Neural Network Models for Object Recognition using MNIST

A Comprehensive Approach for Handwritten Digit Recognition

Roshni Kasturi
November 2024

Introduction

- AI in object detection
- Objective

Significance of the MNIST Dataset
(Modified National Institute of Standards and Technology database)

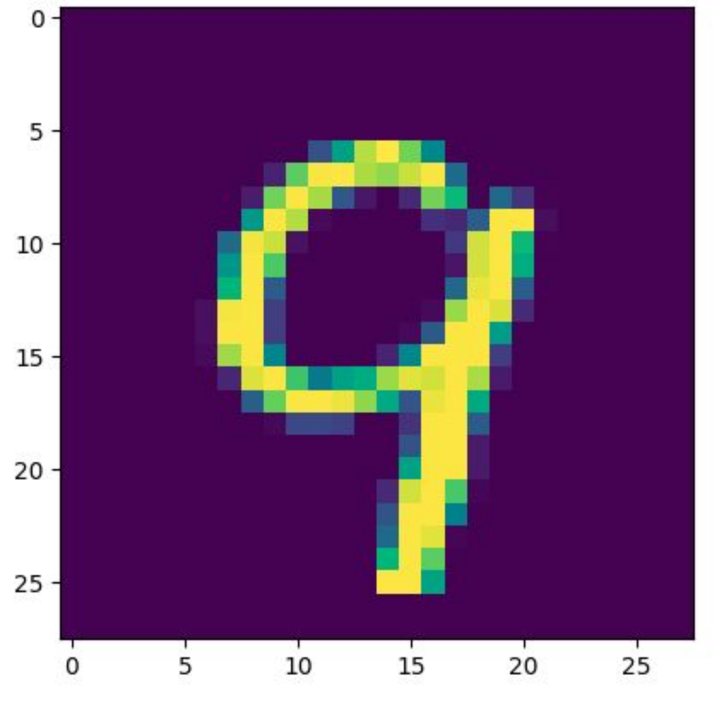
- Commonly used for benchmarking machine learning algorithms in computer vision.
- Regarded as a "Hello World" dataset for introducing image classification tasks (Deng, 2012).

Data Overview and Preparation

- Data Split / Partition
    - Random split of the training set into:
        - training (80%) - 48,000 images
        - validation (20%) datasets - 12,000 images
    - While the test set has 10,000 images
- Image dimensions
    - 28x28 pixels, 1 channel
- 10 object categories (digits 0-9)

Data Preparation
Importance of a Separate Validation Set

- Unbiased performance monitoring
- Hyperparameter tuning
- Detection of overfitting



Neural Network Architecture

- Input Layer
  - Consists of 784 neurons (28 pixels × 28 pixels)
- Hidden Layer (Dense Layer)
  - Features 128 neurons utilizing the ReLU activation function
- Output Layer (Dense Layer)
  - Contains 10 neurons representing the digits 0-9
  - Employes the Softmax activation function to produce a probability distribution

Activation Functions

- ReLU (Rectified Linear Unit)
  - Equation
  - Hidden layers
- Softmax
  - Output layer

$$f(x) = \max(0,x)$$

Image source: (StatQuest with Josh Starmer, 2020)

Loss Function

Function used: Categorical Crossentropy

- Ideal for multi-class classification tasks (Bishop, 2006)
- Assesses the difference between actual labels and predicted probabilities

$$L(y, \hat{y}) = -y \ln(\hat{y}) - (1 - y)\ln(1 - \hat{y})$$

Image Source: (Derivation for Log Loss Function in Classification, 2023)

Training the Model

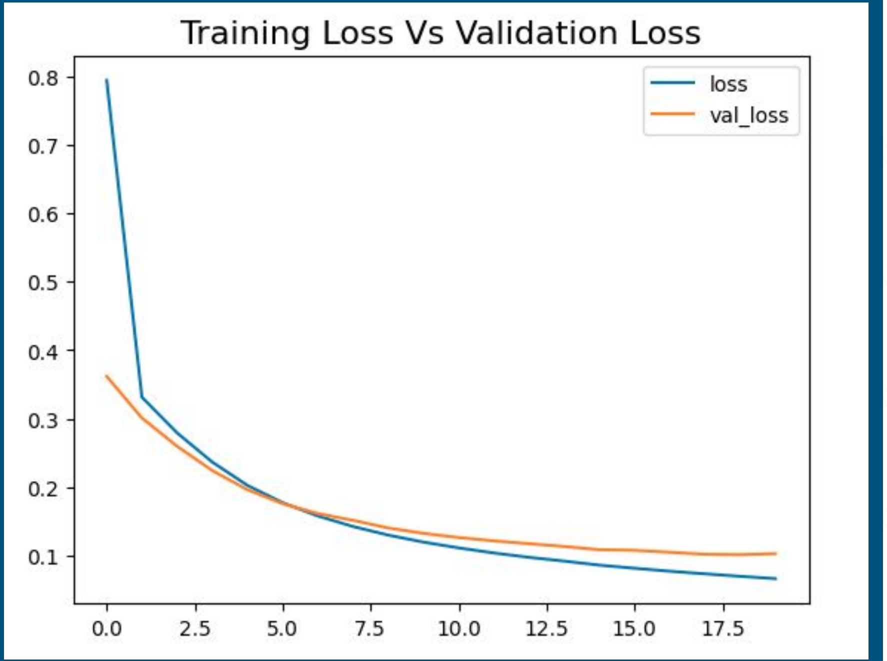
- Epochs: 20
  - Balanced Training Duration
  - Empirical Success
  - Validation Monitoring
- Batch size: 32
- Learning rate: 0.001 (default):
  - Proven Effectiveness of Defaults
  - Simplifies Model Development
  - Baseline Performance
  - Rapid Prototyping

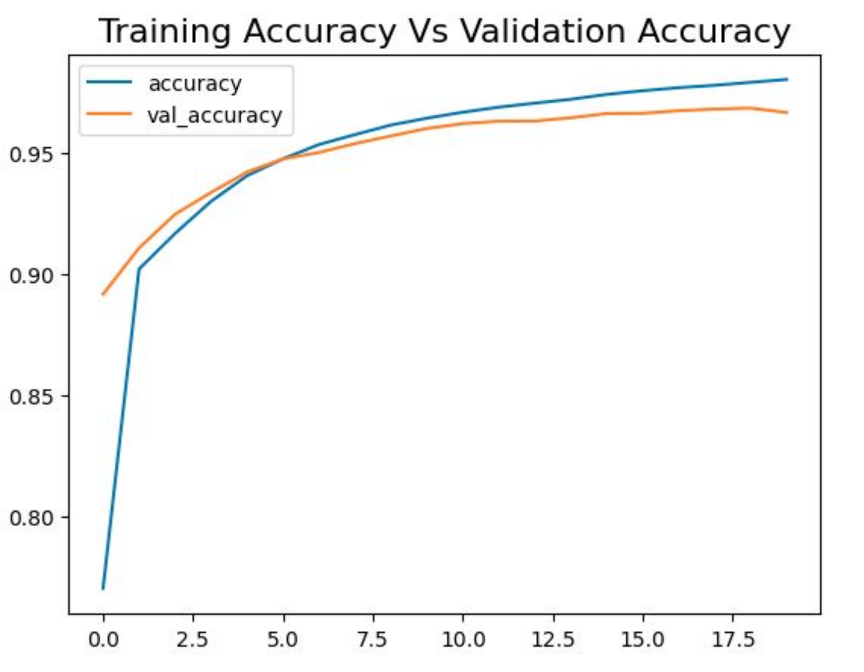
Design Strategy

- Layer combinations for Feature Extraction
  - Conv2D
  - Pooling Layer
  - Flattening Layer
- Regularization to prevent overfitting
  - EarlyStopping
- Optimization Techniques
  - Loss Function (categorical_crossentropy)
  - Optimizer (adam)
  - Metrics (accuracy)

Model Evaluation

Training vs Validation Performance Metrics





Model Evaluation

Classification Report

	precision	recall	f1-score	support
0	0.97	0.99	0.98	980
1	0.98	0.99	0.99	1135
2	0.95	0.97	0.96	1032
3	0.97	0.96	0.96	1010
4	0.97	0.97	0.97	982
5	0.96	0.98	0.97	892
6	0.98	0.97	0.98	958
7	0.99	0.93	0.96	1028
8	0.96	0.95	0.96	974
9	0.94	0.97	0.96	1009
accuracy			0.97	10000
macro avg	0.97	0.97	0.97	10000
weighted avg	0.97	0.97	0.97	10000

Model Evaluation

Confusion Matrix

```
array([[ 970,    1,    0,    0,    2,    1,    2,    0,    2,    2],
       [   0, 1120,    5,    1,    0,    1,    2,    0,    6,    0],
       [   6,    0, 1006,    2,    3,    1,    2,    2,    9,    1],
       [   1,    0,   13,  969,    1,   12,    0,    3,    6,    5],
       [   1,    0,    3,    0,  949,    1,    4,    1,    2,   21],
       [   2,    1,    0,    7,    0,  871,    3,    1,    3,    4],
       [   7,    3,    1,    0,    4,    9,  929,    1,    4,    0],
       [   0,    8,   22,   11,    2,    1,    0,  956,    2,   26],
       [   9,    1,    7,    7,    6,    6,    4,    2,  928,    4],
       [   1,    5,    0,    2,   10,    7,    1,    1,    0,  982]])
```

Summary
Project Overview

- Neural network model created for object recognition
  - Specifically, handwritten numerical digits
- Focus:
  - Architecture
  - Training
  - Evaluation
- Importance of a separate validation set

Summary

Potential Improvements

Experimentation with:

- Different CNN architectures / variations
    - ResNet
    - DenseNet
- With hyperparameter tuning

….for optimized model performance

Summary

Reflections on Learning

- Data preprocessing and preparation is key!
- Model evaluation can also help eliminate errors
- Visualization of performance metrics provided greater understanding
  - Compared to numerical performance metrics alone

Thank you!

References

- MNIST Keras Dataset. Available from: https://keras.io/api/datasets/mnist/ [Accessed 04 November 2024].
- University of Essex Online. (2024) *Unit 9 CNN Object Recognition.* [Jupyter notebook]. ML_PCOM7E JULY 2024 Machine Learning July 2024. University of Essex Online.
- Open AI ChatGPT-4o mini (2024) ChatGPT response to Roshni Kasturi, 03 November.
- StatQuest with Josh Starmer. (2020) Neural Networks Pt. 3: ReLU in Action!!!.

  Available from: www.youtube.com/watch?v=68BZ5f7P94E [Accessed 04

  November 2024]

References

- Bishop, C. M. (2006). Pattern Recognition and Machine Learning. Springer. [Accessed 04 November 2024]
- Deng, L. (2012). The MNIST Database of Handwritten Digit Images for Machine Learning Research. IEEE Transactions on Neural Networks and Learning Systems, 22(10), 1617-1620
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. [Chapter 6 discusses best practices in hyperparameter tuning.]
- Bergstra, J., & Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. Journal of Machine Learning Research, 13, 281-305.
- Loshchilov, I., & Hutter, F. (2016). SGDR: Stochastic Gradient Descent with Warm Restarts. arXiv preprint arXiv:1608.03983.
- Chollet, F. (2017). Deep Learning with Python. Manning Publications.

References

- Ciresan, D. C., Meier, U., & Schmidhuber, J. (2012). A committee of neural networks for handwritten digit classification. IEEE Transactions on Neural Networks and Learning Systems, 23(2), 331-340.
- Patel, A. N., & Lio, P. (2016). Early stopping as a form of regularization for deep neural networks. International Journal of Neural Networks and Applications, 1(2), 45-54.
- “Derivation for Log Loss Function in Classification.” DeepLearning.AI, 7 May 2023, community.deeplearning.ai/t/derivation-for-log-loss-function-in-classification/328003. Accessed 4 Nov. 2024.
- Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML) (pp. 807-814).
- Russell, S., & Norvig, P. (2020). Artificial Intelligence: A Modern Approach (4th ed.). Pearson.