Image classification in Machine Learning

CS 273A - Machine Learning, Project Report

Shreyas Badiger, Nevedha Ravi, Lakshmipriyadarshini.

https://github.com/hard-fault/Residual_Neural_Networks

Abstract: In this report, we have evaluated some of the popular image classification techniques in machine learning and have finally used ResNet to analyze the CIFAR-10 dataset with Stochastic Gradient Descent(SGD) optimization. Furthermore, we have made attempts to better understand ResNet through some observations and experiments.

I Introduction

Image recognition is a vital component of many modern systems. Robotics, autonomous vehicles, security systems, social media, and many other applications rely on some nor the other form of image recognition system. With the onset of affordable smartphones which are enabled with high-quality cameras, the **images captured are of higher resolution** than it ever used to be. Even the CCTV cameras are pervasive for the sake of monitoring and security. With such rapid **growth of data, both in terms of volume and complexity**, it is critical to have highly efficient image classification systems. A system that is **accurate** with **little-to-no errors, resource-efficient** and less time-consuming. This was our motivation to choose the **CIFAR-10 dataset** for our project where we could try, evaluate and understand some of the popular ML techniques.

II Dataset Exploration

The CIFAR10 dataset consists of 60000 32x32 color images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.

Each data point in this dataset is a 32x32 image wherein each of the pixels is represented by 3 values denoting the values of red, blue and green that contribute to the color of the pixel. Thus each data point has 32*32*3 numbers(features) associated with it. Each data point can be visualized as shown in Fig 2.1

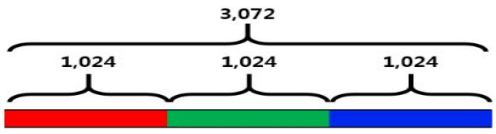

Fig 2.1 Composition of a data point in CIFAR10, 3072 features.

CIFAR-10 has 10 different classes of images - Aeroplane, Bird, Cat, etc. *[1]* The class labels are represented as numbers from 0-9. It is one of the **most frequently used image datasets** to introduce and validate new machine learning topics. One of the first classifications done on the CIFAR-10 dataset used a multinomial regression model. This model achieved an accuracy of **64.87%** *[2]*. Since then, various research papers have been published on image classification techniques and have now achieved an accuracy of around **97%** on CIFAR-10.*[3]*

We used the pickle library to load the dataset as mentioned in the documentation for CIFAR-10 and did some preliminary analysis to get a clear idea about the dataset.

```
Done loading batch 1...

Features.shape: (10000, 32, 32, 3)  # 10k data points in a batch, each with 32*32*3 features

Number of data points:  10000

Number of features:  3072
```

Fig 2.2 Shape of CIFAR-10 dataset

As shown above, each batch has 10000 data points. A sample image from the dataset :

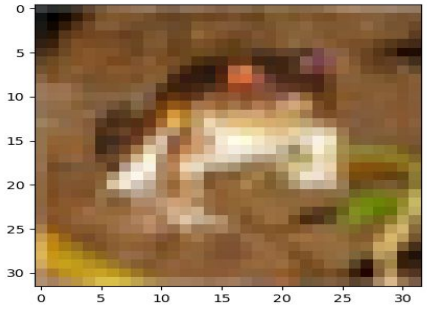

Fig 2.3 Sample image from CIFAR-10

Challenges with CIFAR-10:

As mentioned in the earlier sections, the image classification problem is a hard problem to be solved. This is due to the fact that an image could be captured from any angle /distance which would make it look like a different object altogether. CIFAR-10 has such images.

Comparing CIFAR-10 with faces dataset, we can clearly see how CIFAR-10 is more challenging to explore. Unlike human faces, the images in CIFAR-10 doesn't have a lot of common features within the same class. For example, images of horses are very different amongst themselves because of their possible orientations. To demonstrate this, we have showed the mean of all the horse images in CIFAR-10 and mean of all human faces from faces dataset. From Fig-2.4, it is clear that images of horses in CIFAR-10 are harder to learn than the human faces.

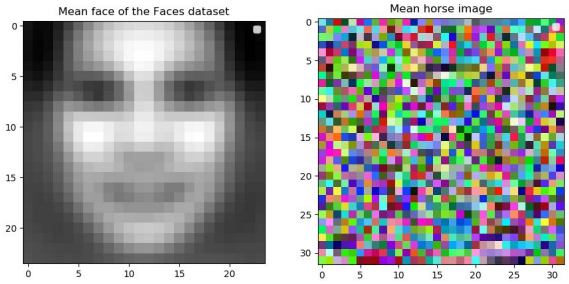

Fig 2.4 Comparing the mean pixels.

III MODEL SELECTION

Given the complexity of the problem and the dataset representing the same, we wanted to research how different machine learning models would fair. In the table below, we have listed the models we considered to explore and the percentage of accuracy obtained by using the same.

Classifiers	Accuracy(%)
SVM	37
Random Forest	48.9
MLP	49.6
CNN	89.8

Fig 3.1 Performance of different classifiers on CIFAR-10

Neural networks **was the obvious choice** because of its ability to deal with complex datasets. However, neural networks itself has **several variations** and we have explained in

the upcoming sections, the reasons for why we zeroed on **Residual Neural Networks** (**ResNet**).

IV Residual Neural Network

Neural networks are a set of computation entities, modeled closely after the human brain. They are **designed to recognize patterns**. Each node in the neural network performs similar functionality to a neuron in our brain. These nodes are stacked in multiple layers. Ideally, by increasing the number of layers(depth), we must see an increase in the accuracy. However, that is not true in the case of plain neural networks as shown in Fig 4.1.*[4]*. This is the case because when we increase the depth in neural network we face the problem of **diminishing gradient**. Whereas, when we increase the depth in residual neural networks(ResNet), it uses residual learning **ReLu** from the previous layers through **skip connections**, which **increases the accuracy without actually increasing the complexity**. In Fig 4.1 (right half) it is clear that ResNet-110 has an error% much lesser than ResNet-20 whereas error% for plain-56 is higher than plain-20. Since ResNet was introduced, it has been one of the most popular image classification neural network architecture. The **original paper received more than 20,000 citations** and the authors have achieved staggering accuracy in image classification**. Hence we chose to classify CIFAR10 using ResNet**.

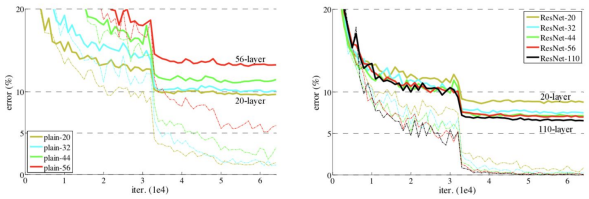

Fig 4.1 Comparison of plain Neural Networks and Residual Neural Networks on ImageNet dataset. *[4]*.

V Implementation Of ResNet

The **heavy-lifting** for the ResNet modeling is done by a python module called **Keras**. It is a high-level neural networks API that abstracts many aspects of the ResNet. Key functionalities offered by Keras - **initialize** the ResNet model instance, **preprocessing** the image, batch **normalization**, use **SGD** Optimizer, etc. The code we implemented is largely derived from the official webpage of Keras. *[5]*

The main steps in the ResNet modeling is to

- *Load data*
- *Preprocess the data for colors (RGB values) build the network (init model)*
- *Create a tensorboard*
- *Start training*
- *Check the performance against the validation set.*

SGD optimizer:

Each node in the ResNet works like a perceptron. This gave us an opportunity to optimize the parameters using the Stochastic Gradient Descent (SGD) method. As mentioned earlier, Keras provides us with a SGD implementation where we can set the desirable learning rate to train the model.

```
# set optimizer
sgd = optimizers.SGD(lr=.1, momentum=0.9, nesterov=True)
resnet.compile(loss='categorical_crossentropy', optimizer=sgd, metrics=['accuracy'])
```

The stochastic gradient descent algorithm was used here to find the optimal

parameter values for the model. The algorithm drives towards the best parameter values iteratively by trying to minimize the loss function. Given the large size of the dataset, it was best to use SGD over gradient descent as it doesn't use all the data points to calculate the gradient in an iteration.

VI OBSERVATIONS

We were curious to know how the hyperparameters (like **epoch value**, #of hidden **layers**, #of **nodes**) affect the classification accuracy for CIFAR-10.

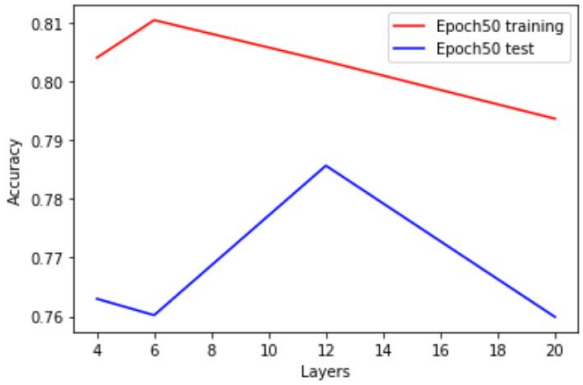

Fig 6.1 Accuracy vs #Layers when Epoch=50

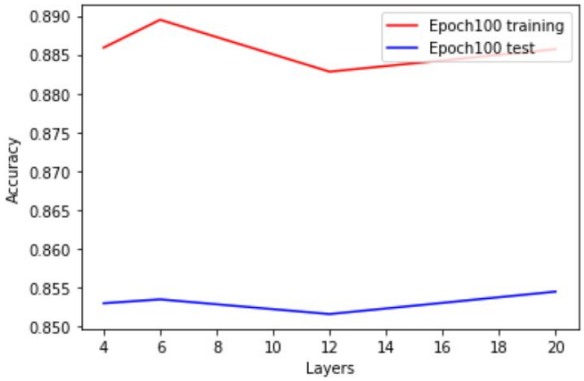

Fig 6.2 Accuracy vs #Layers when Epoch=100

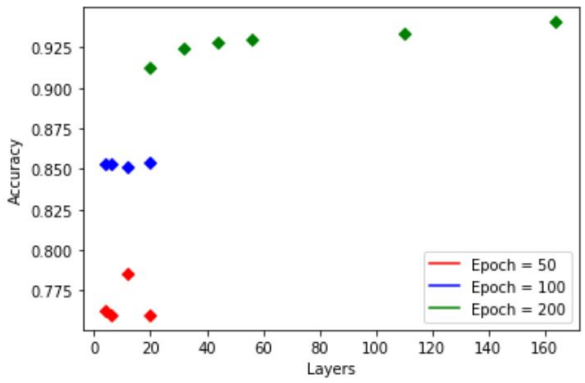

Fig 6.3 Accuracy vs #Layers (All)

From the experiments we ran, we can ascertain that small number of epochs doesn't perform the best due to underfitting. Similarly training the model using too many epochs causes the model to overfit to the training data. Hence we have used early stopping to stop training when performance on a validation dataset started to degrade.

Note: While we ran the experiments to obtain the observations in Fig 6.1 and 6.2. (The raw values are available in Github.) But the Green dots (higher epoch and layers) in Fig 6.3 were obtained from the paper summaries in Keras official page.[5]

VII PERFORMANCE VALIDATION

We fed data into the network in batches because it is not possible to feed all data at once. The number of iterations denotes the number of batches we feed in one epoch. When we trained the model with all the data, we had, i.e 391 iterations to feed the entire training data in batches. We observed the best performance with **92.53% validation accuracy**. But when we reduced the number of iterations (to 40), we fed lesser data for an epoch and the validation accuracy dropped to **86.83%.** This trend was expected because at

every epoch with a lesser number of data, the model doesn't learn well.

Fig 7.1 Layers=32, Epoch=200, iterations=40
VAL_ACCURACY = 86.83%

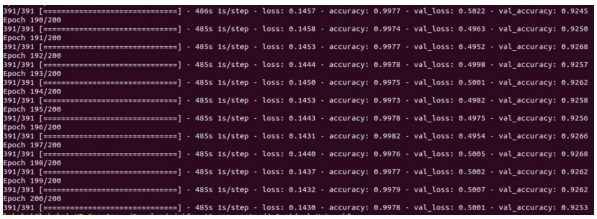

Fig 7.2 Layers=32, Epoch=200, iterations=391
VAL_ACCURACY = 92.53%

VIII Conclusion

The fundamental breakthrough with ResNet is that it allows one to train models with extremely deep neural networks (around 150+layers) successfully. This is indeed a very effective method to deal with complex computer vision problems. Prior to ResNet training very deep neural networks was difficult due to the problem of vanishing gradients.

References

[1] CIFAR-10 dataset official page
https://www.cs.toronto.edu/~kriz/cifar.html

[2] Learning Multiple Layers of Features from Tiny Images.
https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[3] CIFAR-10 top results.
http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html#43494641522d3130

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun. Deep Residual Learning for Image Recognition.
https://arxiv.org/pdf/1512.03385.pdf

[5] Keras example for CIFAR-10
https://keras.io/examples/cifar10_resnet/