CNN vs. ViT architecture on MNIST

Arlen Dumas and Tuyetlinh Nguyen

University of Rhode Island

May 10, 2023

Overview

Table of Contents

The Problem Statement

A subset of handwriting recognition (HRW), mathematical expression recognition (MER) focuses on the automatic detection and transcription of handwritten mathematical expressions into their corresponding digital form.

The use of recurrent neural networks and feed-forward neural networks began in 2009 exploded the scope of research.

MER began receiving significantly more attention after the inception of **CROHME (Competition on Recognition of Online Handwritten Mathematical Expression)** in 2011.

CROHME Dataset

We planned to use the CROHME 2023 dataset available through the TC11 datasets repository.

Ink Markup Language (InkML)

InkML is a data format for representing "ink" entered with an electronic pen or stylus. The format essentially stores the data as a list of (x, y) coordinates representing sampled points on the digital writing over time.

The dataset can be organized into three subsets: online, offline, and bimodal.

- online: 13,325 inkml files
- offline: 23,325 offline images consisting of rendered inkml and scanned images
- bimodal: 13,325 files, consisting of all aforementioned formats

Table of Contents

The Proposed Project

Our proposal outlined a plan to create a model with end-to-end capabilities that can both recognize mathematical expressions and map them to corresponding LaTeX output.

Given that generating LaTeX formulas will require a somewhat intensive mapping process and we were inclined to use a CNN.

Unaware at the time of proposal, this task requires a much more complex architecture than a standalone CNN.

Planned Architecture

Through an extensive literature search, an end-to-end model would most likely incorporate:

- a CNN model to extract image features
- an encoder-decoder to work with the features and learn corresponding Latex sequences
- a generator to predict these sequences at evaluation time

Complications

We ran into a few complications that forces our project to change direction.

- The CROHMELib and IgEval tools required for the data to be usable did not work and lacked troubleshooting documentation
- Due to the lack of interest in MER for the last five years, there are no other tools that allow the InkML images to be used as intended

Since the dataset we had chosen was such a pivotal part of our project, we needed to step back and reevaluate what could be done in the remaining time.

Table of Contents

Pivoting to im2latex

After struggling with CROHME, we referred our literature review and found "What You Get Is What You See: A Visual Markup Decomplier" [Deng et al.]. This im2latex dataset consisted of images of compiled Latex and the corresponding sequence used to generate it.

However, our attempts at creating a data processor and dataloader for the raw dataset were futile, and again, we needed to pivot.

New Problem Statement and Dataset

Given our past complications, we decided to use the reliable MNIST dataset.

We elected to experiment with using a CNN and a pre-trained ViT model. Our primary goal was to compare two different approaches for an image recognition task: training from scratch and transfer learning.

Our Model

Experiments using CNN-based models preformed well, reach a 90% to 95% accuracy quickly.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_7 (Conv2D)            (None, 26, 26, 32)        320

batch_normalization_7 (Batc  (None, 26, 26, 32)        128
hNormalization)

conv2d_8 (Conv2D)            (None, 24, 24, 32)        9248

batch_normalization_8 (Batc  (None, 24, 24, 32)        128
hNormalization)

conv2d_9 (Conv2D)            (None, 12, 12, 32)        25632

batch_normalization_9 (Batc  (None, 12, 12, 32)        128
hNormalization)

dropout_3 (Dropout)          (None, 12, 12, 32)        0

conv2d_10 (Conv2D)           (None, 10, 10, 64)        18496

batch_normalization_10 (Bat  (None, 10, 10, 64)        256
chNormalization)

conv2d_11 (Conv2D)           (None, 8, 8, 64)          36928

batch_normalization_11 (Bat  (None, 8, 8, 64)          256
chNormalization)

conv2d_12 (Conv2D)           (None, 4, 4, 64)          102464

batch_normalization_12 (Bat  (None, 4, 4, 64)          256
chNormalization)

dropout_4 (Dropout)          (None, 4, 4, 64)          0

conv2d_13 (Conv2D)           (None, 1, 1, 128)         131200

batch_normalization_13 (Bat  (None, 1, 1, 128)         512
chNormalization)

flatten_1 (Flatten)          (None, 128)               0

dropout_5 (Dropout)          (None, 128)               0

dense_1 (Dense)              (None, 10)                1290

=================================================================
Total params: 327,242
Trainable params: 326,410
Non-trainable params: 832
```

Figure: Our CNN model summary

Results

The following graphs were generated visualizing our model accuracy and loss over ten epochs.

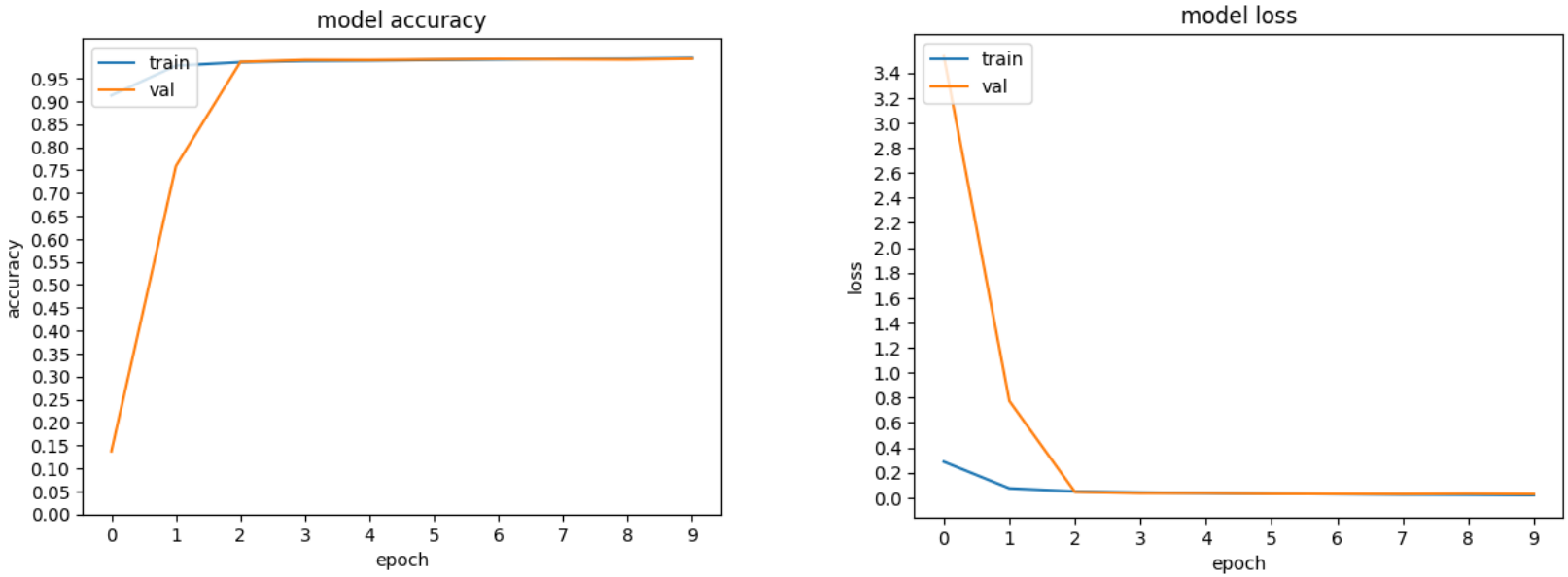

Figure: Our CNN model results

Transfer Learning - Comparison with a ViT

When evaluated on the test dataset of MNIST, our CNN model had 99.24% accuracy and a loss of 0.0246, as compared to the pre-trained ViT mentioned earlier, which has an accuracy of 98.29%.

Given current literature, it is expected that the ViT perform at least as well as a CNN, which do not correlate with our results.

However, Huggingface documentation says this model is a version of the google/vit-base-patch16-224-in21k model that has been finetuned on MNIST. According to documentation, the ViT model achieved an evaluation loss of 0.0236 and evaluation accuracy of 99.49%, which does support our initial expectation of better performance from the ViT model over the CNN

Table of Contents

References

Zhelezniakov et al (2021)

Online Handwritten Mathematical Expression Recognition and Applications: A Survey

IEEE Access vol. 9, pp. 38352- 38373, 2021, doi: 10.1109/ACCESS.2021.3063413

Deng et al. (2016)

What You Get Is What You See: A Visual Markup Decompiler

Deng et al. (2017)

Image-to-markup generation with coarse-to-fine attention

International Conference on Machine Learning, 4PMLR, 2017

Huang et al. (2007)

Preprocessing techniques for online handwriting recognition

Proc. 7th Int. Conf. Intell. Syst. Design Appl. (ISDA), , pp. 793-800, Oct. 2007.

(Remaining sources can be found in project report)