Juliette Raubolt

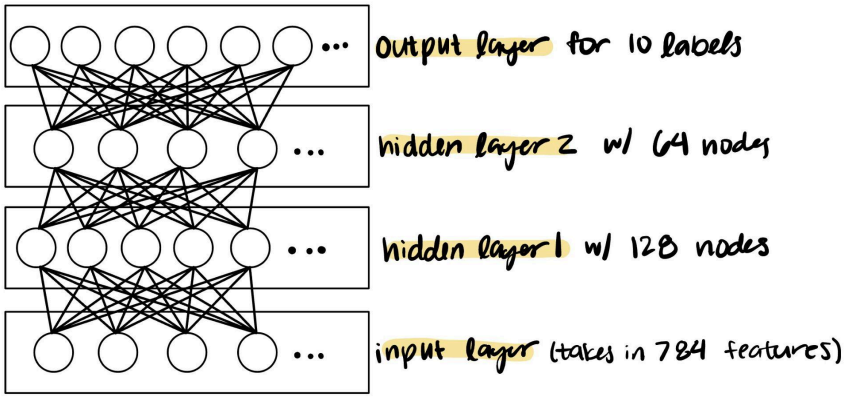
ICS 661
Advanced AI

Fall 2024

Section 1: Task Description

Given a set of training and testing data develop a Multilayer Perceptron (MLP) model. Within the developed MLP model use evaluation metrics such as accuracy, precision, recall and F1 score. Develop a report that details and analyzes the model.

Section 2: Model Description



The model used includes four layers in total, one input layer that takes in the 784 features, two hidden layers (128 and then 64 nodes), and an output layer for the 10 labels (0-9). Further details regarding the implementation discussed in section 3.2.

Section 3: Experiment Settings

3.1 Dataset Description

The datasets include a training set and a testing set. The training set contains 60,000 instances and the testing set contains 10,000 instances. Every instance has 785 elements. The first element represents the label (a value 0 to 9) and the rest of the 784 elements are features, all of which are integers.

3.2 Detailed Experimental Setups

My Multilayer Perceptron model consists of an input layer that takes in the 784 features, two hidden layers (with 128 and 64 nodes), and the output layer that corresponds with the 10 labels (0-9). For the hidden layers I used the relu activation function, for the output layer I used softmax. I added an early

stopping callback where if the model does not improve after three epochs it stops, generally I have a maximum of 100 epochs set. I used adam for the optimizer.

While there was some strategy in determining these hyper-parameters, such as 10 neurons for the output layer, for others there was not much reasoning since the dataset used is so simple. For instance, there is no reason why I chose relu and adam over other options.

For the loss functions and how I handled the labels I used two different methods. This is not so much for performance considering the simplicity of the dataset but more for personal experimentation and understanding. The main method I used and experimented with was using integer labels. Alternatively I also handled the output labels with one-hot encoding.

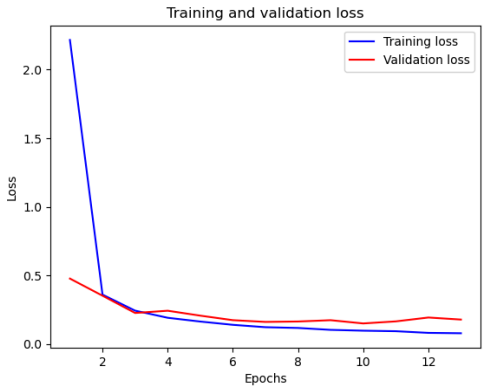
3.3 Evaluation Metrics

In the experiment, we will use Accuracy (how often the model was correct), Precision (how often positive is predicted correctly), Recall (true positives, how well the model can predict all the positives), F1 score (evaluation metric that combines precision and recall) as our evaluation metrics.

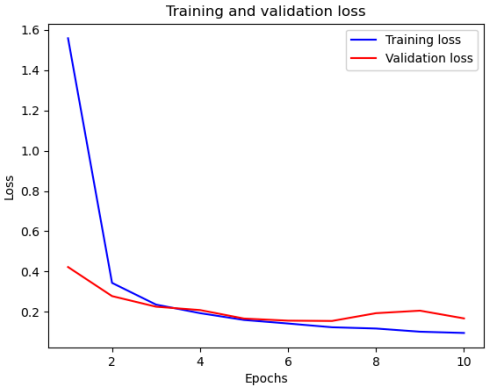
3.4 Source Code

https://github.com/julietteraubolt/MLP-Model

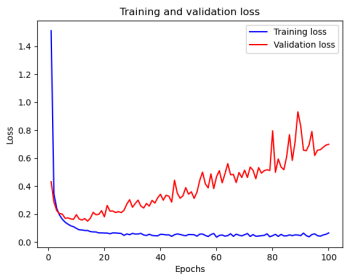
3.5 Training Convergence Plot

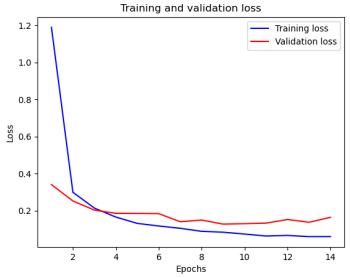


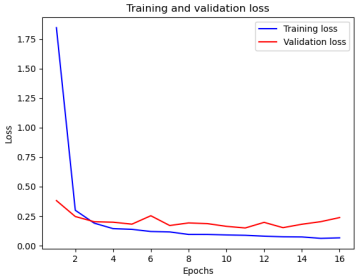
integer_label



one-hot encoding







integer_label_alt1 integer_label_alt2 integer_label_alt3

3.6 Model Performance

Integer_label:

- Test loss: 0.1505608707666397
- Test accuracy: 0.9648000001907349
- Test Precision: 0.9643759547281745
- Test Recall: 0.9646998870061957
- F1 Score: 0.9645378936696746

Integer_label_alt1:

- Test loss: 0.6771061420440674
- Test accuracy: 0.9656000137329102
- Test Precision: 0.9666572007667422
- Test Recall: 0.9652080769159666
- F1 Score: 0.9659320953355626

Integer_label_alt2:

- Test loss: 0.12037331610918045
- Test accuracy: 0.9704999923706055
- Test Precision: 0.9700035727914769
- Test Recall: 0.9703538872426467
- F1 Score: 0.9701786983939673

Integer_label_alt3:

- Test loss: 0.15714268386363983
- Test accuracy: 0.9667999744415283
- Test Precision: 0.9667799349426346
- Test Recall: 0.9663066561271787
- F1 Score: 0.9665432375983304

Hot_encoding:

- Test loss: 0.15975213050842285
- Test accuracy: 0.9635999798774719
- Test Precision: 0.9740816354751587
- Test Recall: 0.9545999765396118
- F1 Score: 0.9642424136567462

3.7 Ablation Studies

Integer_label:

- baseline starting point, four layers, utilizing integer labeling, early stopping callbacks.

Integer_label_alt1:

- No early stopping.

Integer_label_alt2:

- Three hidden layers (128, 64, 32 nodes), early stopping with patience=5.

Integer_label_alt3:

- Two hidden layers (256 & 128 nodes), early stopping with patience=5.

Hot_encoding:

- One-hot encoding for labels

Considerably the simplicity of the dataset used meant that the varying hyperparameters did not make much of an impact. Early stopping made the biggest difference as it prevented overfitting. We saw this represented in the test loss value in integer_label_alt1 compared to all other models. Then, while it was a small margin the best performing model was the one with the most layers, integer_label_alt2. This model had the lowest loss and the greatest accuracy, precision, recall, and F1 score.