Irrelevant Variability Normalization via Hierarchical Deep Neural Networks for Online Handwritten Chinese Character Recognition

Jun Du
University of Science and Technology of China

ICFHR 2014, Crete, Greece, September 1-4, 2014

Background

- Popular input mode on mobile devices in China

- Solved problem?
  - More and more diversified real data from users
  - How to further improve the recognition accuracy?
    - Writer adaptation
    - Designing a more robust character classifier

Irrelevant Variability Normalization (IVN)

- A general concept for pattern recognition problem
  - Remove any variabilities irrelevant to the content
- First proposed in speech recognition area (1999)
  - Speaker variability (SAT: Speaker Adaptive Training, 1996)
  - Environment variability (NAT: Noise Adaptive Training, 2000)
  - RDT: Region-Dependent Transformation (2006)
- Related work in handwriting recognition area
  - WAT: Writer Adaptive Training (2009) and RDT (2012)
  - Style Normalized Transformation (2011)
  - IVN based feature transformation (2013)

Linear or piecewise linear transformations!

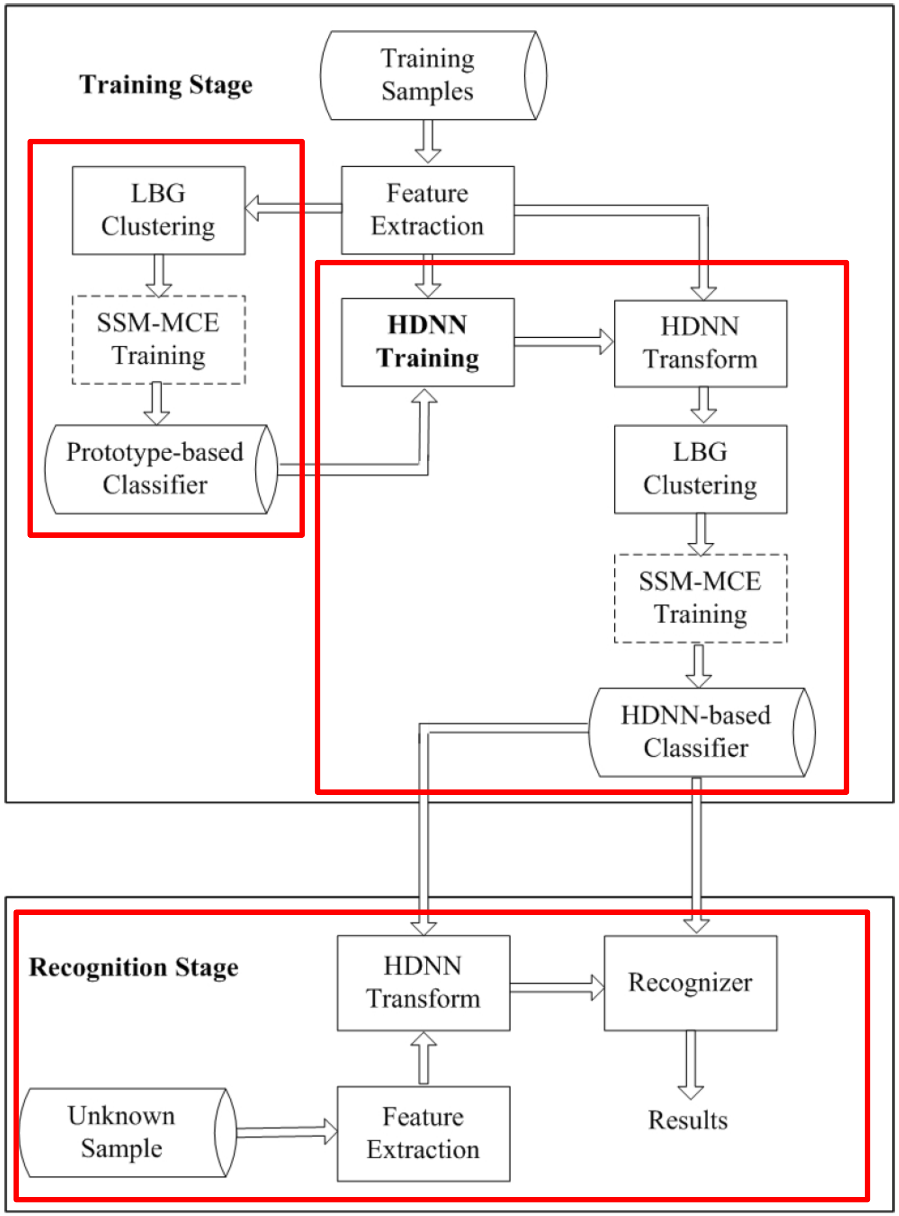
Core Innovations

- Hierarchical Deep Neural Network (HDNN)
  - Extension from DNN for regression problem
  - A novel architecture focusing on both “depth” and “width”

- HDNN as a highly nonlinear feature transformation
  - Incorporate with multi-prototype based classifier
  - Application for Chinese handwriting recognition

System Overview

- Baseline classifier
  - LBG Clustering
  - SSM-MCE training
- HDNN-based classifier
  - HDNN training
  - Classifier training
- Online recognition
  - HDNN transform



SSM-MCE training

- Classification with discriminant functions

$$r(\mathbf{x};\mathbf{\Lambda}) = \arg\max_i g_i(\mathbf{x};\lambda_i)$$

$$g_i(\mathbf{x};\lambda_i) = -\min_k \parallel \mathbf{x} - \mathbf{m}_{ik} \parallel^2$$

- Minimum Classification Error (MCE) criterion

$$l(\mathcal{X};\mathbf{\Lambda}) = \frac{1}{R}\sum_{r=1}^{R} \frac{1}{1+\exp[-\alpha d(\mathbf{x}_r;\mathbf{\Lambda})+\beta]}$$

- Misclassification measure
  - Sample Separation Margin (SSM)

$$d(\mathbf{x}_r;\mathbf{\Lambda}) = \frac{-g_p(\mathbf{x}_r;\lambda_p) + g_q(\mathbf{x}_r;\lambda_q)}{2 \parallel \mathbf{m}_{p\hat{k}} - \mathbf{m}_{q\bar{k}} \parallel}$$

IVN-based Feature Transformation

- Feature transformation
  - Normalizing the irrelevant variabilities in handwritten samples

$$\mathbf{x}_r^{\text{ivn}} = \mathcal{F}(\mathbf{x}_r; \mathbf{\Theta})$$

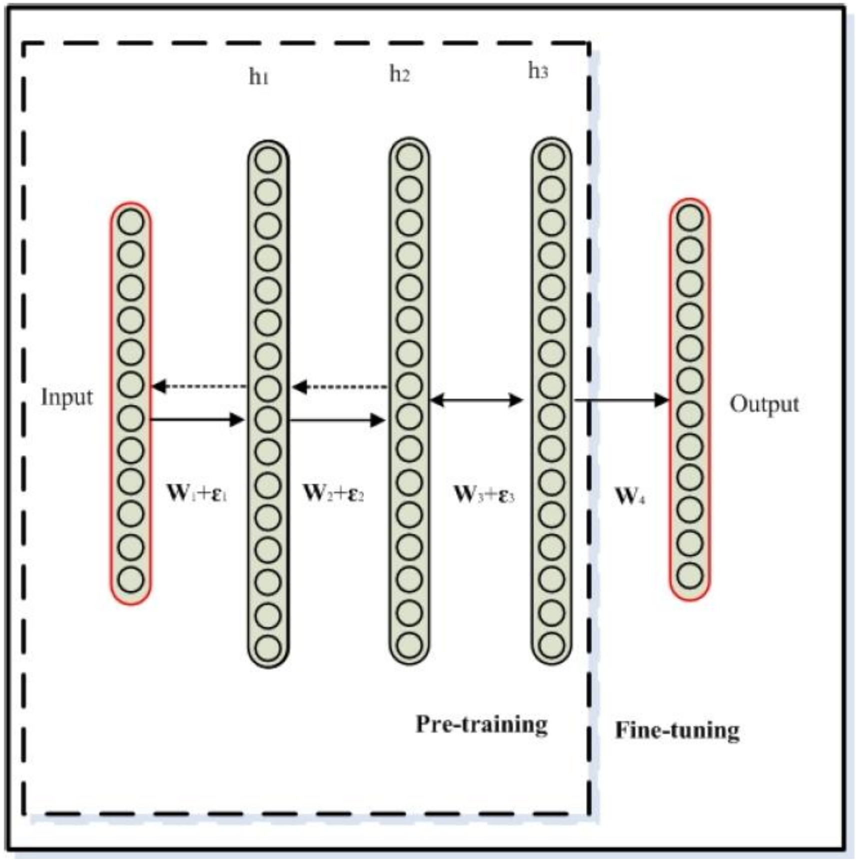
- Objective function for parameter learning
  - Minimizing the Euclidean distance between the IVN transformed feature vector and the prototype of the reference class

$$E = \frac{1}{R} \sum_{r=1}^{R} \|\mathbf{x}_r^{\text{ivn}} - \mathbf{x}_r^{\text{ref}}\|_2^2$$

- Specific forms of transformation function
  - DNN
  - HDNN

DNN Training

- Hinton's recipe
  - Layer-by-layer RBM pre-training
  - Supervised fine-tuning



Why HDNN

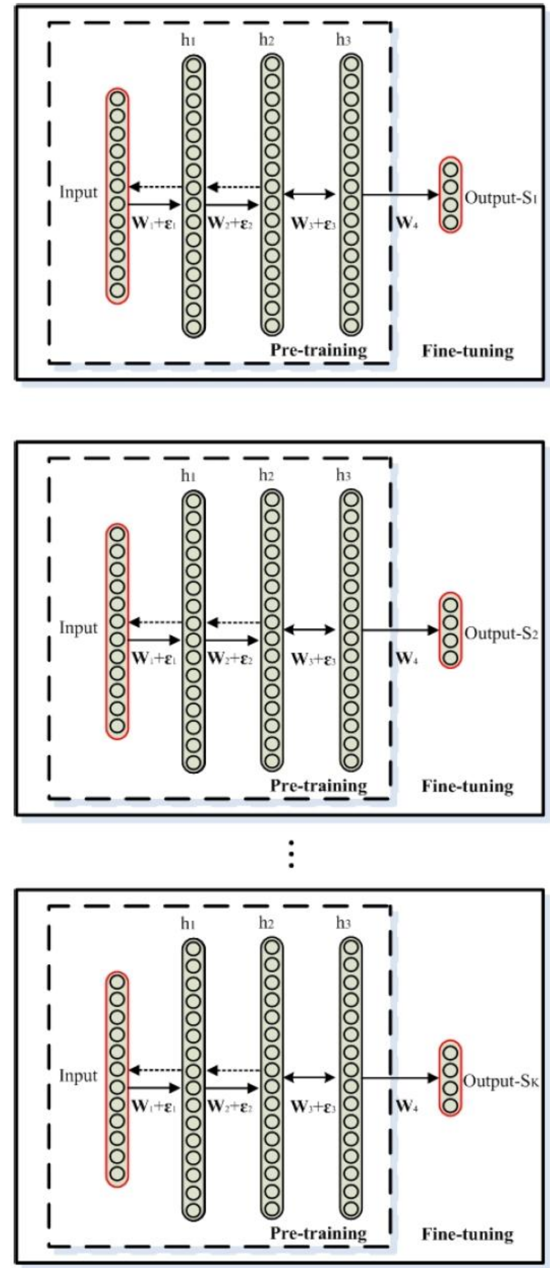
- DNN is widely used for classification
- DNN might be failed for regression as
  - Unbounded output
  - Highly nonlinear relationship between input and output
  - High dimension for both input and output
- HDNN: divide and conquer
  - Divide the output vector into *K* subvectors
  - Learning is relatively easy between input and each subvector

$$
\begin{aligned}
E &= \frac{1}{R}\sum_{r=1}^{R}\|\mathbf{x}_r^{\mathrm{ivn}} - \mathbf{x}_r^{\mathrm{ref}}\|_2^2 = \frac{1}{R}\sum_{r=1}^{R}\sum_{k=1}^{K}\|\mathbf{x}_{r,k}^{\mathrm{ivn}} - \mathbf{x}_{r,k}^{\mathrm{ref}}\|_2^2 \\
&= \sum_{k=1}^{K} E_k
\end{aligned}
$$

HDNN Training

- HDNN is both deep and wide
- Training of *K* subnets
  - Share the same pre-training as DNN
  - Fine-tuning for each subnet

- Implementation issues
  - How to design *K*
- Input is LDA transformed feature vector
  - Only transform first *M* dimension in output
  - The remaining *D-M* dimensions are noisy



Experimental Setup

- CASIA benchmark
  - Vocabulary: 3926 character classes
  - Training: totally 939561 samples
  - Test: totally 234798 samples
- Feature extraction
  - 512-dimensional raw feature: 8-directional features
  - LDA transformation: 512 -> 128
- Configurations for DNN and HDNN
  - 1024 nodes for each hidden layer of DNN and HDNN subnets
  - M is set as 48

DNN vs. HDNN

- DNN underperforms baseline even using deep layers
  - The mean square error of DNN can not be small enough
  - Even on the training set

- HDNN significantly outperforms baseline

Table 1. Performance (character error rate in %) comparison of different systems prototype-based classifiers with LBG clustering on the testing set.

Methods	Baseline	DNN-1L	DNN-2L	DNN-3L	HDNN-1L	HDNN-2L
CER(%)	16.13	29.26	23.30	25.63	13.44	12.37

HDNN with Different Configurations

- HDNN always achieves better performance with the same
  - Prototype setting
  - Training criterion for classifier

Table 2. Performance (character error rate in %) comparison of systems using prototype-based classifiers with different features and different training criteria on the testing set.

	#prototype	LBG	SSM-MCE
Baseline	1	16.13	12.26
	4	13.68	11.64
HDNN (LBG)	1	12.37	11.64
	4	11.84	11.32
HDNN (SSM-MCE)	1	11.38	10.82
	4	10.96	10.61

Summary and Future Work

- HDNN can potentially outperform DNN in the case of
  – Unbounded regression problem
  – Highly nonlinear relationship between input and output
  – High dimension for both input and output

- Future work
  – Improve HDNN training by designing better objective function
  – Incorporate with deep learning based classifiers