Abhinav Java

Delhi Technological University (erstwhile DCE)

Deep Learning Internship, IITB

August 2020

Attention based Optical Character Recognition

Optical Character Recognition is a deeply studied topic in Computer Vision and NLP and has a wide range of real world applications. A robust handwritten form reader facilitated by the means of high quality synthetic data is developed for this project. The impact of **normalizing the content-based factor and smoothening the activation** on **hybrid Bahdanau Attention** is studied in the following report, both of which indicate a considerable **performance gain**. Another trivial yet extremely important observation is the effect of different data distributions on real world test accuracies using comprehensive data analysis techniques highlighting the importance of a dataset capable of capturing a wide range of distributions.

Synthetic Dataset Generation

To capture the wide variety of variations present in real world data like handwriting (font), noise, perspective of camera and illumination conditions among others the importance of a good synthetic dataset cannot be stressed enough. In order to evaluate the same a number of datasets were generated using an open source tool called text-renderer [1].

For the purposes of this study, we were required to generate text samples with characters inside boxes. In order to achieve this, the open source text-renderer is modified as follows:

1. **`PIL ImageDraw.rectangle`** function is used as each character is written on the text mask, ensuring that all rectangles are of approximately the same size. This operational modification is done for **`draw_text_on_bg`** in **`text-renderer/text-renderer/utils/draw_utils.py`**
2. Furthermore, the color of the bounding box is selected randomly from a list of shades of black (#000) to ensure it's robustness due to variation in color of the real dataset.
3. A series of augmentation steps are undertaken in order to capture the diverse set of distributions. The main generator function is in **`text-renderer/ocr_data`** along with the set of subdirectories for background, fonts, character list and text. Other steps to ensure robustness of dataset are
    a. Randomly generated list of words and english paragraphs.
    b. Using 5 different background images, with varying degree of luminance (based on visual cue)
    c. Using 5 different script fonts which have close visual similarity to human handwriting.
4. In order to convert the `json` output annotations, a simple conversion script is written in **`text-renderer/dataset_labels/convert_labels.py`**

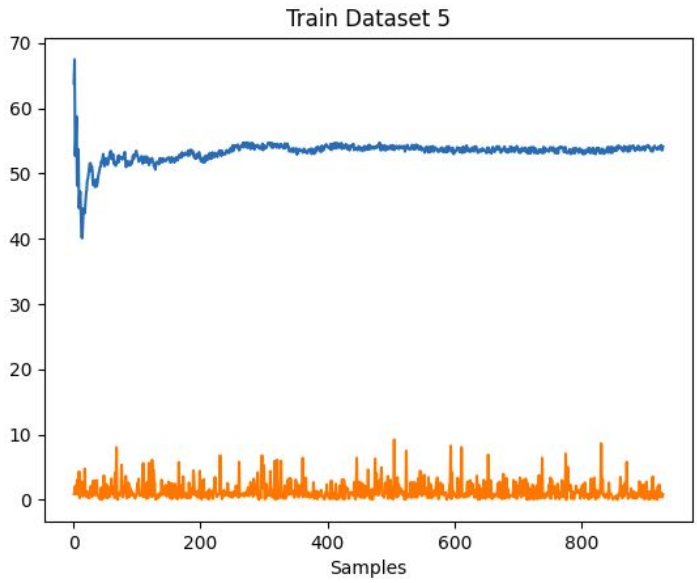
Using the above mentioned augmentation strategies, multiple datasets were generated with varying levels of augmentation and specifications and were iteratively regenerated after evaluation on the public dataset.

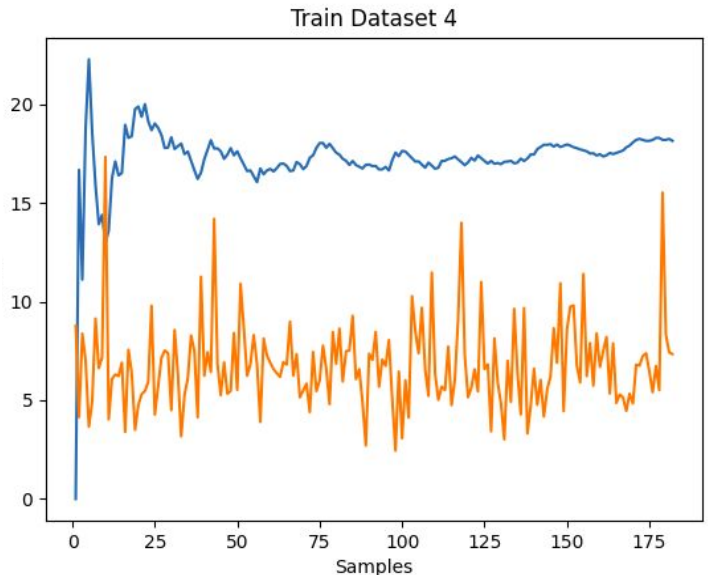
Synthetically Generated Images (above)

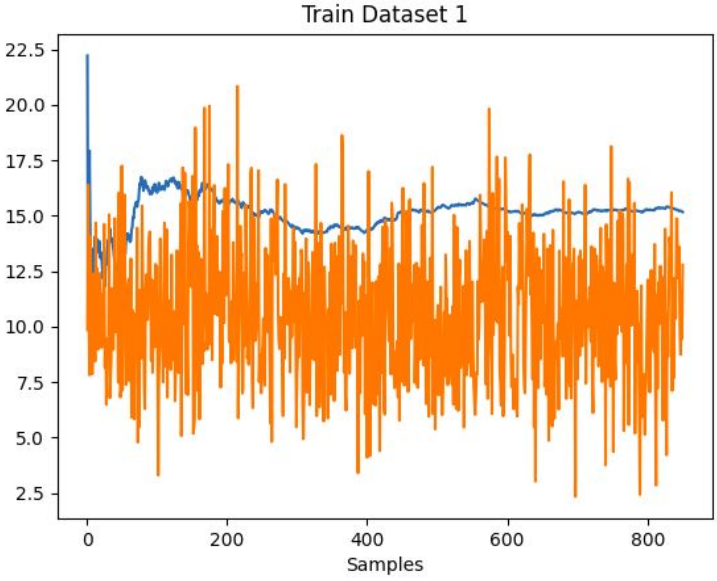
Public Images (above)

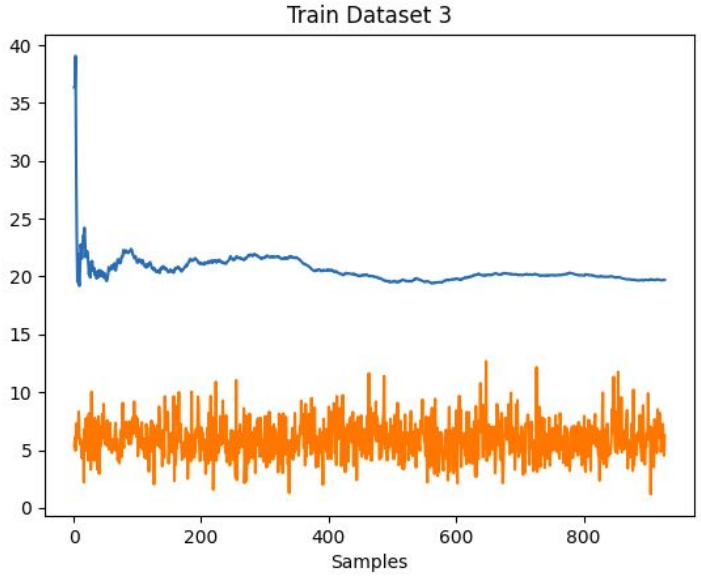
Despite the high similarity of generated data with real data (based on visual cue), the existing OCR model [2] was unable to generalize even after hyperparameter tuning which was the basic motivation behind the generation of multiple datasets and the subsequent selection of the best one from the set. This was a particularly time consuming process.

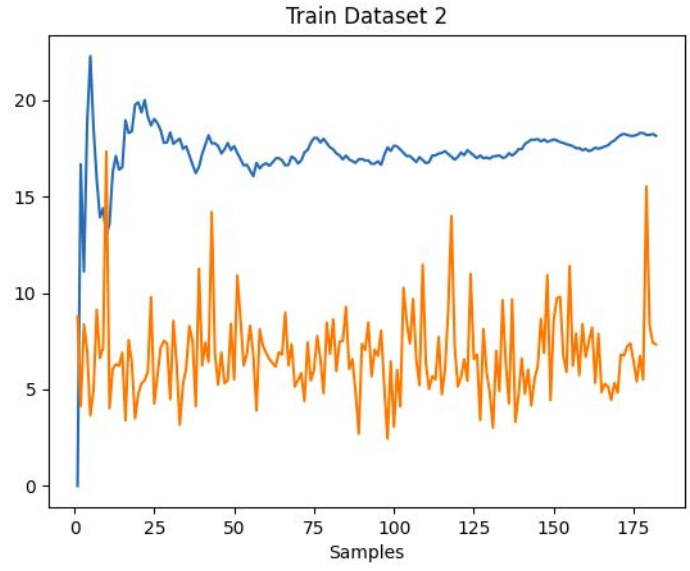
Dataset-wise Comparison of Attention OCR











Orange: Loss

Blue: Accuracy on Public Dataset

The plots above show a comparison between the accuracies and losses obtained on the public dataset for different distribution of train images. The upper bound of accuracies obtained for **dataset 1, 2, 3, 4 is 40%**, which is extremely undesirable. In order to rectify this behavior a much diverse dataset, **dataset 5** was generated using all the augmentation techniques mentioned above. And naturally it's train accuracy was not particularly at power with the other datasets (which essentially overfit), however it's accuracy on public datasets was much better as shown above.

Please note that in some of the plots outputs have been truncated in order to give the general idea.

Normalized Attention OCR

Upon observing the output of the trained attention ocr the following observations were made:

1. The model is unable to generalize well to a real world distribution due to insufficient dataset size and variation. This can be validated further since the validation split of the dataset (synthetic) also gave extremely high accuracy, yet a large gap was observed while testing on public dataset.
2. Negligible impact of embedding vector size on output accuracy.
3. **Small attention masks unable to capture the entire character**
4. **Repeated attention if the same character is encountered again.**

To make a robust pipeline all of these issues were attempted to be rectified. Since the model was trained and tested on Google Colab and there were major computational and time limitations the dataset size was not increased, instead a larger variety of dataset was used as mentioned previously.

In order to fix the attention mask size **attention smoothening** as described in [3] was used. In order to smoothen the attention weights, sigmoid function was used instead of the softmax in Bahdanau Attention as follows:

$$a_{i,j} = \sigma(e_{i,j}) \Big/ \sum_{j=1}^{L} \sigma(e_{i,j}).$$

where e represents the Bahdanau score and sigma is the sigmoid function.

This resulted in a sufficiently large attention map which was able to correctly capture the character in the input image as shown below. Note that this smoothening is only used during training due to its regularizing effect.

The other important observation of that the network was attending to the previously seen character in case the same character was encountered again. For instance, if the input is `SBINOOO123` then the sliding window keeps attending to the first `O` instead of moving forward. In order to make sure that the hybrid attention mechanism did justice to both **content and location** the factor regulating the hidden representation was normalized as follows:

```
v[a] = g*(v[a]/tf.norm(v[a])) # Normalization term
s = tf.reduce_sum(v[a] * tf.tanh(hidden_features[a] + y), [2, 3])
```

where **v[a]** is a vector and **tanh** is applied element-wise to the same. The normalization of vector v ensures that the **hidden_features (content-based factor)** do not dominate the equation and **y**, which is the location aware factor, can express the attention map as required. In the above equation **g** is a learnable scalar.

Largely smoothened attention

Sigmoid attention

Pure Softmax Attention

Normalized, smooth Attention

Various experiments with different forms of activation were conducted and it was observed that the **normalized and smoothened attention** strategy worked the best compared to the pure softmax or purely smooth attention. In the above diagram we notice a slight deviation however, note that the size of the attention mask is perfect to capture the character, which is what happens in most cases giving high accuracy. Lastly sigmoid attention worked even worse than the standard implementation.

Both the visual evidence and accuracies were the basic motivation behind this modification.

Weight Normalization

For a standard ANN the computation of each neuron consists in taking a weighted sum of input features and element wise nonlinearity:

$$y = g\,(w * x + b)$$

In order to make this optimization process better, weights are often expressed as new parameters using:

$$w = g * (v/||v||)$$

The normalization used for our purposes is different from the one done at each step of stochastic gradient descent, instead **explicit reparameterization** is done for this task, by making g learnable. Similar work has been proposed in [6]

The other alternatives for the attention mechanism with the perspective of handwriting recognition are listed below. The attention mechanisms have been studied extensively in [4] and were used as motivation for this report:

1. **Content Based Attention:**

$$e_{t,j} = \begin{cases} s_t^{\mathsf{T}} W_h h_j & \text{Luong} \\ v^{\mathsf{T}} \tanh(W_s s_t + W_h h_j + b) & \text{Bahdanau} \end{cases}$$
$$\alpha_{t,j} = \text{softmax}(e_{t,j}),$$

(No involvement of the location information)

2. **Penalized Attention:**

If the timestep is unity the attention is calculated differently compared to the other timesteps

3. **Location based attention:**

Dropping the hidden_features or h_j factor and only using the queries while score calculation is the fundamental of location based attention.

4. **Hybrid Attention:**

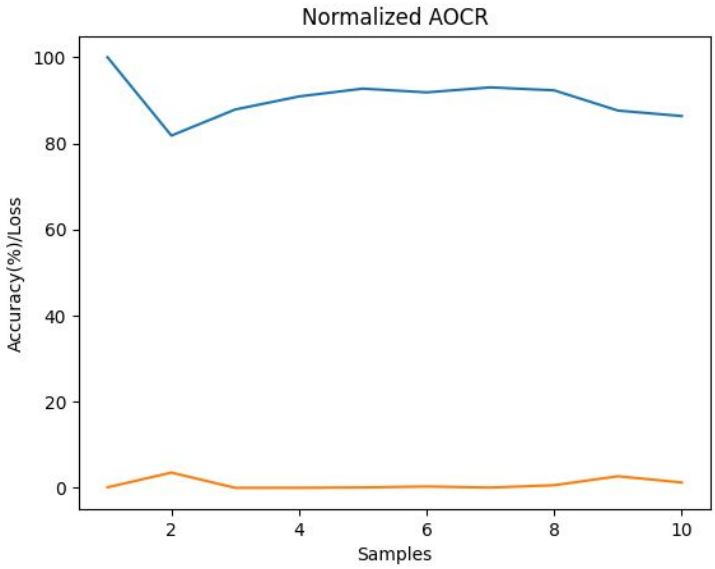
The attention mechanism used in attention ocr which is both content and location aware gives the best performance according to [4], which is why the other attention mechanisms were not analysed for this report.

Lastly, the convolutional layers of the `CRNN` feature extracted were made deeper in order to capture higher level image information for the encoder-decoder. This decision was made since the model initially kept attending to the box edges, and it is known that lower level features of neural networks are typically edge detectors (edges). This motivated the use of more complex feature maps for inference.

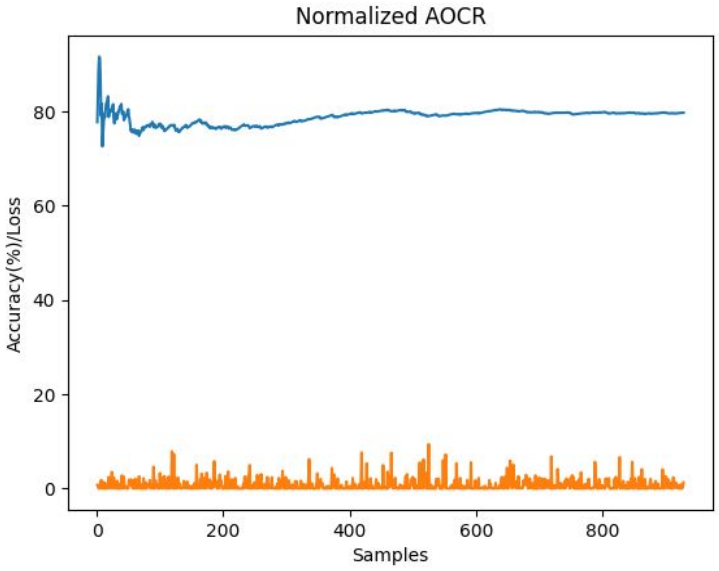
Comparison on Public Dataset (Real World)

These simple modifications give a considerable performance gain in the normalized Attention OCR as shown by the figures below.

Normalized Attention Optical Character Recognition



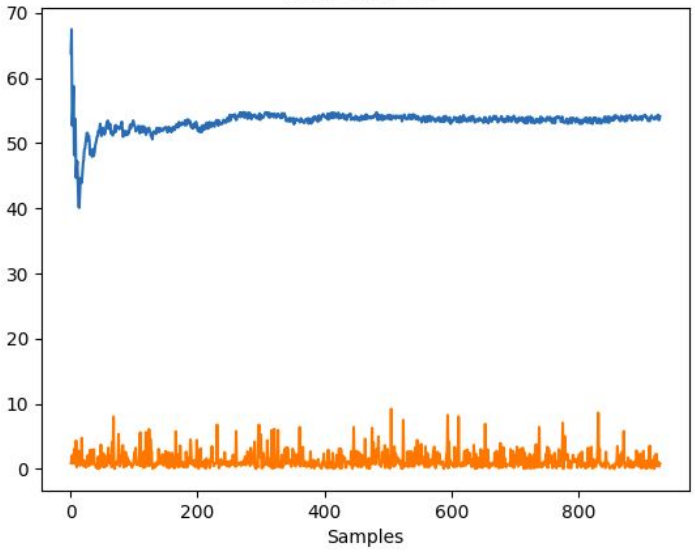
Assignment Noisy Samples



Public Dataset

Attention Optical Character Recognition

(Official Implementation)



Orange: Loss

Blue: Accuracy on Public Dataset

Conclusion and Future Prospects

The analysis clearly indicates that the content and location awareness factors in the attention mechanism, if normalized and smoothened can produce a large accuracy gain for the purposes of HTR. Another important inference is the low variance produced by the normalization of the weights. This study still needs to be refined in order to understand optimal conditions for a more generalized convergence. Despite **high test accuracy (71%)** on the public dataset, it is possible that the model might have overfit to a certain extent. This needs to be rectified by using more data which is able to capture even higher levels of variations like the real world data.

Furthermore, a number of other experiments could not be carried out due to the strict deadline. These experiments include the departure from the RNN Seq2Seq style encoding and decoding to Transformer and positional encoding [5] based neural network. Transformers have been instrumental due their parallel computation capability and independence of the conventional RNN which is known to not be able to capture long dependencies. Another work that can be explored is temporal attention based networks.

References

a. [1] https://github.com/oh-my-ocr/text_renderer/tree/master/text_renderer

b. [2] https://github.com/emedvedev/attention-ocr

c. [3] Sankaran, Baskaran, et al. "Temporal attention model for neural machine translation."

d. [4] Michael, Johannes, et al. "Evaluating sequence-to-sequence models for handwritten text recognition." 2019 International Conference on Document Analysis and Recognition (ICDAR). IEEE, 2019.

e. [5] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems. 2017.

f. [6] Salimans, Tim, and Durk P. Kingma. "Weight normalization: A simple reparameterization to accelerate training of deep neural networks." Advances in neural information processing systems. 2016.

Other sources of information:

a. Wojna, Zbigniew, et al. "Attention-based extraction of structured information from street view imagery." 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). Vol. 1. IEEE, 2017.

b. Chowdhury, Arindam, and Lovekesh Vig. "An efficient end-to-end neural model for handwritten text recognition." arXiv preprint arXiv:1807.07965 (2018).

c. Poulos, Jason, and Rafael Valle. "Character-based handwritten text transcription with attention networks." arXiv preprint arXiv:1712.04046 (2017).