iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** V **Month of publication:** May 2024

DOI: https://doi.org/10.22214/ijraset.2024.62553

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue V May 2024- Available at www.ijraset.com

MNIST Handwritten Digit Classification Using Neural Network

Kukkala Mahender[1], A. Mahesh Chandra[2], O. Mallikarjun[3], M. Mani Kiran Goud[4], Thanish Kumar[5]
[1, 2, 3, 4]*School of Engineering Malla Reddy University*
[5]*Asst professor, Department of AIML, School of Engineering, Malla Reddy University*

Abstract: ***Handwritten digit classification is a fundamental problem in the field of machine learning and computer vision. The MNIST dataset, consisting of 60,000 training images and 10,000 test images of handwritten digits, has been widely used as a benchmark for evaluating classification algorithms. In this study, we propose a neural network-based approach for classifying the MNIST digits. Our model consists of multiple layers, including an input layer, one or more hidden layers, and an output layer. We preprocess the images by normalizing the pixel values and convert them into a suitable format for the neural network. The model is trained using the training images and their corresponding labels, adjusting the weights of the connections to minimize the difference between the predicted outputs and the actual labels. We evaluate the performance of the model using metrics such as accuracy on the test images. Our results demonstrate the effectiveness of the neural network approach in accurately classifying handwritten digits. This research contributes to the advancement of digit recognition techniques and provides insights into the application of neural networks in image classification tasks.***

I. INTRODUCTION

The classification of handwritten digits within the MNIST dataset using neural networks stands as a cornerstone in the realm of pattern recognition and machine learning. MNIST, comprising a collection of 28x28 pixel grayscale images representing digits 0 through 9 alongside their respective labels, serves as a benchmark for evaluating the efficacy of various classification algorithms. Neural networks, with their ability to learn complex relationships within data, have emerged as a prominent approach for tackling this task. The architecture typically encompasses an input layer, one or more hidden layers responsible for feature extraction and non-linear transformations, and an output layer for digit classification. Through iterative training procedures such as backpropagation and gradient descent, the network optimizes its parameters to minimize classification errors. This research delves into the nuanced intricacies of neural network-based MNIST digit classification, exploring novel architectures, optimization techniques, and performance evaluations, thereby contributing to the advancement of both theoretical understanding and practical applications in the field

II. LITERATURE REVIEW

This literature review on MNIST Handwritten Digit Classification using neural networks demonstrates a consensus on the efficacy of this approach while also showcasing a diversity of techniques aimed at improving classification accuracy and efficiency. Early studies, such as LeNet-5, established convolutional neural networks (CNNs) as a powerful tool for digit recognition, laying the groundwork for subsequent research. Since then, advancements in network architectures, optimization algorithms, and evaluation metrics have led to notable improvements in performance. Techniques like residual connections, attention mechanisms, and capsule networks have been explored to enhance model capacity and robustness. Optimization methods like SGD variants, Adam, and RMSprop have been investigated for faster convergence and better generalization. Moreover, ensemble learning strategies and novel evaluation metrics have been proposed to leverage model diversity and assess model performance comprehensively. Recent trends also emphasize the development of lightweight models for deployment in resource-constrained environments, reflecting a shift towards practical applications

III. PROBLEM STATEMENT

The problem statement in MNIST Handwritten Digit Classification Using Neural Network is to develop a machine learning model that can accurately classify images of handwritten digits into their respective numeric representations (0 through 9). Writing numbers by hand is common, but turning that into digital form can be slow and prone to mistakes. Imagine if there was a way to use technology to quickly convert handwritten numbers into digital once. This challenges associated with the project include variations in handwriting styles, diverse writing instruments, and potential destortions in the input image.

©IJRASET: All Rights are Reserved

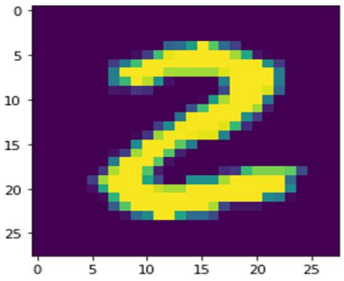
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue V May 2024- Available at www.ijraset.com

That's what the project is all about. We want to create a smart system using deep learning and special algorithm called neural networks(NN) to automatically read and translate handwritten numbers into digital format

IV. METHODOLOGY

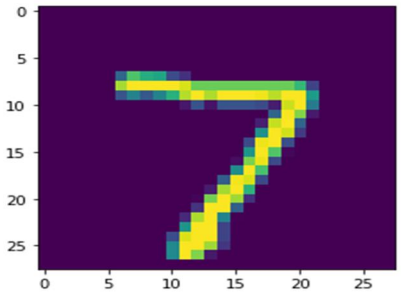
1) *Dataset Preparation:* Download the MNIST dataset. It consists of 28x28 pixel grayscale images of handwritten digits (0-9) along with their corresponding labels. Split the dataset into training, validation, and test sets. A common split is 60% for training, 20% for validation, and 20% for testing.
2) *Data Preprocessing:* Normalize the pixel values to be between 0 and 1 by dividing each pixel value by 255 (since pixel values range from 0 to 255).Flatten the 28x28 images into a 1D array of 784 elements. This will be the input to our neural network.
3) *Model Architecture:* Define the architecture of the neural network. A simple architecture might consist of an input layer of 784 neurons (one for each pixel), one or more hidden layers with ReLU activation functions, and an output layer with 10 neurons (one for each digit) and a softmax activation function. Experiment with different architectures, layer sizes, and activation functions to improve performance.
4) *Training:* Choose a loss function suitable for multi-class classification, such as categorical cross-entropy. Choose an optimizer, such as Adam or SGD (Stochastic Gradient Descent).Train the model using the training data. Monitor the validation loss to prevent overfitting. Experiment with different learning rates and batch sizes to find the optimal training hyperparameters.
5) *Evaluation:* Evaluate the trained model on the test set to measure its performance. Calculate metrics such as accuracy, precision, recall, and F1-score to assess the model's performance. Visualize the performance metrics using plots or confusion matrices.
6) *Hyperparameter Tuning:* Experiment with different hyperparameters like learning rate, batch size, number of layers, number of neurons per layer, etc., to improve performance. Utilize techniques like grid search, random search, or Bayesian optimization for hyperparameter tuning.
7) *Regularization:* Implement regularization techniques like L1/L2 regularization or dropout to prevent overfitting.Tune the regularization hyperparameters based on the validation performance.
8) *Deployment:* Once satisfied with the model performance, deploy the model for inference.You can deploy it as a web service, mobile app, or integrate it into other applications

V. EXPERMENTAL RESULTS

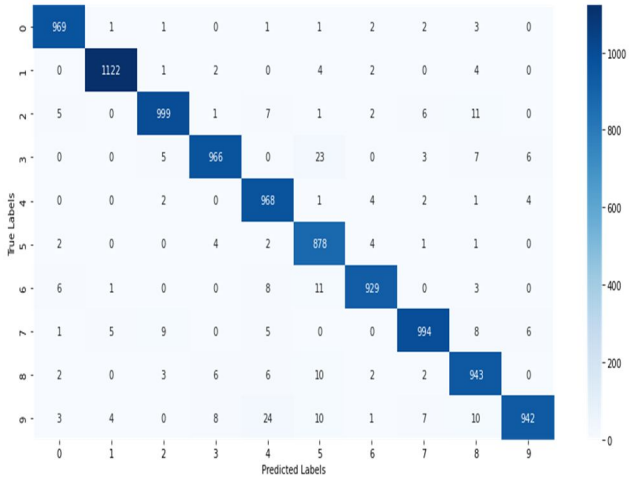


Training data accuracy = 98.9%

Test data accuracy = 97.1%



©IJRASET: All Rights are Reserved



Predicted Labels



The Handwritten digit regognized as 3

VI. CONCLUSION

Here, we implemented two handwritten digit recognition models MNIST datasets based on deep and machine learning algorithms We compared them based on their characteristics so that the most accurate model can be evaluated from them. Support vector machines are one of the basic classifiers, therefore it is faster than most algorithms and in this case, it provides the maximum degree of training accuracy, but due to its amplicity it cannot be clasafied complex and ambiguous images as accurately as achieved using MLP and CNN algorithms We found that CNN provided the most accurate results for handwritten digit recognition This leads us to conclude that CNN is best suited for any type of prediction problem including image data as input Furthermore, by comparing the execution time of the algorithms we concluded that increasing the number of epochs without changing the configuration the algorithm is useless due to the limitation of a certain model and we noticed that after a certain number of epocha the model starts to overfit the data set and gives us a bias forecast

,

VII. FUTURE ENHANCEMENT

A. Advanced Architectures

Explore more advanced neural network architectures such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), or transformer models. CNNs, in particular, are well-suited for image classification tasks like MNIST due to their ability to capture spatial dependencies.

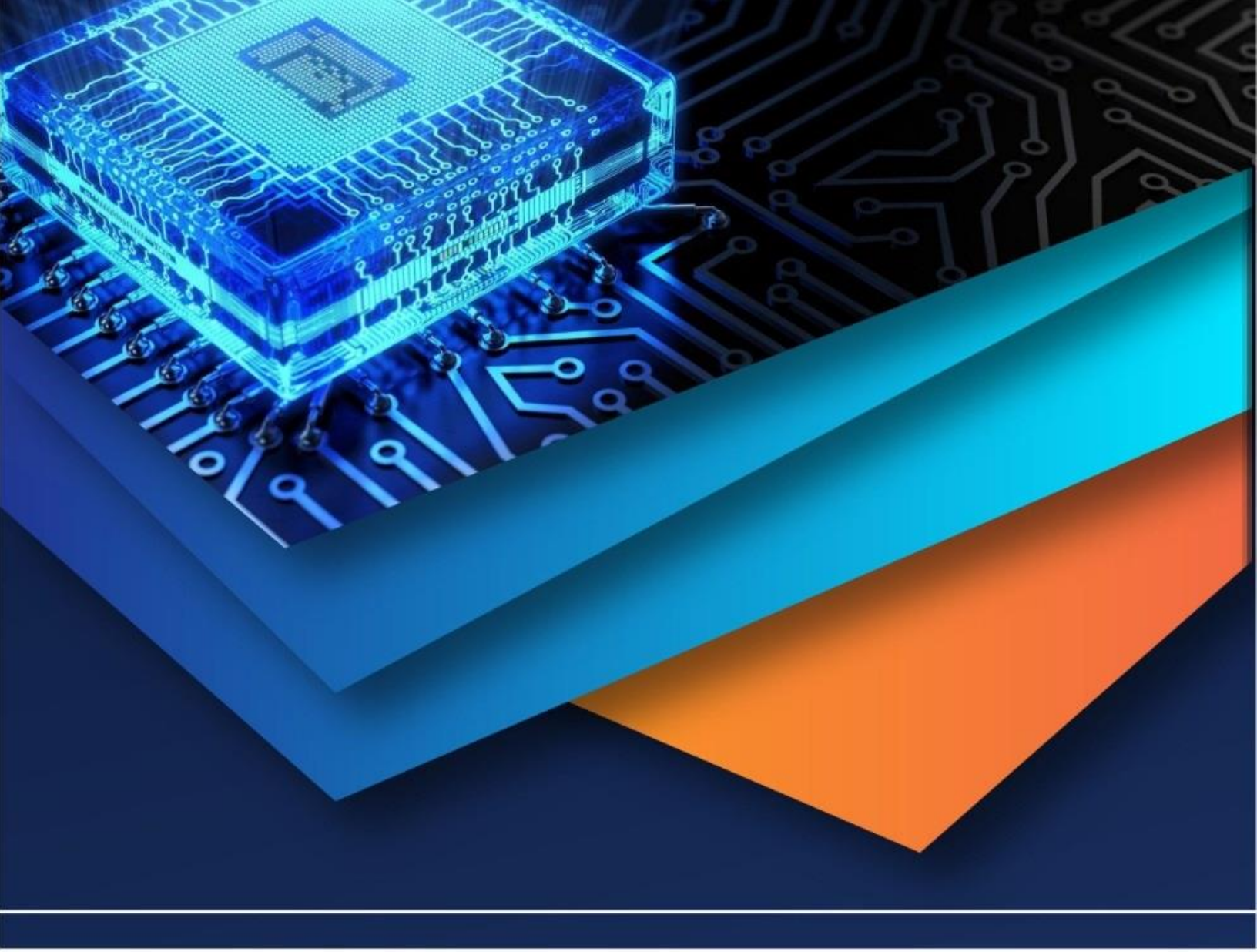
1) *Data Augmentation:* Implement data augmentation techniques to artificially increase the size of the training dataset. This can include random rotations, translations, scaling, and adding noise to the images, which helps improve the model's generalization capability.

2) *Ensemble Learning:* Combine multiple neural networks into an ensemble to leverage the diversity of individual models and improve overall performance. Ensemble methods such as bagging, boosting, or stacking can be applied to enhance classification accuracy.
3) *Transfer Learning:* Utilize transfer learning by fine-tuning pre-trained neural network models on similar tasks or datasets before training on MNIST. Pre-trained models trained on larger datasets like ImageNet can capture generic features that are useful for digit classification tasks.
4) *Hyperparameter Optimization:* Apply more sophisticated hyperparameter optimization techniques such as Bayesian optimization or evolutionary algorithms to efficiently search the hyperparameter space and find the optimal configuration for the neural network.

REFERENCES

[1] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.
[2] LeCun, Y., Cortes, C., & Burges, C. J. (2010). MNIST handwritten digit database. AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist.
[3] Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep Learning (Vol. 1). MIT press Cambridge.
[4] Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems. O'Reilly Media. Chollet, F. (2017). Deep Learning with Python. Manning Publications.
[5] Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2017). Understanding deep learning requires rethinking generalization. arXiv preprint arXiv:1611.03530.
[6] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(1), 1929-1958.
[7] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
[8] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)