Imperial College London

UROP

IMPERIAL COLLEGE LONDON

DEPARTMENT OF MATHEMATICS

Signature method on Character recognition

Author:
Zhuobin Zhou

Supervisor:
Dr.Henri Elad Altman

August 24, 2020

Contents

1 Introduction **3**

2 Signature **4**
- 2.1 Signature of path . . . 4
  - 2.1.1 Path integral . . . 4
  - 2.1.2 Signature of path . . . 4
  - 2.1.3 Chen's identity . . . 4
- 2.2 Algorithm . . . 5

3 Preparation **6**
- 3.1 Dataset . . . 6
- 3.2 Model . . . 6
- 3.3 Input . . . 6
- 3.4 Signature layer . . . 6

4 Evaluation **7**
- 4.1 DeepCNet vs Letnet . . . 7
- 4.2 Model result . . . 8
- 4.3 Low quality of higher level of signature . . . 9
- 4.4 Evaluation . . . 10

A Appendix **11**
- A.1 All in one . . . 11

Chapter 1

Introduction

The purpose of this project is to evaluate the application of signature method on character recognition. At the beginning of the article, we introduce some basic concepts of signature. In the second part, we evaluate the data(Pendigits) and the model (CNN) for character recognition. Then we compare the result from the model with and without the application of signature.

Chapter 2

Signature

In this chapter, we only review the key concepts used in the application of character recognition.[CK16]

2.1 Signature of path

2.1.1 Path integral

First, we briefly go through the concept of path integral. For a one-dimensional piecewise-differentiable path, $X : [a,b] \mapsto \mathbb{R}$, the path integral of $f(X_t)$ against X_t is defined by

$$\int_a^b f(X_t)\, dX_t = \int_a^b f(X_t)\dot{X}_t\, dt,$$

where the $\dot{X}_t$ is the derivative of X_t with respect to t. If we substitute $f(X_t)$ with another path , Y_t, this becomes the path integral of a path with respect to the other path.

2.1.2 Signature of path

Now, we are ready to define signature of path. For a path $X : [a,b] \to \mathbb{R}^n$, we denote the coordinate paths by $(X^1, X^2, ..., X^n)$. For i in (1,2,...,n),we can define

$$S(X)^i_{a,b} = \int_{a<t<b} dX^i_t.$$

This is the first level of signature.Then we define the double-iterated integral for any pair (i,j) in (1,2,...,n)

$$S(X)^{i,j}_{a,b} = \int_{a<t<b} S(X^i)_{a,t} dX^j_t$$

Likewise for n level of signature, we define the multi-iterated integral

$$S(X)^{(i,..,j,k)}_{a<b} = \int_{a<t<b} S(X)^{(i,...,j)}_{a,t} dX^k_t$$

Summarizing all level of signature, we define the signature of path as:

$$S(X)_{a,b} = (1, S(X)^1_{a,b}, ..., S(X)^{1,1}, S(X)^{1,2}, ...)$$

The first n-level signature are call n level truncated signature:

$$S(X)^n_{a,b} = (1, S(X)^1_{a,b}, ..., S(X)^{1,1}, S(X)^{1,2}, ..., S(X)^{i_1,...,i_n})$$

2.1.3 Chen's identity

Chen's identity builds up a algebraic relation between path and signature. In simple words, concatenation of path equals to the tensor product of signature.

First we need to define formal power series. Let $e_1 ... e_n$ be d formal indeterminates and the algebra of non-commuting formal power series

$$\sum_{k=0}^{\infty} \sum_{i_1,...,i_k \in (1,..,d)} \lambda^{i_1,...,i_k} e_i ... e_d.$$

In our case, the $\lambda^{i_1,...,i_k}$ is $S(X)_{a,b}^{i_1,...,i_k} e_{i_1} ... e_{i_k}$. Therefore, the above equation becomes a representation of signature.

$$\sum_{k=0}^{\infty} \sum_{i_1,...,i_k \in (1,..,d)} S(X)_{a,b}^{i_1,...,i_k} e_{i_1} ... e_{i_k}$$

The algebra is naturally equipped with vector space structure

$$\lambda \sum_{k=0}^{\infty} \sum_{i_1,...,i_k \in (1,..,d)} S(X)_{a,b}^{i_1,...,i_k} e_{i_1} ... e_{i_k} + \sum_{k=0}^{\infty} \sum_{i_1,...,i_k \in (1,..,d)} S'(X)_{a,b}^{i_1,...,i_k} e_{i_1} ... e_{i_k} =$$
$$\sum_{k=0}^{\infty} \sum_{i_1,...,i_k \in (1,..,d)} (\lambda S(X)_{a,b}^{i_1,...,i_k} + S'(X)_{a,b}^{i_1,...,i_k}) e_{i_1} ... e_{i_k}$$

and the product $\otimes$

$$e_{i_1} ... e_{i_k} \otimes e_{j_1} ... e_{j_l} = e_{i_1} ... e_{i_k} e_{j_1} ... e_{j_l}.$$

Next, we define mathematically concatenation of path, $*$. For two path,$X : [a, b] \mapsto \mathbb{R}^d$ and $Y : [b, c] \mapsto \mathbb{R}^d$, we define their concatenation as the path$X*Y : [a, c] \mapsto \mathbb{R}^d$ for which $(X*Y)_t := X_t$ for $t \in [a, b]$ and $(X * Y)_t = X_b + (Y_t - Y_b)$ for $t \in [b, c]$.
Chen's identity is defined as follow:

$$S(X * Y)_{a,c} = S(X)_{a,b} \otimes S(Y)_{b,c}$$

2.2 Algorithm

Chen's identity can be used to calculate the signature of piecewise-linearly path efficiently. Since signature of a linear path $X : [a, b] \mapsto \mathbb{R}^n$ is easy to calculate:

$$S(X)_{a,b}^{i_1,...,i_k} = (X_b^{i_1} - X_a^{i_k})...(X_b^{i_k} - X_a^{i_1})/k!$$

We can calculate the n level truncated signature of all linear piece of the path and do outer product. Then according to Chen's identity, we get the n-level truncated signature concatenated path. We use 'iisignature' library in python for the calculation of signature of path.[RG18]

Chapter 3

Preparation

3.1 Dataset

Pendigit dataset has 7000 character sample in train set and 4000 in test set. Each sample consists of data streams of two dimension integer coordinate ranged from 0 to 500 and the label.The characters are integers from 0 to 9. Most of the data points lie on the 300 by 300 central area.

3.2 Model

Initially, we try to train DeepCNN model as illustrated in the paper,[Gra13] ,but the performance was much lower than expected.

$$c2 - p3 - c2 - p2 - c2 - p2 - c2 - p2 - d200 - d10 \ c = convolution, p = maxpool, d = dense$$

Therefore, we used a network similar to Lenet[LBBH01]. We call it Letnet in the following text. The 5 by 5 convolution kernel and large fully connected network improve the performance significantly.

$$c1 - p2 - c1 - p3 - c1 - p2 - d1024 - d10$$

3.3 Input

The input of CNN used in this project is in the shape of (?,50,50),where ? represents the number of layers. The first layer is 50 by 50 rescaled image of character. Since we will compare the model with different level of signature, $ depends on the level of signature.

3.4 Signature layer

Signature layer is constructed as follow:

1. Create a 50 by 50 empty image.
2. At time t, take $t \pm 3$ segment of the two-dimension path .
3. calculate n-level truncated signature and put it on the $t \pm 3$ position
4. t+1

The n-level truncated signature of d dimension has dimension $(n^{d+1}-1)/(d-1)$. Since $S^0(X)$ is conventionally 1, we ignore it and replace with character image layer, which means the dimension of input is $\$ = (n^{d+1} - 1)/(d - 1)$.

Chapter 4

Evaluation

Since the training process of model are time-consuming, we stop the training process when accuracy rate on test sample decrease less than 0.0001 for 1000 iteration. The accuracy rate at this time is regarded as the final score of this model.

4.1 DeepCNet vs Letnet

We run both models without using signature method. From the fig 4.4,we can see that Letnet performs much better than DeepCNet,up to 15 percent difference between the final score. Since DeepCNet has one convolution layer more than Letnet, it also needs more time to converge.

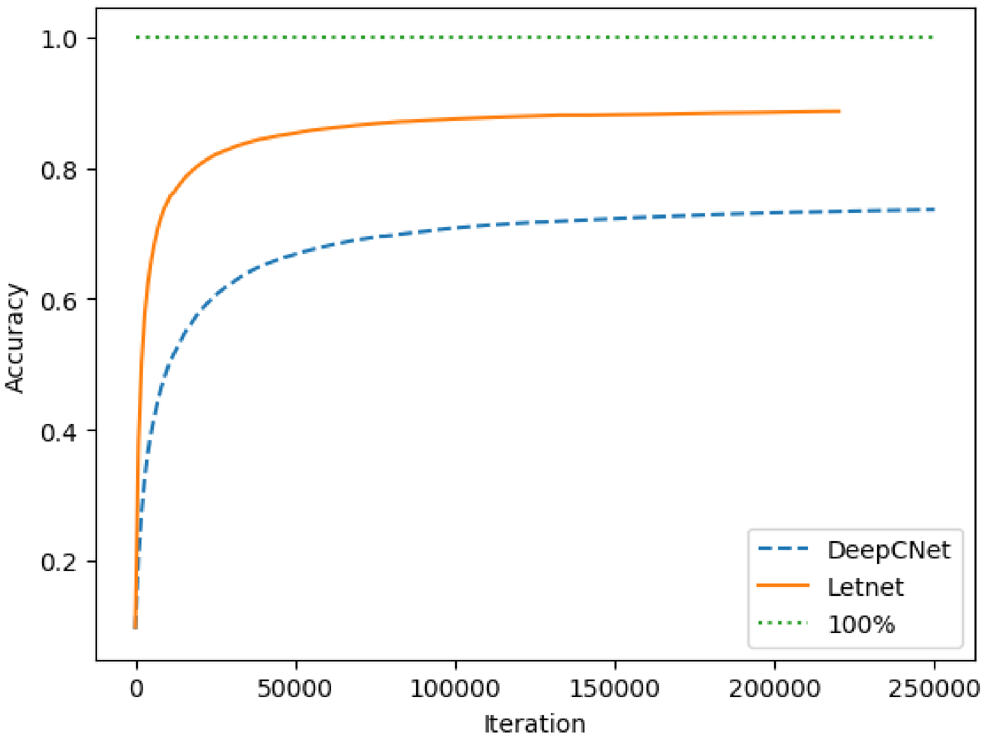
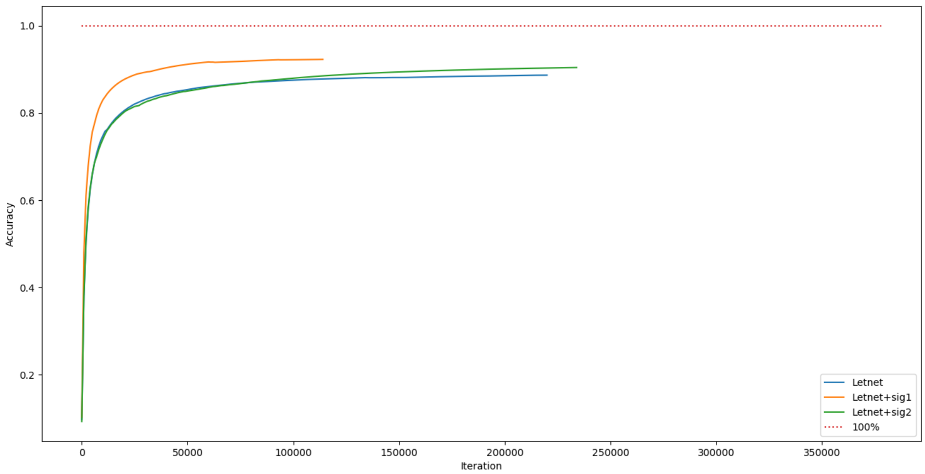


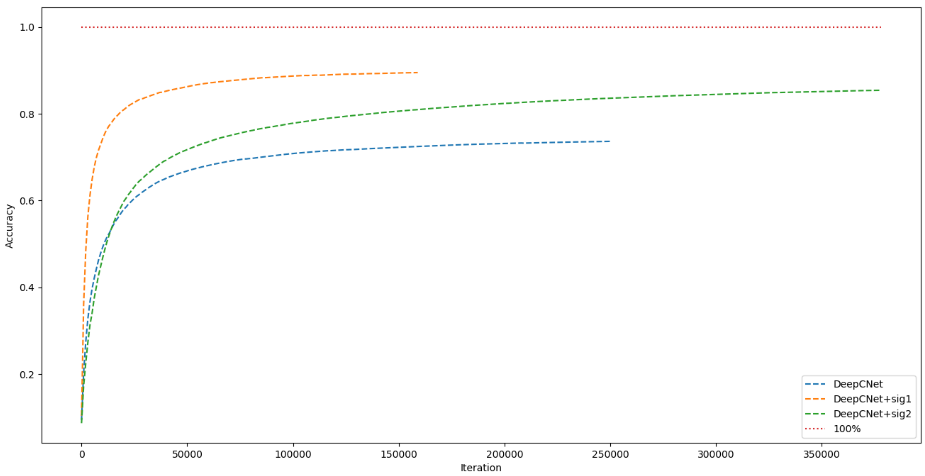
Figure 4.1: Letnet vs DeepCNet

4.2 Model result

In fig4.2(a)4.2(b), the blue represents the model without signature method. The orange and green lines represent 1-level and 2-leve truncated signature model respectively. Undoubtedly, the model with signature method has higher accuracy than the normal one. However,the model using 2-level truncated signature performed worse than 1-level one and took more iteration to converge. We can also view the difference from the table 4.1. By running on two different model, we avoid the possibility that such an unexpected result is due to specific model.



(a) LetNet



(b) DeepCNet

Figure 4.2: Letnet and DeepCNet

	standard	sig1	sig2
LetNet	0.8840	0.9218	0.9031
DeepCNet	0.7361	0.8944	0.8535

Table 4.1: Final score

4.3 Low quality of higher level of signature

For the term 'low quality', we mean the features is weakly related to the result and has negative effect on converging process. The figure4.3 gives a clear evidence that higher level signatures do no good on the model.

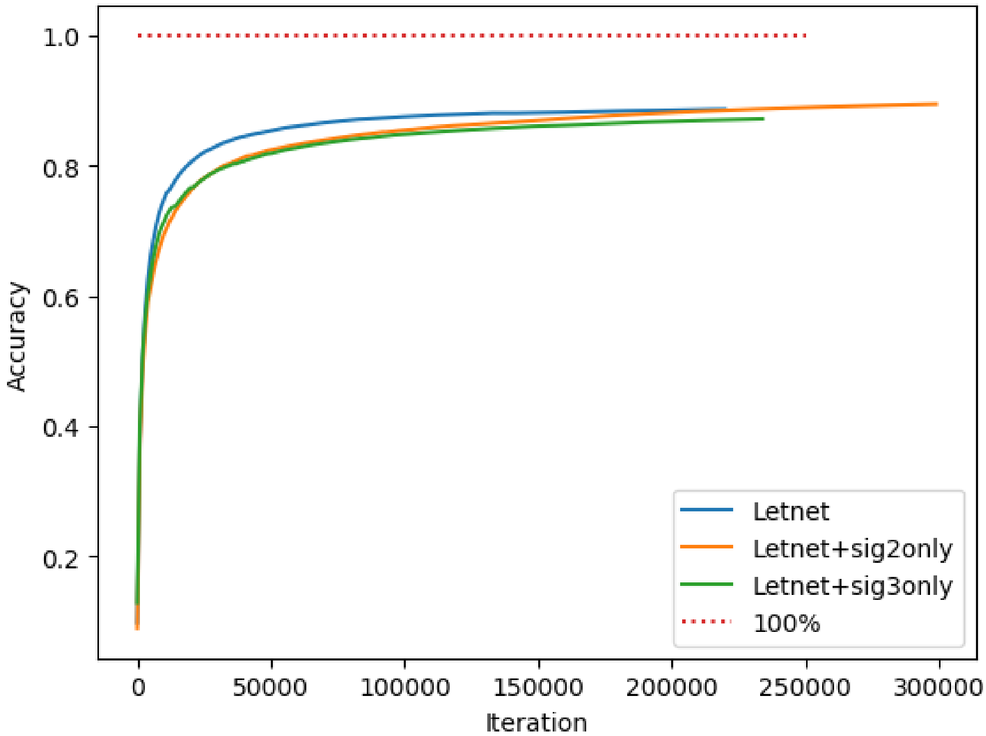


Figure 4.3: Letnet vs DeepCNet

4.4 Evaluation

To explain such a result 4.1, we first evaluate the factors that affect the final score. In our case, since we use the same model structure, only two factors change, the number of parameters and the features used. In usual, increasing the number of parameters increase the fitting ability of the model,while the correlation between the features and the result determine the speed of converge and partially affect the final score. Therefore we deduce: Higher level truncated signature increases the score by increasing the number of parameters in the first layer of model. However, except the first level of signature, which has obvious geometric property(Displacement), higher level signature is weakly relative to the result which slows down the converging process. In other words, due the low quality of sig2(the second level of signature), the model needs much more iterations to get the same score as the model with sig1. As for the possible maximum score, the model with sig2 may performance better than sig1 due the number of parameter.[YJL15][Gra13]

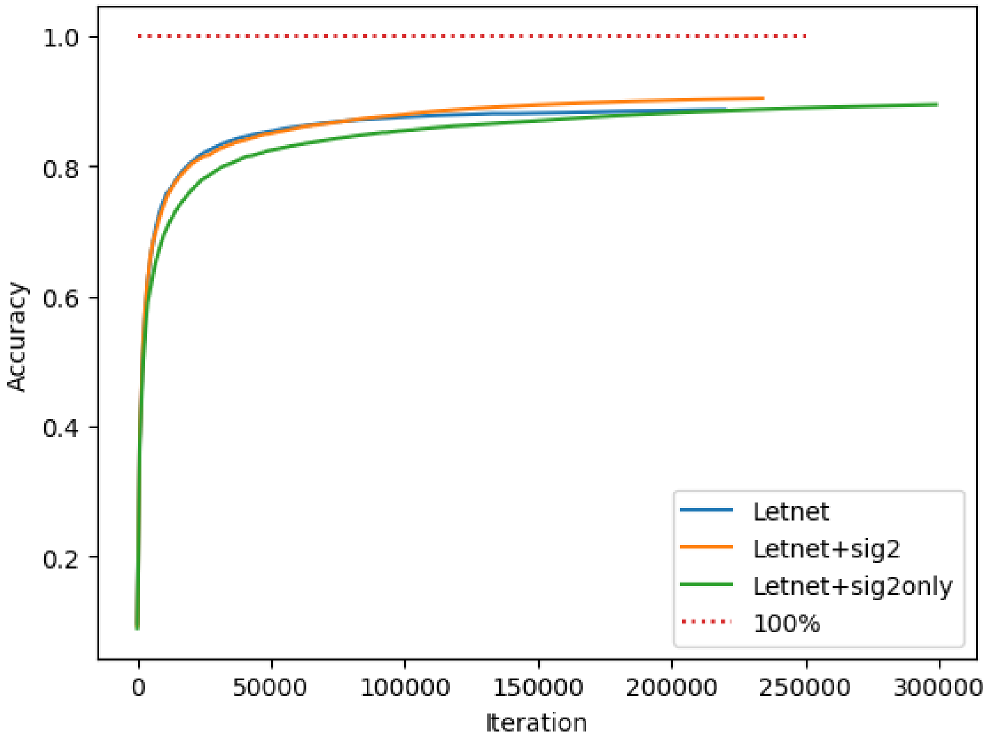


Figure 4.4: Letnet vs DeepCNet

Appendix A

Appendix

A.1 All in one

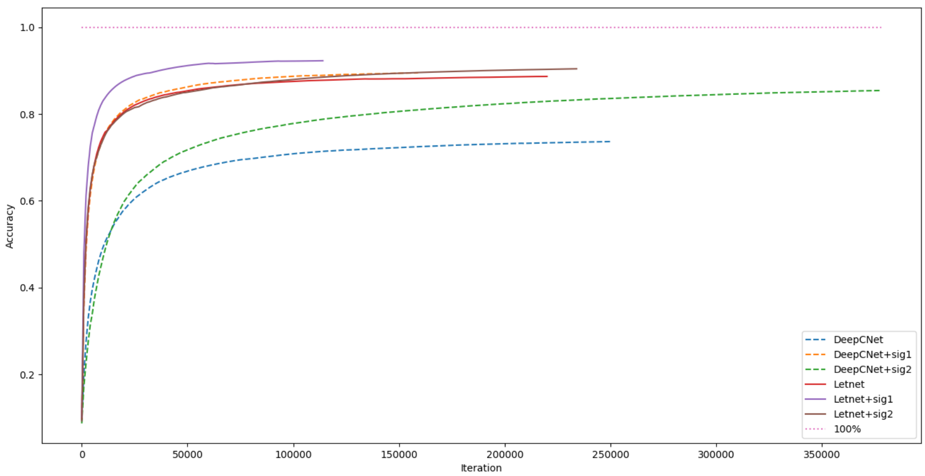


Figure A.1: Letnet vs DeepCNet

Bibliography

[CK16] Ilya Chevyrev and Andrey Kormilitzin. A primer on the signature method in machine learning, 2016.

[Gra13] Benjamin Graham. Sparse arrays of signatures for online character recognition, 2013.

[LBBH01] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Intelligent Signal Processing*, pages 306–351. IEEE Press, 2001.

[RG18] Jeremy Reizenstein and Benjamin Graham. The iisignature library: efficient calculation of iterated-integral signatures and log signatures, 2018.

[YJL15] Weixin Yang, Lianwen Jin, and Manfei Liu. Deepwriterid: An end-to-end online text-independent writer identification system, 2015.