ECS 189G-001

Deep Learning

Winter 2024

Course Project: Stage 2 Report Example

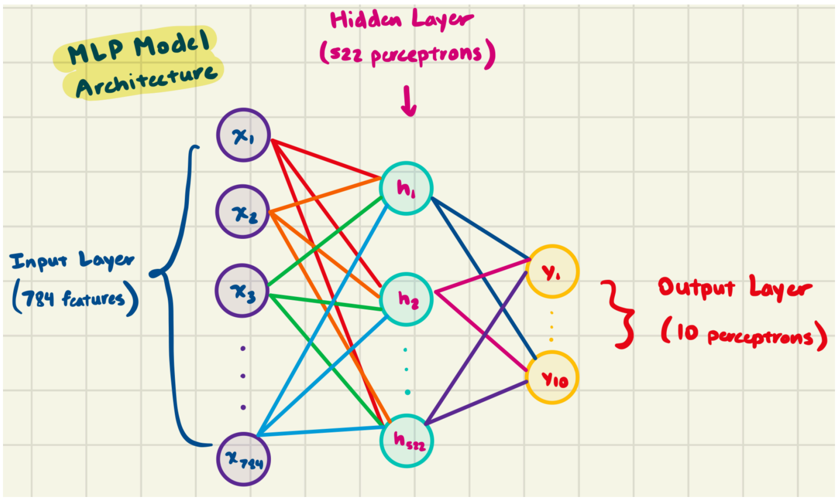
Team Information

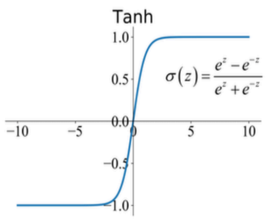
Team Name: TheTransformers		
Aditya Seth	xxxxxxxxxx	aseth@ucdavis.edu
Parth Pawar	xxxxxxxxxx	pupawar@ucdavis.edu
Brian Li	xxxxxxxxxx	xqnli@ucdavis.edu
Ryan Yu	xxxxxxxxxx	ryuyu@ucdavis.edu

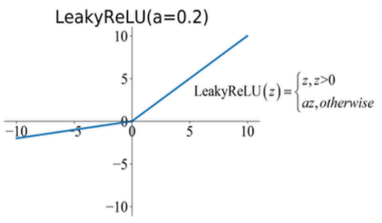
Section 1: Task Description

Build a Multi-layer Perceptron (MLP) model classifying instances in a provided dataset.

Section 2: Model Description


MLP Model Architecture
Hidden Layer (522 perceptrons)
Input Layer (784 features)
x_1
x_2
x_3
x_{784}
h_1
h_2
h_{522}
y_1
y_{10}
Output Layer (10 perceptrons)





Our MLP model consists of an input layer, 1 hidden layer, and an output layer. The input layer contains 784 neurons because there are 784 input features, meaning each neuron corresponds to 1 feature. The input data undergoes a linear transformation involving the bias vector being added to the product of the input vector and the transpose of the weight vector. The result of the linear transformation is fed into the Tanh activation function, which introduces non-linearity to the model to help capture complex data relationships. It determines the activation of neurons in the next layer, which is the hidden layer in this case. The hidden layer consists of 522 neurons, which is ⅔ the size of the input added to the size of the output layer. Each neuron in the hidden layer undergoes the same linear transformation, but the result is instead passed through the leaky ReLU activation function. The output layer consists of 10 neurons, one for each unique output label. The neuron in the output layer with the highest probability is selected as the predicted label for the input data.

Section 3: Experiment Settings

3.1 Dataset Description

The provided datasets are already pre-partitioned into train and test datasets. The train and test datasets consist of 60,000 and 10,000 rows respectively, with each row containing 784 features per label. There are 10 unique labels in each dataset (integers 0 to 9). The values of features range from integers 0 to 255.

3.2 Detailed Experimental Setups

Hyperparameters:
The learning rate is 0.01, which refers to the step size during weight updates. We used full-batch gradient descent, so that the whole input is used as one batch. The max number of epochs is set to 350, which is the number of times the model processes the entire dataset. Cross Entropy Loss is used for our loss function. The Adam optimization algorithm is used to minimize the loss function during the training process. We chose this algorithm as it combines the benefits of AdaGrad and Momentum.

We implemented a convergence condition, checking if the difference between the current and previous loss is less than 0.000001 after each epoch. This ensures faster training and guards against overfitting.

We used the default weight initialization method provided by nn.Linear, which is Kaiming initialization or He initialization. This method of weight initialization assigns weights from the uniform distribution from $-k$ to k, where $k = \frac{1}{number\ of\ input\ features}$ (source).

Please refer to section 2 for details regarding activation functions and layer dimensions.

3.3 Evaluation Metrics

Using the classification_report function from the sklearn.metrics library, we obtained the accuracy, precision, recall, and F1 score for all the epochs during training and testing.

Of these four evaluation metrics, we aimed for models with a high accuracy and F1 score.

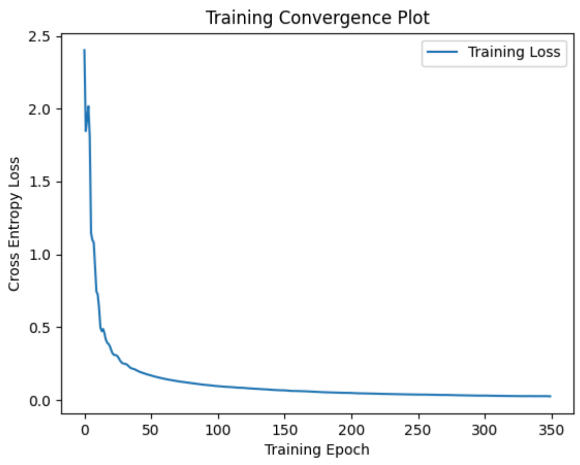
From the classification confusion matrix, accuracy is defined as the ratio of correctly predicted instances to the total number of instances. If a model has high accuracy, it is likely that it will correctly predict data instances. One limitation of using accuracy as an evaluation metric is that if the dataset is imbalanced (i.e: one class has significantly more instances than the others), accuracy may not provide an accurate representation of the model's performance. This is because in this scenario, the model can achieve a high accuracy by only correctly predicting the class with the majority of data instances for every data instance.

To deal with this limitation, we also considered models with a high F1 score. The F1 score incorporates precision, the ratio of correctly predicted positive instances, and recall, the ratio of correctly predicted actual positive instances, to represent the balance between precision and recall. This is done through considering both false positives and false negatives. The F1 score ranges from 0 to 1, where 0 indicates a poor balance between precision and recall, whereas 1 indicates an ideal balance between precision and recall. By considering models with a high F1 score, these models are more robust against imbalance datasets.

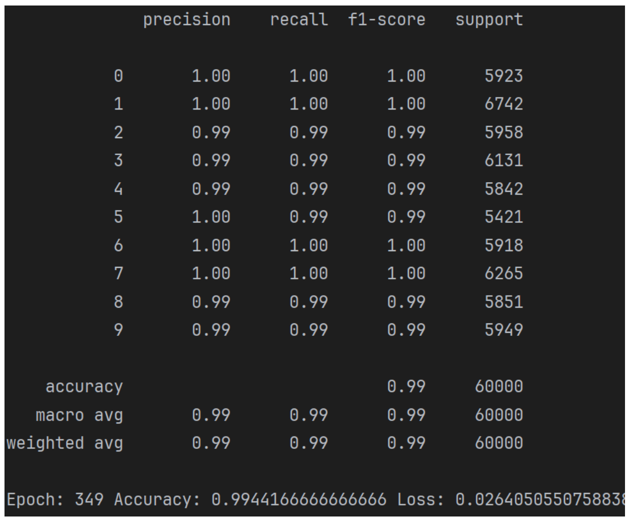
3.4 Source Code

All code can be found on github. Note: The repository does not contain the test and train.csv because their size exceeds 100 MB.

3.5 Training Convergence Plot

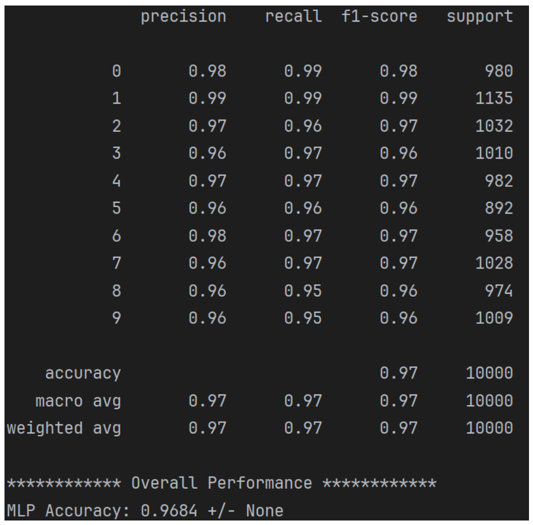


3.6 Model Performance



```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      5923
           1       1.00      1.00      1.00      6742
           2       0.99      0.99      0.99      5958
           3       0.99      0.99      0.99      6131
           4       0.99      0.99      0.99      5842
           5       1.00      0.99      0.99      5421
           6       1.00      1.00      1.00      5918
           7       1.00      1.00      1.00      6265
           8       0.99      0.99      0.99      5851
           9       0.99      0.99      0.99      5949

    accuracy                           0.99     60000
   macro avg       0.99      0.99      0.99     60000
weighted avg       0.99      0.99      0.99     60000

Epoch: 349 Accuracy: 0.9944166666666666 Loss: 0.0264050550758838
```

350th epoch on the train.csv dataset



```
              precision    recall  f1-score   support

           0       0.98      0.99      0.98       980
           1       0.99      0.99      0.99      1135
           2       0.97      0.96      0.97      1032
           3       0.96      0.97      0.96      1010
           4       0.97      0.97      0.97       982
           5       0.96      0.96      0.96       892
           6       0.98      0.97      0.97       958
           7       0.96      0.97      0.97      1028
           8       0.96      0.95      0.96       974
           9       0.96      0.95      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

************ Overall Performance ************
MLP Accuracy: 0.9684 +/- None
```

Test.csv dataset

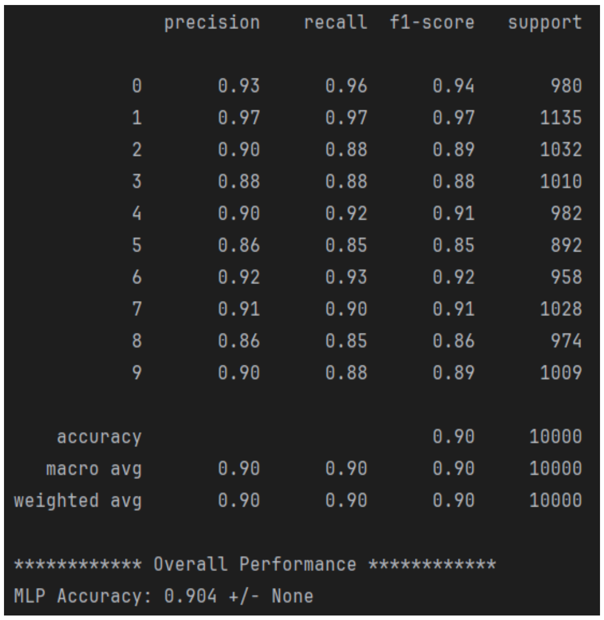
We obtained these scores from the classification report from the sklearn library. Support refers to the true number of occurrences of each label in the dataset. Our model achieves high precision and high recall, thus achieving a high f1 score. During training, our model achieved an extremely low loss rate of 0.026, indicating the model makes very few mistakes in classification.

3.7 Ablation Studies

Note: Each trial shows the results on the test.csv dataset only.

Trial A:

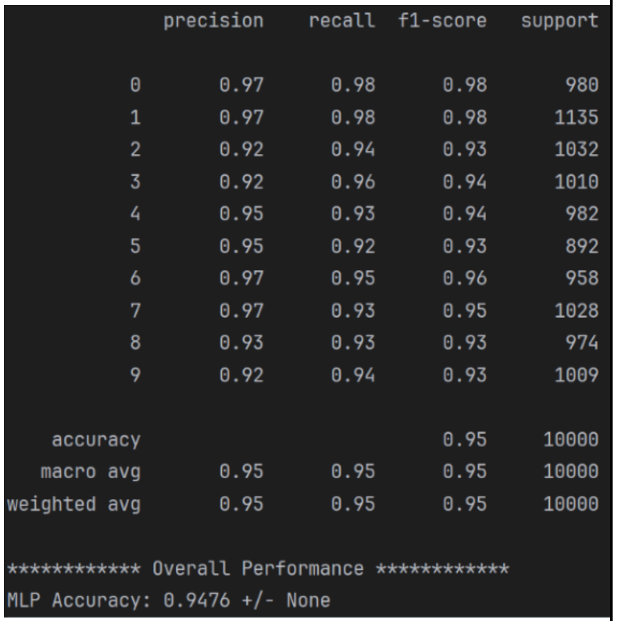
Tanh in the first linear layer, followed by ReLU in the second linear layer. Learning rate of 1e^{-3}. 50 neurons in the hidden layer. 75 epochs.



```
              precision    recall  f1-score   support

           0       0.93      0.96      0.94       980
           1       0.97      0.97      0.97      1135
           2       0.90      0.88      0.89      1032
           3       0.88      0.88      0.88      1010
           4       0.90      0.92      0.91       982
           5       0.86      0.85      0.85       892
           6       0.92      0.93      0.92       958
           7       0.91      0.90      0.91      1028
           8       0.86      0.85      0.86       974
           9       0.90      0.88      0.89      1009

    accuracy                           0.90     10000
   macro avg       0.90      0.90      0.90     10000
weighted avg       0.90      0.90      0.90     10000

************ Overall Performance ************
MLP Accuracy: 0.904 +/- None
```

Trial B:

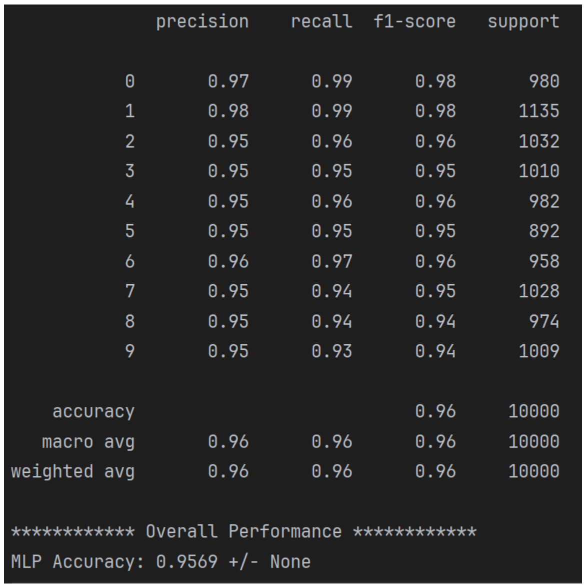
Tanh in the first linear layer, followed by ReLU in the second linear layer. Learning rate of 1e^{-3}. 50 neurons in the hidden layer. 500 epochs.



```
              precision    recall  f1-score   support

           0       0.97      0.98      0.98       980
           1       0.97      0.98      0.98      1135
           2       0.92      0.94      0.93      1032
           3       0.92      0.96      0.94      1010
           4       0.95      0.93      0.94       982
           5       0.95      0.92      0.93       892
           6       0.97      0.95      0.96       958
           7       0.97      0.93      0.95      1028
           8       0.93      0.93      0.93       974
           9       0.92      0.94      0.93      1009

    accuracy                           0.95     10000
   macro avg       0.95      0.95      0.95     10000
weighted avg       0.95      0.95      0.95     10000

************ Overall Performance ************
MLP Accuracy: 0.9476 +/- None
```

Trial C:

337 neurons in the hidden layer. 500 epochs, but training stopped at 120th epoch due to the convergence condition being met.



```
              precision    recall  f1-score   support

           0       0.97      0.99      0.98       980
           1       0.98      0.99      0.98      1135
           2       0.95      0.96      0.96      1032
           3       0.95      0.95      0.95      1010
           4       0.95      0.96      0.96       982
           5       0.95      0.95      0.95       892
           6       0.96      0.97      0.96       958
           7       0.95      0.94      0.95      1028
           8       0.95      0.94      0.94       974
           9       0.95      0.93      0.94      1009

    accuracy                           0.96     10000
   macro avg       0.96      0.96      0.96     10000
weighted avg       0.96      0.96      0.96     10000

************ Overall Performance ************
MLP Accuracy: 0.9569 +/- None
```