Deep Neural Networks for Large Vocabulary Handwritten Text Recognition
(2015)

A Resume

November 28, 2018

Abstract

A thesis on handritten text recognition.

1 Introduction

Handwriting recognition is the process of transforming a digital representation of the physical result of handwriting into a digital text, generally for further treatments, such as indexing, classification, or translation. The handwritten text is to be extracted from the image, using image processing techniques or relevant feature extraction. This is done in two ways, Online (touch screens), or offline (scanned documments), for offline, we talk about optical charachters recognition (either scanned prited text or recognition of hand writing).

Difficulties : Different writing styles, varying character shpes even with the same person.

2 Preliminary Steps to Offline Handwriting Recognition Handwriting

First, the text in the document should be localized, and possibly grouped into text zones. Sometimes, one should prepare the documents beforehand. The digitalization process may introduce noise in the image. The resolution can be an important aspect of the image, and may need to be normalized, and when the language of the document is not known, some methods can be applied at the image level to determine the language, as to pass the image to a specific recognition system.

So before doing character recognition, a segmentation step is performed, involving a connected componenet analysis or vertical projection profile, the result is obtained by deciding whether distances between componenets, or non white columns of pixels is indicative of a word separation. The objective is to split text into the text lines into word images (given that spliting an image into images of character is not an easy task).

3 Reducing handwriting variability

Normalizing Contrast Assuming that the light (white) pixels correspond to back- ground and dark ones to foreground, i.e. writing, the dissimilarities across images can be reduced by contrast enhancement techniques. Binarization is the crisper approach, mapping the range of pixel values from [0, 255] (0 is black) to {0, 1} using thresholding techniques.

Normalizing Skew Text lines are generally not straight lines. In handwritng the writing goes down, and the bounding box of a line also includes part of the surrounding lines. Recognition systems must follow the text line, which is easier when the line is really horizontal.

Normalizing Slant The writing of some people tends to be inclined. Slant correction methods roughly consists in finding the angle between the vertical axis and the strokes that should be vertical, which are mainly found in ascenders and descenders, and in applying a shear transformation to correct that angle.

Normalizing Size Writing Writing size is also an important source of diversity across writers and documents. It includes the general area occupied by a word, as well as the size of some components (such as ascenders (top of h) and descenders(bottom of y)), the thickness of the stoke, and the tightness of writing. For height, rescaling all images to a common value is the simplest solution. However, the presence of ascenders/descenders will affect the size and the position of the core region.

4 Features extraction

Unlike speech recognition, where popular features have strong theoretical foundations in physiology or signal processing, the majority of features proposed for hand- writing recognition are heuristic ones, handcrafted by researchers.

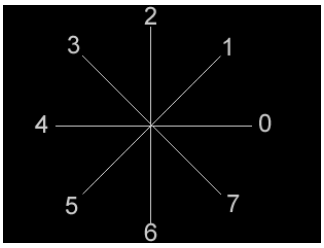
Text Segmentation for Feature Extraction When the system does not directly model words in which case the features can be extracted from the whole word image but characters, one should segment the image.

There are mainly two approaches for segmentation, both considering smaller regions of the image, corresponding to at most one character, and letting the recognition system merge sub-parts into characters. Basically, the first one consists in heuristically segmenting the ink into small chunks, at positions which are likely to be connections between characters. Features are extracted from the smaller images. The second method does not make any assumptions or use any heuristic. It consists of scanning a sliding window through the image, and compute the features in each extracted frame.

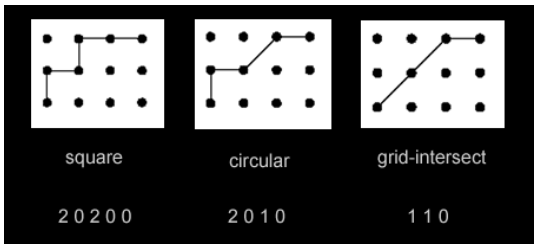
Features for Handwriting Representation Features may be very simple (such as pixel intensities) or a higher-level representation of the image. Different complexities involve different amount of computation and image processing effort to obtain them.

- **Pixels:** In some situations, using the pixel intensities may be sufficient, with a lot of effort on the image pre-processing, a simple recognition system yield good results such as using PCA to reduce and decorrelate the pixel dimensions.
- **Low-level pixel features Simple:** Simple features can be derived from the image easily by mere counting or averaging operations, such as a count of black pixels, pixels density, run-length analysis.
- **High-level features:** More sophisticated features, giving a higher representation of the shape, may be computed with simple operations on images. such as derivatives of pixel intensities, the orientation of contours, histograms of oriented gradients,
- **Shape features:** To capture a more global view of the shapes in the image, one can also look for common structural elements of handwriting, such as loops (Knerr et al., 1998; Guillevic & Suen, 1998), T-crossings or presence of ascenders and descenders.
- **Learning Features:** either using unsupervised learning, where the obly objective is to map the image onto a compact feature vector which accurately represent the image, and supervised learning, where the obtained representation is the by product of the discriminative task of classiying the shape.

Side note : Chain code Chain Coding is a method used to represent line drawings that help efficient transmission and storage of such data. The approximation of a line drawing using the sequence of curve points found using any of the three quantization schemes (square, circular, grid-intersect) can be coded using at most 8 bits per line segment. This is because for any curve point the next curve point in the sequence can be in at most eight directions as shown in the figure below.



Each line segment in the approximation is given a code according to the eight directions the segment can take, giving rise to a chain code representation for each line drawing. Each of the three quantization schemes desribed yield the chain codes shown in the next figure.



The different quantization schemes often yield very different chain codes for the same line drawing. One of the factors in which the schemes differ is the number of diagonal line segments they contain. It can be readily seen that the square quantization does not give any diagonal segments (assuming that the line drawing never passes through the intersection of two mesh lines), whereas they do occur in circular and grid-intersect quantizations. According to Freeman, diagonal elements should occur half the time for random configurations for a good or close approximation.

5 Modeling handwriting

The transformation of feature vectors into digital text. We call this module optical model. The choice of optical model is related to the previous steps especially the features and the segmentation of the input.

- **Whole-Word Models:** In this first class of methods, the goal is to recognize a word directly as a whole, without relying on a segmentation or on an explicit representation of the parts. Often, a simplified representation of the word shape is extracted from the images and matched against a lexicon, The disadvantage of whole word modeling is the inherent limitation of the vocabulary size. The number of models grows linearly with the number of words. Moreover, for methods based on a matching to a prototype, a new prototype must be created with every added word.

- **Part-Based Methods:** the image is divided into sub-regions corresponding to at most one character. Gigen the difficulty to segment characters from cursive words, without knowing the word's identity. Thus these systems often perform an over-segmentation of the image, or consider several segmentation alternatives. The goal is to merge different part to get complete characters, or to find the correct segmentation into characters among the different hypotheses.

- **Segmentation-Free Approach:** In the segmentation-free approach, the recognition is accomplished without an explicit segmentation of the image, thus without relying on heuristics to find character boundaries, and limiting the risk of under-segmentation.

6 Language Modeling for a contrained recognition

6.1 Vocabulary

When the system models words directly, the vocabulary is already embedded in the recognition engine, and limited to the set of words modeled, which is usually relatively small. In other cases, a vocabulary constrains the sequences of characters to form words from a predefined set. It helps alleviate ambiguities arising in the recognition procedure, and limits the size of the search space.
As the vocabulary size grows, so does the search space. On the other hand, the chance to encounter an Out-of-Vocabulary word (OOV) decreases. As a consequence, the choice of a vocabulary is usually a tradeoff between size and coverage. the coverage is measured in terms of OOV rate, i.e. the proportion of words in the test dataset that are not in the vocabulary. The OOV rate is a lower bound on the error rate.
To reduce the search space (avoid linear complexity), the vocabulary is organized as a tree, which root cor- responds to the beginning of a word. Each branch is associated with a character, and a terminal nodes contains the word made of the letters along the path.

6.2 Language Modeling

Even when the system outputs tokens from a vocabulary, not all sequences of words are valid, i.e. grammatically or semantically correct. Checking the grammatical validity of a sentence may not be easy, but many techniques are developed in the field of Natural Language Processing (NLP). More generally, language modeling for handwriting recognition usually consists in giving a score to different word sequence alternatives. The score measures how likely is the observation of the given sequence of words.

To measure the suitability of a language model to a given corpus, one usually computes its perplexity. It is derived from the entropy of the probability model, and can be expressed as follows:

$$PPL = \left(p\left(w_1, \ldots, w_{N_w}\right)\right)^{-\frac{1}{N_w}} = 2^{\frac{1}{N_w}\sum_{k=1}^{N_w} \log_2 p(w_k|w_{k-1},\ldots)}$$

Statistical n-gram Language Models The space of all possible word sequences is very large. Estimating a probability distribution over this space, even with very large corpora, is difficult, especially because most word sequences will not appear. Factorizing $p(W)$ with the chain rule:

$$p(\mathbf{W}) = p\left(w_1, \ldots, w_N\right) = p\left(w_1\right) p\left(w_2|w_1\right) \ldots p\left(w_N|w_1, \ldots, w_{N-1}\right)$$

yields a representation in which the probability of observing a word only depends on the previous words. The n-gram approach addresses the data scarcity problem by making Markovian assumptions: the probability of a word does not depend on the position of the word in the sequence, and only on an history of n 1 previous words:

$$p\left(w_k|w_1, \ldots, w_{k-1}\right) = p\left(w_k|w_{k-1}, \ldots, w_{k-n+1}\right)$$

The Maximum Likelihood estimator of n-gram probabilities is achieved by mere counting in the corpus:

$$p\left(w_k|w_{k-1}, \ldots, w_{k-n+1}\right) = \frac{C\left(w_k, w_{k-1}, \ldots, w_{k-n+1}\right)}{\sum_w C\left(w, w_{k-1}, \ldots, w_{k-n+1}\right)}$$

Neural Network Language Models The basic idea of these methods is to project each word of the history in a continuous space and to perform a classification with neural networks to predict the next word.

6.3 Open-Vocabulary approaches

n-grams are introduced for modeling word sequences, but they can as well serve to model sequences of characters. The advantages of working with characters rather than words are multiple. First, as already mentioned for recognition models, there are less different tokens, and much more data for character modeling. Therefore, we can estimate more reliable probability distributions, and build higher order n-grams. Moreover, although the outputed words will not necessarily be valid ones, there would be not such things as OOVs

7 Measuring the quality of recognition

In order to assess the quality of a recognition system, a measure of the performance is required. For isolated character or word recognition, the mere accuracy (proportion of correctly recognized items) is sufficien. But for sentences, counting the number of completely correct sequences is too coarse, because there would be no difference between a sentence with no correct word and another with only one misrecognition, and we do not only find incorrect words, but there might also be inserted or deleted words.
Measures such as precision and recall may take these types of errors into account, but not the sequential aspect. The most popular measure of error, used in international evaluations of handwriting or speech recognition, is based on the Levenstein edit distance.
This distance counts the number of edit operations required to transform one string into another. The possible edits are: substitution of one item for another, deletion of one item of the sequence or an insertion of one item in the sequence.
The minimum edit distance between two strings can be retrieved efficiently with a dynamic programming algorithm. The Word Error Rate (WER) is obtained by com- puting the minimum number of edits from the reference string to the output transcript,

$$WER = \frac{n_{sub} + n_{ins} + n_{del}}{n_{ref}}$$