COMP 551 A3: *Classification of Image Data*

Rebecca Mizrahi (260975001) rebecca.mizrahi@mail.mcgill.ca
Software Engineering

Amélie Barsoum (290988658) amelie.barsoum@mail.mcgill.ca
Software Engineering

Samantha Handal (260983914) samantha.handal@mail.mcgill.ca
Software Engineering

1. Abstract

This report describes our implementation and evaluation of multilayer perceptrons (MLPs) and convolutional neural networks (ConvNets) for image classification on the Sign Language MNIST dataset. We constructed MLPs with varying depths and activations and compared their performance with a ConvNet architecture. Our findings reveal the large impact of non-linearity and network depth on MLP performance, the superiority of ReLU and its variants over sigmoid activations, and the balance of model complexity and generalization achieved through L2 regularization. Notably, ConvNets demonstrated a marked advantage over MLPs for dealing with image-related tasks.

2. Introduction

This project gives a comparative analysis of MLPs and ConvNets to classify image data from the Sign Language MNIST dataset, an image dataset that presents challenges due to its representation of hand gestures for sign language alphabets. Through experimentation, we explore the influence of network architecture, activation functions, and regularization on model performance, we gain insights into each architecture's performance, offering a deeper understanding of their potentials and limitations in image classification tasks. Increasing depth and adding hidden layers improved performance up to a certain point, until it caused overfitting. This could be mitigated by L2 regularization, as we observed in experiment 3. Finally, we conpared MLPs to ConvNet architecture, which showed us that although ConvNet architectures performed better than MLPs on our datasets due to their compatibility with image datasets, modifying the MLP architecture also gave us a competitive performance. Our observations mostly align with the findings in [1].

3. Datasets

The Sign Language MNIST dataset provides grayscale images of hand signs representing the 26 letters of the American Sign Language alphabet, containing a high degree of complexity. To pre-process this data for neural network processing, we normalized the pixel intensities and reshaped the images for MLP and ConvNet inputs. Through normalization,

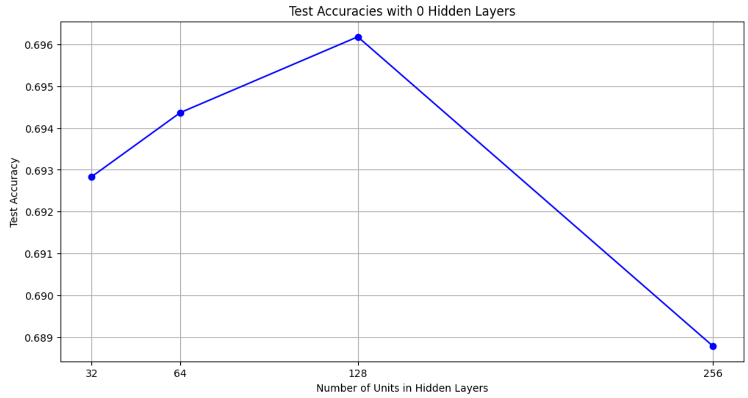

Figure 1: Test Accuracies with 0 Hidden Layers

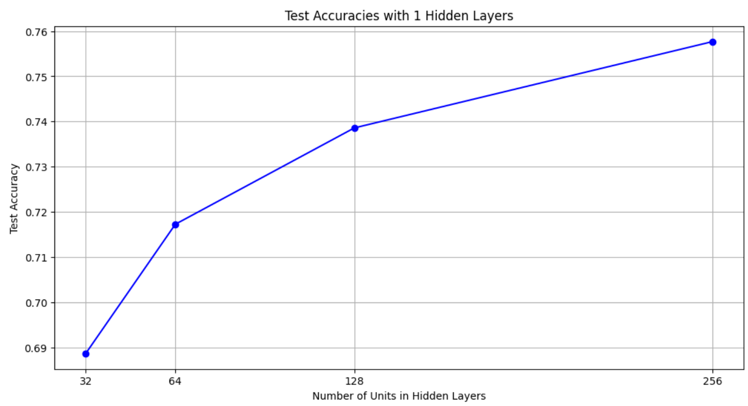

Figure 2: Test Accuracies with 1 Hidden Layer

pixel values were scaled to a [0,1] range, reducing variance. The reshaped data was then fed into MLPs directly while maintaining spatial dimensions for ConvNets. These preprocessing steps were taken from the provided tutorials and documentation.

4. Results

4.1 Experiment with 3 Different Models

By looking at the test accuracy graphs when using a neural network with 0 hidden layers, 1 hidden layer, and 2 hidden layers, we can observe some patterns that tell us how the architecture of a neural network impacts its performance on the Sign Language MNIST dataset.

The model with 0 hidden layers, depicted in Figure 1, is relatively stable regardless of the number of units used in the hidden layer, which makes sense. The best test accuracy achieved is with 128 units (around 0.6962). Since there are no hidden layers, this model is basically just a linear classifier, relying only on the input layer and output layer. However, we can see that even a simple linear classifier can achieve decent performance. The model

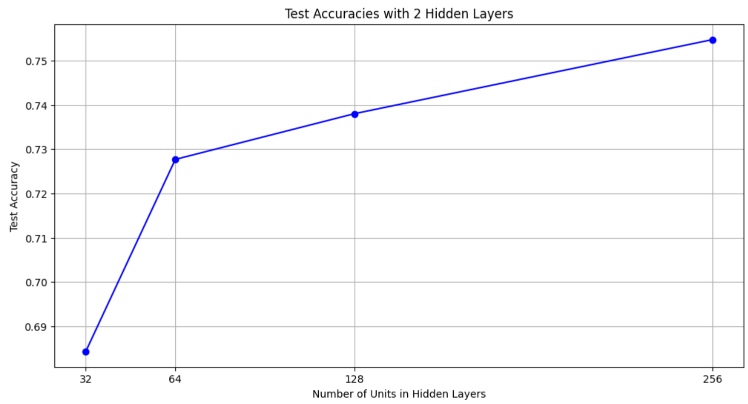

Figure 3: Test Accuracies with 2 Hidden Layers

with 1 hidden layer (with ReLU activation), in Figure 2, shows a performance improvement, reaching a best test accuracy of 0.7577 with 256 units. This increased accuracy is due to the non-linearity that comes with the hidden layer and ReLU activations. There is a trend where increasing the number of hidden units leads to better performance, indicating that the model becomes more capable of learning complex patterns with more units. The model with 2 hidden layers (Figure 3)also leads to improvements from 0 hidden layers, with the best test accuracy observed to be 0.7547 with 256 units. However, the improvements from additional depth (from 1 to 2 layers) are not as significant as those from introducing the first hidden layer. This might be because of the complex function that the network needs to model. Here, we see that additional layers may lead to issues like overfitting.

Introducing the first hidden layer significantly improves the accuracy of the models. Neural networks with non-linear activation functions can usually model complex functions that linear models cannot. The depth of the network also affects the accuracy, with deeper networks generally performing better. This is because the additional layers allow the network to learn more complex representations of the data. However, as we have often observed previsouly, there are diminishing returns with additional depth and units. Beyond a certain point, adding more layers or units does not result in significant performance gains and might even lead to overfitting. The plots show a trend where increasing the number of units in hidden layers leads to higher test accuracy. This is expected because more units can learn more features or patterns in the dataset.

The results from the models with 0 hidden layers show that even without hidden layers, a neural network can perform well on the task. However, the addition of hidden layers and the use of non-linear activations demonstrate the power of neural networks to model more complex relationships in the data, also leading to improved performance. These results align with the expections for MLPs, where increased capacity (through more units and layers) generally leads to better performance, up to the point where the network starts to overfit.

4.2 2 Hidden Layer MLP with Sigmoid and Leaky-ReLU Activation Functions

Based on the output of this experiment depicted in Figure 4, among the three activation functions—ReLU, Sigmoid, and Leaky ReLU—the ReLU model performs the best on the

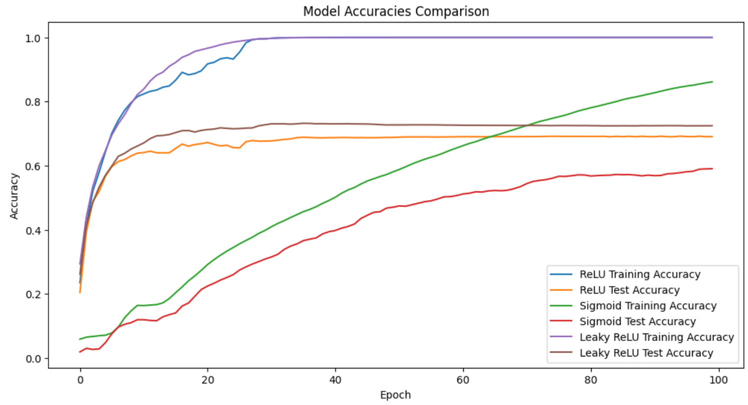

Figure 4: Model Accuracies Comparison

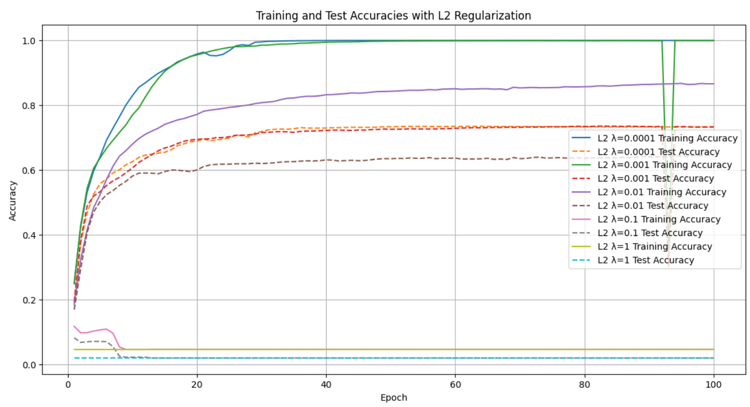

Figure 5: Training and Test Accuracies with L2 Regularization

training data. However, when we look at the test accuracies, the Leaky ReLU model has highest score, followed by the ReLU and then the Sigmoid.

In Figure 4, we see that the ReLU and Leaky ReLU models learn much faster than the Sigmoid model, since they have steeper slope for their accuracy curves, indicating rapid learning. On the other hand, the Sigmoid model's training accuracy curve is much less steep, meaning it is learning slower. Leaky ReLU outperformed ReLU on the test set, despite getting similar training accuracies, which could be due to the fact that Leaky ReLU permits a small gradient when the unit is inactive. This can let the model maintain a better gradient flow, which may lead to better generalization on the test set.

The results obtained are expected because ReLU and its variants (like Leaky ReLU), usually perform better than the Sigmoid function for deeper networks due to the reasons mentioned above. However, the difference in test accuracies between ReLU and Leaky ReLU shows that there might be an advantage to using Leaky ReLU in this case.

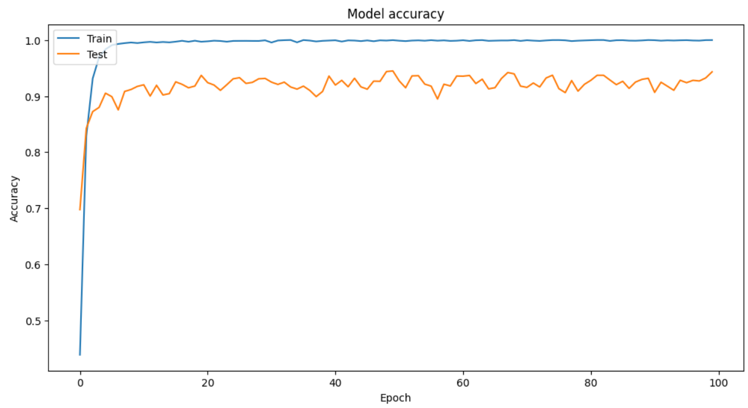

Figure 6: ConvNet Model Accuracy

4.3 2 Hidden Layer MLP with L2-Regularization

Here, we are looking at the effects of different L2 regularization strengths (λ values) on training and test accuracies. The results are depicted in Figure 5. Firstly, for a low λ (here 0.0001 and 0.001), the models show high training accuracy, which shows that they have good fitting on the training data. However, the test accuracy is lower compared to the training accuracy, telling us that there is some overfitting. Next, for a moderate λ (here 0.01), there is a decrease in training accuracy compared to the models with lower λ, which means that the regularization is taking effect and preventing overfitting. Now, the test accuracy is closer to the training accuracy, which means that we have a more generalized model that should perform better on unseen data. This λ value seems to strike a good balance between bias and variance. Finally, for a high λ (here 0.1 and 1), the models show much lower training accuracies, meaning that the penalty for complexity caused by the L2 regularization is too strong, leading to underfitting. The test accuracy is flat, showing that the models are not learning the patterns in the data well.

To conclude, a small amount of regularization (low λ) helps to prevent overfitting, but a moderate amount of regularization (λ=0.01) seems to offer the best trade-off of reducing overfitting while still getting a high test accuracy. Too much regularization (high λ) leads to underfitting, where the model is too simple.

4.4 ConvNet vs. MLP

In this experiment, we want to compare a Convolutional Neural Network (ConvNet) model to an Multilayer Perceptron (MLP) model. Based on our results depicted in Figure 6 using a ConvNet appears to increase accuracy compared to using MLPs. This can be due to the fact that ConvNets are designed to recognize patterns in visual data, which is the data found in the Sign Language MNIST dataset. MLPs do not have this ability, as the input data needs to be flattened, and we lose spatial context.

The validation accuracy reaches a higher percentage with the ConvNet, which means that it generalizes better on the test data than the MLP model. While there is a rapid gain in accuracy during the first few epochs as can be seen in Figure 6, which means ConvNet

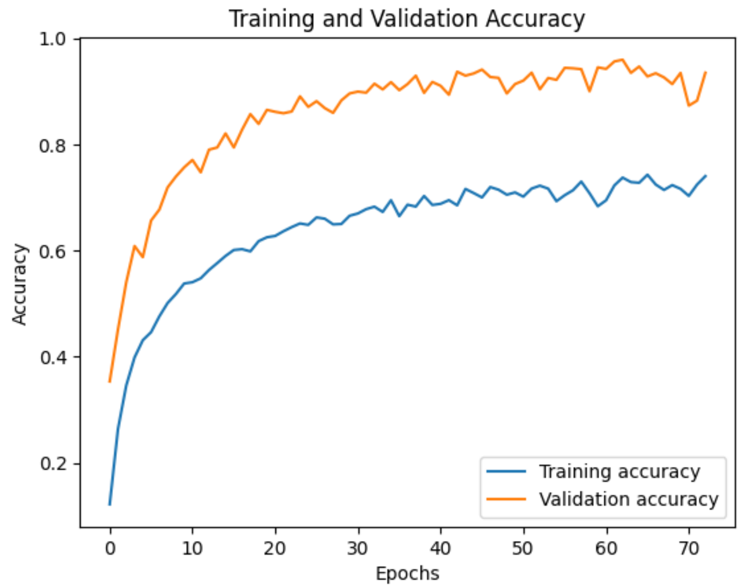

Figure 7: Training and Validation Accuracy with Modified MLP Architecture

is learning fast, the accuracy on the test set plateaus and even fluctuates. This could indicate of overfitting. In summary, the ConvNet model has better performance on the visual data of the Sign Language MNIST dataset compared to MLPs, which is consistent with expectations given the advantages of ConvNets for image-related tasks.

4.5 Best MLP Architecture

In this experiment, we tried to find an MLP architecture that optimizes performance. Here, we chose an architecture that has three dense layers with ReLU activation functions and dropout regularization. The input layer is a flattened version of the normalized image data. Each image's pixel values are scaled down to a range of 0 to 1 (by dividing by 255) and then flattened to form a 1D array (as required for the MLP's input). The model also has two hidden dense layers with 512 and 256 units respectively. The ReLU activation function is used for these layers, allowing the model to learn more complex data patterns. Between the layers, we apply dropout regularization with a rate of 0.5 to prevent overfitting. The final dense layer is the number of classes with a softmax activation function, which outputs a probability distribution over the classes to determine the right class label. The MLP was trained for up to 100 epochs with a batch size of 128. It achieved a peak validation accuracy, showing its ability to generalize to new data, which can be seen in Figure 7.

4.6 Training the MLP and ConvNet with Subset of Images

In this extra experiment, we tested the ConvNet model's performance on subsets of the image data. We can see in Figure 8 that there is an increase in the model's learning performance over five epochs. The proximity of the train and test lines tells us that that the model generalizes well; it performs similarly on both seen (training) and unseen (test) data.

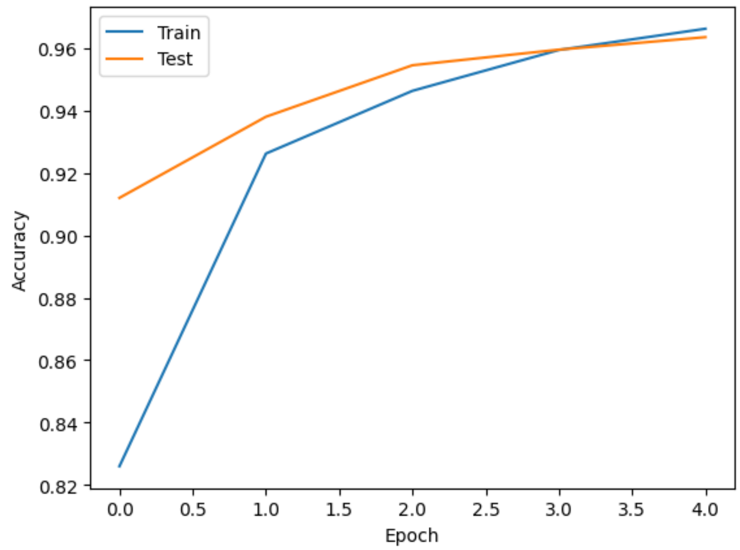

Figure 8: Accuracy with Subset of Image Dataset

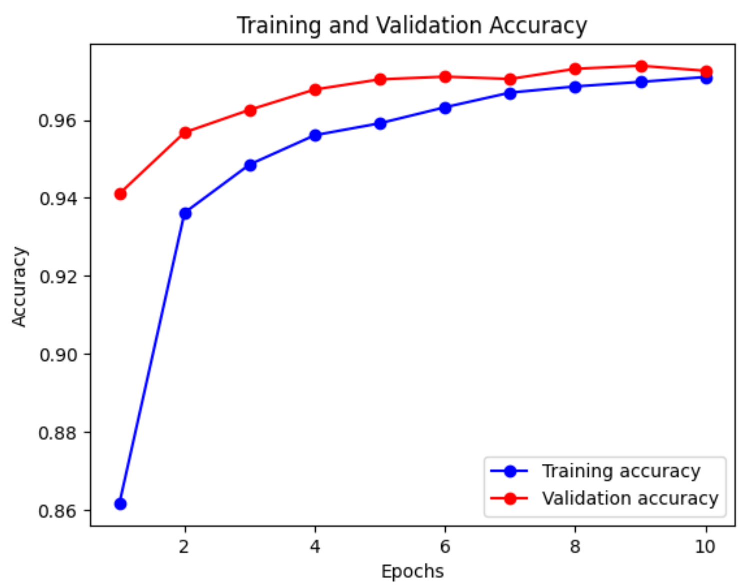

Figure 9

However, after the third epoch, the test accuracy begins to plateau around 96%, while the training accuracy continues to rise, still at a slower rate. This might suggest overfitting. Overall, the model exhibits a strong performance, with test accuracy reaching above 96% by the fifth epoch, which is indicative of an effective learning process up to this point.

4.7 Impact of Dropout Rates

Experimenting with different dropout rates is important to refine NNs, especially in tasks like image classification where overfitting can stunt performance. Dropout randomly sets a fraction of input units to 0 at each update during training time, which helps prevent overfitting by making the network less sensitive to the specific weights of neurons. By varying the dropout rates, we want to find a sweet spot where the network maintains enough capacity to learn without memorizing the noise. Lower dropout rates might lead to overfitting, while higher rates might lead to underfitting.

We experimented with dropout rates of 0.2 and 0.5. At a dropout rate of 0.2, the model achieved a test accuracy of 0.9727, showing a strong capacity to generalize well to unseen data. This performance is shown by the steady increase of both training and validation accuracy over epochs, as illustrated in Figure 9.

Increasing the dropout rate to 0.5 led to a slight reduction in test accuracy to 0.9664. This outcome indicates that while a higher dropout rate can contribute to reducing overfitting, it may also hinder the network's ability to learn, possibly leading to underfitting.

In summary, a lower dropout rate of 0.2 was more conducive to achieving higher accuracy in this scenario, suggesting that moderate regularization is good to mitigate overfitting while preserving the network's learning capabilities. This experiment validates the significance of dropout and highlights the importance of careful dropout rate selection.

5. Discussion and Conclusion

The results of our experiments have yielded several interesting outcomes. MLPs have significant gains in accuracy as the network depth increased, confirming that deeper networks can model more complex relationships. However, these gains plateaued beyond a certain depth, probably due to overfitting. Activation functions were also important, with ReLU and its variants (Leaky ReLU) outperforming the sigmoid function, which has slower learning.

The regularization experiments using L2 regularization showed its effectiveness in combating overfitting when using a moderate regularization strength (λ=0.001) offering a balance that improved test accuracy. Too much regularization (λ=0.1) led to underfitting, with high training loss and poor test accuracy. Our ConvNet experiments showed that they are superior over MLPs, with ConvNets having higher test accuracy, meaning they generalize better. ConvNets can learn spatial patterns in the image data much better than MLPs, which is beneficial for image classification tasks.

Our findings reinforce the importance of tuning neural network architectures and parameters to the specifics of the dataset at hand. In the future we can investigate different normalization strategies, explore more advanced regularization methods, and evaluate the performance of other neural network architectures for image classification tasks. We should also explore the performance of these models on a larger and more diverse set of image classification tasks to and expand upon our findings here.

6. Statement of Contributions

For this project, all three team members first agreed on our list of experiments, including the extra ones we would be conducting. Once decided, Rebecca started working on task 1, while Samantha started working on task 2, which involved the data acquisition and preprocessing for each dataset respectively, as well as the MLP model implementation. Amélie, Samantha and Rebecca then divided amongst them the implementations for the suite of experiments for task 3. All three team members then created an outline for the report together, and wrote the sections on the parts they implemented.

References

[1] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 24261–24272. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper_files/paper/2021/file/cba0a4ee5ccd02fda0fe3f9a3e7b89fe-Paper.pdf`.