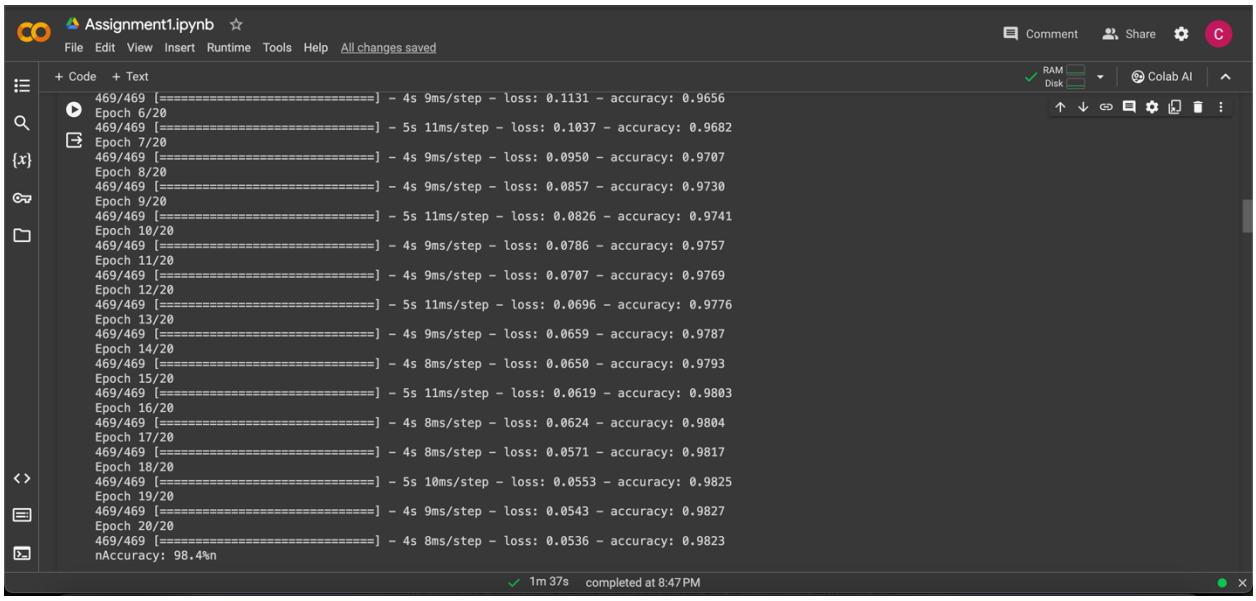
Chandana Sree Krishna
1226847
07 February 2024

COMP5313 Artificial Intelligence
Department of Computer Science
Assignment 1: MLP for Image Classification: Connectionist AI

1. Simple Three-Layer MLP
    a. With Keras (As per tutorial [1])
    The code implements a simple Multi-Layer Perceptron (MLP) using TensorFlow and Keras for classifying handwritten digits from the MNIST dataset. Key steps include loading the MNIST dataset, converting labels to one-hot encoding, reshaping the input images, and normalizing pixel values. The MLP consists of two hidden layers with ReLU activation and dropout for regularization, followed by an output layer with softmax activation. The model is compiled using categorical crossentropy loss and the Adam optimizer. It is then trained on the training set for 20 epochs with a batch size of 128. Finally, the model's accuracy is evaluated on the test set. The code uses Keras functionalities for model creation, training, and evaluation.

    The obtained accuracy is 98.4%

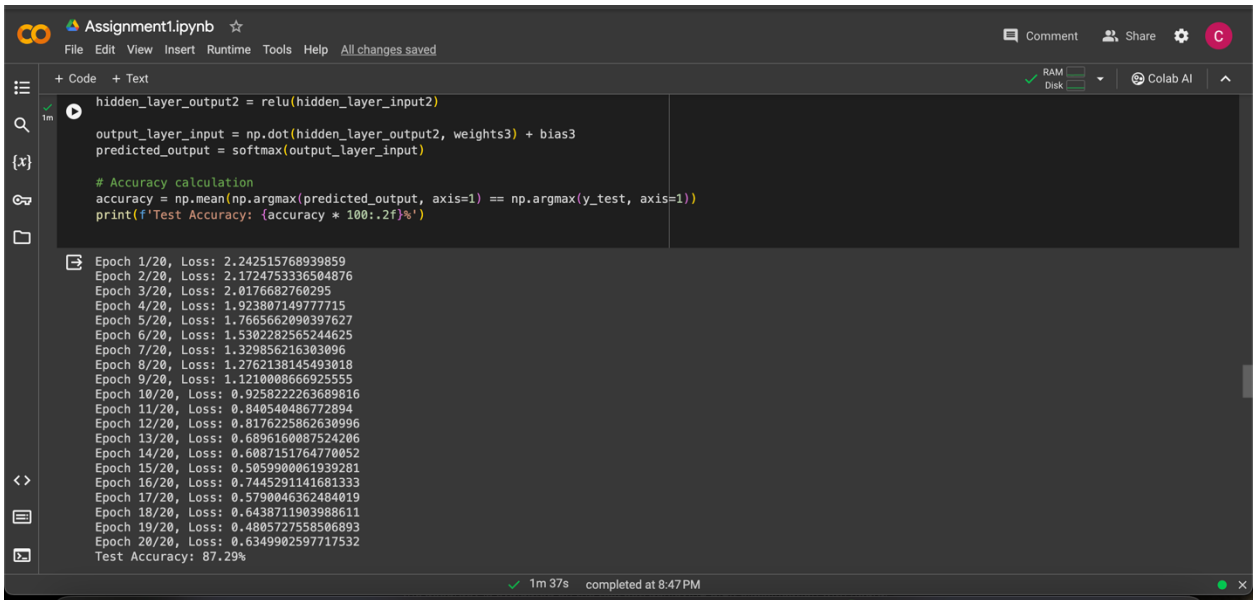


    b. Without Keras
    The 2nd code implements a three-layer Multi-Layer Perceptron (MLP) using NumPy to classify handwritten digits from the MNIST dataset. It preprocesses the

Chandana Sree Krishna
1226847
07 February 2024

data, initializes weights using Xavier/Glorot initialization, and employs mini-batch gradient descent for training. ReLU is used as the activation function for hidden layers, softmax for the output layer, and dropout for regularization. The model is trained for 20 epochs, and the accuracy is evaluated on the test set, achieving digit classification with neural network techniques. The code doesn't rely on deep learning frameworks like TensorFlow or Keras.

The obtained accuracy is 87.29%

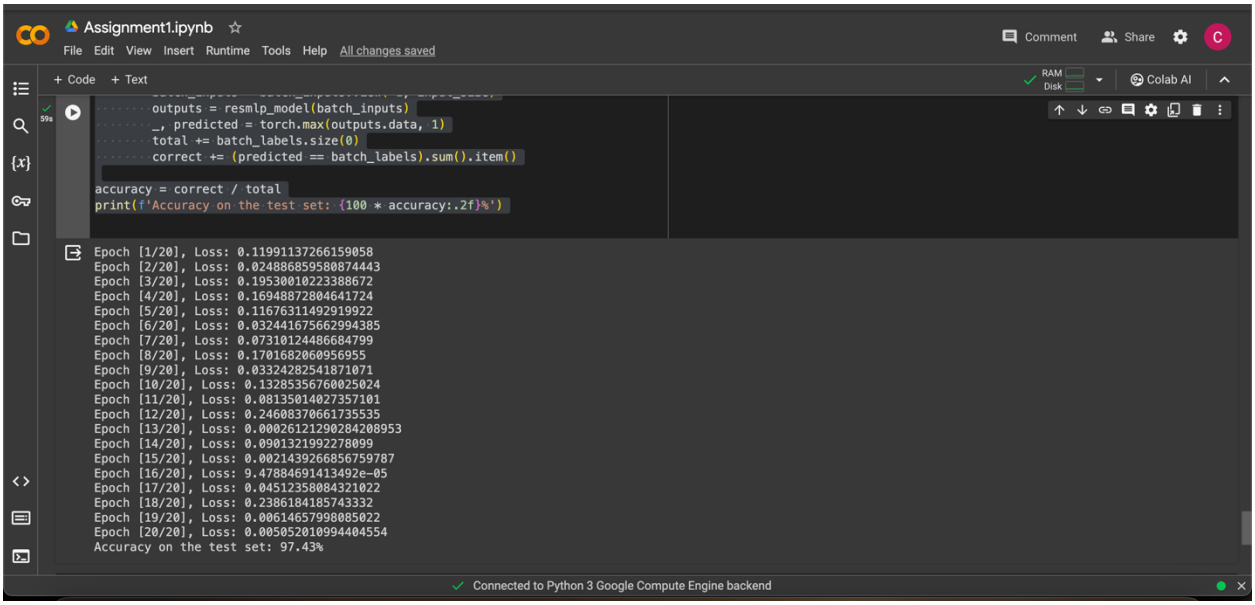


2. ResMLP

The code implements a Residual Multi-Layer Perceptron (ResMLP) model using PyTorch for classifying the MNIST dataset, which consists of handwritten digits (0-9). The ResMLP architecture includes a sequence of linear layers with GELU activations and residual connections through ResMLPBlock instances. The model is trained using the Adam optimizer and cross-entropy loss. The MNIST dataset is loaded using torchvision, transformed to tensors, and normalized. The training loop iterates over batches, computes the forward and backward passes, updates the model parameters, and prints the loss for each epoch. After training, the model is evaluated on the test set, and the accuracy is calculated and printed. The model achieves accuracy results on the MNIST test set.

The obtained accuracy is 97.43%

Chandana Sree Krishna
1226847
07 February 2024



Comparison:

The Simple Three layer MLP model built with Keras performed the best with the highest accuracy of 98.4%.
The MLP model without Keras had a lower accuracy of 87.29%.
The ResMLP model with PyTorch achieved a good accuracy of 97.43%.

In summary, the Keras-based model stood out for having the highest accuracy, but other factors like simplicity and efficiency should also be considered depending on the specific use case.

Simple Three-Layer MLP with Keras:
Pros:

- Utilizes the high-level API Keras, making it concise and easy to understand.
- Achieves a high accuracy of 98.4% on the MNIST dataset.

Cons:

- May not provide as much flexibility for customization as lower-level implementations.

Simple Three-Layer MLP without Keras (NumPy-based):
Pros:

- Provides a low-level implementation using NumPy, offering more control over the model.
- Achieves a decent accuracy of 87.29% on the MNIST dataset.

Cons:

- Requires more manual coding compared to the Keras-based implementation.
- May not be as efficient or scalable for larger datasets.

Chandana Sree Krishna
1226847
07 February 2024

Residual Multi-Layer Perceptron (ResMLP) with PyTorch:

Pros:

- Implements a more advanced ResMLP model with residual connections using PyTorch.
- Achieves a good accuracy of 97.43% on the MNIST dataset.
- Utilizes PyTorch's automatic differentiation for efficient backpropagation.

Cons:

- Involves a slightly more complex implementation compared to the simple MLP models.

Conclusion:

If simplicity and high-level abstraction are prioritized, the Simple Three-Layer MLP with Keras might be preferable, especially with its impressive accuracy.
If a lower-level implementation with more control is preferred over the model, the Simple Three-Layer MLP without Keras (NumPy-based) is an option.
The Residual Multi-Layer Perceptron (ResMLP) with PyTorch offers a balance between abstraction and control, achieving good accuracy with the added benefit of using PyTorch's features.

References:

1. https://www.analyticsvidhya.com/blog/2020/12/mlp-multilayer- perceptron-simple-overview/
2. https://keras.io/examples/vision/mlp_image_classification/
3. https://sh-tsang.medium.com/review-resmlp-feedforward-networks-for-image-classification-with-data-efficient-training-4eeb1eb5efa6
4. Lecture slides of Dr. Sabah Mohammad