华南理工大學

DLVCLAB

Accelerating and Compressing LSTM based Model for Online Handwritten Chinese Character Recognition

Reporter: Zecheng Xie

South China University of Technology

August 5th , 2018

Outline

- Motivation
- Difficulties
- Our approach
- Experiments
- Conclusion

Motivation

- Online handwritten Chinese character recognition (HCCR) is widely used in pen input devices and touch screen devices

DLVCLAB

Motivation

- The difficulties of online HCCR
  - Large number of character classes
  - Similarity between characters
  - Diversity of writing styles
- Deep learning models are powerful but raise other problems
  - Models are too large → require large footprint and memory
  - Computational expensive → consume much energy
- The advantages of deploying models on mobile devices
  - Ease server pressure
  - Better service latency
  - Can work offline
  - Privacy protection
  - …

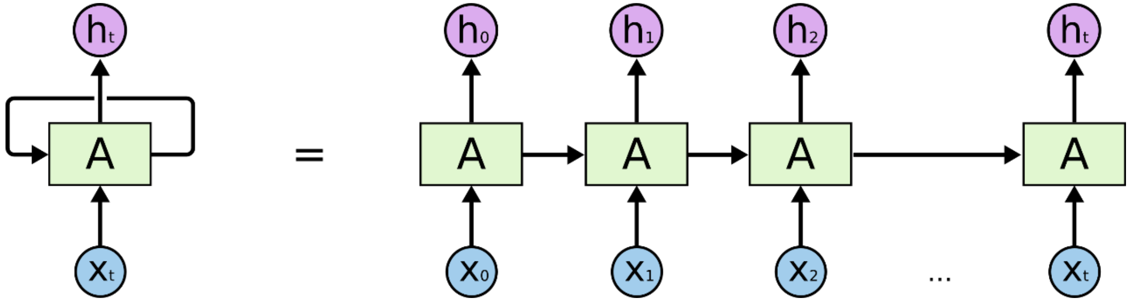
Our goal: build fast and compact models for on-device inference

DLVCLAB

Difficulties of deploying LSTM based online HCCR models on mobile devices

- 3755 classes
  - Model tends to be large

- Dependences between time steps
  - Make the inference slow
  - Nature of RNNs, unlikely to be changed



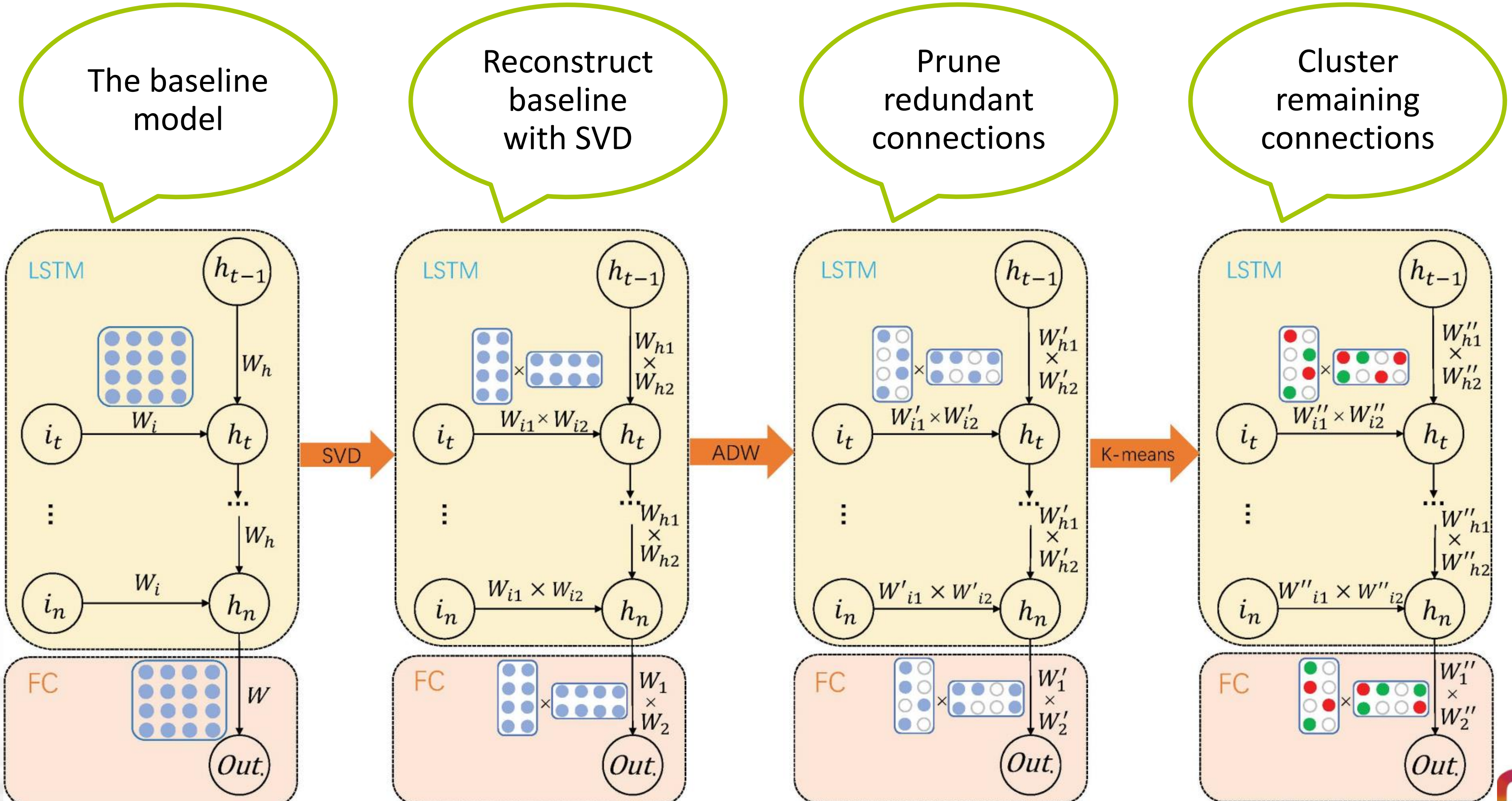
Unroll of RNN [1]

[1] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

DLVCLAB

Our approach

- The proposed framework



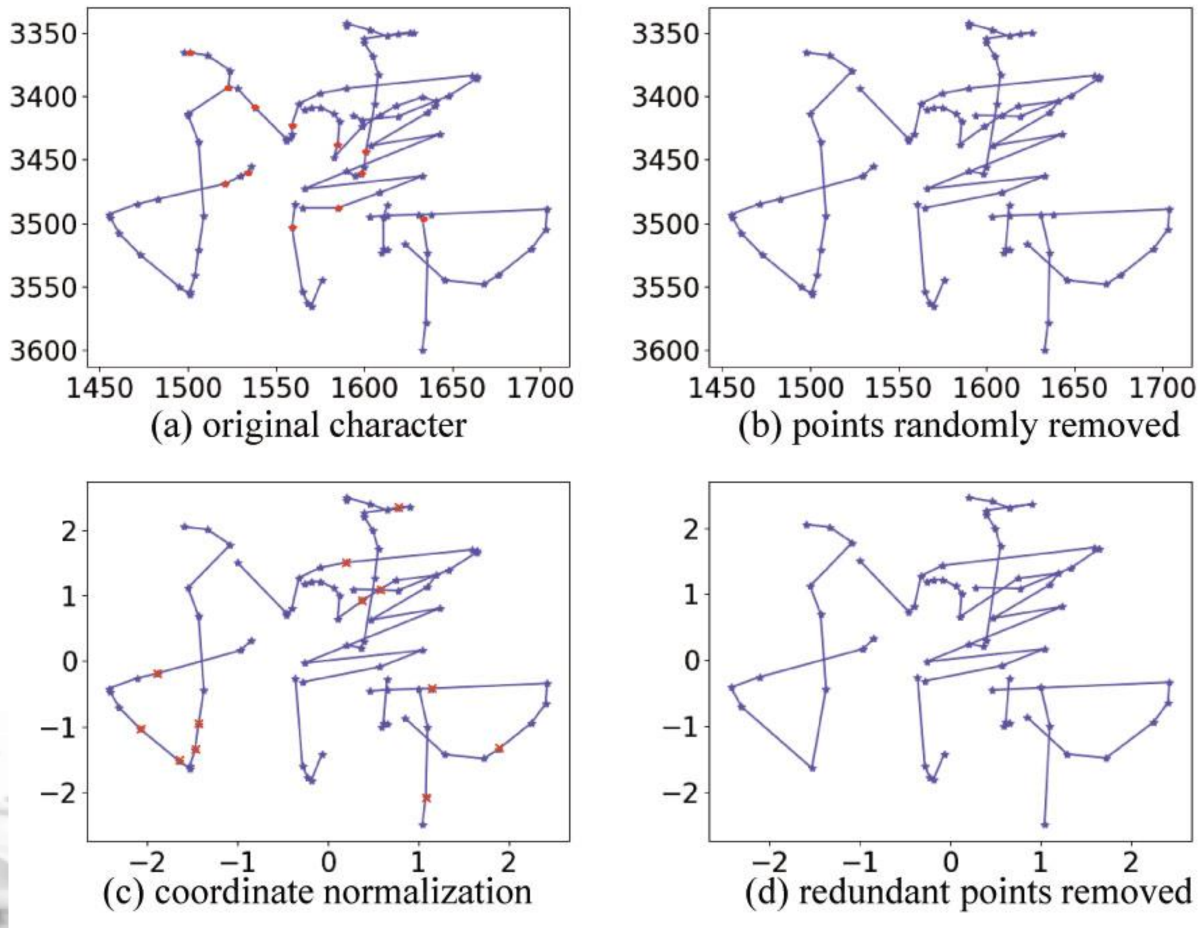
Our approach

- Data preprocessing and augmentation
  - Randomly remove 30% of the points in each character
  - Perform coordinate normalization
  - Remove redundant points using method proposed in [1]
    - Point that is too close to the point before it
    - Middle point that nearly stands in line with the two points before and after it
  - Data transform & feature extraction[1]
    - $[[x_i, y_i, s_i]], \quad i = 1, 2, 3, \ldots$
    - $[[x_i, y_i, \Delta x_i, \Delta y_i, (s_i = s_{i+1}), (s_i \neq s_{i+1})]], \quad i = 1, 2, 3, \ldots$

[1] X.-Y. Zhang et al., "Drawing and recognizing Chinese characters with recurrent neural network", TPAMI, 2017

Our approach

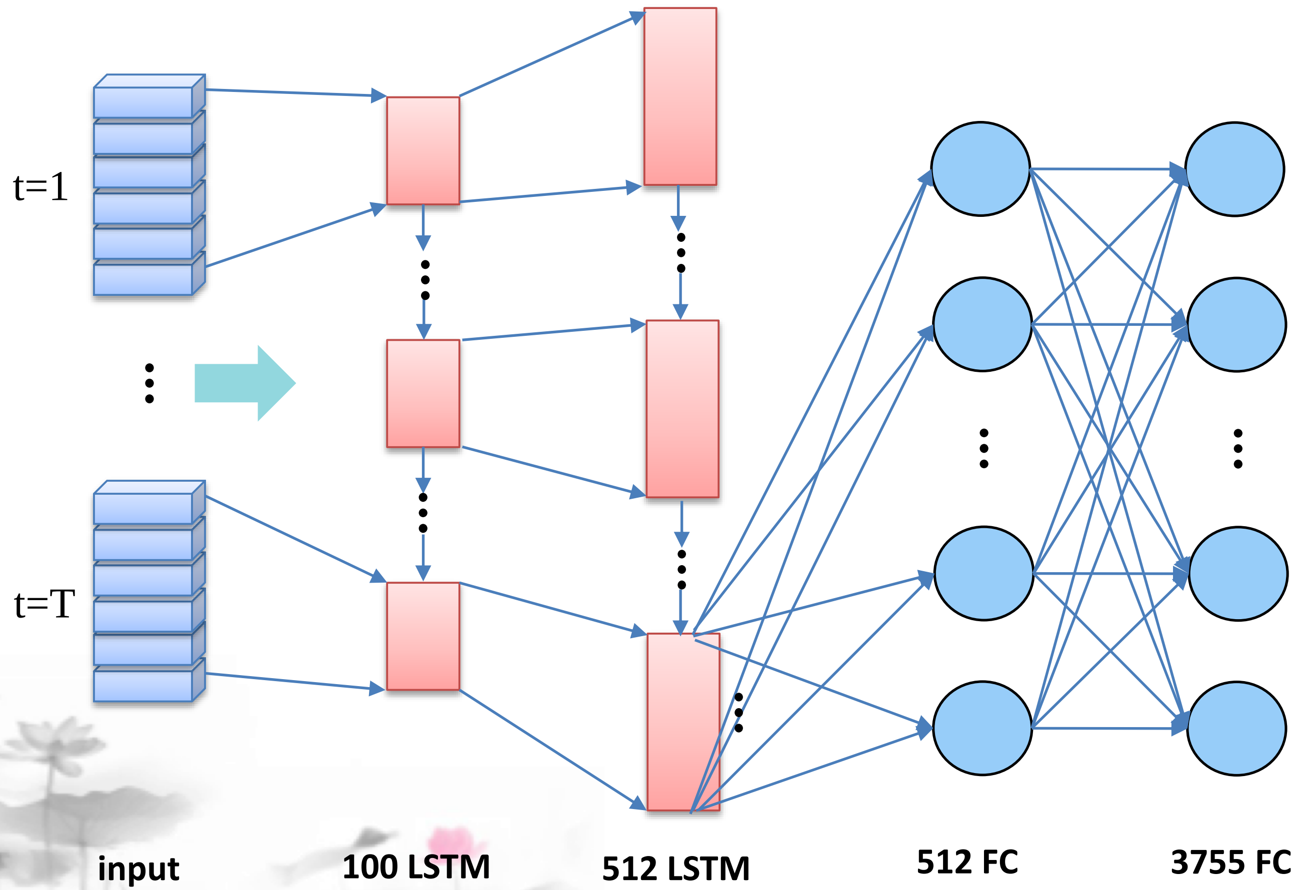
- Data preprocessing and augmentation



[1] X.-Y. Zhang et al., "Drawing and recognizing Chinese characters with recurrent neural network", TPAMI, 2017

Our approach

- Baseline model architecture
  - Input-100LSTM-512LSTM-512FC-3755FC-Output



DLVCLAB

Our approach

- Reconstruct network with singular value decomposition (SVD)

$$i_t = \sigma(W_{ii}x_t + W_{hi}h_{t-1} + b_i)$$
$$f_t = \sigma\left(W_{if}x_t + W_{hf}h_{t-1} + b_f\right)$$
$$g_t = \tanh\left(W_{ig}x_t + W_{hg}h_{t-1} + b_g\right)$$
$$o_t = \sigma(W_{io}x_t + W_{ho}h_{t-1} + b_o)$$
$$c_t = f_t * c_{t-1} + i_t * g_t$$
$$h_t = o_t * \tanh(c_t)$$

Main computation

$$\begin{bmatrix} i_t \\ f_t \\ g_t \\ o_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \tanh \\ \sigma \end{bmatrix} * \left(\begin{bmatrix} W_{ii} \\ W_{if} \\ W_{ig} \\ W_{io} \end{bmatrix} x_t + \begin{bmatrix} W_{hi} \\ W_{hf} \\ W_{hg} \\ W_{ho} \end{bmatrix} h_{t-1} + \begin{bmatrix} b_i \\ b_f \\ b_g \\ b_o \end{bmatrix} \right)$$

DLVCLAB

Our approach

- Reconstruct network with singular value decomposition (SVD)

$$\begin{bmatrix} i_t \\ f_t \\ g_t \\ o_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \tanh \\ \sigma \end{bmatrix} * \left(\underbrace{\begin{bmatrix} W_{ii} \\ W_{if} \\ W_{ig} \\ W_{io} \end{bmatrix} x_t}_{W_i x_t} + \underbrace{\begin{bmatrix} W_{hi} \\ W_{hf} \\ W_{hg} \\ W_{ho} \end{bmatrix} h_{t-1}}_{W_h h_{t-1}} + \begin{bmatrix} b_i \\ b_f \\ b_g \\ b_o \end{bmatrix} \right)$$

- Apply SVD to W_i and W_h
  - W_i: input connections
  - W_h: hidden-hidden connections

Our approach

- Efficiency analysis of SVD method
  - Suppose $W \in \mathbb{R}^{m\times n}$, by SVD we have

$$W_{m\times n} = U_{m\times n}\Sigma_{n\times n}V_{n\times n}^T$$

  - By reserving proper number of singular values

$$W_{m\times n} \approx U_{m\times r}\Sigma_{r\times r}V_{n\times r}^T = U_{m\times r}N_{r\times n}$$

  - Replace $W_{m\times n}$ with $U_{m\times r}N_{r\times n}$
    - $Wx \rightarrow UNx$

Our approach

- Efficiency analysis of SVD method

  - For a matrix-vector multiplication Wx, $W \in \mathbb{R}^{m\times n}$, $x \in \mathbb{R}^{n\times 1}$, the acceleration rate and compression rate with *r* singular values reserved is given by

$$R_a = R_c = \frac{mn}{mr + rn}$$

  - If $m = 512, n = 128, r = 32$, then $R_a = R_c = 3.2$

Our approach

- Adaptive drop weight (ADW) [1]
  - Improvement on “Deep Compression” [2] in which a hard threshold is set
  - ADW gradually prunes away redundant connections in each layer, which have small absolute values (by sort them during retraining)

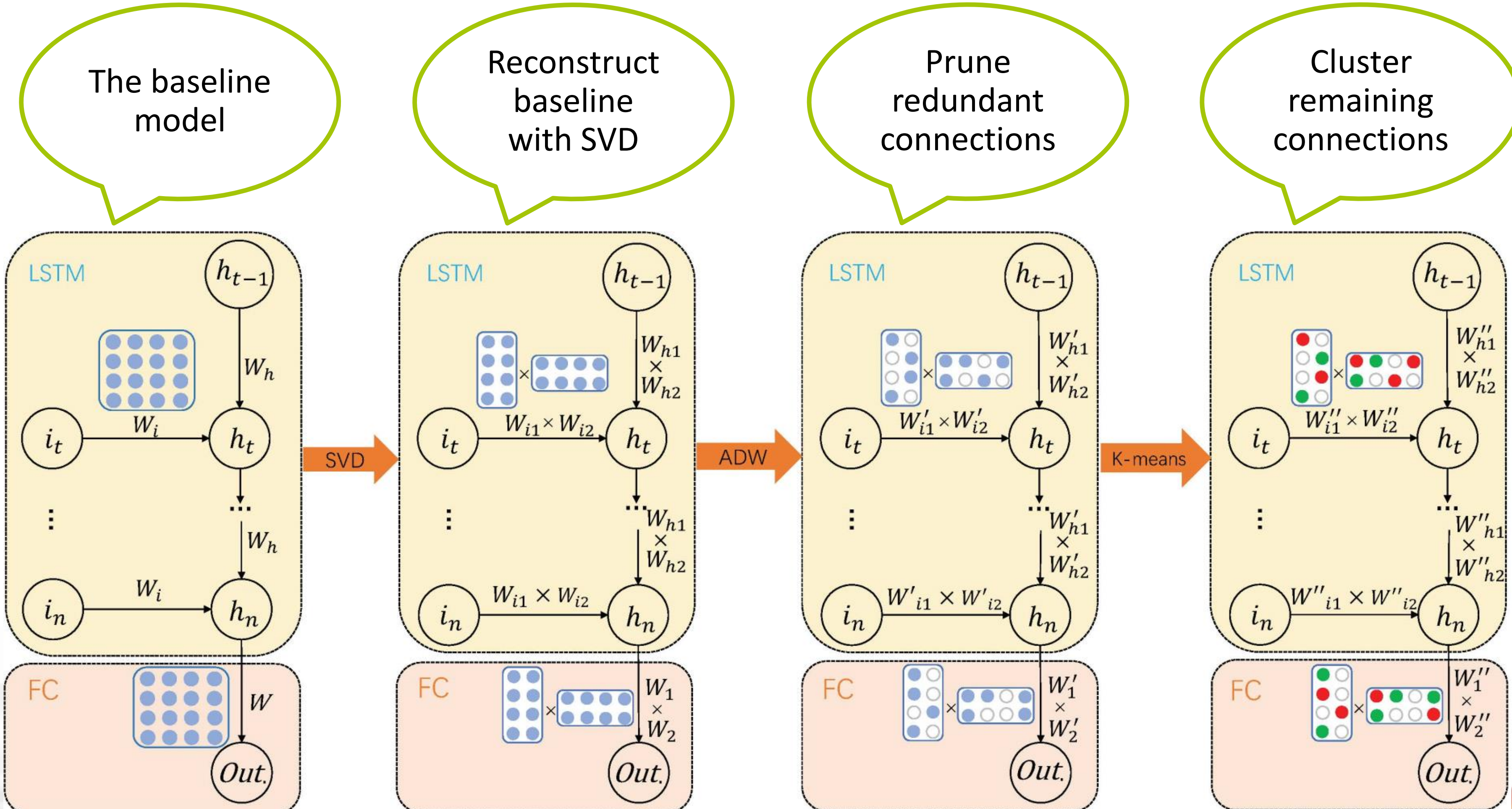
- After ADW, the network become sparse, K-means based quantization is applied to each layer to gain further compression

[1] X. Xiao, L. Jin, et al., “Building fast and compact convolutional neural networks for offline handwritten Chinese character recognition”, Pattern Recognition, 2017

[2] S. Han,et al., “Deep compression: compressing deep neural network with pruning, trained quantization and Huffman coding”, ICLR, 2016

Our approach

- The proposed framework - review



Experiments

- Training set
  - CASIA OLHWDB1.0 & OLHWDB1.1
  - 720 writers, 2,693,183 samples, 3755 classes

- Test set
  - ICDAR2013 online competition dataset
  - 60 writers, 224,590 samples, 3755 classes

- Data preprocessing and augmentation as mentioned before

Experiments

- Details of the baseline model

Layer	#Output	Param. matrix shape		#Param. ($\times 10^3$)	FLOPs ($\times 10^8$)
LSTM1	100	W_i W_h	400×6 400×100	2.4 40	0.06
LSTM2	512	W_i W_h	2048×100 2048×512	204.8 1048	1.88
FC1	512	W	512×512	262.1	0.003
FC2	3755	W	3755×512	1922	0.019

- Main storage cost: LSTM2, FC1, FC2
- Main computation cost: LSTM2

Experiments

- Experimental settings

Layer	Param. matrix	SVD setting (r)	ADW pruning ratio (%)	#Cluster centroids	#Quant. bits
LSTM1	W_i	4	0	-	-
	W_h	16	0	-	-
LSTM2	W_i	32	10	256	8
	W_h	16	20	256	8
FC1	W	32	20	256	8
FC2	W	32	30	256	8

- Consideration of the experimental settings
  - In our experiments, we found LSTM is more sensitive to input connections than hidden-hidden connections
  - Most computation latency is introduced by hidden-hidden connections

Experiments

- Experimental results
  - Intel Core i7-4790, single thread

Method	Storage (MB)	FLOPs ($\times 10^8$)	Accuracy (%)
baseline	14	1.9	97.83
+SVD	1.2	0.18	97.37
+ADW & quant.	0.45	0.14	97.33

- After SVD, model is 10 × smaller, and FLOPs is also reduced by 10 ×
- After ADW & quantization, model is 31 × smaller, and FLOPs is further reduced
- A minor 0.5% drop of accuracy

DLVCLAB

Experiments

- Experimental results

Method	Ref.	Storage (MB)	Speed (ms)	Accuracy (%)
Traditional Benchmark	[4]	120.0	-	95.31
ICDAR-2011 Winner	[13]	41.62	-	95.77
ICDAR-2013 Winner	[14]	37.80	-	97.39
DropSample	[11]	135	12	97.51
DirectMap+Convnet	[15]	23.50	295.03	97.91
RNN-NET4	[16]	10.38	-	97.76
Ensemble-NET123456	[16]	78.11	-	**98.15**
DropDistortion	[18]	19.03	-	97.79
HCCR-GAP-Pruned	[21]	0.57	-	96.88
Baseline	ours	14	12	97.83
SVD+ADW+Quant.	ours	**0.45**	**2.7**	**97.33**

- Compared with [11], our model is $300\times$ smaller and $4\times$ faster on CPU
- Compared with [15], our model is $52\times$ smaller and $109\times$ faster on CPU

[1] W. Yang, L. Jin, et al., "Dropsample: A new training method to enhance deep convolutional neural networks for largescale unconstrained handwritten Chinese character recognition", Pattern Recognition, 2016

[2] X.-Y. Zhang, et al., "Online and offline handwritten Chinese character recognition: A comprehensive study and new benchmark", Pattern Recognition, 2017

DLVCLAB

Conclusion

- SVD is efficient for accelerating computation
- ADW also works well for LSTMs
- By combining SVD and ADW, we can build fast and compact LSTM based model for online HCCR

Thank you!

Lianwen Jin(金连文), Ph.D, Professor

eelwjin@scut.edu.cn   lianwen.jin@gmail.com

Zecheng Xie(谢泽澄), Ph.D, student

Yafeng Yang(杨亚锋), Master, student

http://www.hcii-lab.net/