Denoising Dirty Documents with Neural Network

Nhat Hoang Pham *
Minh Hoang Phan †

June 2018

Contents

1 Introduction **2**

2 Algorithms **2**

3 Evaluation and Results **3**

*Lead Author
†Co-Author

1 Introduction

This project is based on Kaggle's *Denoising Dirty Documents* competition. The training and testing data are obtained from [1].

2 Algorithms

First, we have the following remarks:

- The actual contents of the text do not matter. The model only needs to learn to differentiate the "word" pixels from the "background/noise" pixels based on the local texture information.
- The classification of each pixel only concerns several pixels around it.

From these two remarks, we came to the conclusion that we can crop an image into smaller subimages and train our denoising model on these subimages. This also has extra benefits:

- Smaller models.
- Additional training examples.
- Prevent overfitting: The model will purely attempt to learn to differentiate the "word" pixels from the "background/noise" pixels instead of trying to "copy" the text from the training input to the output, which will affect generalization to testing data.

To further maximize the amount of training data, instead of simply cropping, we could "slide" a window across the picture, and train on batches of windows. This also allows us to clean documents of any size.
At test time, we perform the same sliding operation, "clean" each windows, and stich them together to reconstruct the clean version of the original photo.
For a pixel overlapped by different windows, we take the average of the corresponding pixels in each window as the final prediction. This also has an "ensembling effect".
The main model that we use to denoise is an autoencoder-like neural network. $f_\theta : \mathbb{R}^{30\times 30} \to \mathbb{R}^{30\times 30}$, given by:

$$f_\theta = f_2 \circ f_1$$

where f_1 is a convolutional encoder, f_2 is a decoder (feedforward or deconvolutional), and θ is the parameters vector of f.
f_1 has the purpose of projecting the original image into a new space of lower dimension, whereas f_2 attempts to restore the original image from this representation. A certain degree of information will be lost in the process, which hopefully is the noise we want to remove.
For f_1, we use a convolutional layer with max-pooling to reduce the dimension

of the picture, before applying several "residual modules", which is modelled after the residual network, as described in [3]. For f_2, we can simply set up a two-layer fully-connected neural network, or use a deconvolutional layer to upsample the transformed images to original size.
To train the network, we use a simple mean squared error (MSE) loss function:

$$J(\theta) = \sum_{(X,X')\in\mathcal{B}} \sum_{i=1}^{30}\sum_{j=1}^{30}(X'_{ij} - f(X)_{ij})^2$$

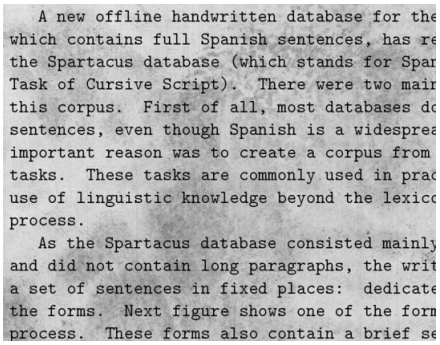
Where:

- $\mathcal{B}$ is the training batch of data.
- $X \in \mathbb{R}^{30\times30}$ is the dirty image.
- $X' \in \mathbb{R}^{30\times30}$ is the corresponding clean image.
- M_{ij} is the pixel at the position (i, j) of the image M, for $M = X'$ or $f(X)$.

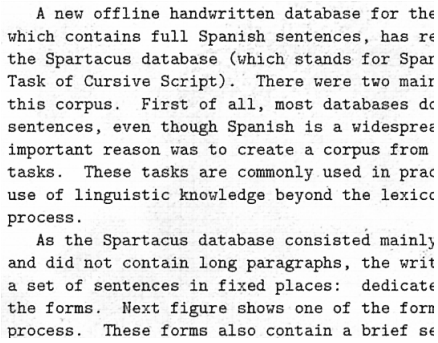
We minimize this loss function using the Adam optimization algorithm, as described in [2].
After we have trained the network, at test time we "reconstruct" the images from their subimages. Additionally, we can also apply a binary thresholding algorithm to the images to improve the the "crispness" of the restored photo, but this makes the images look somewhat unnatural.

3 Evaluation and Results

Some of the results are shown below:
Before:



After:



A new offline handwritten database for the
which contains full Spanish sentences, has re
the Spartacus database (which stands for Spar
Task of Cursive Script). There were two mair
this corpus. First of all, most databases dc
sentences, even though Spanish is a widespree
important reason was to create a corpus from
tasks. These tasks are commonly used in prac
use of linguistic knowledge beyond the lexicc
process.

As the Spartacus database consisted mainly
and did not contain long paragraphs, the writ
a set of sentences in fixed places: dedicate
the forms. Next figure shows one of the forn
process. These forms also contain a brief se

References

[1] K. Bache & M. Lichman. UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science (2013).

[2] Diederik P. Kingma, Jimmy Ba. Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980 (2014).

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition. arXiv preprint arXiv:1512.03385 (2015).