neurons and increasing number of hidden layers just add processing overhead without significant improvements in accuracy and loss. Therefore MLP is not a suitable solution for solving image classification problem.

which designs and builds an 8-layer

If the inputs are ambiguous and variable, the MLP is successful in classifying the data. Since MLP is one of the most important tools used in modeling and decision making, it can produce the best results from limited and incomplete databases.

multi-layer perceptron to solve image classification problem. For this task, you will be employing the CIFAR-10 dataset.

Artificial neural networks that are frequently used in machine learning and

inspired by the biological nervous system are computer systems. Learning in these

networks is achieved through examples [37,38]. Additional layer(s) are placed between

the input layer and the output layer to solve non-linear problems. MLP neural network

is preferred in this study because it gives high success rate especially in problems

requiring classification, recognition and generalization. Medical decision-making is a

versatile process and the goal is to make a correct diagnosis. In order to achieve this

purpose, it is necessary to find the appropriate data, to feature extraction of the data and

to analyze the new data. If the inputs are ambiguous and variable, the MLP is successful in classifying the data. Since MLP is one of the most important tools used in modeling and decision making, it can produce the best results from limited and incomplete databases. In addition, it is possible to increase the success if it is trained with different training algorithms. The overall architecture of the network is multi-layered as shown in

Figure 3.

While network architects are being determining, the number of hidden layers is usually determined by trial and error [39,40]. The network architecture used in this study is as follows; N\_Count\_Input\_X\_1\_Count\_Hiddenlayer\_X\_N\_Count\_Output. The "N" is a dynamic number that varies according to the number of segments.

The optimized ANN model in this work has two method which based only on the dominant pixel RGB (mean) and applying principle component analysis (PCA) on the pixel gradation values of each image. The optimized model was evaluated and validated through analysis of the performance indicators. Findings in this work have shown that both models have produced about 70% in diagnostic accuracy with more than 80% achievement for sensitivity. However, model with the applied PCA has lower network size.

Multilayer perceptron is a neural network model with multiple hidden layers [48], and the neurons between adjacent layers are connected [44]. The architecture of the model is shown in Figure 4, and the parameters of the MLP are summarized in Table 2. The parameter selection involved in the proposed MLP is based on experience and experiment. As for the hidden layers’ selection, the comparison experiment is conducted by setting 2, 4, 6 and 8 hidden layers, and the result (Figure 5a) shows that with the layer increased, the time cost will sharply increase, while the accuracy will not be improved. When the layer is set as two, the classification accuracy is low. Therefore, to balance the accuracy and time cost [49,50], four hidden layers are selected in this experiment. The number of neurons for each hidden layer is set according to the experience of multiple trials, and the principle is still balancing time cost and accuracy. The activation function and loss function are ReLU and softmax cross-entropy with logits, respectively. The flow of the extraction contains three steps: sample selection, model training and classification generation. • Sample selection: The training samples for each scene are manually labeled and cover all water types and non-water types. In this paper, all training samples are manually labeled in ENVI 5.3 using ROI for water and no-water. Each ROI consists of pixels within a polygon feature. The distribution of these samples is based on manual experience, and we try our best to improve representativeness and randomness of the spatial distribution. To ensure the accuracy of the sample, only identified non-water and water bodies will be selected as samples. The mixed pixels in coastal area, river banks and wetlands will not be considered. The sample numbers for all water and non-water types in each image are shown in Table 3. These samples are randomly divided into training samples and validation samples. According to the experiment, by setting different training sample percentages (Figure 5b), 80% of the total sample is used to train the model and to generate the fitting accuracy, and 20% of the total sample is used to verify the accuracy. To ensure the comparability of the algorithm accuracy, the training samples of the water and non-water bodies via the MLP and support vector machine are the same.

The simulation results demonstrate promising performance on the image classification models. At the end of the discussion, a deep comparative analysis has been carried out to identify the significance of the presented models. Among the presented models, the DCNN yields better accuracy of 99.83% for MNIST and 90.14% for CIFAR-10 than other methods.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Layer -1 | Layer -2 | Output Layer | epochs |
| Function | Sigmoid | Sigmoid | Softmax | 50 |
|  |  |  |  |  |
|  |  |  |  |  |