Assignment 3: Classification of Image Data

Terrance Yan, Yunjia Zheng, Annie Kang

COMP 551 Winter 2024, McGill University

Abstract

Neural Network(NN) or Multi-layer Perceptron(MLP) has long been regarded as a fundamental but powerful method and received a lot of investigation with application in both the industrial and academic contexts. In this assignment, We started by preprocessing the data, including normalization and vectorization. we implemented a simple Feedforward Neural Network (FNN) from scratch under the guidance of the assignment for practice. To combine it with an application scenario, we acquired the ASL-MNIST dataset, carried out pre-processing steps, and utilized it as an experimental subject. We offered several loss and activation functions for flexibility, which were tuned and selected with other hyper-parameters for better performance. Adaptive Gradient Descent(Adagrad) optimizer was selected for this task due to consideration of convergence and performance. Additionally, we performed gradient verification to validate our backpropagation(BP) algorithm. Through experiments, we found that adding hidden layers and using Leaky ReLU as an activation function slightly improved MLP performance. Key findings include that models with more hidden layers yielded higher accuracy, with the best MLPs approaching 80%. However, ConvNets significantly outperformed MLPs by achieving a higher accuracy of 95.7%, underscoring their suitability for image classification tasks. Additionally, increasing the amount of training data consistently enhanced model accuracy for both network types.

1 Introduction

Neural Networks (NNs) or Multi-layer Perceptrons (MLPs) have gained popularity in machine learning due to their complex layer structures, non-linear activation functions, and the backpropagation algorithm, which simplify training and are well-suited to handling large datasets.[1]

This assignment tasked us with exploring the practical application and effectiveness of Multi-layer Perceptrons (MLPs) in image classification, a critical area within machine learning that has seen extensive research and development. Specifically, we focused on the ASL-MNIST dataset[2], a collection of images representing American Sign Language (ASL) hand signs that offers a unique challenge due to its visual complexity and the importance of accurate classification for communication applications.

A study in 2022 introduced a Two-Stream Mixed (TSM) method within a Convolutional Neural Network (CNN) framework to enhance sign language recognition with the ASL-MNIST dataset By incorporating feature extraction and fusion operations for sequential images, the TSM-CNN model, especially TSM-ResNet50, achieved an impressive accuracy of 97.57% on the MNIST and ASL datasets.[3] Another study explores the use of capsule networks to demonstrate their superior performance over the LeNet model in sign language character recognition. [4]

After preprocessing the dataset to correct for scale disparities and remove extraneous data components, we eliminated the scaling error and unnecessary components inherent in the data and flattened it for further classification. Then a Feedforward Neural Network (FNN) classifier was designed from scratch. By defining several loss functions and activation functions, we constructed an FNN class with flexibility utilizing modified Adagrad as an optimizer. To ensure the reliability of our model, we implement gradient verification, which justifies our BP and thus the optimizer. After that, the hyper-parameter tuning was carried out in a grid search to decide the layer structures and activation functions. This process was instrumental in refining our model to achieve the best possible classification performance on the ASL-MNIST dataset.

Our experiments yielded significant insights, particularly highlighting the advantages of deeper network architectures and the effectiveness of certain activation functions, such as Leaky ReLU, in enhancing model performance. Moreover, we extended our analysis to compare MLPs with Convolutional Neural Networks (ConvNets), which are renowned for their prowess in image-based tasks. The comparison underscored ConvNets' superior accuracy in classifying the ASL-MNIST images, thereby reinforcing the critical role of model selection in machine learning applications. This exploration not only reaffirmed the capabilities of neural networks in

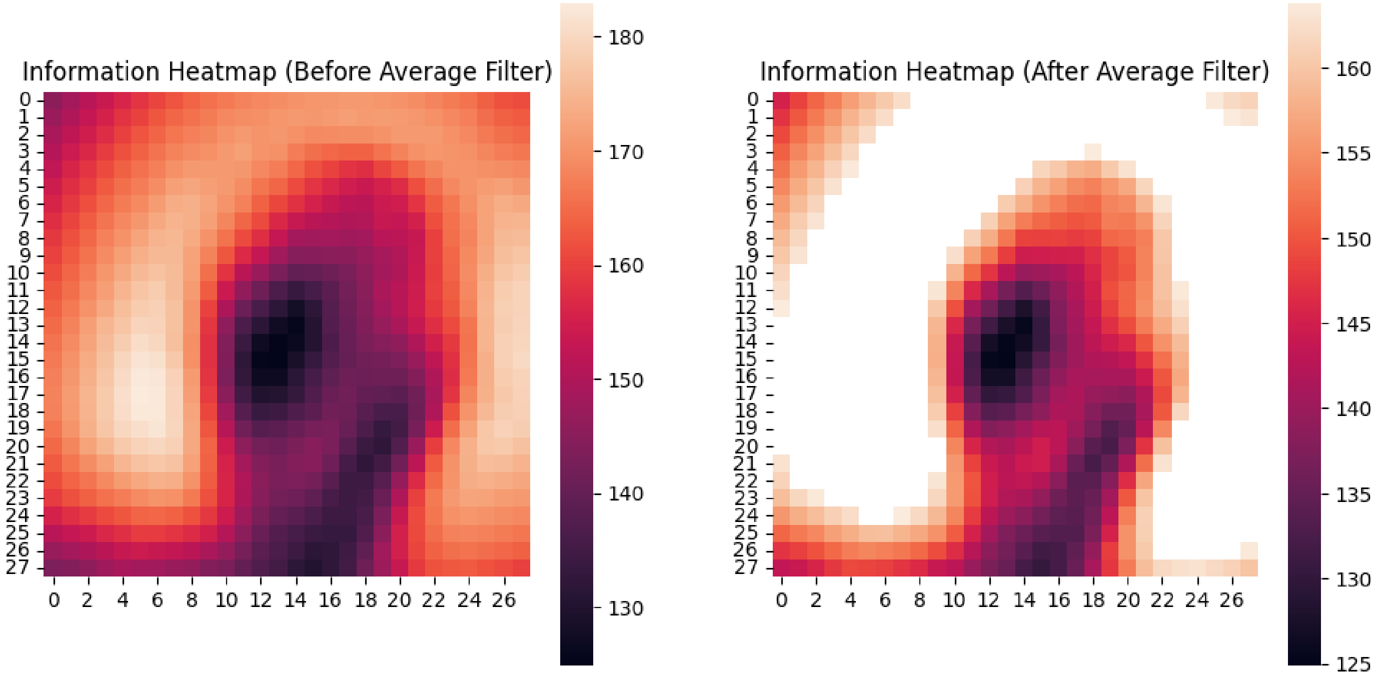


Figure 1: Information Heatmap with(right) and without(left) Average Filter

handling complex classification tasks but also contributed to the broader discourse on the optimal strategies for implementing such models in real-world scenarios.

2 Datasets

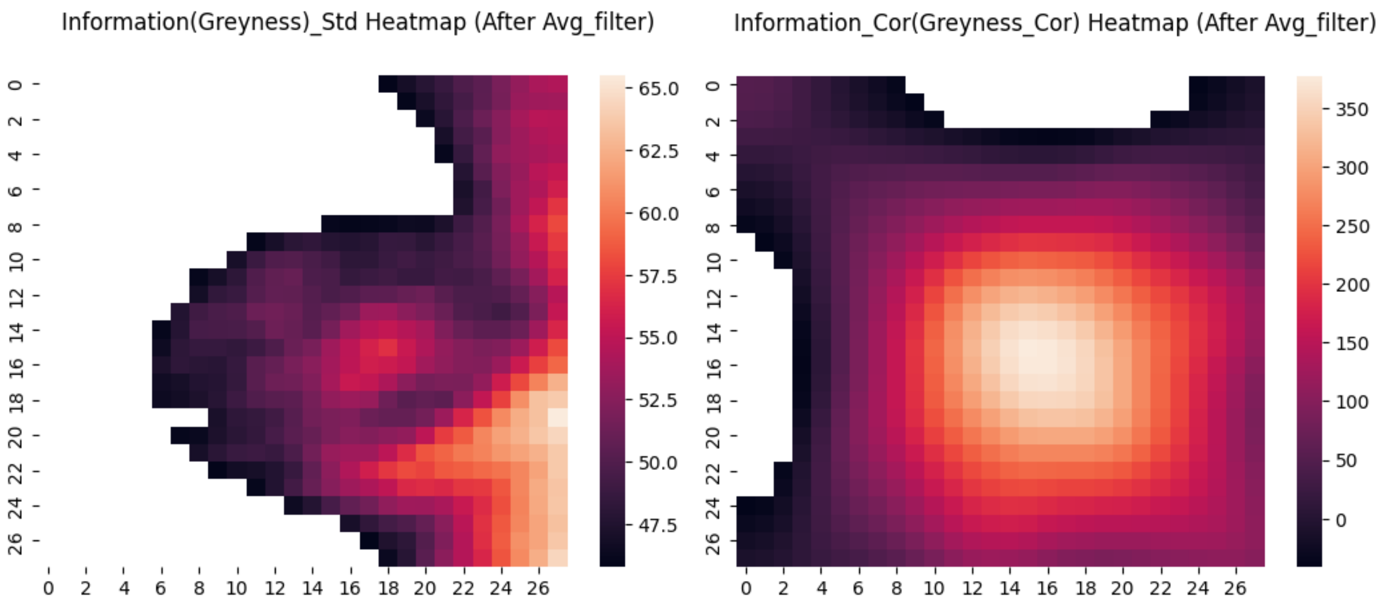
Under the guidance of the assignment explanation, we chose the ASL-MNIST as our data source. ASL-MNIST can be regarded as one of the classical extrapolations to the famous MNIST dataset, incorporating the image with the size of 28x28 inducing 784 pixels total and with only a gray channel used in this case(thus the process is simplified). We performed our data pre-processing initially by separating the labels which were then followed by carrying out the necessary vectorization(flattening) and one-hot encoding for the MLP to successfully execute. To prevent the scaling error inherent in the data, we performed normalization then to get rid of the potential menace.

FNN is an affine-transformational variant algorithm, requiring the structure to be rigidly located to learn the scrutable features, thus we need to check whether our dataset satisfies this requirement.[5] We did a brief visualization-based analysis on our dataset for that. We utilized the information heatmap and its derivatives, as offering intuitiveness, and successfully observed a structural pattern existing in our dataset that justifies our selection of MLP as a learning structure. In Figure 1, a palmar shape is observed, testifying to the significance of the dataset per se and positional rigidity of the dataset, which is further emphasized by average filter f (defined as $f(x) = \max\{x - mean(x), 0\}$). We further bear out this observation and conclusion by Figure 2a that demonstrates the ambient variation of information in the information-accumulating region(the region that the palmar shape reveals), with concomitantly suggesting different classes exist. We further measured the informational structural association by utilizing the correlation operand and observed that after the average filter, most pixels(informational elements) strongly covariate around especially again in the information-accumulating region. This result displayed in Figure 2b suggests CNN(Convolutional Neural Network) could be a better solution for this task since it utilizes this covariant informational structure.

3 Results

Firstly, we tested on MLPs with 0,1 and 2 hidden layers, where each layer contains 32, 64, 128, 256 hidden units. The result shows that the best model with the highest test accuracy for MLPs with 1 and 2 hidden layers are those with 256 hidden units. The hidden layers allow the model to learn more complex patterns than a linear model.

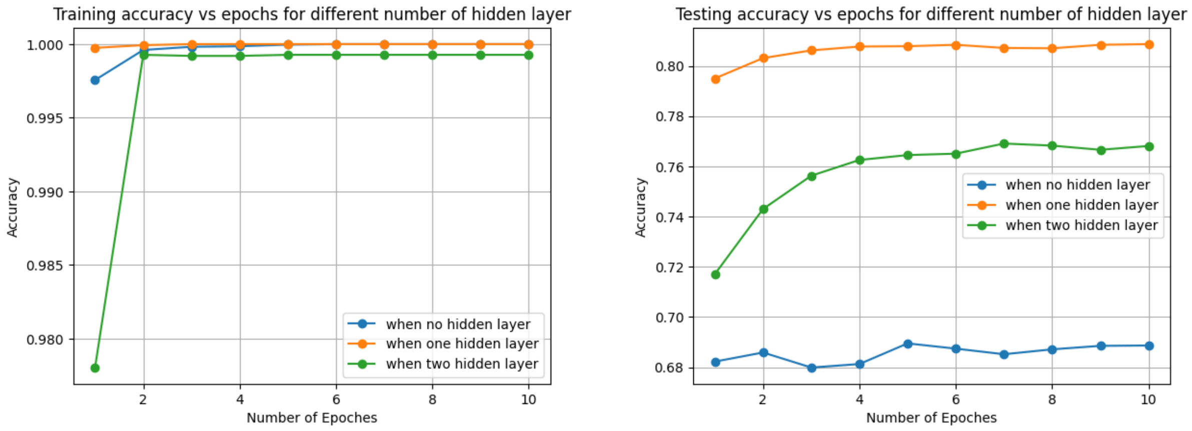
However, the test accuracy for the best models above are 69.2%, 80.9%, and 77.2% for 0, 1, and 2 hidden



(a) Information Variation Heatmap with Average Filter (b) Structural Information Correlation with Average Filter

Figure 2: Self and Structural Information Covariation

layers respectively, shown in Figure 3. This phenomenon indicates that increasing the depth of the network by adding more hidden layers allows the model to represent more complex functions and interactions among inputs. However, as the network gets very deep, overfitting can occur. These results align with common expectations in deep learning that adding non-linearities and increasing network depth generally leads to better performance by capturing more complex relationships between the given datasets.



(a) Train Accuracy with respect to Epochs for 0,1 and 2 hidden-layer models (b) Test Accuracy with respect to Epochs for 0,1 and 2 hidden-layer models

Figure 3: Accuracy and Convergence for three models

Then we substitute ReLU to Sigmoid and Leaky ReLU for 2-hidden-layer MLP with 256 hidden units in each layer. The accuracy is 78.1% for sigmoid and 79.9% for Leaky Relu. Thus among the three, Leaky ReLU achieves the highest accuracy and it is reasonable as ReLU prevents the nodes with negative output from being ignored compared to ReLU. Since our neural network is still quite shallow and we stopped training at 10 epochs, the potential overfitting problem for Leaky ReLU did not appear at this stage. In general, Leaky ReLU is quite popular because it also prevents the vanishing gradient problem of using the sigmoid function as an activation function in deep neural networks.

The third experiment we conducted is integrating L2 regularization into the 2-hidden-layer MLP with lambda = 0,0.01,0.1,0.5. To show the effect of L2 regularization, we trained the model on 30 epochs. Notice that, when there is no regularization, the test accuracy keeps increasing during the whole process, indicating that the threshold of overfitting has not occurred yet. In the circumstances, the main effect of adding L2 regularization will be increasing bias leading to lower accuracy, as shown in Figure 4.

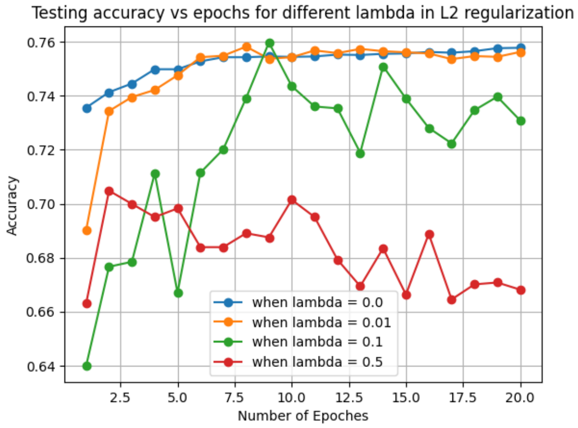
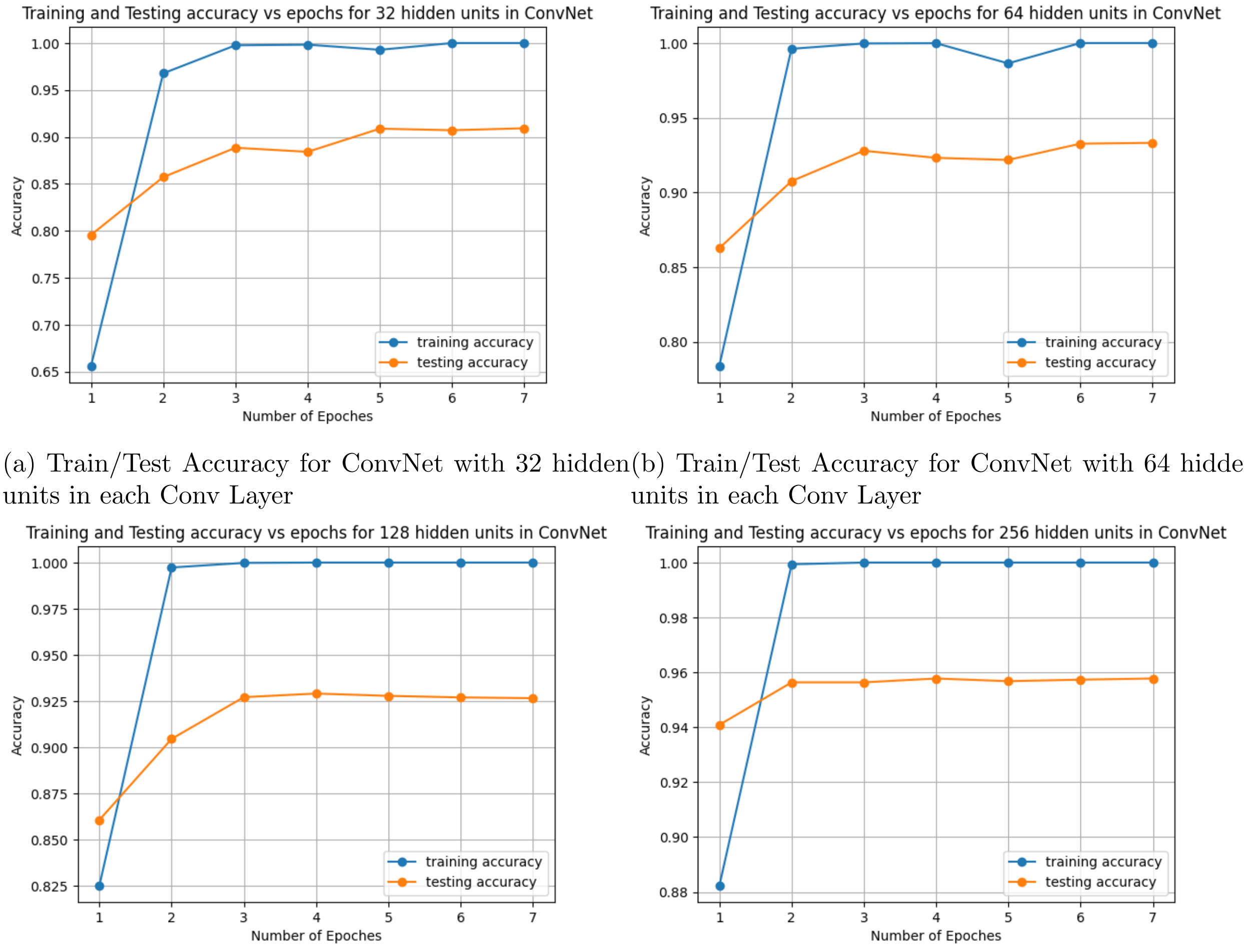


Figure 4: Accuracy and Convergence for different L2 Regularization

Next, we compared the performance of using a ConvNet model and our MLP implementation. The kernel size for each Conv2D layer is 3*3 along with a 1*1 stride and the pool size for the MaxPool layer is 2*2. The test accuracy for ConvNet with 32,64,128 and 256 hidden units are 90.6%, 92.3%, 92.6%, and 95.7% respectively. We can see that more non-linearity also gives ConvNet more power to find the graph pattern and thus achieve higher accuracy. Compared the figures below with Figure 3, we can see that not only the accuracy is increased by using ConvNet, but ConvNet also converges faster than MLP, for example, ConvNet with 256 hidden units almost converges in one epoch.



(a) Train/Test Accuracy for ConvNet with 32 hidden units in each Conv Layer

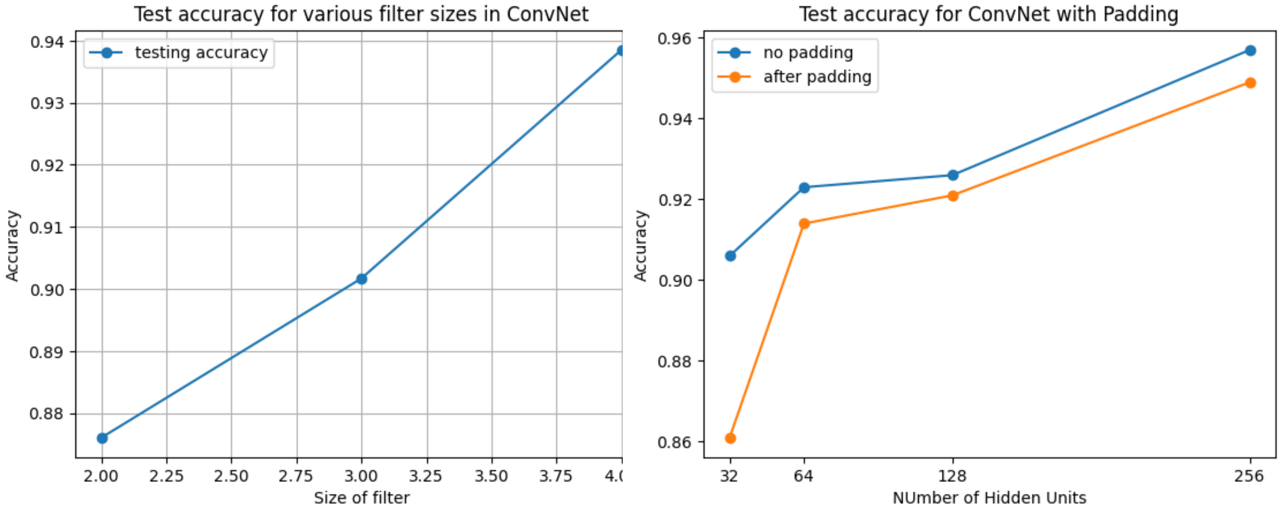
(b) Train/Test Accuracy for ConvNet with 64 hidden units in each Conv Layer

(c) Train/Test Accuracy for ConvNet with 128 hidden units in each Conv Layer

(d) Train/Test Accuracy for ConvNet with 256 hidden units in each Conv Layer

Figure 5: Accuracy and Convergence for ConvNets

It is not surprising that ConvNet performs better than our MLP models. ConvNet is designed to recognize local patterns by pooling, which makes it possible to detect patterns in all locations and orientations. This feature gives a more accurate interpretation of a picture and thus achieves higher accuracy.



(a) How filter size affects the test accuracy

(b) How padding affects the test accuracy

Figure 6: ConvNet Hyperparameters

We also looked at how filter size and padding affect the test accuracy shown in Figure 6. For the sake of training time, we used 64 hidden units for each layer when testing various filter sizes. A larger filter size helps increase the test accuracy by involving a larger proportion of the picture. Hence, capturing more information from the picture makes it possible to detect the patterns at a higher accuracy. However, adding padding does not help for all 4 cases of hidden unit numbers. Potential reasons could be that our images are quite small (28*28 pixels) and adding ”blank” lines around might confuse the model, or that additional zeros at the boundary add up the noises.

As regards the best model we could have using our MLP implementation, we chose a one-hidden-layer model with 256 hidden units and no L2 regularization along with LeakyReLu as an activation function since it is the best model from our previous experiments. As for hyper-parameters like learning rate and batch size, we performed a grid search with 3-fold cross-validation where the validation accuracy is listed below. Thus we chose batch_size = 32 and learning_rate = 0.05 as the hyper-parameters for our model and the final test accuracy is 81.2%, slightly higher than the one using ReLu but still worse than the ConvNet.

batch size	learning rate	validation accuracy
32	0.01	0.995
32	0.05	0.997
32	0.1	0.996
32	0.5	0.997
64	0.01	0.996
64	0.05	0.995
64	0.1	0.997
64	0.5	0.995
128	0.01	0.991
128	0.05	0.994
128	0.1	0.992
128	0.5	0.993

Table 1: HyperParameters for MLP

The final tests are investigating the influence of input data size on the test accuracy, shown in Figure 7. We fed 1, 10, 100,1000, and 10000 data points into our best MLP model just found and a ConvNet with 256 hidden units/layer. Note we need a validation set for ConvNet so the accuracy when there is only one input image is 0. The result shows that the more data used for training, the more accurate the model predicts the testing data correctly. Also, ConvNet is more data-driven than MLP because it only outperforms MLP when there are enough input images.

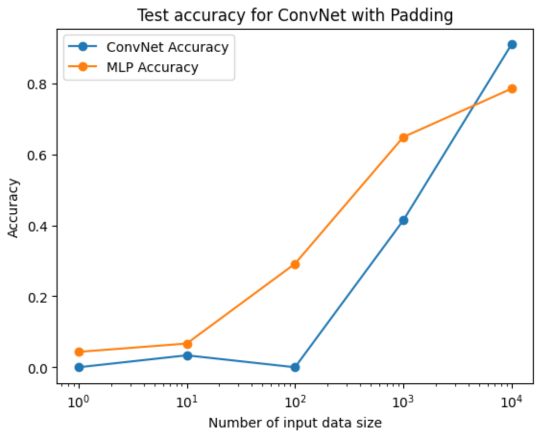


Figure 7: Test Accuracy for different input datasize

4 Conclusion and discussion

This study highlights the crucial role of preprocessing and data augmentation in boosting the accuracy and efficiency of models, showcasing how the architecture of the model, the quality of data, and the approach to training work together.

The key takeaways from this assignment include our deep dive into the effectiveness of neural network architectures like MLPs and ConvNets for the ASL-MNIST dataset. Our results show how important it is to choose the right network depth and activation functions, with the use of Leaky ReLU and additional hidden layers being key to improving MLP performance. Additionally, we found ConvNets to be significantly better at processing image data, due to their structure which is adept at recognizing patterns in a way that mimics human vision.

Future research might begin by exploring the use of Transfer Learning which offers a performance boost by leveraging pre-trained models, thereby reducing training time and enhancing accuracy on smaller datasets. Additionally, adopting advanced regularization and optimization techniques might refine models further, making them more generalizable and efficient and ultimately leading to more accurate and robust sign language recognition systems.

5 Statement of Contributions

This section outlines the contributions of each team member to the project.

Terrance Yan

- Data pre-processing and analysis
- MLP class with corresponding components
- Writing of Abstract and Dataset

Yunjia Zheng

- Communication and final checks
- Part3(starting 3.2) and additional experiments with corresponding report sections

Annie Kang

- Abstract, Introduction, Conclusion, and Discussion
- Gradient verification, hyper-parameter tuning, and Part 3.1

References

[1] Z.-H. Zhou, *Machine Learning*, trans. by S. Liu. Springer Singapore, 2021, ISBN: 978-981-15-1967-3.

[2] Tecperson, *Sign language mnist*, `https://www.kaggle.com/datasets/datamunge/sign-language-mnist/data`, Accessed: 25 March 2024, 2017.

[3] Y. Ma, T. Xu, and K. Kim, "Two-stream mixed convolutional neural network for american sign language recognition," *Sensors*, vol. 22, no. 16, p. 5959, Aug. 9, 2022. DOI: `10.3390/s22165959`.

[4] M. Bilgin and K. Mutludoğan, "American sign language character recognition with capsule networks," Dec. 2019. DOI: `10.1109/ISMSIT.2019.8932829`.

[5] K. P. Murphy, *Probabilistic Machine Learning: An introduction*. MIT Press, 2022. [Online]. Available: `probml.ai`.