Introduction to Machine Learning: Mini Project 2

March 9, 2023

Usman Hameed Ullah: 260925886 | Liv Toft: 260856481 | Simaan Arshad: 260921696

Abstract

We investigated the optimal architectures for multilayer perceptron (MLP) and convolutional neural network (CNN) machine learning models for image classification using the CIFAR-10 dataset. We implemented MLP models with 0, 1, or 2 hidden layers and find that models with 1 and 2 hidden layers have comparable performance and achieve the highest test accuracies at 53.70% and 53.52%, respectively, with the 2-layer model exhibiting slightly less overfitting. In addition, 2-layer models using ReLU, leaky ReLU, or tanh hidden layer activation are implemented. We found that the model with ReLU hidden layer activation gives the highest test performance at 53.52% accuracy. Furthermore, we implemented both L1 and L2 regularisation and demonstrated that L2 regularisation improves model performance (53.80% test accuracy) and decreases overfitting. We also demonstrated that data normalization leads to a higher performing model, with test accuracy on unnormalized data reaching only 39.57%.

We implemented a CNN model with 2 convolutional layers followed by 2 fully connected layers with ReLU activation. The convolutional layers were separated by max pooling layers of size 0.5 to reduce the spatial dimensions by half. Dropout layers were also added between fully connected layers to prevent overfitting. This CNN model gave a test accuracy of 75%, performing better than the highest performing MLP model. Finally, we loaded a ResNet50 model pre-trained on the ImageNet dataset for object recognition and froze all convolutional layers. We replaced the pre-existing fully connected layers with 3 UpSampling2D layers, 2 fully connected layers with ReLU activation, and dropout layers. This model resulted in a test accuracy of 82%, performing the best out of all MLP and CNN models. Finally, we reduce the width of the 2-layer MLP model to 64 and 128 hidden units per hidden layer and demonstrated how this effectively reduces the discrepancy between train and test performance from 10.51% to 7.16%. Our investigation demonstrates how model architecture can be fine-tuned to increase performance and directly compares these two important machine learning models.

1 Introduction

Here we analyze and compare the performance of two machine learning models – multilayer perceptron (MLP) and convolutional neural network (CNN) – as image classifiers. MLPs are a type of artificial neural network which consists of several layers of nodes. An activation function is used to non-linearly transform the inputs to each node layer using the nodes of the preceding layer and weights learned using mini-batch stochastic gradient descent [1]. On the other hand, CNNs are neural networks which are designed for image processing [2]. In addition to the fully connected layers of MLPs, CNNs also contain convolutional layers where each neuron is only connected to a small region of the input image. CNNs are designed to automatically learn features from the input images by applying filters to each small region of the image and then aggregating these features across the entire image [2]. Both models are quite important in machine learning; MLPs are particularly useful for general-purpose machine learning tasks, such as classification and regression, while CNNs are specifically designed for image processing tasks [2].

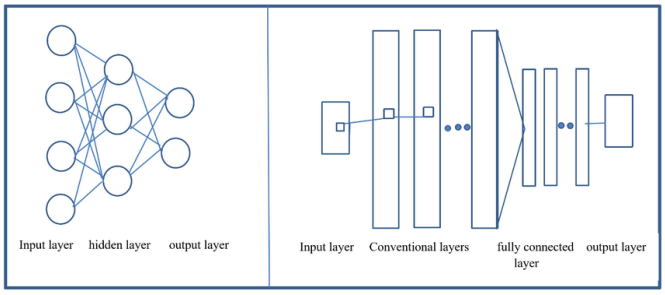


Figure 1 – Multilayer perceptron (MLP) architecture vs convolutional neural network (CNN) architecture [3]

We explore these two models using the CIFAR-10 dataset: a benchmark dataset containing 6,000 images of each object class. We first analyse the effect of number of hidden layers on the performance of the MLP model. We implement models with 0, 1, and 2 hidden layers and find that the highest performing model is that with 1 and 2 hidden layers, giving test accuracies of 53.70% and 53.52%, respectively. This demonstrates how increasing model depth can increase performance. Following this, we investigate the effect of different hidden layer activation functions on model performance and show that using ReLU activation gives the highest performing model with a test accuracy of 53.52%. Furthermore, we implement both L1 and L2 regularization. The regularizations work by adding a penalty term to the loss function, which enables the model to learn patterns that generalize the data better in comparison to more complex patterns that fit the training data more closely. While L2 slightly decreased overfitting and improved model performance, L1 seemed to increase

overfitting and yielded poorer performance. We also show the importance of data normalisation by training on the un-normalised training set. This achieves a maximum test accuracy of 39.57%.

Following this, we compare the MLP models to CNN models. First, we develop a CNN model with 2 convolutional and 2 fully connected layers with ReLU activation. We find that max pooling and dropout layers help in reducing computational complexity and prevent overfitting of data. For the convolutional layers, 32 3x3 filters and 64 5x5 filters were chosen. 3x3 filters work best for detecting small features whereas the 5x5 filters are used for detecting larger features of the images. [9] We set the padding to 'same' to indicate that the output feature map has the same dimensions as the input. 256 units were chosen for both fully connected layers, followed by an output layer with 10 units and a SoftMax function. The model yields a train and test accuracy of 82% and 75%, respectively. For our second CNN model, we use the ResNet50 model from the TensorFlow package which is pre-trained on the ImageNet dataset. The convolutional layers of this pre-trained model are frozen, and all its fully connected layers removed. We add 3 UpSampling2D layers to increase the spatial resolution of the input images by a factor of 2, 2 fully connected, ReLU activated layers with 128 and 64 neurons, followed by an output layer of 10 neurons with SoftMax activation. This model yields the best accuracy amongst all models with a train and test accuracy of 84% and 82%, respectively.

Throughout these experiments we noticed significant overfitting of 2-layer MLP models with ReLU hidden layer activation. While higher model depth can increase predictive power, wide models can lead to overfitting. We subsequently decrease the number of hidden layers to 64 and 128 and find that the 128-hidden layer MLP model leads to decreased overfitting, while maintaining relatively the same test accuracy at 52.05%. In this way, the various model parameters of both MLPs and CNNs were explored, giving insight into how they affect model performance as well as how they can be fine-tuned the for-image classification.

2 Datasets

The CIFAR-10 dataset is comprised of 60,000 images corresponding to 10 different classes. Each image is 32 pixels by 32 pixels, with each pixel corresponding to a 3x1 vector containing its RGB values. Each class has 5,000 associated images in the training set and 1,000 associated images in the test set, giving balanced class instances in both the training and test sets.

This dataset was normalized to improve performance, stability, and generalization of the MLP and CNN models. To make the dataset suitable for the MLP models, the images were first vectorized and z-score normalization was applied by subtracting the mean of the data from each data point and then dividing by the standard deviation of the data. For the purposes of the CNN model, the data did not need to be vectorized first and was directly normalized from its raw format following the same steps. Since we are working with categorical data, the classifications were one-hot encoded to represent the categorical variable as binary vectors such that the MLP and CNN models can learn the correct relationships and avoid making false numerical relations.

3 Results

3.1 Analysing the effect of network depth

In order to analyse the effect of network depth on MLP model performance, MLP models with 0, 1, or 2 hidden layers were implemented using pre-determined optimal batch sizes and learning rates (see Appendix). All hidden layers had 256 hidden units, ReLU activation, and bias. The models were trained for 50 epochs and the associated train and test accuracies as a function of epoch number are shown in Figure 2.

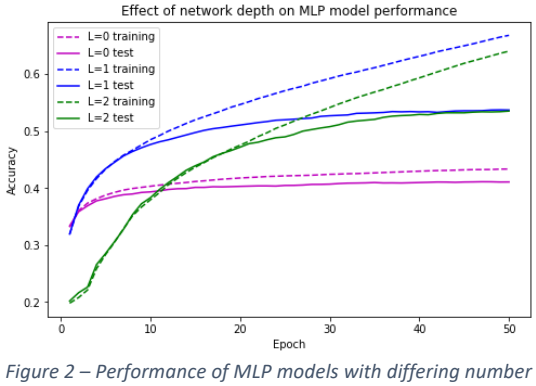


Figure 2 – Performance of MLP models with differing number of hidden layers (L) as a function of epoch. (M=256, ReLU activation, bias)

The 1-layer and 2-layer models exhibited similar performance with test accuracies of 66.81% and 64.03% on the training set and 53.70% and 53.52% on the training set, respectively. Both models performed considerably better compared to the 0-layer model. This can be attributed to how increased network depth can better model more complex data as more hidden units and activation functions lead to more complicated manipulations of the input data [4]. The lack of performance increase between the 1-layer and 2-layer models may occur if using 2 layers with 256 hidden units creates too complicated of a model for the dataset [4]. In this case, the performance of

the model no longer increases with increasing network depth, however overfitting is reduced suggesting that the 2-layer model more accurately models the trends in the data.

3.2 Analysing the effect of hidden layer activation function.

Three MLP models with 2 hidden layers, 256 hidden units per layer and ReLU, Leaky ReLU, or tanh hidden layer activation functions were implemented using pre-determined optimal mini-batch sizes and learning rates (see Appendix). The models' performances are summarised in Figure 3.

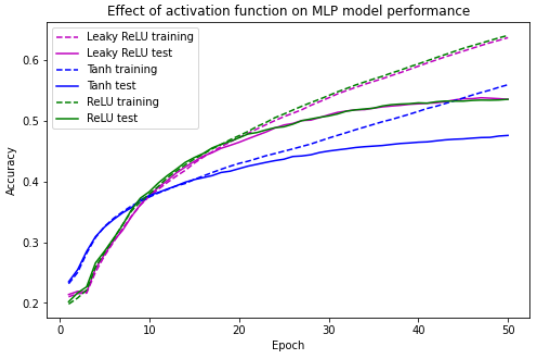


Figure 3 – Performance MLP models using 3 different hidden layer activation functions (L=2, M=256, bias)

The highest performing model was that with ReLU hidden layer activation with a test accuracy of 53.52%. Tanh is used to deal with binary classification problems while leaky ReLU is used when the input distribution is negative [5, 6]. Therefore, tanh and leaky ReLU activation functions are not suited for our dataset. ReLU hidden layer activation is better suited for this dataset because it is less likely to counter the problem of vanishing gradients, where the gradients become too small to update the weights during the backpropagation process, and due to its ability to effectively learn non-linear features from the data. Models with ReLU hidden layer activation also have a sparsity property, which means that some of the neurons in the network can become inactive, reducing the computational cost of the model [6].

3.3 Regularization

We implemented the ability to apply L1 and L2 regularization to the models by independently applying either an L1 or L2 penalty to the model weight gradients calculated during gradient descent. Using the pre-determined optimal mini-batch size and learning rates (see Appendix), we trained the 2-layer model detailed in Section 3.1 for 50 epochs using L1 or L2 regularization (Figure 4).

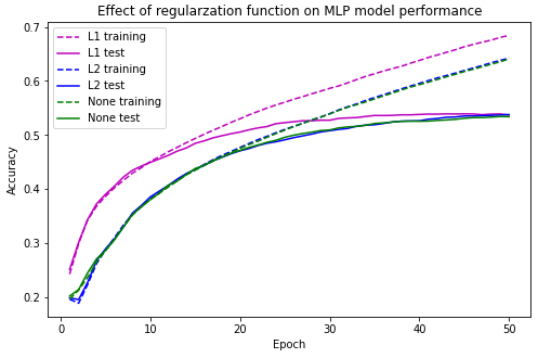


Figure 4 – Performance of MLP model with L1, L2, or no regularization (L=2, M=256, ReLU activation, bias)

Looking at the results we can see that adding L2 regularization improves the performance slightly and reduces overfitting as compared to L1. The L1 regularization can reduce the accuracy of the dataset, because L1 regularization can cause some of the weights to become zero, which effectively removes the corresponding input features from the model and causes the accuracy to decrease. On the other hand, L2 regularization does not cause the weights to become zero but instead encourages the weights to be smaller overall. This leads to a smoother and more stable optimization problem, and it has been shown to be effective in improving the generalization performance of a model without reducing its accuracy.

3.4 Un-normalized images

After training the model using un-normalised input data, we can see that the accuracy is much lower than the model that was trained with normalised images (40.00% train accuracy, 39.57% test accuracy). This is because normalising the data helps with standardizing the input and model can more easily identify patterns and relationships between the features (pixels) of the image.

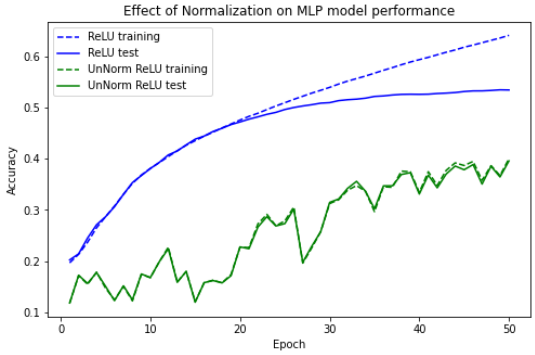


Figure 5 – Performance of MLP model using normalized images versus un-normalized images. (L=2, M=256, ReLU activation, bias)

As we can see from the above figure, the accuracy is far higher for the model trained on normalized images. Normalisation helps to standardize the range of pixel values in the image, which can improve convergence during training. When pixel values are too different, the gradients computed during backpropagation can be too large or too small, making it difficult for the model to learn.

3.5 Comparison to convolutional neural networks

The CNN model was implemented with 2 convolutional and 2 fully connected layers with 256 hidden units. ReLU activation was applied to all hidden layers. This model was fit using stochastic gradient optimization with a learning rate of 0.001 and a momentum of 0.9. A momentum of 0.9 is commonly used in practice and is shown to work well for deep CNN image classification tasks. [7] Furthermore, a learning rate of 0.001 gave better computational performance in comparison to a lower rate of 0.0001 and converged within the optimal solution range. The loss function was set to categorical cross entropy with accuracy as the monitoring metric.

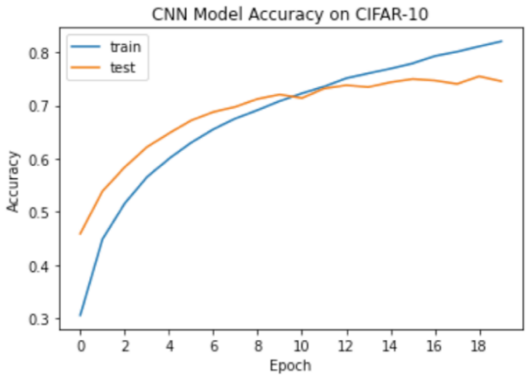


Figure 6 – Performance of CNN model with 2 convolutional and 2 fully connected layers

As seen in Figure 6, the train and test accuracies of this model are 82.05% and 74.54%, respectively. These accuracies are higher than that of the best performing MLP models described previously. This is because MLP models by nature are unable to capture the spatial relationships between pixels in an image. It treats each input feature independently and has no knowledge of its relation to other features making them unable to distinguish between patterns that are spatially close to each other but in different positions in an image. Conversely, CNN models are commonly used for image classification as their convolutional layers are specifically designed to capture spatial relationships between pixels. It uses filters to extract features from small regions and combines these features to learn about the complex patterns in the image. [8]

3.6 Comparison to ResNet50 pre-trained model

We loaded the ResNet50 model from the TensorFlow package. This model has been pre-trained on ImageNet dataset, however, in order to fit this model to the CIFAR-10 dataset, we first froze all its convolutional layers such that they would not be trained and would maintain their existing weights. Then all fully connected layers were removed, and new fully connected layers were added to the output of the pre-trained model. Three UpSampling2D layers were added to increase the spatial resolution of the input images. Two fully connected layers were added with 128 and 64 units, respectively, and ReLU activation was applied to both layers. Dropout layers were added to prevent overfitting followed by a final SoftMax layer with 10 units. Hyperparameter testing performed to find optimal hyperparameters. From this testing, we found that the optimal learning rate is 0.1, optimal batch size is 128, and the optimal dropout rate is 0.2 We first started with learning rates of 0.0001, 0.001, 0.01, and 0.1 alongside batch sizes 32, 64, and 128 to find 0.1 and 128 to be the optimal values, respectively. Using this, we then tested on dropout rates to determine 0.2 to be the optimal rate. [See Appendix]

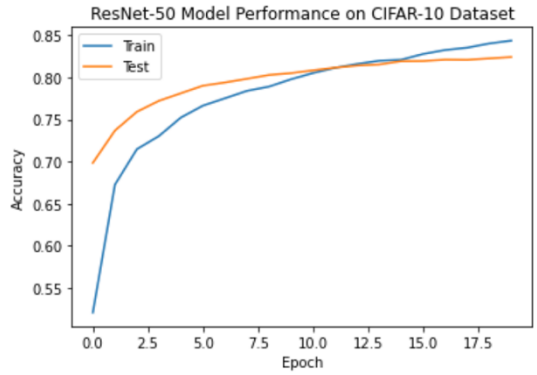


Figure 7 – Performance of the ResNet50 Model on the CIFAR-10 Dataset

The train and test accuracies were 84% and 82%, respectively, giving the best performance amongst all models. As discussed earlier, CNN models perform better on image classification tasks than MLP models due their ability to capture spatial relations. Additionally, since we are using a pre-trained model, it has already learned from ImageNet which is a large dataset consisting of 1000 categories and 1 million images. By fine-tuning this model with newly added layers on the CIFAR-10 dataset, we are leveraging the features learned from the pre-model to achieve better performance. We also applied UpSampling2D layers on the input data as we are working with images of size 32x32 whereas ResNet-50 is pre-trained on images of size 224x224. Thus, by increasing the spatial resolution of the feature maps, the model can capture more fine-grained details in the input image and produce more accurate predictions. Therefore, this model outperforms the model in 3.5. Additionally, the training time of this model is very comparative to that of the MLP and other CNN models in parts 3.1 – 3.5. To train over 20 epochs it took 1 hour 15 minutes, while the next highest training time is that of the CNN model described in 3.5 at around 30 minutes. The third slowest training time, but far quicker than either of these, is the MLP model with 2 hidden layers and 256 hidden units at 16 minutes for 50 epochs. These increased training times compared to MLPs is because CNN models contain many layers that involve a large number of matrix multiplications and convolutions, which can take a long time to compute. CNN models are specifically designed for processing image data thus the computational complexity is the cost to pay to achieve optimal performances.

3.7 Analysing the effect of model width

The 2-layer, 256-hidden unit MLP models with ReLU hidden layer activation exhibit high degrees of overfitting, with discrepancies between training and test accuracies reaching 10.75%. While we have demonstrated that adding L2 regularization aids in remedying this, we still observe a difference of 10% between train and test accuracies. We speculated that his may be because 256 hidden units per hidden layer leads to an over-complex model that does not generalize. Therefore, we reduced the number of hidden layer units to 64 or 128 in an attempt to reduce overfitting. Figure 8 depicts the respective models' accuracies compared to the previously implemented 2-layer MLP model with 256 hidden units per hidden layer.

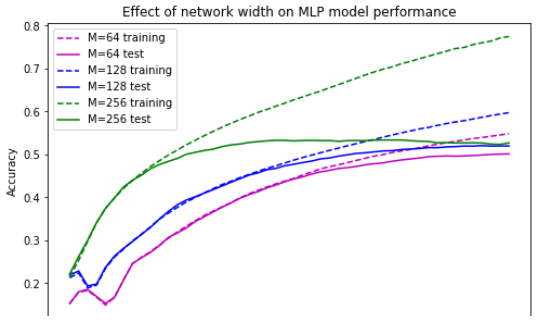


Figure 8 – Effect of number of hidden units (M) on model performance. (L=2, ReLU activation, and bias).

While we observe a decrease in train accuracy for both models, the test accuracy decreases by only 1.47% when reducing the number of hidden layers to 128. This gives a 7.16% percent decrease in performance between train and test sets compared to the 10.51% percent decrease observed when using a model with 256 hidden units per hidden layer. Therefore, we demonstrate how reducing the width of the model can reduce overfitting and produce a model that generalizes better. Reducing the width too much, however, creates a model that is too simple and cannot model the data as effectively, as demonstrated by the decrease in both train and test accuracy of the 64-hidden unit model compared to the 128- and 256-hidden unit counterparts.

5 Discussions and Conclusion

We investigated how model architecture affects the behaviour and performance of two machine learning models – multilayer perceptron and convolutional neural network – as image classifiers using the CIFAR-10 dataset. Upon implementing MLP models with 0, 1, and 2 hidden layers we found that increasing model depth increased performance before a maximum threshold was met giving similar test accuracy for both the 1-layer and 2-layer models, but reduced overfitting by the 2-layer model. In addition, we investigated the effect of using ReLU, tanh, and leaky ReLU activation functions on 2-layer MLP model performance and found that ReLU activation gives the highest performance. This is in alignment with the fact that this activation function is best suited for the image classification task being performed. Furthermore, we implemented both L1 and L2 regularisation and found that L2 regularisation decreased overfitting and increased model performance. Due to unfavorable characteristics of L1 regularisation, we found no improvement in either accuracy or overfitting upon its implementation. Finally, we investigated the effect of training on un-normalized data and found that MLP model performance significantly decreased.

We found that the optimal architecture of the CNN model was that with 2 convolutional layers, separated by max pooling layers of size 0.5, followed by 2 fully connected layers with ReLU activation and dropout layers. This architecture gave a far higher test accuracy than any of the implemented MLP models, demonstrating how convolutional layers increase the model's performance as an image classifier. Following this, we used TensorFlow's ResNet50 model which is pretrained on ImageNet. We replaced the two fully connected layers with layers with one 128- and one 64 neuron-layer, each with ReLU activation. This model yielded the best accuracy amongst all models.

Finally, we investigated the effect of decreasing model width on overfitting and found that a deep model (L=2) with decreased number of hidden units (M=128) per hidden layer reduced overfitting. Too shallow of a model, however, decreased overall performance. We therefore demonstrated how a balance between too complicated and too simple of a model must be found by choosing the optimal number of hidden units. Through these experiments we investigated various aspects of neural network architecture and their subsequent effects on both the performance and accuracy of MLPs and CNNs.

6 Statement of Contributions

Liv implemented the MLP class, gradient descent, as well as the code and write up associated with sections 3.1, 3.2, and 3.7. Simaan implemented the code and associated write up for sections 3.3 and 3.4. Usman wrote the code and write up sections associated with parts 3.5 and 3.6, as well as performed the data normalization.

7 Appendix

7.1 MLP model accuracies

Hidden layers	Hidden units	Hidden layer activation	Regularization	Unnormalized data?	Train accuracy	Test accuracy
0	0	ReLU	None	No	0.4335	0.4108
1	256	ReLU	None	No	0.6681	0.5370
2	256	ReLU	None	No	0.6403	0.5352
2	256	Tanh	None	No	0.5592	0.4757
2	256	Leaky ReLU	None	No	0.6367	0.5351
2	256	ReLU	L1	No	0.6860	0.5345
2	256	ReLU	L2	No	0.6395	0.5380
2	256	ReLU	None	Yes	0.4000	0.3957
2	64	ReLU	None	No	0.5507	0.5028
2	128	ReLU	None	No	0.5921	0.5205

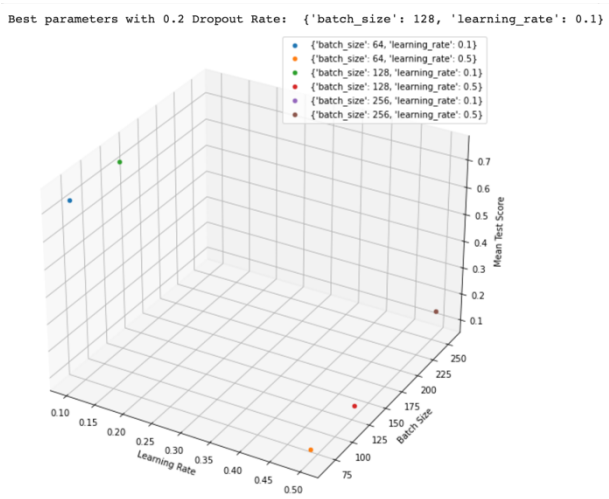
7.2 CNN model accuracies and ResNet-50 Hyperparameter Selection

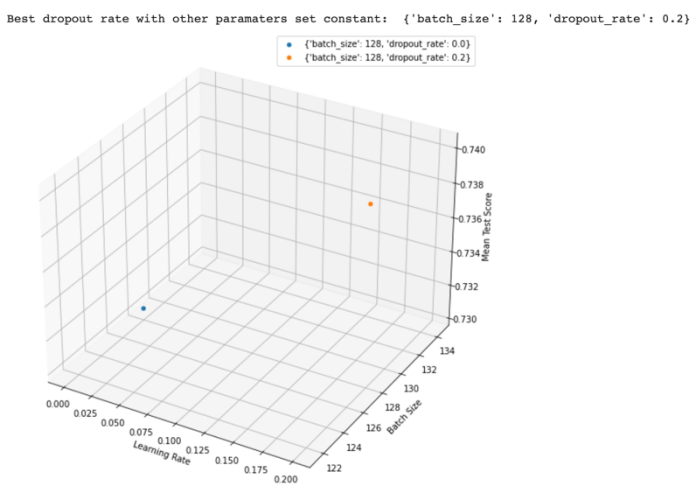
CNN model with 2 fully connected and 2 convolutional layers with ReLU activations applied.

Convolutional Layers				Fully Connected Layers			Train Accuracy	Test Accuracy
Filters	Activation Function	Pooling Operation	Padding	Number of Units	Activation	Dropout		
32, (3, 3)	ReLU	MaxPooling2D(2, 2)	Same	256	ReLU	0.5	82.05%	74.54%
64, (5, 5)	ReLU	MaxPooling2D(2, 2)	Same	256	ReLU	0.5		
-	-			10	SoftMax	-		

ResNet50 Pre-trained model.

Fully Connected Layers			Train Accuracy	Test Accuracy
Number of Units	Activation	Dropout		
128	ReLU	0.2	84.37%	82.42%
64	ReLU	0.2		
10	SoftMax	-		





7.3 Hyperparameter selection

All hyperparameters were chosen using the same basic procedure. First, the models were trained using a mini-batch size of 128 and a learning rate of 0.0001, 0.001, 0.01, or 0.1 for 50 epochs. The learning rate giving the highest test accuracy was then used to train models with a mini-batch size of 64, 128, or 256 for 50 epochs. Figure 9 depicts the effect of learning rate and mini-batch size on the models with differing number of hidden layers.

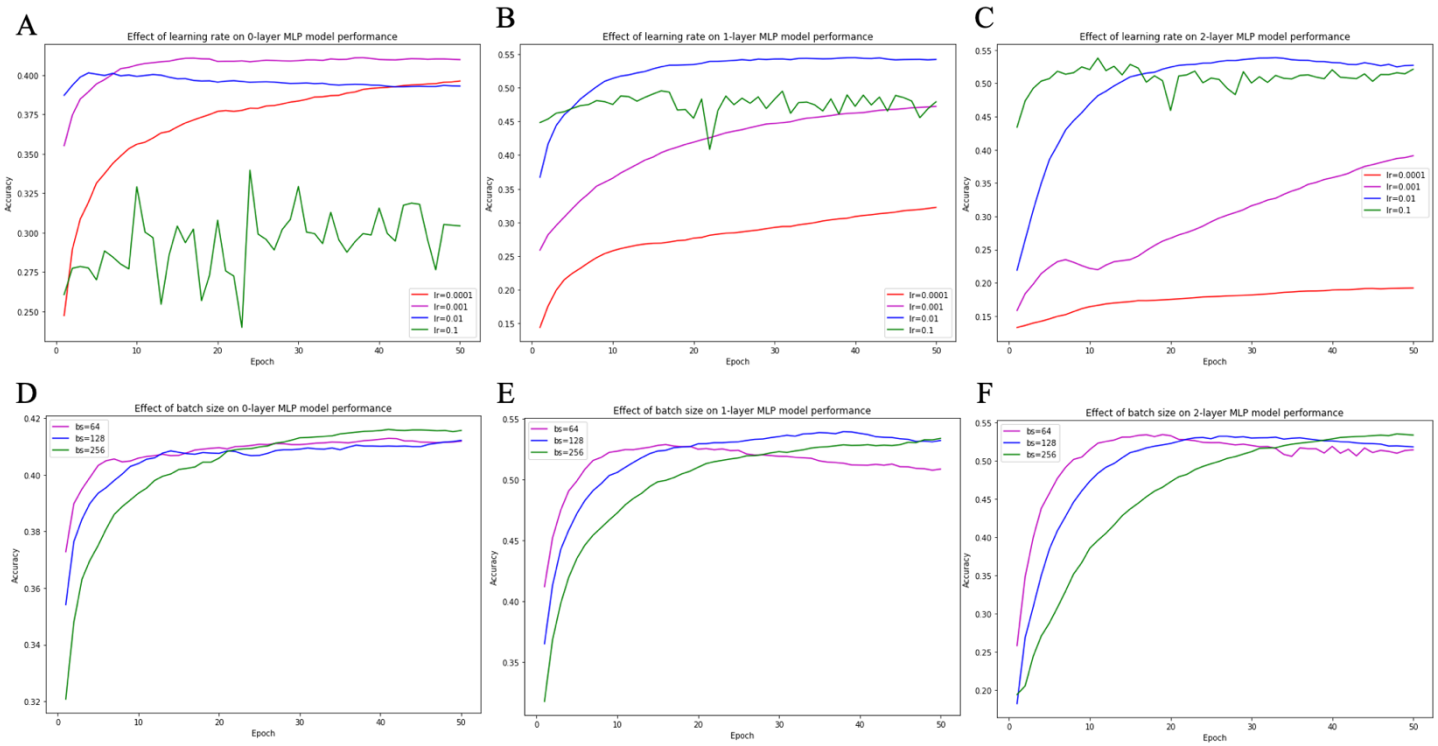
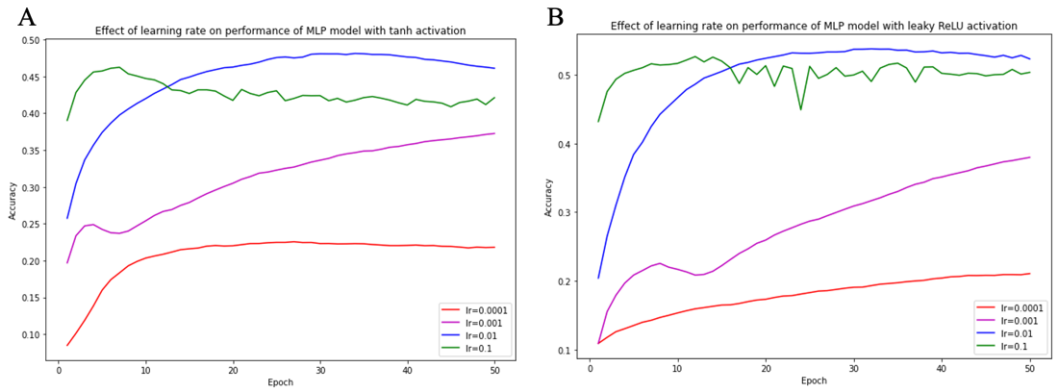


Figure 9 – Accuracy as a function of epoch number. ***(A-C)*** *Accuracy of 0, 1, or 2-layer MLP models with differing learning rates, bs=128 (M=256, ReLU activation, bias).* ***(D-F)*** *Accuracy of 0, 1, or 2-layer MLP models with differing mini-batch sizes, lr=0.01 (M=256, ReLU activation, bias).*

Figure 10 depicts the effect of learning rate and mini-batch size on the models with differing hidden layer activation functions.



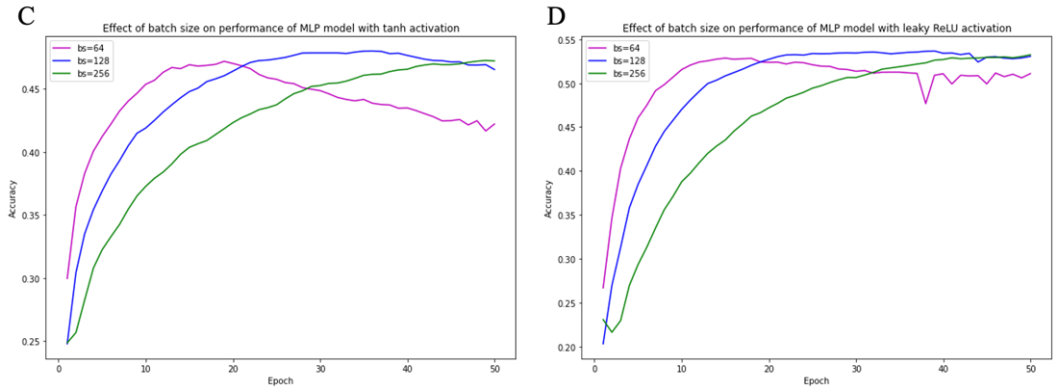


Figure 10 – Accuracy as a function of epoch number. ***(A,B)*** *Accuracy MLP models with tanh or leaky ReLU hidden layer activation and differing learning rates, bs=128 (L=2, M=256, bias).* ***(C,D)*** *Accuracy of MLP models with tanh or leaky ReLU hidden layer activation and differing mini-batch sizes, lr=0.01 (L=2, M=256, bias).*

Figure 11 depicts the effect of learning rate and mini-batch size on the models with differing number of hidden units per hidden layer.

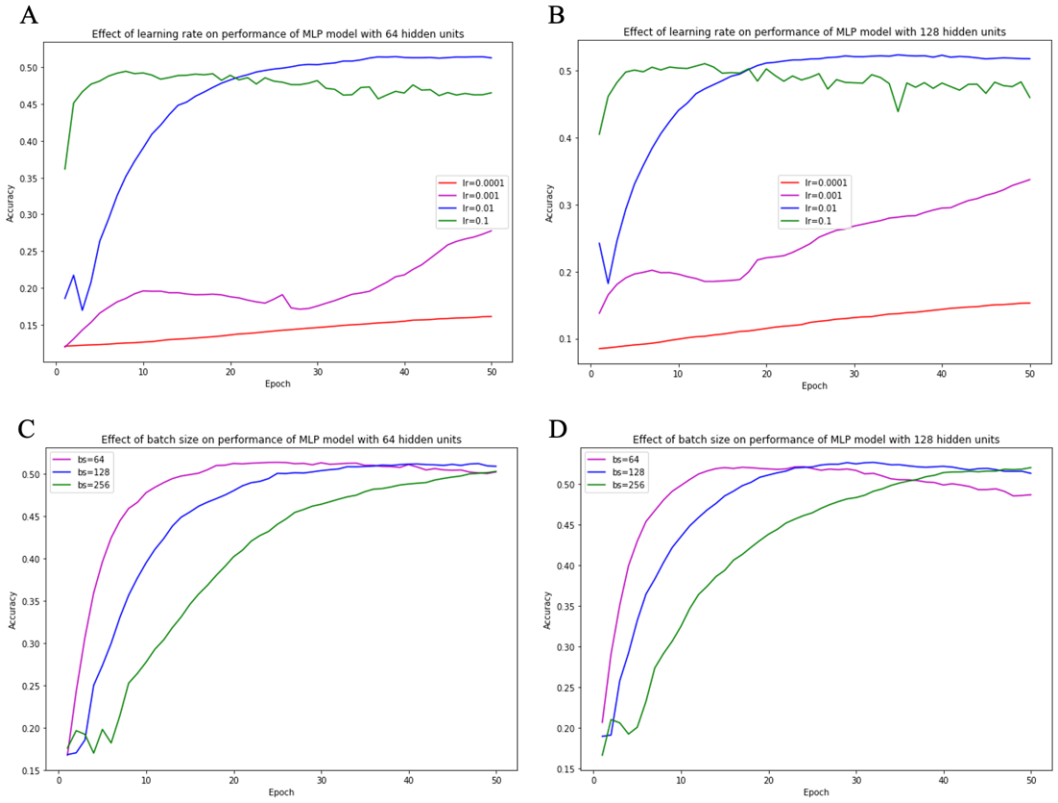


Figure 11 – Accuracy as a function of epoch number. ***(A,B)*** *Accuracy 64- or 128-hidden unit MLP models with differing learning rates, bs=128 (L=2, ReLU activation, bias).* ***(C,D)*** *Accuracy of 64- or 128-hidden unit MLP models with differing mini-batch sizes, lr=0.01 (L=2, ReLU activation, bias).*

From Figures 9-11 the following optimal hyperparameters were determined:

Hidden layers	Hidden units	Hidden layer activation function	Optimal learning rate	Optimal mini-batch size
0	0	ReLU	0.001	256
1	256	ReLU	0.01	256
2	256	ReLU	0.01	256
2	256	Tanh	0.01	256
2	256	Leaky ReLU	0.01	256
2	64	ReLU	0.01	256
2	128	ReLU	0.01	256

8 References

[1] F. Murtagh, "Multilayer perceptrons for classification and regression," *Neurocomputing,* vol. 2, no. 5-6, pp. 183-197, 1991, doi: 10.1016/0925-2312(91)90023-5.

[2] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A survey of convolutional neural networks: Analysis, applications, and prospects," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 33, no. 12, pp. 6999-7019, 2022, doi: 10.1109/TNNLS.2021.3084827.

[3] A. Haya, "Convolutional neural network application in biomedical signals," *Journal of Computer Science and Information Technology,* 2018, doi: 10.15640/jcsit.v6n2a5.

[4] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Feedforward Networks," in *Deep Learning*: MIT Press, 2016.

[5] S. Sharma. "Activation Functions in Neural Networks." Towards Data Science. https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6 (accessed March 8, 2023).

[6] J. Brownlee. "A Gentle Introduction to the Rectified Linear Unit." Machine Learning Mastery. https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/ (Accessed March 8, 2023).

[7] C. Wolfe. "Why 0.9? Towards Better Momentum Strategies in Deep Learning." Towards Data Science. https://towardsdatascience.com/why-0-9-towards-better-momentum-strategies-in-deep-learning-827408503650 (Accessed March 8, 2023).

[8] R. Logan *et al.*, "Deep convolutional neural networks with ensemble learning and generative adversarial networks for Alzheimer's disease image data classification," *Frontiers in Aging Neuroscience,* vol. 13, 2021, doi: 10.3389/fnagi.2021.720226.

[9] O. Khanday and M. Egyetem, "Convolution neural networks and impact of filter sizes on image classification," *Multidiszciplináris Tudományok,* vol. 10, pp. 55-60, 2020, doi: 10.35925/j.multi.2020.1.7.