Comparison of Deep-Learning Neural Network Architectures for Image Classification

University of Texas at Dallas

Kenan Stredic
KLS190005
Errik Johnson School of Engineering

Zachary Young
ZFY200000
Errik Johnson School of Engineering
University of Texas at Dallas

Abstract*— Machine learning has seen significant technological advancements regarding neural networks and computer vision. The concepts of machine learning and artificial neural networks can be used for a variety of applications, including image recognition, natural language processing, healthcare, finance, and autonomous driving. This paper takes a deep dive into multiple types of neural network architectures and how they behave when presented with different data sets for training, as well as an analysis of a new neural network architecture that presents key differences and challenges when compared to existing architectures. The MNIST and Fashion-MNIST datasets have played an important role in advancing the field and continue to be useful resources for researchers and practitioners alike. It has also been used as a benchmark dataset for computer vision and has been used to evaluate the performance of various machine learning models, including deep learning models such as convolutional neural networks. This study analyzes the hidden layers and output layers of the MLP and LeNet architectures and tests them against different datasets according to a variety of metrics. The LeNet, MLP, and custom neural network architectures are compared and analyzed to gain insight into the strengths and shortcomings of various Machine Learning Approaches. Overall, this study systematically compares the performance and training efficiency of each neural network as it pertains to each data set and draws meaningful conclusions based on a statistical analysis of the results.

***Keywords*—neural network, machine learning, architecture, convolutional, layers, datasets, training models, training metrics**

I. Introduction (*K: 60%, Z: 40%*)

Machine learning has advanced significantly in recent years, fueled by improvements in computing power, data availability, and algorithmic innovations. This report will take a deep dive into some of these recent technological advancements. There exist several key concepts that are necessary to understand the project moving forward that need to be aptly defined. Key concepts include:

- Deep Learning
  - Deep learning is a type of machine learning that involves training artificial neural networks with multiple layers. Deep learning has been particularly successful in image recognition, natural language processing, and speech recognition.[4]
- Reinforcement Learning
  - Reinforcement learning is a type of machine learning that involves an agent learning to interact with an environment to maximize a reward signal.[5] Reinforcement learning has been successful in game playing, robotics, and autonomous driving.
- Transfer Learning
  - Transfer learning is a technique that involves leveraging knowledge gained from one task to improve performance on another related task.[6] Transfer learning has been used to improve performance in image recognition, natural language processing, and other applications.
- Generative Models
  - Generative models are a type of machine learning model that can generate new data samples that are like training data.[7] Generative models have been used for image and text generation, data augmentation, and other applications.

- Backpropagation
    - Backpropagation is a widely used algorithm for training artificial neural networks, including deep neural networks, convolutional neural networks, and recurrent neural networks. The algorithm uses the chain rule of calculus to compute the gradients of a loss function with respect to the weights of the neural network.[8]

- Gradient Descent
    - Gradient descent is a widely used optimization algorithm in machine learning for finding the optimal parameters of a model that minimize a given cost function. The algorithm works by iteratively adjusting the model parameters in the direction of the steepest descent of the cost function.[9]

These advancements and many more have led to significant progress in a wide range of applications, including image recognition, natural language processing, healthcare, finance, and autonomous driving.[10] As machine learning continues to evolve, it is likely to play an increasingly important role in many areas of our lives. The experiments to follow will examine multiple datasets to accurately reflect the wide variety of applications for deep learning. These datasets are MNIST and Fashion-MNIST.

The MNIST dataset is a well-known benchmark dataset in the field of machine learning, specifically in image classification, consisting of 70,000 grayscale images of handwritten digits from 0 to 9, each of size 28x28 pixels.[11] MNIST stands for Modified National Institute of Standards and Technology. The MNIST dataset is popular because it is relatively small and simple, making it easy to experiment with different machine learning models and techniques. It has also been extensively studied, and many state-of-the-art machine learning models have been trained and evaluated on this dataset. This dataset has been used as a benchmark for evaluating machine learning models in tasks such as image recognition, classification, and feature extraction.[11] It has also been used as a starting point for many researchers and students learning about machine learning. Overall, the MNIST dataset has played an important role in advancing the field of machine learning and continues to be a useful resource for researchers and practitioners alike.

The Fashion-MNIST dataset is like the original MNIST dataset and was created as an alternative to evaluate more complex machine learning algorithms using more diverse images. The 10 classes in the Fashion-MNIST dataset are T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, and Ankle boot. Each image in the dataset is a 28x28 pixel grayscale image. The dataset is split into a training set of 60,000 images and a test set of 10,000 images, which are used to train and evaluate machine learning models.

The Fashion-MNIST dataset has become a popular benchmark dataset in the field of computer vision and has been used to evaluate the performance of various machine learning models, including deep learning models such as convolutional neural networks. This dataset is useful for researchers and practitioners interested in evaluating the performance of machine learning models on real-world images of clothing items.[12] It has also been used as a teaching tool for students learning about computer vision and machine learning. Overall, the Fashion-MNIST dataset has played an important role in advancing the field of machine learning and computer vision.

MNIST and Fashion-MNIST were chosen for the following experiments due to their diverse subject matter. It was determined that the handwritten numbers included in MNIST, and the articles of clothing included in Fashion-MNIST were adequately unique, and therefore able to showcase the various neural network architectures. In addition, each dataset consists of a different number of images while maintaining the same image size, with MNIST and Fashion-MNIST, which consist of 28x28 pixel grayscale images.[12]

We decided to test these datasets against three kinds of neural networks to gain insight into the strengths and shortcomings of various Machine Learning Approaches. LeNet and MLP neural networks were chosen for the following experiments alongside a custom neutral network of our own design. These three neural network architectures take a different approach to the image classification algorithm and present their own strengths and weaknesses throughout the development and training processes of a neural network.

The LeNet neural network is a type of convolutional neural network that was developed in the early 1990s by Yann LeCunn and his colleagues.[1] It was one of the first successful neural networks used for image recognition tasks. This type of neural network consists of several layers, including convolutional layers, pooling layers, and fully connected layers. The input to the network is a grayscale image, and the output is a probability distribution over the different classes that the image could belong to.

The first layer of the network is a convolutional layer that applies a set of filters to the input image. Each filter is a set of weights that are learned during the training process. The output of the convolutional layer is a set of feature maps, which are the result of convolving each filter with the input image.

The next layer of the network is a pooling layer, which reduces the dimensionality of the feature maps by down sampling them. This helps to make the network more computationally efficient and reduces the risk of overfitting. After several convolutional and pooling layers, the feature maps are flattened and fed into one or more fully connected layers.[1] These layers perform a nonlinear transformation of the input and produce the final output of the network, which is a probability distribution over the different classes.

The LeNet neural network was originally developed for the task of recognizing handwritten digits in the MNIST dataset, but it has since been adapted for other image recognition tasks as well. This is why LeNet was chosen for the following experiment: to compare its performance on the MNIST dataset, for which it was originally developed, and its performance on the other selected dataset, revealing how this architecture can by adapted for a variety of image classification applications.

The Multi-Layer Perceptron (MLP) neural network is a type of artificial neural network that is widely used for supervised learning tasks such as classification and regression.[13] It consists of three or more layers of artificial neurons: an input layer, one or more hidden layers, and an output layer. Each neuron in the MLP is connected to every neuron in the previous layer and the next layer, forming a fully connected network. The input layer receives input data, which is passed through the hidden layers, where nonlinear transformations are applied to the data. The output layer produces the final output of the network, which is a prediction or decision based on the input data.

The neurons in the MLP are organized in layers, where each neuron in a layer is connected to all neurons in the previous layer. Each connection between two neurons has a weight associated with it, which is learned during the training process. The weights determine the strength of the connection between neurons and can be adjusted to optimize the performance of the network. During training, the MLP adjusts the weights of the connections between neurons in order to minimize a loss function, which measures the difference between the predicted output of the network and the true output. This process is known as backpropagation and is based on the chain rule of calculus.

The activation function used in the MLP is typically a sigmoid function or a rectified linear unit (ReLU) function. The sigmoid function maps any input value to a value between 0 and 1, while the ReLU function returns the input value if it is positive, and 0 otherwise.[13] These nonlinear activation functions allow the MLP to learn complex nonlinear relationships between the input and output data.

The MLP architecture is a versatile and powerful model that can be used for a wide range of supervised learning tasks, including image recognition, natural language processing, and financial forecasting. However, it can be prone to overfitting if the model is too complex or if the training data is limited. Regularization techniques such as weight decay and dropout can be used to address this issue.

In designing our custom neural network architecture, we wanted to test a specific aspect of existing neural network architectures. Based on our research, the impact of the final fully connected layers in many architectures seemed like an intriguing place to introduce variants two existing models. We elected to build our custom architecture off the LeNet architecture for comparison reasons, as well as the fact that it is like various architectures that were covered in class and our various homework assignments. This gave us confidence when trying to tackle the idea of a unique architecture and served as a solid basis and starting point for neural network development. Through our research, we noticed that many architectures employed multiple fully connected layers to gradually flatten the data into an acceptable output. Our custom neural network is going to shortcut this process with a single fully connected layer as well as simple dense layers to flatten our output data. We hope to see a significant change in the accuracy and validity of the output data when fully connected layers are removed, and this process is circumvented. Although, if the accuracy of the model remains similar, this will have interesting implications on the necessity of fully connected layers to flatten data.

The following set of experiments will compare the performance and training efficiency of the LeNet, MLP, and custom neural network architectural structures as they process unique and varied data from the MNIST and fashion- MNIST datasets, each pertaining to a distinct application end use case. In order to test the effectiveness of each neural network architecture a variety of loss functions, training metrics, and runtime statistics will be employed and generated in order to accurately measure the performance of each neural network as it pertains to each data set. The employed metrics include but are not limited to confusion matrices, precision, recall, F1-score, and AU-ROC. It is our intention to showcase the capabilities of each neural network and each data set using varied and rigorous training accurately and elastically across several variables.

II. Related work (*K: 30%, Z: 70%*)

A. Gradient-Based Learning Applieed to Document Recognition by Yann LeCunn, Leon Buttou, Yoshua Bengio, and Patrick Haffner [1]

When looking for ways to structure our study and design our experiments, we were inspired by the works of LeCunn, Buttou, Bengio, and Haffner regarding Gradient-Based Learning algorithms. Their study took a deep dive into neural networks that employ backpropagation and gradient based algorithms for two-dimensional recognition and classification. Through their studies and testing LeCunn's team found that neural networks designed with these qualities in mind, specifically those tailored to two-dimensional image recognition, significantly outperform other forms of machine learning. When looking into their work, what truly interested us and pushed our research in the direction of image recognition was the application that LeCunn's team focused on. Their neural network was trained to identify handwritten documents, which was very similar to the subject matter found in the MNIST data set of handwritten numbers. While this study makes a very good point regarding the flexibility of global training and graph transformer algorithms, we elected not to take our research in that direction due to the scope of our machine learning class [1].

B. Performance Analysis of Various Activation Functions in Generalized MLP Architectures of Neural Networks by Bekir Karlik and A Vehbi Olgac[2]

This research paper takes a deep dive into the various activation functions that can be used with artificial neural networks, such as those described earlier in the document. Specifically, Karlik and Olgac look at the MLP neural network architecture and its activation functions. This study analyzes the hidden layers and output layers of the MLP architecture and tests them against a variety of activation functions. For the purposes of our experiment Karlik and Olgac conclusions about sigmoid functions are most important, but all the activation functions used in their study are noteworthy. This study found the MLP architecture and its derivative to be very fast computing and agile. Notably, the study looks at what activation functions are most effective in different problems or applications, which proved very useful for the purposes of our experiment. [2]

C. *Towards Accountability for Machine Learning Datasets: Practices from Software Engineering and Infrastructure by Ben Hutchinson, Andrew Smart, Alex Hanna, Emily Denton, Christina Greer, Oddur Kjartansson, Parker Barnes, and Margaret Mitchell [3]*

During the development of our experiment, the issue of how to choose our datasets became quite prevalent. Many questions about the validity of a machine learning data set needed to be answered. This research paper not only takes into question the validity of machine learning data sets but dives into the ethics and processes by which machine learning data sets are created, shared, and used for research and commercial purposes. Towards accountability for machine learning datasets questions the visibility of the creation and development of machine learning data sets and how more visibility would affect their use. Through their research, Hutchinson 's team developed a framework for the development of datasets and introducing more transparency to this process, as well as the use of machine learning data sets and how to maintain their transparency and visibility as they are shared and reused. Based on the framework presented in this research paper, we felt confident about the ethics and development process of the MNIST and fashion-MNIST datasets for our research and experiments. [3]

III. Proposed Approach *(K: 70%, Z: 30%)*

To compare the performance of a LeNet neural network and a MLP neural network using the MNIST and Fashion-MNIST datasets, we would need to design an experiment that follows a systematic and rigorous approach. Here is a general outline of the experiment setup:

1. Data Preparation:
    - Download the MNIST and Fashion-MNIST datasets from a reliable source.
    - Preprocess the data by normalizing the pixel values to the range [0,1] and dividing it into training and testing sets.
    - Clean, convert, and prepare the data as necessary for each data set neural network combination. This includes but is not limited to converting the data to one-hot encoded vectors and reshaping data to 4D tensors.

2. Neural Network Architecture:
    - Design a LeNet neural network architecture with convolutional layers, pooling layers, and fully connected layers.
    - Design a MLP neural network architecture with one or more hidden layers, depending on the complexity of the dataset.
    - Design a custom neural network architecture utilizing convolutional layers pooling layers and fully connected layers.

3. Hyperparameters:
    - Define a range of hyperparameters such as learning rate, number of epochs, batch size, activation functions, regularization parameters, and optimizer.
    - Use a hyperparameter search technique such as grid search or random search to find the best set of hyperparameters for each model and dataset combination.

4. Model Training:
    - Train the LeNet and MLP models separately on each dataset using the best set of hyperparameters.
    - Monitor the training process by measuring the loss and accuracy of the models on both the training and validation sets.
    - Save the best model weights based on the performance on the validation set.

5. Model Evaluation:
    - Evaluate the performance of the LeNet and MLP models on the test set of each dataset by measuring their accuracy and confusion matrix.
    - Compare the performance of the models on each dataset by computing statistical significance tests such as t-tests or ANOVA.

6. Result Analysis:
    - Analyze the results and draw conclusions based on the performance of the models on each dataset.
    - Identify the strengths and weaknesses of each model architecture and how they compare to each other.
    - Suggest possible future directions for improving the performance of the models on the datasets.

Overall, this experiment setup would allow us to systematically compare the performance of a LeNet neural network, MLP neural network, and custom neural network on the MNIST and Fashion-MNIST datasets and draw meaningful conclusions based on statistical analysis of the results.

When looking at the types of neural networks that will be analyzed in this experiment it is important to note both the benefits and drawbacks of each neural network architecture. The LeNet architecture was originally designed for image recognition tasks and therefore boasts A variety of benefits such as its ability to capture spatial features. The convolutional layers can detect local patterns and edges in the images of each data set. These images are then combined through pooling layers into more complex features. LeNet neural networks also use parameters efficiently, with a smaller number of parameters than other neural network architectures.

The LeNet architecture is easier to implement than other architectures through tools such as PyTorch and TensorFlow making it a good choice for applications with limited computational resources. LeNet architectures are also very robust against variations in input the unique series of convolutional and pooling layers allow the network to handle changes in image size and content. One drawback of this architecture is its limited applications. It was designed specifically for image recognition mission tasks, but this is not an issue in our experiment which focuses specifically on image recognition.

Compared to newer neural network structures LeNet has a rather shallow architecture which can limit its ability to learn complex features. This is why we chose MLP as our other architecture. LeNet performance is highly dependent on the choice of hyperparameters which is why we put forth extensive research in this section of the experiment.

The MLP neural network architecture is a type of feed forward neural network, giving it a unique set of benefits and drawbacks when compared to other architectures. Most importantly the MLP neural network architecture can be applied to a variety of tasks including classification, regression, and reinforcement learning as well as different applications.[13] We wanted to contrast this with the LeNet architecture which is designed specifically for image classification. MLP neural networks are very flexible and able to adapt to a variety of input data. This neural network architecture can also be used in combination with other architectures.

MLP neural networks can be trained very efficiently via the back propagation optimization algorithm, and they are relatively easy to interpret regarding their decision-making process. Through our research we discovered that it is quite common for MLP neural networks to be overfitted especially in the cases of many parameters and small data sets. Another drawback we found when researching MLP neural networks is the model's reliance on large datasets, but we found our chosen datasets to be adequately large. The performance of an MLP model is highly defendant on the tuning of hyperparameters such as initial weights and biases of the network, which called for extensive research on our part.

The custom neural network architecture is not a fully-fledged neural network architecture but rather an extension of the LeNet architecture used in the experiment. The design of the custom neural network calls for a reduction in fully connected layers during the data flattening stage. The main purpose of including the custom neural network architecture is to test the application of fully connected layers in neural network architecture. Therefore, the expected outcomes and performance metrics of this architecture will not be revealed until the experiments completion.

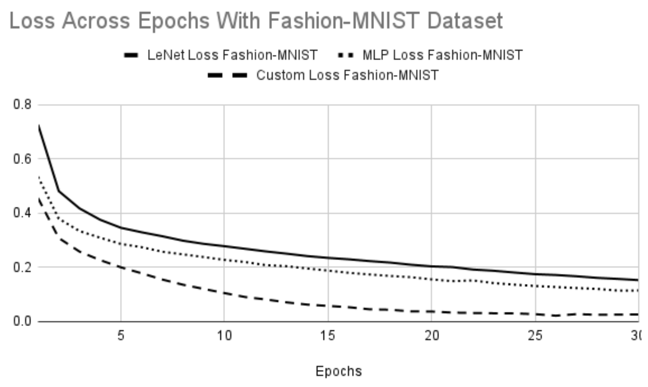
IV. Experiments *(K: 40%, Z: 60%)*

Ultimately, we elected to train each neural network for 30 epochs and compare their performance based on loss, accuracy, validation loss, validation accuracy, Precision, Recall, F1-Score, and AU-ROC to foster a wide understanding of each neural network architectures performance with each data set. We then compiled the various metrics for each neural network architecture and prepared them for comparison and statistical analysis. It is also necessary to look at the confusion matrix of each neural network architecture in order to fully understand the performance of each model on each data set.

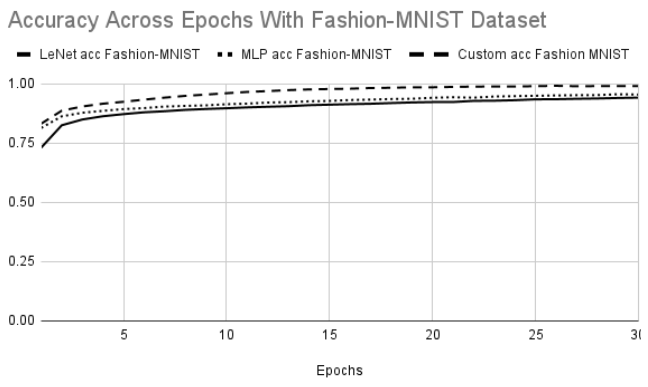
The outcomes and metrics of each neural network architecture will be compiled and grouped together by their corresponding Datasets for comparison.

A. Fashion-MNIST

Training on the Fashion-MNIST data set proved to be slightly more complex than MNIST for our neural network architectures, taking longer to reach an optimal accuracy and loss function than the MNIST data set. This is in line with the complexity of the data set as it is characteristically designed to greatly challenge machine learning models.

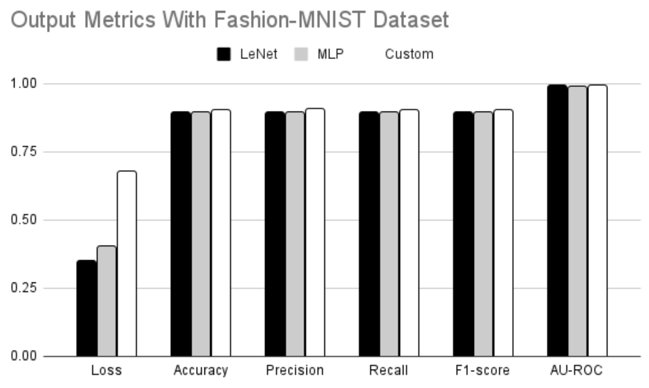


As seen in the above Diagram, our custom architecture achieved the lowest loss values by the 30th epoch, while the LeNet architecture had a steep initial optimization curve that flattened out, producing the highest loss values of the three architectures. The MLP architecture loss curve sits in between the other two architectures with a curve more closely resembling the LeNet architecture, perhaps due to their shared use of extra fully connected layers.



The Accuracy curves for the Fashion-MNIST dataset follow a similar trend to their loss curve counterparts. The Custom architecture achieves the highest accuracy at the 30th epoch, appearing to approach 1.00 asymptotically. Once again, the MLP and LeNeet architectures have a similar shape in their accuracy curves, with the MLP architecture's accuracy slightly above that of the LeNet architecture, in between the other two models

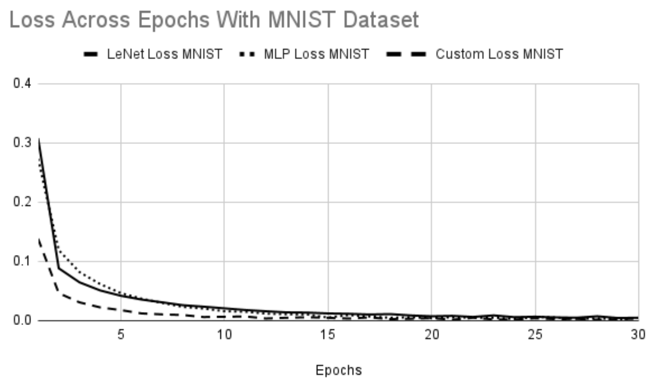
The next parameters that need to be examined are Loss and Accuracy of the trained models, Precision, Recall, F1-Score, and AU-ROC. As these are metrics for the trained models, they cannot be recorded across epochs, so they will be presented differently from the previous training metrics. These output metrics will be grouped together based on Dataset and architecture for comparison and analysis.



Examining the output metrics across the three neural network architectures, the shortcomings of the custom architecture become apparent. The Loss value of the trained custom model is significantly higher than the other two architectures. This is most likely due to the lack of additional fully connected layers to flatten the output data. During training, losses are very high, which is reflected in the final, trained model. The other metrics are not significantly different across the three architectures. Notably, the LeNet neural network has the lowest loss, reflective of our trends seen in training.

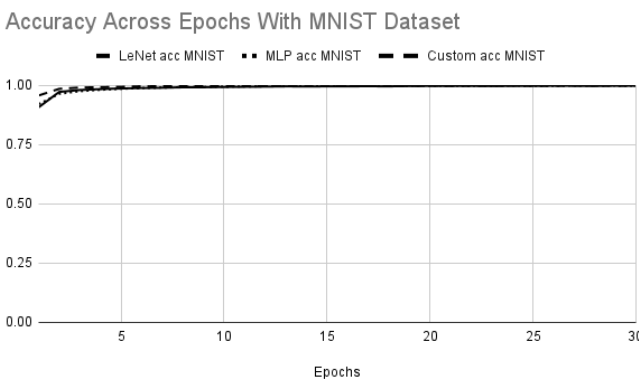
B. MNIST Dataset

Training with the MNIST dataset was much more efficient, likely due to the simplicity of the dataset. This is to be expected considering the nature of the MNIST dataset when compared to the Fashion-MNIST dataset, which was designed to be more complex than the MNIST dataset and better challenge machine learning models.



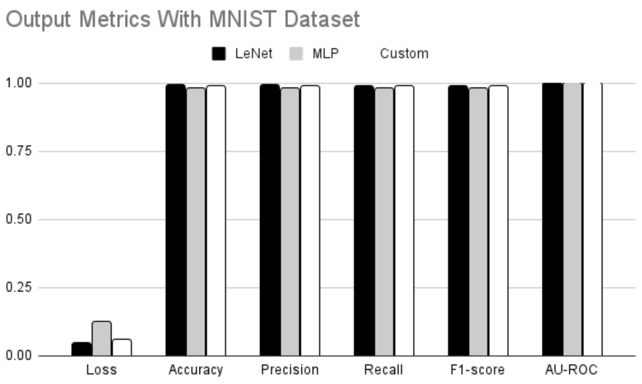
The Loss curves for all three architectures are refined much more quickly when trained using the MNIST dataset. This is to be expected considering the origins of Fashion-MNIST (it was specifically designed to be more complex than MNIST) The Curves are significantly steeper than those trained with the Fashion-MNIST dataset, but when comparing them the results are somewhat changed. We can see that the loss values of LeNet now decrease quickly, leaving the

MLP loss curve above that of the other architectures until approximately the 7th epoch, where LeNet and MLP become practically indistinguishable. In contrast, the curve for our custom architecture does not converge with the other architectures until approximately the 25th epoch.



Based on this graph, the three architectures are nearly indistinguishable. Although the Custom architecture appears to start with a higher initial accuracy, all three architectures flatten very quickly, becoming asymptotic with 1.00.

The next parameters that need to be examined are Loss and Accuracy of the trained models, Precision, Recall, F1-Score, and AU-ROC. As these are metrics for the trained models, they cannot be recorded across epochs, so they will be presented differently from the previous training metrics. These output metrics will be grouped together based on Dataset and architecture for comparison and analysis.



Now that the models are being trained with the MNIST dataset, we see much more stable losses, and extremely refined Accuracy, Precision, Recall, F1-Scores, and AU-ROC values. The lower loss value for the Custom neural network architecture may be due to the MNIST dataset's reduced complexity. This serves as additional insight into the capabilities of our custom architecture. The lack of additional fully connected layers appears to cause significant loss when applied to complex datasets but proves relatively efficient with simple datasets.

V. Conclusion *(K:55%, 45%)*

In conclusion, we conducted an experiment comparing the performance of the LeNet neural network, MLP neural network, and a custom architecture with limited fully connected layers, trained on the MNIST and Fashion-MNIST datasets. Our results show that the LeNet neural network and MLP neural network outperformed the custom architecture, particularly when applied to more complex datasets such as Fashion-MNIST. This suggests that the custom architecture may not be well-suited for more challenging machine-learning tasks.

Our findings also highlight the importance of choosing the appropriate neural network architecture for a specific task or dataset. While the custom architecture may have had some benefits, such as faster training times or improved interpretability, it ultimately failed to achieve competitive performance compared to the LeNet and MLP architectures. Therefore, we recommend that future research focus on developing more sophisticated neural network architectures or adapting existing architectures for specific tasks, rather than relying on simpler custom architectures with limited fully connected layers.

VI. Acknowledgment *(K:50%, Z:50%)*

Kenan Stredic and Zachary Young would like to thank:

- Prof. Feng Chen, Eric Johnson School of Computer Science and Engineering, The University of Texas at Dallas, For In-Depth and informative lectures throughout the past five months. This Report could not have been completed without the knowledge you imparted to us this semester.

VII. References *(K:45%, Z:55%)*

[1] "Convolutional networks and applications in vision - Yann LeCun," yann.lecun.com, http://yann.lecun.com/exdb/publis/pdf/lecun-iscas-10.pdf (accessed May 10, 2023).

[2] B. Karlik and A. V. Olgac, "Performance analysis of various activation functions in generalized MLP ...," research gate, https://www.researchgate.net/publication/228813985_Performance_Analysis_of_Various_Activation_Functions_in_Generalized_MLP_Architectures_of_Neural_Networks (accessed May 10, 2023).

[3] B. Hutchinson et al., "Towards Accountability for Machine Learning Datasets: Proceedings of the 2021 ACM Conference on Fairness, accountability, and transparency," ACM Conferences, https://dl.acm.org/doi/abs/10.1145/3442188.3445918 (accessed May 9, 2023).

[4] A. Saroha, "50+ key terms/ topics in deep learning [complete DL revision]," OpenGenus IQ: Computing Expertise & Legacy, https://iq.opengenus.org/key-terms-in-deep-learning/ (accessed May 10, 2023).

[5] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge (Mass.): The MIT Press, 2018.

[6] "What is transfer learning? exploring the popular deep learning approach.," Built In, https://builtin.com/data-science/transfer-learning (accessed May 10, 2023).

[7] 2023 January 08 et al., "Unsupervised machine learning algorithms and their implementation in tensorflow2.0," Unsupervised Machine Learning Algorithms and their implementation in TensorFlow2.0, https://python-blogs-everything.blogspot.com/2023/01/unsupervised-machine-learning.html (accessed May 10, 2023).

[8] Y. Liu, R. Ravichandran, K. Chen, and P. Patnaik, "Application of machine learning to solid particle erosion of Aps-TBC and EB-PVD TBC at elevated temperatures," MDPI, https://www.mdpi.com/2079-6412/11/7/845/html (accessed May 10, 2023).

[9] T. A. Team, "The gradient descent algorithm," Towards AI, https://towardsai.net/p/tutorials/the-gradient-descent-algorithm (accessed May 10, 2023).

[10] R. Lapid and M. Sipper, "Patch of invisibility: Naturalistic black-box adversarial attacks on object detectors," arXiv.org, https://arxiv.org/abs/2303.04238 (accessed May 10, 2023).

[11] "The mnist database," MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges, http://yann.lecun.com/exdb/mnist/ (accessed May 10, 2023).

[12] Z. Research, "Fashion mnist," Kaggle, https://www.kaggle.com/datasets/zalando-research/fashionmnist (accessed May 10, 2023).

[13] "Multilayer Perceptron," Multilayer Perceptron - an overview | ScienceDirect Topics, https://www.sciencedirect.com/topics/computer-science/multilayer-perceptron (accessed May 10, 2023).