Internet of Things and Deep Learning

Mingxing Duan, Kenli Li, and Keqin Li

1 Introduction

1.1 The Era of Big Data

This is the era of big data. Big data are not only meaning large volume but also including velocity, variety, veracity. With the rapid development of science and technology, we are surrounded by the amount of structured and unstructured data. Big data contains text, image, video, and other forms of data, which are collected from multiple datasets, and are explosively increasing in size and getting more complexity in context. It has aroused a large amount of researchers from different area and it affects our lives. Specially, machine learning have developed into an effective method to deal with big data for mining a valuable information. Deep learning is a hot research area and has facilitated our lives.

M. Duan (✉)
Collaborative Innovation Center of High Performance Computing, National University of Defense Technology, Changsha, Hunan, China
e-mail: duanmingxing16@nudt.edu.cn

K. Li
College of Computer Science and Electronic Engineering, Hunan University, Changsha, Hunan, China

K. Li
Department of Computer Science, State University of New York, New Paltz, NY, USA
e-mail: lik@newpaltz.edu

R. Ranjan et al. (eds.), *Handbook of Integration of Cloud Computing, Cyber Physical Systems and Internet of Things*, Scalable Computing and Communications,
https://doi.org/10.1007/978-3-030-43795-4_6

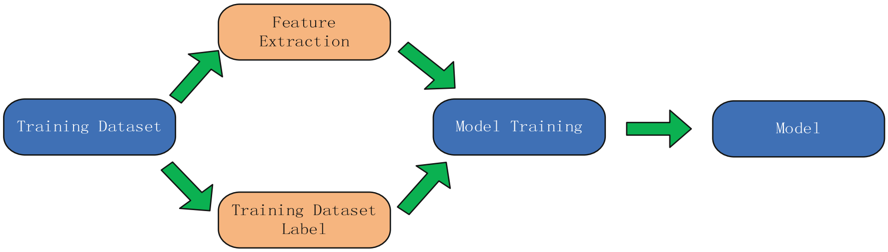

Fig. 1 The process of supervised learning algorithms

1.2 Supervised Learning Algorithms

Roughly speaking, supervised learning algorithms need training dataset which have corresponding label. Through these dataset, we can train the model. For example, given a training set **x** and corresponding **y**, training the model is through the loss that summing the deviation of real output and ideal output. However, in real world, the corresponding labels are difficult to collect and usually are provided by a human. Figure 1 present the process of supervised learning algorithms.

A large amount of supervised learning algorithms have been widely used, such as probabilistic supervised learning, support vector machine, k-nearest neighbors, and so on. These methods should be trained using corresponding labels and the well-tuned models used to predict the test tasks.

1.3 Unsupervised Learning Algorithms

Usually, an unsupervised learning algorithm is trained with the unlabel training dataset and there are no distinction between supervised learning algorithms and unsupervised learning algorithms by distinguishing whether the value is a feature or a label. In general, an unsupervised learning algorithm tries to extract information from a distribution without labels and these process includes density estimation, learning to extract information from a distribution, and clustering the data into groups of related examples. Figure 2 presents the process of supervised learning algorithms.

Many supervised learning algorithms have been used successfully, such as, Principal Components Analysis (PCA), k-means Clustering, Stochastic Gradient Descent (SGD), and so on. We can build a machine learning algorithm to deal with tasks with a training dataset, a cost function, and a optimization model. Deep learning methods are similar to these process and using the well-tuned model, we can test our testing dataset.

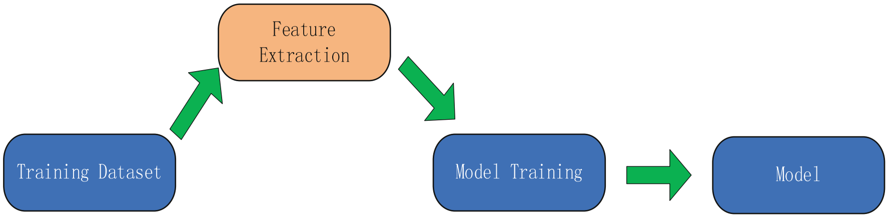

Fig. 2 The process of unsupervised learning algorithms

1.4 Common Deep Learning Models

Recently, a large amount of deep learning models have been successfully used in video surveillance, internet access control, security, and demography, such as CNN, RNN, GAN, ELM, and so on. We will simple present the ELM and CNN.

1.4.1 Extreme Machine Learning Model

ELM was first proposed by Huang et al. [28], which was used to process regression, and classification based on single-hidden layer feed forward neural networks (SLFNs). Huang et al. [29] pointed out that the hidden layer of SLFNs, which needed not be tuned is the essence of ELM. Liang et al. [40] proved that ELM had good generalization performance, fast and efficient learning speed. Simple and efficient learning steps with increased network architecture (I-ELM), and fixed-network architecture (ELM) were put forward by Huang et al. [27, 28]. In the two methods, the parameters of a hidden node were generated randomly, and training was performed only at the output layer, which reduced the training time. Zhou et al. [65] suggested that the testing accuracy increased with the increasing number of hidden layer nodes.

In order to reduce the residual errors, Feng et al. [11] proposed a dynamic adjustment ELM mechanism (DA-ELM), which could further tune the input parameters of insignificant hidden nodes, and they proved that it was an efficient method. Other improved methods based on ELM are proposed, such as enhanced incremental ELM (EI-ELM) [26], optimally pruned ELM (OP-ELM) [45], error minimized ELM (EM-ELM) [10], meta-cognitive ELM [52] and so on. Most of the methods mentioned above improved the ELM performance by decreasing the residual error of NN (Neural Network) to zero. When we use the above methods to process relatively small big data classification, and regression problems, they show good performance, fast and efficient learning speed. However, as the dataset getting larger and larger, serial algorithms cannot learn such massive data efficiently.

He et al. [22] first proposed the parallel ELM for regression problems based on MapReduce. The essential of the method was how to parallelly calculate the generalized inverse matrix. In PELM method, two MapReduce stages were used to compute the final results. Therefore, there were lots of I/O spending and communication costs during the two stages, which increased the runtime of ELM based on MapReduce framework. Comparing with PELM, Xin et al. [58, 59] proposed ELM* and ELM-Improved algorithms, which used one MapReduce stage instead of two and reduced the transmitting cost, even enhanced the processing efficiency. However, They needed several copies for each task when MapReduce worked, and if one node could not work, the tasks in this node would be assigned to other nodes, and re-processed again, leading to more costs during the process. Even more, lots of I/O overhead and communication costs were spent in the map and reduce stages, which reduced the learning speed and efficiency of the system.

1.4.2 Deep Neural Network System

CNN has been successfully applied to various fields, and specially, image recognition is a hot research field. However, few researchers have paid attention on hybrid neural network. Lawrence et al. [36] presented a hybrid neural-network solution for face recognition which made full use of advantages of self-organizing map (SOM) neural network and CNN. That approach showed a higher accuracy compared with other methods using for face recognition at that time. In 2012, Niu et al. [48] introduced a hybrid classification system for objection recognition by integrating the synergy of CNN and SVM, and experimental results showed that the method improved the classification accuracy. Liu et al. [41] used CNN to extract features while Conditional Random Rield (CRF) was used to classify the deep features. With extensive experiments on different datasets, such as Weizmann horse, Graz-02, MSRC-21, Stanford Background, and PASCAL VOC 2011, the hybrid structure got better segmentation performance compared with other methods on the same datasets. In [57], Xie et al. used a hybrid representation method to process scene recognition and domain adaption. In that method, CNN used to extract the features meanwhile mid-level local representation (MLR) and convolutional Fisher vector representation (CFV) made the most of local discriminative information in the input images. After that, SVM classifier was used to classified the hybrid representation and achieved better accuracy. Recently, Tang et al. [55] put forward a hybrid structure including Deep Neural Network (DNN) and ELM to detect ship on spaceborne images. In this time, DNN was used to process high-level feature representation and classification while ELM was worked as effective feature pooling and decision making. What's more, extensive experiments were presented to demonstrate that the hybrid structure required least detection time and achieved highter detection accuracy compared with existing relevant methods. Based on the analysis above, we can integrate CNN with other classifiers to improve the classification accuracy.

2 Extreme Machine Learning Model

ELM was first proposed by Huang et al. [18, 47] which was used for the single-hidden-layer feedforward neural networks (SLFNs). The input weights and hidden layer biases are randomly assigned at first, and then the training datasets to determine the output weights of SLFNs are combined. Figure 3 is a basic structure of ELM. For N arbitrary distinct samples (x_i, t_i), $i = 1, 2, \ldots, N$, where $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^T$, $\mathbf{t}_i = [t_{i1}, t_{i2}, \ldots, t_{im}]^T$. Therefore, the ELM model can be written as:

$$\sum_{j=1}^{L} \beta_j g_j(\mathbf{x}_i) = \sum_{j=1}^{L} \beta_j g(\mathbf{w}_j \cdot \mathbf{x}_i + b_j) = \mathbf{o}_i \ \ (i = 1, 2, \ldots, N), \tag{1}$$

where $\beta_j = [\beta_{j1}, \beta_{j2}, \ldots, \beta_{jm}]^T$ expresses the jth hidden node weight vector while the weight vector between the jth hidden node and the output layer can be described as $\mathbf{w}_j = [w_{1j}, w_{2j}, \ldots, w_{nj}]^T$. The threshold of the jth hidden node can be written as b_j and $\mathbf{o}_i = [o_{i1}, o_{i2}, \ldots, o_{im}]^T$ denotes the ith output vector of ELM.

We can approximate the output of ELM if activation function $g(x)$ with zero error which means as Equation (2):

$$\sum_{i=1}^{N} ||\mathbf{o}_i - \mathbf{t}_i|| = 0. \tag{2}$$

Therefore, Equation (1) can be described as Equation (3):

$$\sum_{j=1}^{L} \beta_j g_j(\mathbf{x}_i) = \sum_{j=1}^{L} \beta_j g(\mathbf{w}_j \cdot \mathbf{x}_i + b_j) = \mathbf{t}_i \ \ (i = 1, 2, \ldots, N). \tag{3}$$

Finally, Equation (3) can be simply expressed as Equation (4):

$$\mathbf{H}\beta = \mathbf{T}, \tag{4}$$

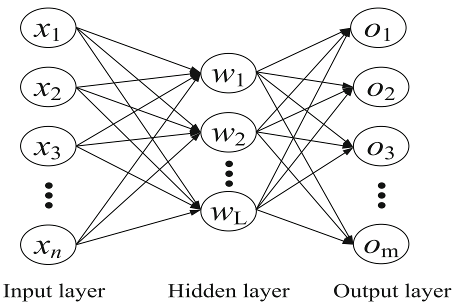

Fig. 3 A basic structure of ELM

where $\mathbf{H}$ expresses the hidden layer output matrix, and $\mathbf{H} = \mathbf{H}(\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_L, b_1, b_2, \ldots, b_L, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$. Therefore, $\mathbf{H}$, β, and $\mathbf{T}$ can be written as follows:

$$[h_{ij}] = \begin{bmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \ldots & g(\mathbf{w}_L \cdot \mathbf{x}_1 + b_L) \\ \vdots & \ddots & \vdots \\ g(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & \ldots & g(\mathbf{w}_L \cdot \mathbf{x}_N + b_L) \end{bmatrix}, \tag{5}$$

$$\beta = \begin{bmatrix} \beta_{11} & \beta_{12} & \ldots & \beta_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{L1} & \beta_{L2} & \ldots & \beta_{Lm} \end{bmatrix}, \tag{6}$$

and

$$\mathbf{T} = \begin{bmatrix} t_{11} & t_{12} & \ldots & t_{1m} \\ \vdots & \vdots & \ddots & \vdots \\ t_{N1} & t_{N2} & \ldots & t_{Nm} \end{bmatrix}. \tag{7}$$

After that, the smallest norm least-squares solution of Equation (4) is:

$$\overset{\wedge}{\beta} = \mathbf{H}^{\dagger}\mathbf{T}, \tag{8}$$

where $\mathbf{H}^{\dagger}$ denotes the Moore-Penrose generalized the inverse of matrix $\mathbf{H}$. The output of ELM can be expressed as Equation (9):

$$f(\mathbf{x}) = h(\mathbf{x})\beta = h(\mathbf{x})\mathbf{H}^{\dagger}\mathbf{T}. \tag{9}$$

From the description above, the process of ELM can be described as follows. At the beginning, ELM was randomly assigned the input weights and the hidden layer biases $(\mathbf{w}_i, b_i)$. After that, we calculates the hidden layer output matrix $\mathbf{H}$ according to Equation (5). Then, by using Equation (8), we can obtain the output weight vector β. Finally, we can classify the new dataset according to the above training process.

ELM is not only widely used to process binary classification [30, 42, 61, 66], but also used for multi-classification due to its good properties. As we have mentioned above, CNNs show excellent performance on extracting feature from the input images, which can reflect the important character attributes of the input images.

3 Convolutional Neural Network

Convolutional Neural Network [37], which usually includes input layer, multi-hidden layers, and output layer, is a deep supervised learning architecture and often made up of two parts: an automatic feature extractor and a trainable classifier. CNN has shown remarkable performance on visual recognition [31]. When we use CNNs to process visual tasks, they first extract local features from the input images. In order to obtain higher order features, the subsequent layers of CNNs will then combine these features. After that, these feature maps are finally encoded into 1-D vectors and a trainable classifier will deal with these vectors. Because of considering size, slant, and position variations for images, feature extraction is a key step during classification of images. Therefore, with the purpose of ensuring some degree of shift, scale, and distortion invariance, CNNs offer local receptive fields, shared weights, and downsampling. Figure 4 is a basic architecture of CNNs.

It can be seen from Fig. 4 that CNNs mainly include three parts: convolution layers, subsampling layers and classification layer. The main purpose of convolutional layers is to extract local patterns and the convolutional operations can enhance the original signal and lower the noise. Moreover, the weights of each filtering kernels in each feature maps are shared, which not only reduce the free parameters of networks, but also lower the complication of relevant layers. The outputs of the convolutional operations contain several feature maps and each neuron in entire feature maps connects the local region of the front layers. Subsampling is similar to a fuzzy filter which is primary to re-extract features from the convolutional layers. With the local correlation principle, the operations of subsampling not only eliminate non-maximal values and reduce computations for previous layer, but also improve the ability of distortion tolerance of the networks and provide additional robustness to position. These features will be encoded into a 1-D vectors in the full connection layer. After that, these vectors will be categorized by a trainable classifier. Finally, the whole neural network will be trained by a standard error back propagation algorithm with stochastic gradient descent [3]. The purpose of training CNNs is to adjust the entire parameters of the system, i.e., the weights and biases of the convolution kernel, and we will use the well-tuned CNNs to predict the classes, such as label, age, and so on, from an unknown input image datasets.

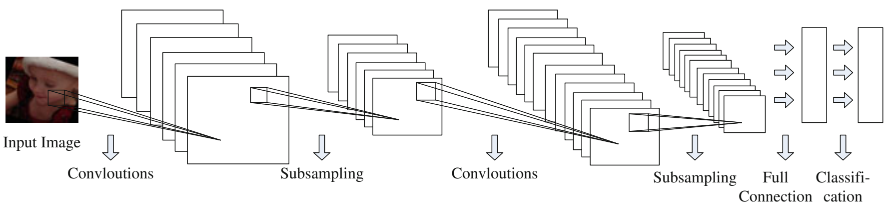

Fig. 4 Structure of CNN for visual recognition

3.1 Convolutional Layer

In the convolutional layer, convolutions which are performed between the previous layer and a series of filters, extract features from the input feature maps [8, 54]. After that, the outputs of the convolutions will add an additive bias and an element-wise nonlinear activation function is applied on the front results. Without loss of generality, we have used the ReLU function as the nonlinear function in our experiment. In general, η_{ij}^{mn} denotes the value of an unit at position (m, n) in the jth feature map in the ith layer and it can be expressed as Equation (10):

$$\eta_{ij}^{mn} = \sigma\left(b_{ij} + \sum_{\delta}\sum_{p=0}^{P_i-1}\sum_{q=0}^{Q_i-1} w_{ij\,\delta}^{pq}\eta_{(i-1)\delta}^{(m+p)(n+q)}\right), \tag{10}$$

where b_{ij} represents the bias of this feature map while δ indexes over the set of the feature maps in the $(i-1)$th layer which are connected to this convolutional layer. $w_{ij\delta}^{pq}$ denotes the value at the position (p, q) of the kernel which is connected to the kth feature map and the height and width of the filter kernel are P_i and Q_i.

The convolutional layer offers a nonlinear mapping from the low level representation of the images to the high level semantic understanding. In order to be convenient to later computations, Equation (10) can be simply denoted as follows:

$$\eta_j = \sigma\left(\sum w_{ij} \bigotimes \eta_{(i-1)}\right), \tag{11}$$

where $\bigotimes$ expresses the convolutional operation while w_{ij}, which will be randomly initialized at first and then trained with **BP** neural network [35, 53], denotes the value of the ith layer in the jth feature map. $\eta_{(i-1)}$ is the outputs of the $(i-1)$ layer and η_j is defined as the outputs of the jth feature map in the convolutional layer. Different sizes of the input feature maps have various effects on the accuracy of classification. Large size of a feature map means good features learned by the convolutional operations with the high cost of the computations while small size reduces the computation cost degrading the accuracy of the classification. Making a comprehensive considerstion of the factors mentioned above and by lots of experiments, we set the size of the input feature map as 227×227.

3.2 Contrast Normalization Layer

The goal of the local contrast normalization layer is not only to enhance the local competitions between one neuron and its neighbors, but also to force features of different feature maps in the same spatial location to be computed, which is motivated by the computational neuroscience [49, 53]. In order to achieve the target, two normalization operations, i.e., subtractive and divisive, are performed. In this time, η_{mnk} denotes the value of an unit at position (m, n) in the kth feature map. We have

$$z_{mnk} = \eta_{mnk} - \sum\nolimits_{p=-\frac{P_i-1}{2}}^{\frac{P_i-1}{2}} \sum\nolimits_{q=-\frac{Q_i-1}{2}}^{\frac{Q_i-1}{2}} \sum_{j=1}^{J_i} \varepsilon_{pq} \eta_{(m+p)(n+q)j}, \tag{12}$$

where ε_{pq} is a normalized Gaussian filter with the size of 7×7 at the first stage and 5×5 at the second stage. z_{mnk} not only represents the input of the divisive normalization operations, but also denotes the output of the subtractive normalization operations. Equation (13) expresses the operator of the divisive normalization:

$$\eta_{mnk} = \frac{z_{mnk}}{\max\left(M, M(m, n)\right)}, \tag{13}$$

where

$$M(m,\ n) = \sqrt{\sum\nolimits_{p=-\frac{P_i-1}{2}}^{\frac{P_i-1}{2}} \sum\nolimits_{q=-\frac{Q_i-1}{2}}^{\frac{Q_i-1}{2}} \sum_{j=1}^{J_i} \varepsilon_{pq} \eta^2_{(m+p)(n+q)j}}, \tag{14}$$

and

$$M = \left(\sum\nolimits_{m=1}^{s1} \sum\nolimits_{n=1}^{s2} M(m,\ n)\right) / (s1 \times s2). \tag{15}$$

During the whole contrast normalization operations above, the Gaussian filter ε_{pq} is calculated with the zero-padded edges, which means that the size of the output of the contrast normalization operations is as same as its input.

3.3 *Maxing Pooling Layer*

Generally speaking, the purpose of pooling strategy is to transform the joint feature representation into a novel, more useful one which keeps crucial information while discards irrelevant details. Each feature map in the subsampling layer is getting by max pooling operations which are carried out on the corresponding feature map in convolutional layers. Equation (16) is the value of a unit at position (m, n) in the jth feature map in the ith layer or subsampling layer after max pooling operation:

$$\eta_{ij}^{mn} = \max\{\eta_{(i-1)j}^{mn},\ \eta_{(i-1)j}^{(m+1)(n+1)},\ \ldots,\ \eta_{(i-1)j}^{(m+P_i)(n+Q_i)}\}. \tag{16}$$

The max pooling operation generates position invariance over larger local regions and downsamples the input feature maps. In this time, the numbers of feature maps in the subsampling layer are 96 while the size of the filter is 3 and the stride of the sliding window is 2. The aim of max pooling action is to detect the maximum response of the generated feature maps while reduces the resolution of the feature map. Moreover, the pooling operation also offers built-in invariance to small shifts and distortions. The procedures of other convolutional layers and subsampling layers which we have not told are as same as the layers mentioned above, except with a different kernel size or stride.

3.4 Softmax

Softmax function is widely used to present a probability distribution over a discrete variable and sigmoid function is usually to denote a probability distribution over a binary variable. In CNN model, softmax functions are often used as the classifiers with representing the probability distribution. Moreover, we can use the softmax functions inside the CNNs model, which we wish the model to select one of n different options.

We wish to produce a single number when the outputs are binary variables as follows:

$$y = P(y = 1|x). \tag{17}$$

When there are a discrete variable with n value, a vector $\mathbf{y}$ is need to be produced. In this time, each element of y should be between 0 and 1 and summing the whole vectors should be 1 so that it can denote a valid probability distribution. We can achieve a linear layer predicts unnormalized probabilities as follows:

$$z = w^T h + b \tag{18}$$

Therefore, we can obtain the softmax function as follows:

$$soft\max(z)_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)} \tag{19}$$

4 Regularizations for Deep Learning

4.1 Dataset Augmentation

To train a best prediction model, it needs more training dataset while there are limit dataset for training the model in real application. The effective way is to create a fake data which has the same distribution. This method is easiest for classification. A general classifier need a large, high dimensional input $\mathbf{x}$ with a single label y.

That means a classifier is to be invariant to a wide variant to the wide variety of transformations of training dataset. Therefore, we can create new ($\mathbf{x}$, y) pairs for enlarging the training dataset.

Dataset augmentation has been an effective approach for classification problem: object recognition. Images are high dimensional dataset. Images are affected by factors of variation and translating the limit training dataset in different direction can create more generalization dataset. Other operations including scaling the images and rotating the images have also proven an effective approach.

4.2 Noise Robustness

For some models, when we add some noise with infinitesimal variance in the input model, this operation is equivalent to imposing a penalty on the norm of the weights. Generally, injecting the noise may be more useful than simply shrinking the parameters. What's more, the noise added to the hidden units affects the whole performance. Noise has become a hot topic due to its merit and the dropout method is the development od the approach.

The other way to regularize the model by adding the noise to the weight and that process can be as the a stochastic implementation of a Bayesian inference over the weights. This method is usually used in RNN model.

4.3 Dropout

Dropout is a model with a computationally inexpensive but power method of regularizing a broad. Dropout can be seen as a approach of making bagging practical for ensembles of very many large neural networks. Bagging not only includes training multiple models but also evaluates multiple models on each test example. It seen impossible when the trained model is a large network, since dealing with this neural network is costly. Dropout training is not the same as bagging training and it trains all sub-networks that also includes removing non-output units.

4.4 Other Regularizations

There are many other regularization methods widely used in the CNN, such as semi-supervised learning, multi-task learning, sparse representations, and so on. These approaches have been proven to improve the performance of neural network and they have been highlighted in machine learning and pattern recognition fields. They achieved state-of-the-art performance in image recognition and can automatically extract the information.

5 Applications

5.1 Large Scale Deep Learning for Age Estimation

Recently, age and gender classification has received huge attention, which provides direct and quickest way for obtaining implicit and critical social information [13]. Fu et al. [12] made a detailed investigation of age classification and we can learn more information about recent situation from [38]. Classifying age from the human facial images was first introduced by Kwon et al. [33] and it was presented that calculating ratios and detecting the appearance of wrinkles could classify facial features into different age categorization. After that, the same method was used to model craniofacial growth with a view to both psychophysical evidences and anthropometric evidences [50] while this approach demanded accurate localization of facial features.

Geng et al. [15] proposed a subspace method called AGing pattErn Subspace which was used to estimate age automatically while age manifold learning scheme was presented in [19] to extract face aging features and a locally adjusted robust regressor was designed to prediction human ages. Although these methods have shown many advantages, the requirement that input images need to be near-frontal and well-aligned is their weakness. It is not difficult to find that the datasets in their experiments are constrained, so that these approaches are not suited for many practical applications including unconstrained image tasks.

Last year, many methods have been proposed to classify age and gender. Chang et al. [4] introduced a cost-sensitive ordinal hyperplanes ranking method to estimate human age from facial images while a novel multistage learning system which is called grouping estimation fusion (DEF) was proposed to classify human age. Li et al. [39] estimated age using a novel feature selection method and shown advantage of the proposed algorithm from the experiments. Although these method mentioned above have shown lots of advantages, they are still relied on constrained images datasets, such as FG-NET [34], MORPH [51], FACES [20].

All of these methods mentioned above have been verified effectively on constrained datasets for age classification which are not suitable for unconstrain images in practical applications. Our proposed method not only automatically classifies age and gender from face images, but also deals with the unconstrain face image tasks effectively.

5.2 Large Scale Deep Learning for Gender Estimation

Although more and more researchers have found that gender classification has played an important role in our daily life, few learning-based machine vision approaches have been put forward. Makinen et al. [43] made a detailed investigation of gender classification while we can learn more about its recent trend from [38]. In the following, we briefly review and summarize relevant methods.

Golomb et al. [16] were some of the early researchers who used a neural network which was trained on a small set of near-frontal facial image dataset to classify gender. Moghaddam et al. [46] used SVM to classify gender from facial images while Baluja et al. [1] adopted AdaBoost to identify person' sex from facial images. After that, Toews et al. [56] presented a viewpoint-invariant appearance model of local scale-invariant features to classify age and gender.

Recently, Yu et al. [63] put forward a study and analysis of gender classification based on human gait while revisiting linear discriminant techniques was used to classify gender [2]. In [9], Eidinger et al. not only presented new and extensive dataset and benchmarks to study age and gender classification, but also designed a classification pipeline to make full use of what little data was available. In [32], a semantic pyramid for gender and action recognition was proposed by Khan et al. and the method is fully automatic while it does not demand any annotations for a person' upper body and face. Chen et al. [5] used first names as facial attributes and modeled the relationship between first names and faces. They used the relationship to classify gender and got higher accuracy compared with other methods. Last year, Han et al. [21] used a generic structure to estimate age, gender, and race. Although most of the approaches mentioned above make lots of progress for age classification, they are aimed at either constrain imaging condition or non-automated classification methods.

5.3 Natural Language Processing

Natural language processing (NLP) denotes the use of human languages and by computer reading and emitting, simple programs can parse language efficiently. Machine translation is one of popular natural language processing that read a sentence in one human language while emit equivalent sentence form other language. Many NLP model need a probability distribution over sequences of works.

Although neural network methods have been successfully applied to NLP, to obtain a excellent performance, some strategies are important. To build an efficient model, we should design an novel model. Maldonado et al. [44] used NLP to automatically detect self-admitted technical debt and achieved a good performance. Groza et al. [17] used NLP and ontologies to mine arguments from cancer documents. With NLP methods, Xing et al. [60] built a recommendation for podcast audio-items. Other researches present the processing of NLP with neural network such as [14, 62], and so on.

5.4 Other Applications

Deep learning have been successfully used in object recognition, speech recognition and natural language processing analyzed above. Many fields such as recommender systems [6, 7, 64], knowledge representation [23–25] and so on. Deep learning has been applied in many other fields and we believe that deep learning will bring more convenience for our lives.

References

1. S. Baluja, H.A. Rowley, Boosting sex identification performance. Int. J. Comput. Vis. **71**(1), 111–119 (2006). https://doi.org/10.1007/s11263-006-8910-9
2. J. Bekios-Calfa, J.M. Buenaposada, L. Baumela, Revisiting linear discriminant techniques in gender recognition. IEEE Trans. Pattern Anal. Mach. Intell. **33**(4), 858–864 (2011). https://doi.org/10.1109/TPAMI.2010.208
3. Y. Cao, Y. Chen, D. Khosla, Spiking deep convolutional neural networks for energy-efficient object recognition. Int. J. Comput. Vis. **113**(1), 54–66 (2015)
4. K.Y. Chang, C.S. Chen, A learning framework for age rank estimation based on face images with scattering transform. IEEE Trans. Image Process. **24**(3), 785–798 (2015). https://doi.org/10.1109/TIP.2014.2387379
5. H. Chen, A. Gallagher, B. Girod, Face modeling with first name attributes. IEEE Trans. Pattern Anal. Mach. Intell. **36**(9), 1860–1873 (2014). https://doi.org/10.1109/TPAMI.2014.2302443
6. C. Christakou, A. Stafylopatis, A hybrid movie recommender system based on neural networks, in *International Conference on Intelligent Systems Design and Applications. Isda'05. Proceedings* (2005), pp. 500–505
7. M.K.K. Devi, R.T. Samy, S.V. Kumar, P. Venkatesh, Probabilistic neural network approach to alleviate sparsity and cold start problems in collaborative recommender systems, in *Computational Intelligence and Computing Research (ICCIC), 2010 IEEE International Conference on* (2010), pp. 1–4
8. Z. Dong, Y. Wu, M. Pei, Y. Jia, Vehicle type classification using a semisupervised convolutional neural network. Intell. Transp. Syst. IEEE Trans. **16**(4), 1–10 (2015)
9. E. Eidinger, R. Enbar, T. Hassner, Age and gender estimation of unfiltered faces. IEEE Trans. Inf. Forensics Secur. **9**(12), 2170–2179 (2014). https://doi.org/10.1109/TIFS.2014.2359646
10. G. Feng, G.-B. Huang, Q. Lin, R. Gay, Error minimized extreme learning machine with growth of hidden nodes and incremental learning. Neural Netw. IEEE Trans. **20**(8), 1352–1357 (2009)
11. G. Feng, Y. Lan, X. Zhang, Z. Qian, Dynamic adjustment of hidden node parameters for extreme learning machine. Cybern. IEEE Trans. **45**(2), 279–288 (2015). https://doi.org/10.1109/TCYB.2014.2325594
12. Y. Fu, G. Guo, T.S. Huang, Age synthesis and estimation via faces: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **32**(11), 1955–1976 (2010). https://doi.org/10.1109/TPAMI.2010.36
13. S. Fu, H. He, Z.G. Hou, Learning race from face: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **36**(12), 2483–2509 (2014). https://doi.org/10.1109/TPAMI.2014.2321570
14. J. Gao, X. He, L. Deng, Deep learning for web search and natural language processing. WSDM (2015). https://www.microsoft.com/en-us/research/publication/deep-learning-for-web-search-and-natural-language-processing/
15. X. Geng, Z.H. Zhou, K. Smith-Miles, Automatic age estimation based on facial aging patterns. IEEE Trans. Pattern Anal. Mach. Intell. **29**(12), 2234–2240 (2007). https://doi.org/10.1109/TPAMI.2007.70733

16. B.A. Golomb, D.T. Lawrence, T.J. Sejnowski, Sexnet: a neural network identifies sex from human faces, in *Proceedings of the 1990 Conference on Advances in Neural Information Processing Systems 3* (1990), pp. 572–577
17. A. Groza, O.M. Popa, Mining arguments from cancer documents using natural language processing and ontologies, in *2016 IEEE 12th International Conference on Intelligent Computer Communication and Processing (ICCP)* (2016), pp. 77–84. https://doi.org/10.1109/ICCP.2016.7737126
18. H. Guang-Bin, C. Lei, S. Chee-Kheong, Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans. Neural Netw. **17**(4), 879–92 (2006)
19. G. Guo, Y. Fu, C.R. Dyer, T.S. Huang, Image-based human age estimation by manifold learning and locally adjusted robust regression. IEEE Trans. Image Process. **17**(7), 1178–1188 (2008). https://doi.org/10.1109/TIP.2008.924280
20. G. Guo, X. Wang, A study on human age estimation under facial expression changes, in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on* (2012), pp. 2547–2553. https://doi.org/10.1109/CVPR.2012.6247972
21. H. Han, C. Otto, X. Liu, A.K. Jain, Demographic estimation from face images: human vs. machine performance. IEEE Trans. Pattern Anal. Mach. Intell. **37**(6), 1148–1161 (2015). https://doi.org/10.1109/TPAMI.2014.2362759
22. Q. He, T. Shang, F. Zhuang, Z. Shi, Parallel extreme learning machine for regression based on mapreduce. Neurocomputing **102**, 52–58 (2013)
23. J.C. Hoskins, D.M. Himmelblau, Artificial neural network models of knowledge representation in chemical engineering. Comput. Chem. Eng. **12**(9C10), 881–890 (1988)
24. J.C. Hoskins, D.M. Himmelblau, Neural network models of knowledge representation in process engineering. Comput. Chem. Eng. **12**(9–10), 881–890 (1988)
25. F. Hrbein, J. Eggert, E. Rner, A cortex-inspired neural-symbolic network for knowledge representation, in *International Conference on Neural-Symbolic Learning and Reasoning* (2007), pp. 34–39
26. G.-B. Huang, L. Chen, Enhanced random search based incremental extreme learning machine. Neurocomputing **71**(16), 3460–3468 (2008)
27. G.-B. Huang, L. Chen, C.-K. Siew, Universal approximation using incremental constructive feedforward networks with random hidden nodes. Neural Netw. IEEE Trans. **17**(4), 879–892 (2006)
28. G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications. Neurocomputing **70**(1), 489–501 (2006)
29. G.-B. Huang, D.H. Wang, Y. Lan, Extreme learning machines: a survey. Int. J. Mach. Learn. Cybern. **2**(2), 107–122 (2011)
30. G.-B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification. Syst. Man Cybern B Cybern IEEE Trans. **42**(2), 513–529 (2012). https://doi.org/10.1109/TSMCB.2011.2168604
31. F. Jialue, X. Wei, W. Ying, G. Yihong, Human tracking using convolutional neural networks. IEEE Trans. Neural Netw. **21**(10), 1610–1623 (2010)
32. F.S. Khan, J. van de Weijer, R.M. Anwer, M. Felsberg, C. Gatta, Semantic pyramids for gender and action recognition. IEEE Trans. Image Process. **23**(8), 3633–3645 (2014). https://doi.org/10.1109/TIP.2014.2331759
33. Y.H. Kwon, N. da Vitoria Lobo, Age classification from facial images, in *Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94. 1994 IEEE Computer Society Conference on* (1994), pp. 762–767. https://doi.org/10.1109/CVPR.1994.323894
34. A. Lanitis, The fg-net aging database (2002). www-prima.inrialpes.fr/FGnet/html/benchmarks.html

35. S. Lawrence, C. Giles, A.C. Tsoi, A. Back, Face recognition: a convolutional neural-network approach. IEEE Trans. Neural Netw. **8**(1), 98–113 (1997)
36. S. Lawrence, C.L. Giles, A.C. Tsoi, A.D. Back, Face recognition: a convolutional neural-network approach. IEEE Trans. Neural Netw. **8**(1), 98–113 (1997). https://doi.org/10.1109/72.554195
37. Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
38. G. Levi, T. Hassncer, Age and gender classification using convolutional neural networks, in *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2015), pp. 34–42
39. C. Li, Q. Liu, W. Dong, X. Zhu, J. Liu, H. Lu, Human age estimation based on locality and ordinal information. IEEE Trans. Cybern. **45**(11), 2522–2534 (2015). https://doi.org/10.1109/TCYB.2014.2376517
40. N.-Y. Liang, G.-B. Huang, P. Saratchandran, N. Sundararajan, A fast and accurate online sequential learning algorithm for feedforward networks. Neural Netw. IEEE Trans. **17**(6), 1411–1423 (2006)
41. F. Liu, G. Lin, C. Shen, CRF learning with {CNN} features for image segmentation. Pattern Recogn. **48**(10), 2983–2992 (2015). https://doi.org/10.1016/j.patcog.2015.04.019, http://www.sciencedirect.com/science/article/pii/S0031320315001582. Discriminative Feature Learning from Big Data for Visual Recognition
42. J. Luo, C.M. Vong, P.K. Wong, Sparse bayesian extreme learning machine for multi-classification. Neural Netw. Learn. Syst. IEEE Trans. **25**(4), 836–843 (2014)
43. E. Makinen, R. Raisamo, Evaluation of gender classification methods with automatically detected and aligned faces. IEEE Trans. Pattern Anal. Mach. Intell. **30**(3), 541–547 (2008). https://doi.org/10.1109/TPAMI.2007.70800
44. E. Maldonado, E. Shihab, N. Tsantalis, Using natural language processing to automatically detect self-admitted technical debt. IEEE Trans. Softw. Eng. **PP**(99), 1–1 (2017). https://doi.org/10.1109/TSE.2017.2654244
45. Y. Miche, A. Sorjamaa, P. Bas, O. Simula, C. Jutten, A. Lendasse, OP-ELM: optimally pruned extreme learning machine. Neural Netw. IEEE Trans. **21**(1), 158–162 (2010)
46. B. Moghaddam, M.-H. Yang, Learning gender with support faces. IEEE Trans. Pattern Anal. Mach. Intell. **24**(5), 707–711 (2002). https://doi.org/10.1109/34.1000244
47. L. Nan-Ying, H. Guang-Bin, P. Saratchandran, N. Sundararajan, A fast and accurate online sequential learning algorithm for feedforward networks. IEEE Trans. Neural Netw. **17**(6), 1411–1423 (2006)
48. X.X. Niu, C.Y. Suen, A novel hybrid CNN-SVM classifier for recognizing handwritten digits. Pattern Recogn. **45**(4), 1318–1325 (2012)
49. N. Pinto, D.D. Cox, J.J. Dicarlo, Why is real-world visual object recognition hard? Plos Comput. Biol. **4**(1), 86–89 (2008)
50. N. Ramanathan, R. Chellappa, Modeling age progression in young faces, in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1 (2006), pp. 387–394. https://doi.org/10.1109/CVPR.2006.187
51. K. Ricanek, T. Tesafaye, Morph: a longitudinal image database of normal adult age-progression, in *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on* (2006), pp. 341–345. https://doi.org/10.1109/FGR.2006.78
52. R. Savitha, S. Suresh, H. Kim, A meta-cognitive learning algorithm for an extreme learning machine classifier. Cogn. Comput. **6**(2), 253–263 (2014)
53. P. Sermanet, Y. Lecun, Traffic sign recognition with multi-scale convolutional networks, in *Neural Networks (IJCNN), The 2011 International Joint Conference on* (2011), pp. 2809–2813
54. J. Shuiwang, Y. Ming, Y. Kai, 3D convolutional neural networks for human action recognition. Pattern Anal. Mach. Intell. IEEE Trans. **35**(1), 221–231 (2013)
55. J. Tang, C. Deng, G.-B. Huang, B. Zhao, Compressed-domain ship detection on spaceborne optical image using deep neural network and extreme learning machine. Geosci. Remote Sens. IEEE Trans. **53**(3), 1174–1185 (2015). https://doi.org/10.1109/TGRS.2014.2335751

56. M. Toews, T. Arbel, Detection, localization, and sex classification of faces from arbitrary viewpoints and under occlusion. IEEE Trans. Pattern Anal. Mach. Intell. **31**(9), 1567–1581 (2009). https://doi.org/10.1109/TPAMI.2008.233
57. G.S. Xie, X.Y. Zhang, S. Yan, C.L. Liu, Hybrid CNN and dictionary-based models for scene recognition and domain adaptation. IEEE Trans. Circuits Syst. Video Technol. **PP**(99), 1–1 (2015). https://doi.org/10.1109/TCSVT.2015.2511543
58. J. Xin, Z. Wang, C. Chen, L. Ding, G. Wang, Y. Zhao, ELM*: distributed extreme learning machine with mapreduce. World Wide Web **17**(5), 1189–1204 (2014)
59. J. Xin, Z. Wang, L. Qu, G. Wang, Elastic extreme learning machine for big data classification. Neurocomputing **149**, 464–471 (2015)
60. Z. Xing, M. Parandehgheibi, F. Xiao, N. Kulkarni, C. Pouliot, Content-based recommendation for podcast audio-items using natural language processing techniques, in *2016 IEEE International Conference on Big Data (Big Data)* (2016), pp. 2378–2383. https://doi.org/10.1109/BigData.2016.7840872
61. Y. Yang, Q.M. Wu, Y. Wang, K.M. Zeeshan, X. Lin, X. Yuan, Data partition learning with multiple extreme learning machines. Cybern. IEEE Trans. **45**(6), 1463–1475 (2014)
62. W. Yin, K. Kann, M. Yu, H. Schütze, Comparative study of CNN and RNN for natural language processing. arXiv preprint arXiv:1702.01923 (2017)
63. S. Yu, T. Tan, K. Huang, K. Jia, X. Wu, A study on gait-based gender classification. IEEE Trans. Image Process. **18**(8), 1905–1910 (2009). https://doi.org/10.1109/TIP.2009.2020535
64. F. Zhang, Q. Zhou, Ensemble detection model for profile injection attacks in collaborative recommender systems based on BP neural network. IET Inf. Secur. **9**(1), 24–31 (2014)
65. H. Zhou, G.-B. Huang, Z. Lin, H. Wang, Y. Soh, Stacked extreme learning machines. Cybern. IEEE Trans. **PP**(99), 1–1 (2014). https://doi.org/10.1109/TCYB.2014.2363492
66. B. Zuo, G.B. Huang, D. Wang, W. Han, M.B. Westover, Sparse extreme learning machine for classification. IEEE Trans. Cybern. **44**(10), 1858–1870 (2014)