GMDL Final Project

Yair Gross, Neta Elmaliach, Sharon Hendy

31/07/2023

1 Introduction

The project focuses on handling the challenging problem of Open Set Recognition (OSR) in the context of image classification using the well-known MNIST dataset.
Unlike conventional image classification tasks, OSR presents a unique challenge where the model must not only accurately classify examples from known classes but also effectively flag and label instances from previously unseen classes during test time.
During testing, the model will use this knowledge to confidently recognize MNIST digits while properly identifying samples from unrelated datasets as 'Unknown' instances.

2 Data and Preprocessing

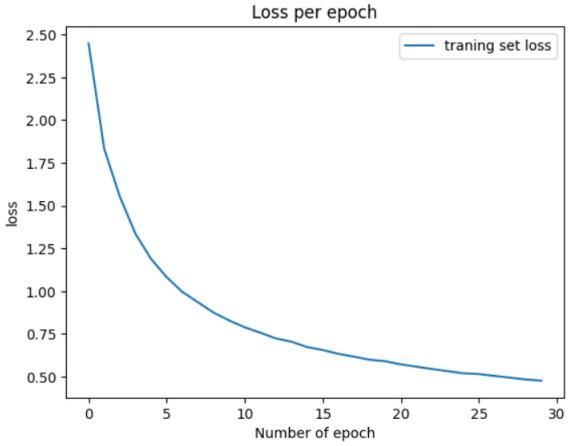
We start by defining two sets of transformations to preprocess the MNIST and OOD (Out-of-Distribution) datasets.
For the MNIST dataset, we apply a basic transformation that converts the images to tensors and normalizes the pixel values. However, for data augmentation, we define a more elaborate set of transformations for the MNIST training set, including random Gaussian blurring and random affine transformations with rotations, translations, and scaling.
Additionally, we introduce a CIFAR-10 dataset as the OOD dataset.
For each OOD example, we assign a fixed label of 10, representing the "Unknown" class.
Finally, we construct data loaders for the various datasets, such as the training set, validation set, augmented training set, and test set, each with appropriate batch sizes and shuffle settings.

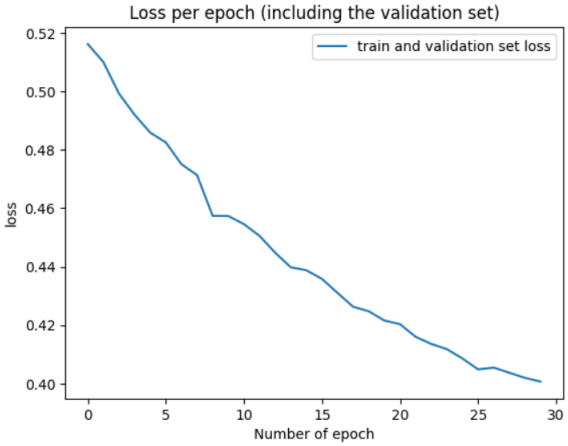
3 Models

In the models phase, we define three classes for our image classification.
The base class, autoencoder, implements an autoencoder architecture that learns to reconstruct input images using an encoder-decoder network.
The second class, ML-autoencoder, extends the autoencoder by incorporating a classification head to perform multi-label classification. This class enables our model to predict class labels for MNIST digits, allowing it to handle traditional image classification tasks effectively.
In the third extension, the ML-autoencoder-OSR class, we provide the model with the ability to handle the OSR problem. It introduces new parameters to control uncertainty thresholds during classification, which is important for identifying and flagging "Unknown" instances during test-time.
The model uses these thresholds to update its predictions, making it adept at distinguishing between known classes and previously unseen data.

4 Training

The training phase of the project involves training a multi-task autoencoder model to perform image reconstruction and multi-label classification on the MNIST dataset.
During training, the model's performance is evaluated on a validation set to compute the mean entropy and mean reconstruction loss, crucial for later Open Set Recognition (OSR) adaptation.
After training the base model, we create an extended version, ML-autoencoder-OSR, specifically designed for Open Set Recognition.
The OSR model incorporates the trained base model and prediction parameters to handle uncertainty and identify "Unknown" instances.





5 Evaluation

OSR rational

Our approach consists of the following key components:

a) Multi-Task Autoencoder: We decided to use a multi-task autoencoder architecture and utilize it for both reconstruction and classification tasks. We detect unknown classes by analyzing the reconstruction loss and the level of uncertainty in the classifications. By combining these techniques, our model is effectively recognizing known classes while detecting and handling unknown samples.

b) Uncertainty Estimation: To detect unknown samples, we calculate the entropy of the model's predictions. Higher entropy implies higher uncertainty, indicating potential unknown samples.

c) Reconstruction Loss: The reconstruction loss is minimized during training. A higher reconstruction loss can signify that the input data is not well-represented by known classes, indicating potential unknown samples.

d) Dropout Layers: Dropout layers are incorporated in the model to improve robustness and prevent overfitting. By introducing randomness during training, the model becomes less confident in its predictions, which aids in open-set recognition.

e) Data Augmentations: Augmentations are applied to the training data to increase diversity and robustness. This exposes the model to various instances of known classes, enhancing its ability to generalize to unseen samples.

Previous Attempts and Lessons Learned:

We explored various strategies for measuring uncertainty in predictions. Initially, we experimented with employing a threshold solely on the maximum probability within the distribution. However, this method did not yield the optimal results we desired, as it failed to consider the variations in probabilities across the distribution.

Subsequently, we decided to use entropy as a measure of uncertainty in the distributions. During the training phase, we compute the mean entropy of predictions on the validation set. This calculated mean entropy then serves as a threshold, in addition to another positive threshold value. When making predictions on new data, we compare the entropy of these predictions to the computed threshold. If the entropy exceeds the threshold, it indicates higher uncertainty in the model's prediction for that sample.

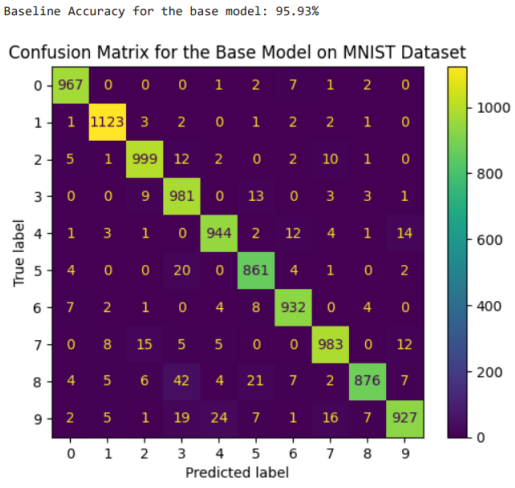
This shift in methodology significantly improved our ability to accurately assess uncertainty in the model's predictions.

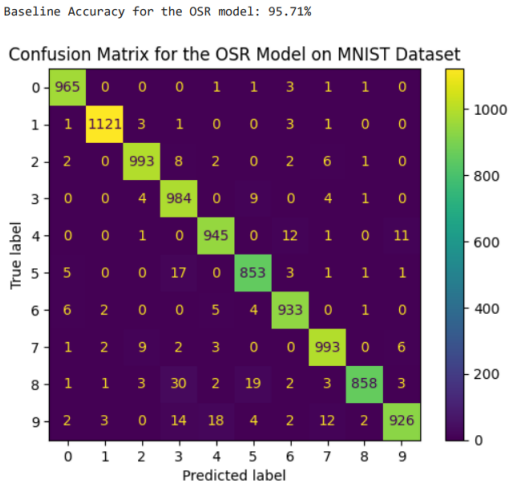
In a similar manner, we also applied the same technique with the loss on reconstruction. During training, we computed the mean loss on reconstruction of the validation set and used it as a threshold for the reconstruction losses of new samples.

In conclusion, after considering both entropy and reconstruction loss thresholds, we made the decision to classify predictions as "unknown" when they exceeded one of the thresholds. By doing so, we effectively identified instances where the model exhibited high uncertainty in its predictions and encountered potential unknown samples.

Baseline results

In the baseline results phase, we evaluate the performance of our trained model on the test set of the MNIST dataset.
The test function takes the trained model, the test dataset, and the test data loader as inputs.
During evaluation, the function calculates the accuracy of the model's predictions by comparing them with the ground truth labels.
Additionally, it records the predicted labels and the true labels for further analysis.
This baseline evaluation allows us to understand how well our model is performing on the standard classification task for the MNIST dataset.

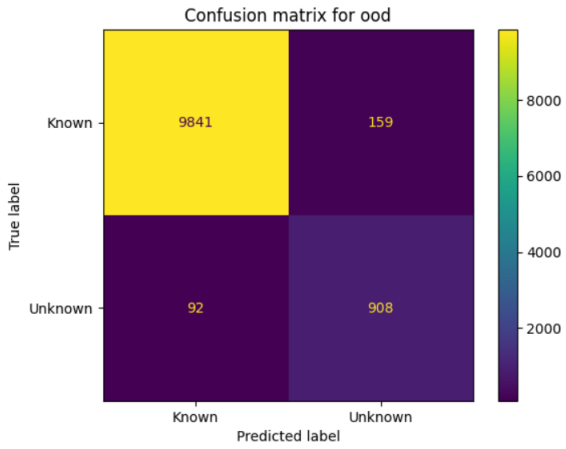




OOD results

In the evaluation of Out-of-Distribution (OOD) results, we assess the performance of our model on the combined dataset, which includes both MNIST examples and OOD samples.
We calculate the binary classification accuracy for both MNIST and OOD instances separately. The MNIST accuracy represents the model's ability to correctly classify examples from known classes, while the OOD accuracy indicates its capability to identify and label "Unknown" instances from unrelated datasets.
Additionally, the confusion matrix is visualized, allowing us to further analyze the model's performance. The matrix differentiates between "Known" and "Unknown" predictions, providing insights into any misclassifications and highlighting the model's ability to correctly identify "Unknown" instances.

```
MNIST Accuracy: 95.71%
OOD Accuracy: 90.80%
```



OSR results

In the OSR evaluation, the model is tested on data that is outside the scope of its training data. This includes samples from classes or categories not present in the training set, as well as data that is substantially different from the familiar distribution. The goal is to determine how well the model generalizes and makes predictions in such unfamiliar situations.
A common metric used for OSR evaluation is the Total Accuracy, which measures the model's accuracy on both the in-distribution (known) data and the out-of-distribution (unknown) data. A higher Total Accuracy indicates better performance in distinguishing between known and unknown samples.

```
Total Accuracy: 95.26%
```