Machine Learning
Project Proposal

Aditya Chetan - 2016217
Brihi Joshi - 2016142
Siddharth Yadav - 2016268
Taejas Gupta - 2016204

October 2, 2018

Motivation and Problem Statement

Problem statement:
To convert the picture of a mathematical expression into its corresponding compilable LaTeX representation.

Learning task:
To learn the LaTeX representation of the symbols present in the images and the relations between them.

Motivation:
There is a vast availability of documents and images containing mathematical expressions and equations on the internet. The reproduction and digitisation of such images in LaTeX is a challenging task. Many people who are not proficient with LaTeX find it difficult to type out mathematical equations due to its steep learning curve. This is why we wish to develop a tool to automate such conversions. Another reason for us to choose this project is to get started with Computer Vision research.

Dataset

We are using datasets of two kinds:

- **Handwritten equations:** The Kaggle Handwritten Mathematical Expressions has handwritten equations in the `.inkml` format. [1]
- **Typeface equations:** OpenAI has started an `Im2latex` challenge, the dataset for which is 100k+ in size, available here. [2]

Preprocessing

We will be using image augmentation for the end-to-end techniques. Also, we will be generating more data from our existing datasets by recompiling their target labels with different typefaces and stylings.

Learning Techniques

We would be following a three-step procedure to implement various learning techniques.

1. **Normal Learning Techniques:** For these techniques we would be mainly using the handwritten dataset from Kaggle. This is because the handwritten dataset does not contain very complex formulae and can entertain heuristic measures for determining the relationships between different symbols, making simple classification algorithms easy to use.
   (a) Naive Bayes Classifier
   (b) Random Forest
   (c) Support Vector Machines
   (d) Multi-Layer Perceptron
   (e) Convolutional Neural Network (CNN)
2. **End-To-End Model** (Only to be implemented if we have enough time and resources):
   (a) **CNN + RNN**: In these methods, we will be using CNN to extract features from images and form the initialisation state for the RNN for producing the latex code
      i. seq2seq GRU RNN: GRU is one of the simplest RNN type
      ii. seq2seq LSTM: It is the most popular RNN variation [6]
      iii. seq2seq Bi-LSTM: We believe that the output of the RNN at a particular instant is dependent on the points lying on the upcoming as well as the previously encountered samples.

iv. seq2seq LSTM + Attention + Beam Search [5]

(b) Convolutional seq2seq [3]

(c) Variants of SqueezeNet [4]

Since the methods above require lots of data (or they will tend to overfit), we will increase the training samples by image augmentation. Image augmentation makes our model more robust.

Model Selection Criteria

Since our data is non-linear, we will be using kernel-based techniques in SVM.
In order to tune the hyperparameters (if any) for the Normal Learning Techniques, we shall use k-fold cross validation. Since the deep learning models require a lot of time and resources to train, we shall not be using k-fold cross validation to tune these. We will merely use the values of the hyperparameters in the papers that we have referred them from.

Evaluation Metrics

For our Normal Learning Techniques, the main evaluation would be for the classification task. Here, we would be using metrics like:

- Accuracy (Both Training and Testing Accuracy to check for overfitting)
- Precision
- Recall
- F1-Score

In case of end-to-end learning, the complete model is trained to maximise the likelihood of the training data. For our end-to-end learning techniques, we would be using metrics like:

- Edit Distance
- BLEU Score
- Exact Match

These metrics will be calculated using k-fold cross validation.

Learning Techniques

We have currently decided on using the Adam Optimizer for training the Neural Network based models.

Individual Responsibilities

- **Aditya** - Preprocessing, Segmentation pipeline, Model Selection, End to End Learning Techniques, System Development
- **Brihi** - Preprocessing, Segmentation pipeline, Normal Learning Technique, Evaluation, Report Writing
- **Siddharth** - End to End Learning Technique, Heuristics for normal learning techniques (Structural Analysis), Evaluation, System Development
- **Taejas** - Normal Learning Techniques, Model Selection, Heuristics for normal learning techniques (Structural Analysis), Report Writing

References

[1] Kaggle, Tatman, Rachael. "Handwritten Mathematical Expressions".

[2] Kanervisto, Anssi. (2016). im2latex-100k, arXiv:1609.04938 [Data set]. Zenodo. http://doi.org/10.5281/zenodo.56198

[3] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, Yann N. Dauphin. Convolutional seq2seq. (2017). arXiv:1705.03122

[4] Forrest N. Iandola, Song Han, Matthew W. Moskewicz, Khalid Ashraf, William J. Dally, Kurt Keutzer. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. (2016). arXiv:1602.07360

[5] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, Yoshua Bengio ; Proceedings of the 32nd International Conference on Machine Learning, PMLR 37:2048-2057, 2015.

[6] Oriol Vinyals, Alexander Toshev, Samy Bengio, Dumitru Erhan. Show and Tell: A Neural Image Caption Generator. (2015). arXiv:1411.4555 .