Machine Learning in Natural Sciences
Spring 2025
Deep Learning Assignment
Part 2 Report

Amey Choudhary
2021113017

1. Task

Given an image with digits in it, the final output should be the sum of the digits.

2. Approaches

a. Baseline

For the baseline approach, we used a **CNN model**. Convolution layers use filters to detect features and extract them. It had 2 convolution layers followed by a Linear layer, whose output would be the sum. We treated this as a regression task, where we tried to predict the sum of the digits. We used Mean Square Error as our loss function and Adam as our optimiser. This had reported an accuracy of **16%** (on the entire dataset, threshold of 0.5).

b. OCR

We used a pre-trained **OCR model** (*pytesseract*). This would recognise the digits in the image and output an array consisting of the digits. We would sum this array to output the result. We treated this as classification and didn't finetune. We found that as digits were handwritten, OCR couldn't properly recognise them, .We obtained an accuracy of **7.8%** on 10,000 samples (one-third dataset).

c. YOLO

YOLO is an object detection model. We tried using YOLO to detect the digits and then sum them. We found that YOLO has predefined categories, which it detects and classifies objects into. As digits were not in these categories, YOLO was unable to classify the digits and thus, couldn't be used.

d. Vision Transformer

We tried fine tuning a **pre-trained vision transformer** on our dataset. Due to the large number of parameters of this, we could only train it for 4 epochs (which took about 12 hours). We obtained an accuracy of

6.4% on the entire dataset (threshold of 0.5)**.** Even though vision transformers have more sophisticated architecture than normal CNNs, they performed worse, possibly due to being trained on lesser epochs.

e. CNNs + Attention

Our final approach was **adding attention to CNNs and adding more convolution layers to them**. We used 5 convolution layers, followed by an attention layer, followed by 2 linear layers. Adding attention to the model, allowed the model to highlight the features we extracted after convolution. By doing so, the linear layers were easily able to relate them to the present digits and output their sum. We performed this as a regression task and reported an accuracy of **96.77% on the train set and 75.25% on the validation set**. **This is our final model**.

Modifications:

We also performed modifications of the above model, where we had **no attention but more layers** , **inserted multi-layer attention** and **attention between convolution layers (layer 3 and 4)**. We report following

Model Name	**Training Accuracy (in%)**	**Validation Accuracy (in %)**
no attention but more layers	94.74%	70.02%
inserted multi-layer attention	86.75%	65.48%
attention between convolution layers	93.09%	70.95%

3. Directory Structure and Instructions to Run

The directory consists of this report, as well as *Training.ipynb*, Jupyter Notebook used for training the CNNs+Attention approach (not the modifications). This saves the model with the best validation dataset score, which is present in the repository. There is also *Inference.ipynb* which, using the saved model weights, calculates accuracy, loss and inferences on random images.

To run, just modify the paths for the dataset. Modify the path to where the best validation score model will be saved too.