COMP 551: MiniProject 2

Lucas Bennett, Bjørn Christensen, Megan Ng

Abstract

In this project we implemented a multilayer perceptron (MLP) from scratch, created a convolutional neural network (CNN) using existing libraries, and loaded a pre-trained model. All models were used to perform multiclass image classification on the CIFAR-10 dataset. Our most important findings were the following: (1) Non-linearity and increasing network depth resulted in higher accuracy, (2) choice in activation function impacts accuracy, (3) regularization prevents over-fitting to training data, (4) and data normalization is a critical pre-processsing step for a stable and optimized model.

1. Introduction

In this project we implemented a multilayer perceptron (MLP) from scratch, created a convolutional neural network (CNN) using existing libraries, and loaded a pre-trained model. All models were used to perform multiclass image classification on the CIFAR-10 dataset. The dataset is a collection of 60,000 32x32 color images in 10 classes, with 6,000 images per class. It is split into 50,000 training images and 10,000 testing images, with a balanced distribution of classes in each set. Our findings suggest that non-linearity and increasing network depth result in higher accuracy in multiclass image classification on the CIFAR-10 dataset using multilayer perceptron (MLP) models. We also found that the choice of activation function has a significant impact on accuracy, with leaky ReLU performing better than tanh and ReLU. Furthermore, regularization techniques such as L1 and L2 regularization were effective in preventing overfitting and improving generalization. Finally, we observed that data normalization is a critical preprocessing step for stable and optimized models. These findings are consistent with other studies that have used MLPs to analyze the CIFAR-10 dataset.

For example, Alizadeh and Rezaei (2016) implemented MLPs with different configurations and analyzed the effect of hyperparameters on classification accuracy [1]. One of their findings was that using activation functions such as ReLU and PReLU improved accuracy compared to sigmoid and hyperbolic tangent functions. This finding is consistent with our own observation regarding the impact of choice in activation function on accuracy. In another study, Xu et al. (2019) proposed a deep MLP architecture and focused on optimizing the hyperparameters of the model [2] They found that regularization techniques such as L2 were effective in preventing over-fitting and improving generalization, which is consistent with our own observation regarding the impact of regularization on accuracy.

2. Datasets

The CIFAR-10 data-set is a collection of 50000 training images and 10000 testing images. The contents of the images are labeled as belonging to one of ten classes: Airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck. Both the training and test sets contain an even distribution of each class, furthermore, each class is mutually exclusive from one another. For each image, 3 separate channels describe its R-G-B values, with each channel containing a 32 x 32 matrix describing the pixels image (each entry representing a single pixel). The values of the pixels range from 0 (no luminance) to 255 (max luminance), allowing us to normalize the data by dividing each entry by 255.

As our MLP network was designed to have a single input channel, the data must undergo vectorization before the model may be implemented. This process entails the concatenation of R-G-B values for a single pixel, along with the concatenation of rows in the pixel matrix. The end result is a 3072 long vector describing each image of the data-set.

3. Results

3.1. Task 3.1

We created three different models. (1) an MLP with no hidden layers, (2) an MLP with a single hidden layer having 256 units and RelU activations, and an MLP with two hidden layers each having 256 units with ReLU activations. All models had a softmax layers at the end. We observed that non-linearity and increasing network depth improves accuracy. These result were expected as both non-linearity and increasing network depth allows the models to learn complex data patterns beyond a linear relationship. Test accuracies in increasing order were as follows: Model (1) 41%, Model (2) 44%, Model (3) 44%. We suspected that the two layer model could perform significantly better given the appropriate activation function and were proven correct in task 3.2.

# of hidden layers	Test Accuracy
0	41%
1	44%
2	44%

Table 1: Testing Accuracies For Different # of Hidden Layers.

3.2. Task 3.1 cont.

In the previous task we only investigated the effect of increasing network depth and thought it would be interesting to test the effect of increasing network depth. Past literature has shown that increasing width increases a model's representational capacity, but also lead to over-fitting. We took our best MLP model, with two hidden layers and ReLU activations, and tested layer sizes of 32, 64, 128, 256, 512. We observed a slight increase in test accuracy with increasing layer size.

# of hidden units	Test Accuracy
32	41%
64	42%
128	44%
256	44%
512	45%

Table 2: Testing Accuracies For Different # of Hidden Units.

3.3. Task 3.2

We created two different MLP models both with two hidden layers, but one with tanh activations and the other with Leaky-ReLU activations. We found that accuracy increased going from. Test accuracies in increasing order were as follows: ReLU 44%, tanh 45%, and Leaky-ReLU 52%. Leaky-ReLU may have given the best results as it fixes the zero gradient problem and ensures that nodes with a negative value always has a non-zero gradient and can continue to learn

Activation Function	Test Accuracy
ReLU	44%
tanh	45%
Leaky-ReLU	52%

Table 3: Testing Accuracies For Different Activation Functions.

3.4. Task 3.3

We created two different MLP models both with two hidden layers each having 256 units and ReLU activations, but one with L1 regularization and the other with L2 regularization. Regularization prevents the model from over-fitting to the training data which is characterized by training accuracy increasing and testing accuracy either plateauing or decreasing. As expected adding regularization prevented over fitting as seen by the minimization of training and testing accuracy differences. Comparing final test accuracies, we found that the regularized models had a slightly lower accuracy. Hyper-tuning for lambda would serve to find the optimal parameter value that yields the best accuracy as a result of balancing over-fitting and generalization.

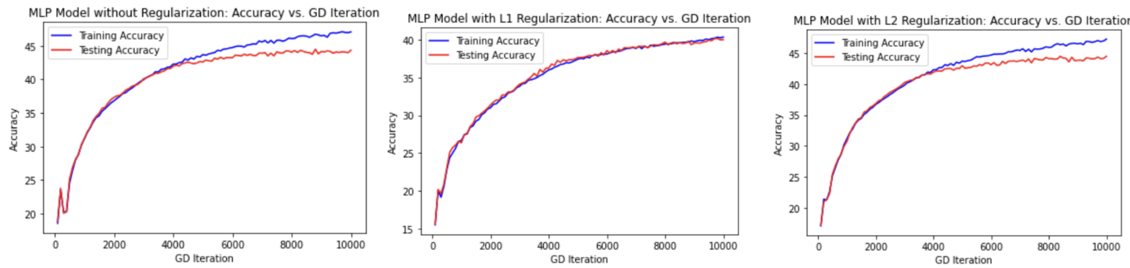
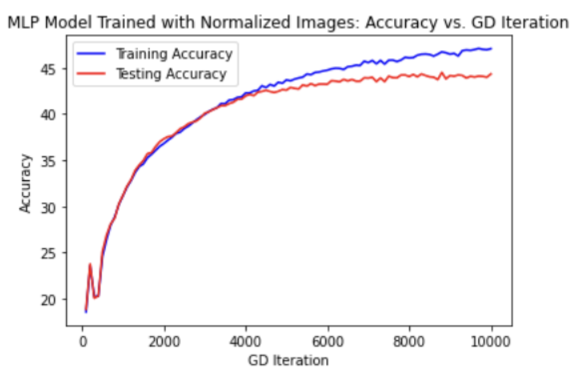


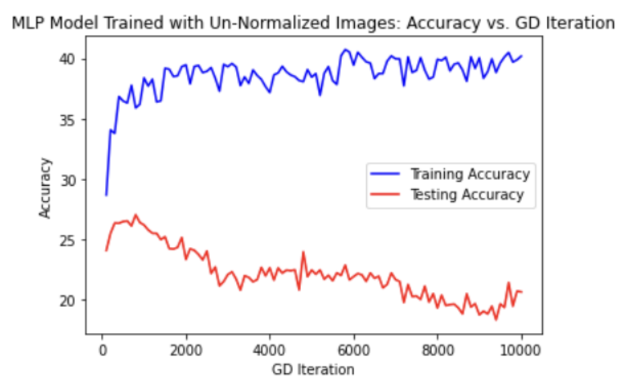
Figure 1: Train and test accuracies plotted over gradient descent iteration for an MLP without and with regularization.

3.5. Task 3.4

We created an MLP model with 2 hidden layers each having 256 units and ReLU activations and trained it with un-normalized images. We found that test accuracy significantly decreased from 44% to 21%, and that in general the model was much more unstable and over-fit. This confirms that data normalization is an critical pre-processing step that helps with optimization and preventing over-fitting.



(a) Normalized images



(b) Un-normalized images

Figure 2: Train and test accuracies plotted over gradient descent iteration for an MLP trained with normalized and un-normalized images, respectively.

3.6. Task 3.5

We created a CNN model using the PyTorch library. The CNN has 2 convolutional and 2 fully connected layers each having 256 units and all activations are ReLU. We found the CNN model had higher accuracy than our best MLP model. The best performing run of the CNN model had train/test accuracy of 61/65%, while the best MLP model had a train/test accuracy of 47/44%. We investigated the CNN at various learning rates in order to optimize performance (Fig.3)

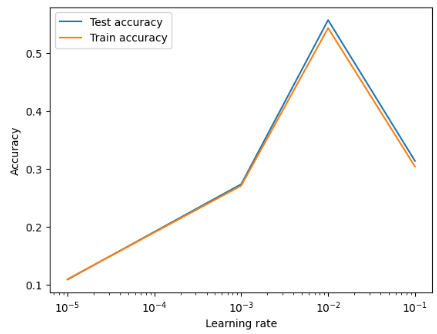


Figure 3: CNN Accuracy versus learning rate

3.7. Task 3.5 cont.

We were also interested in studying the effects of the data augmentation on the CNN model. Data augmentation was done using the torchvision - Transforms library. Specifically, we implemented a predefined augmentation method that follows the policy as described by Cubuk et. al [3]. On a previously trained CNN model, training the model on a fully augmentation data-set reduced the difference between test and train error from >5 to <2. However, the accuracy of both models was decrease, resulting in a train/test accuracy of roughly 53/55%.

Maximum Accuracy	**Test**	**Train**
Standard	**63%**	**68%**
Augmented	**53%**	**55%**

Figure 4: CNN Accuracy for augmented and non-augmented data

3.8. Task 3.6

We loaded the pre-trained ResNet model containing 18 layers using the torchvision library. We froze all the convolutional layers and removed the fully connected ones. We then added two unfrozen, fully connected layers right after the final convolutional layer. Though batch normalization and dropout layers were frozen, they were not deleted. The ResNet model performed better than both the MLP and CNN model. This is corroborated by previous literature [4].

3.9. Task 3.7

All our findings as displayed as a graph or table under each task. Accuracy vs Epoch/Gradient Descent Iterations graphs can be found in our code files and visualize when the models begin to over-fit to the training data.

One result of interest in investigating was the identity of incorrect predictions. To do so, we visualized the confusion matrix for CNN predictions on the test data (Fig. 3). It is of note that while this particular run demonstrated a 60% accuracy, false positives appear to be occurring across classes sharing semantically similar features.

For example, we found the deer class shows the lowest rate of accurate predictions, and often leads to false predictions of a two particular classes - horse or bird. In contrast, there is almost no predictions of classes such as cars or truck. This confusion between classes sharing similar intuitive features can be seen throughout several other pairs in the confusion matrix: cat dog, truck car, plane boat, etc. These results indicate that increasing the depth of the model, allowing for the learning of more features to delineate similar classes, would enhance the model's performance.

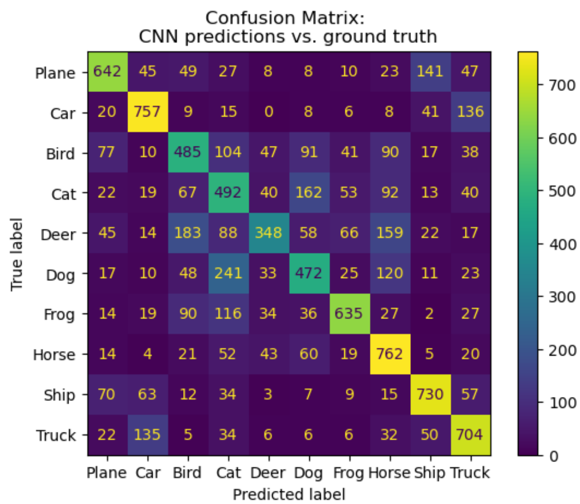


Figure 5: CNN Confusion matrix. Entry values and color bar represent number of hits per cross-catagorization.

4. Discussion and Conclusion

In conclusion, we implemented and evaluated various models for multiclass image classification on the CIFAR-10 dataset, including an MLP built from scratch, a CNN created using existing libraries, and a pre-trained model. Our results indicate that non-linearity and increasing network depth lead to higher accuracy in multiclass image classification with MLP models. We also found that the choice of activation function impacts accuracy, with leaky ReLU outperforming tanh and ReLU. Regularization techniques, such as L1 and L2 regularization, were effective in preventing overfitting and improving generalization. Moreover, data normalization was a critical preprocessing step for stable and optimized models. Our findings are consistent with other studies that have utilized MLPs to analyze the CIFAR-10 dataset.

In terms of future directions, there are several areas where we can continue to explore to further improve our models' performance. One potential avenue is to learn the lambda parameter for regularization. Currently, we have observed that regularization can lead to lower accuracy, and we need to find a balance between overfitting and generalization. Another area for exploration is hidden layers - we can try adding more layers and varying the type of activation function for each layer. Currently, we have restricted ourselves to 0-2 hidden layers and three activation functions, with the added constraint that every hidden layer has the same activation function. Additionally, we can make the bias term for each hidden layer a learnable parameter, as currently it is fixed to one. These future directions have the potential to lead to even better results on the CIFAR-10 dataset.

5. Statement of Contributions

Lucas Bennett implemented the CNN model and ResNet models, ran CNN experiments and data augmentation analysis. Bjørn Christensen ... . Megan Ng implemented the MLP models and ran MLP experiments.

References

[1] S. Alizadeh, M. Rezaei, A comprehensive study of multilayer perceptron neural networks on cifar-10, International Journal of Machine Learning and Cybernetics 7 (4) (2016) 575–586.

[2] K. Xu, Z. Jiang, H. Liu, L. Li, A deep mlp with hyperparameter optimization for image classification on cifar-10, in: 2019 IEEE 12th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM), IEEE, 2019, pp. 39–44.

[3] E. D. Cubuk, B. Zoph, J. Shlens, Q. V. Le, Randaugment: Practical automated data augmentation with a reduced search space, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 3008–3017. doi:10.1109/CVPRW50498.2020.00359.

[4] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778. doi:10.1109/CVPR.2016.90.