Computers & Graphics 35 (2011) 955–966

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Technical Section

Neural network-based symbol recognition using a few labeled samples

Luoting Fu, Levent Burak Kara *

Carnegie Mellon University, Mechanical Engineering Department, 5000 Forbes Ave, Pittsburgh, PA 15213, USA

ARTICLE INFO

Article history:
Received 20 December 2010
Received in revised form
15 June 2011
Accepted 12 July 2011
Available online 23 July 2011

Keywords:
Sketch recognition
Symbol recognition
Deep Belief Network
Unsupervised training
Neural nets
Synthetic training samples

ABSTRACT

The recognition of pen-based visual patterns such as sketched symbols is amenable to supervised machine learning models such as neural networks. However, a sizable, labeled training corpus is often required to learn the high variations of freehand sketches. To circumvent the costs associated with creating a large training corpus, improve the recognition accuracy with only a limited amount of training samples and accelerate the development of sketch recognition system for novel sketch domains, we present a neural network training protocol that consists of three steps. First, a large pool of unlabeled, synthetic samples are generated from a small set of existing, labeled training samples. Then, a Deep Belief Network (DBN) is pre-trained with those synthetic, unlabeled samples. Finally, the pre-trained DBN is fine-tuned using the limited amount of labeled samples for classification. The training protocol is evaluated against supervised baseline approaches such as the nearest neighbor classifier and the neural network classifier. The benchmark data sets used are partitioned such that there are only a few labeled samples for training, yet a large number of labeled test cases featuring rich variations. Results suggest that our training protocol leads to a significant error reduction compared to the baseline approaches.

1. Introduction

Sketch understanding [1,2] aims to enable the computers to interpret man-made, freehand sketches and extract the intended information underlying the input strokes. Fig. 1 shows two exemplary sketches depicting two engineering systems and the corresponding engineering model. If successful, sketch understanding could provide a natural human-computer interface for scenarios in which physical, pen-and-paper sketches have been routinely used, such as the early ideation process or the classroom instruction. Moreover, sketch understanding could automate the mining, organization, search and critique of the information embedded in freehand sketches, potentially resulting in a myriad of intelligent agents, such as a web spider that crawls through the drawings in online textbooks and lecture notes to learn the design rules of electrical systems, an archiver that indexes brainstorming sketches for later retrieval and reuse, and a computer grader for the free-body diagrams that students draw in their statics homework.

One of the core problems in sketch understanding is to devise a symbol recognizer to compute a categorical label for each segment of the input sketch. Used in conjunction with a sketch parser that divides the input sketch into segments and possibly a post-processor that ensures the consistency of the recognition, an interpretation of the input sketch can be produced. Such problem decomposition is recurrent in recent sketch recognition systems [3–8]. For example, in [3], a Convolutional Neural Network recognizer is used with a sliding windows segmenter. The recognition output of the Convolutional Neural Network is post-processed to merge the overlapping and non-maximal labels. In [5], a heuristic-based segmenter is used in conjunction with a Gaussian Bayes classifier. Because of the particular choice of the segmentation heuristics for that domain, each segmentation corresponds to an isolated symbol in the sketch and post-processing is not required. In [7], the up and down motion of the pen-tip is utilized to segment the sketch into a number of strokes, and a Conditional Random Field model plays the dual roles of the recognizer and the post-processor to output a globally consistent interpretation of the input.

Neural network classifiers [9,10] are particularly appealing candidates for the recognition of sketched symbols. They feature a feed-forward classification algorithm capable of rapid classification of the input, and a supervised back-propagation training algorithm capable of the data-driven learning of highly complex decision boundaries between multiple categories. Neural network classifiers are known for the high accuracy achieved in various domains such as freehand-sketched symbols [3,11], handwritten digits [12] and human faces [13]. However, a large, correctly *labeled* training data set with rich, in-class variations is required to train a highly accurate neural network classifier. Such a requirement, arising from the rich stylistic variations of unconstrained user inputs, is reported in empirical [14–16] and theoretical

* Corresponding author.
E-mail addresses: luoting.fu@cmu.edu (L. Fu), lkara@cmu.edu (L.B. Kara).

0097-8493/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cag.2011.07.001

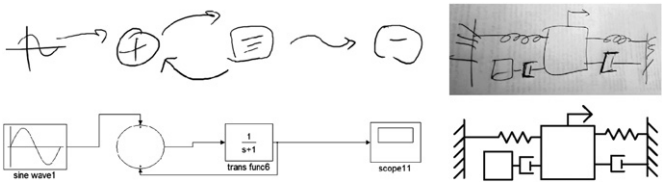

Fig. 1. Top row: two sketches drawn on a tablet PC and on a piece of paper. Bottom row: the control system and mechanical vibratory system models corresponding to the sketches above.

[17] studies. As a rule of thumb in visual classification in which complex, non-linear decision boundaries separate the pattern categories of interest, the performance of a classifier would degrade, if the size of the training data set is significantly below the dimension of the input patterns[1] [18].

Collecting a large number of training samples and manually labeling each instance, is time-consuming, costly, laborious and hence undesirable, especially when targeting a new sketch domain with new symbol definitions. To circumvent such difficulties and accelerate the training and deployment of neural network classifiers for such scenarios, we propose a novel training protocol that relies on only a handful of training samples[2] and achieves higher accuracy than purely supervised approaches.

The proposed training protocol consists of three steps, as illustrated in Fig. 2. Prior to applying our approach, a small, labeled data set of training samples, called the *seed* samples, is collected from the users. At the beginning of our approach, a large pool of synthetic, *unlabeled* samples are generated from the seed samples. Then, a Deep Belief Network (DBN, a neural network variant introduced in [19]) is trained using the unlabeled, synthetic samples. The learning objective of this step is to maximize the probability for the DBN to generate the training samples, such that it would discover the structural regularities underlying the input patterns and learn a non-linear, hierarchical representation of them. Finally, the pre-trained DBN from the previous step is used as a feature extractor and concatenated with an additional layer of decision units. Together they are fine-tuned as a deep, feed-forward neural network classifier using the labeled seed samples. The learning objective here is to minimize the classification errors on the labeled training set, so as to learn the decision boundaries between the pattern classes.

The proposed training protocol can be seen as supervised training preceded and enhanced by the synthesis of unlabeled training samples and the unsupervised pre-training. The incorporation of unsupervised pre-training is inspired by the recent progress on semi-supervised, transfer learning [20,21] and Deep Belief Network [19,22–25].

Our contribution is a protocol suitable for training neural network-based symbol recognizers in novel sketch domains where it is difficult to employ other existing training approaches. Specifically, in the scenario we target, the initial number of labeled samples is limited, no labeled sample synthesizer is available and no relevant domain with similar symbols exists for transfer learning. Our approach reduces the need for labeled training samples and in turn reduces the time or efforts needed to collect or label such samples from the users, thus enabling accelerated deployment of sketch understanding systems. The neural network-based recognizer can work with image-based, off-line sketches as well as trajectory-based, online sketches. We focus on the image-based, off-line sketches in this work for two reasons. First, such off-line sketches can be drawn on a broader range of digital or physical drawing surfaces and then captured using a variety of image or ink acquisition devices, whereas the acquisition of online sketches relies on tablet PCs or multi-touch devices. Second, online sketch recognition is confounded by the issues of stroke-level drawing order and stroke interspersions [26], whereas such issues do not exist in the image-based representation.

The rest of this paper is organized as follows: Section 2 reviews existing training approaches for neural network-based image classifiers, and shows that our approach is suitable for a scenario not covered by the existing approaches. Section 3 presents the step-by-step details of the proposed training protocol. Section 4 evaluates the proposed protocol with three different data sets of sketched symbols against purely supervised baseline techniques, and discusses the implications and future works.

2. Related work

Here we present a summary of related work in Fig. 3 and show their applicability in the form of a decision tree. All the work reviewed here pertains to the classification task defined in the root node of the tree, that is, given a collection of user-generated, labeled training samples $\mathbf{X}^L$, classify unlabeled, user-generated test samples $\mathbf{X}^?$.

If the number of user-generated, labeled training samples $|\mathbf{X}^L|$ far exceeds the dimensionality of a sample $\mathrm{Dim}(x)$, $x \in \mathbf{X}$, it is then straightforward to perform the purely supervised training of a neural network classifier using back-propagation [10]. In this case, the large volume of training set ensures the inclusion of rich stylistic variations within each class, which in turn results in accurate classifiers [14–16]. Otherwise, if the sample size is small, the various approaches reviewed in the subsequent sections can be utilized.

2.1. Synthetic training samples

In case the number of labeled training samples $|\mathbf{X}^L|$ is below the dimensionality of each input, if there exists a sample generator $\mathcal{G}$ that takes existing seed samples $\mathbf{X}^L$ as the inputs and outputs labeled, synthetics samples $\tilde{\mathbf{X}}^L$, then a viable strategy is to computationally generate many supplemental training samples by $\mathcal{G}$, rather than collecting and labeling additional samples by the users. With such synthetic samples $\tilde{\mathbf{X}}^L$, supervised training can be performed on the expanded training set $\tilde{\mathbf{X}}^L \cup \mathbf{X}^L$.

Several distortion-based techniques [27–29] have been developed to generate synthetic samples through global, affine transformations and local, elastic deformations applied to the seed samples. Such distortions emulate the stylistic variations of pen-based input patterns naturally induced by the users. However, the amount of distortions has to be manually tuned through repeated trial-and-error. If too aggressive, the deformations would invalidate the labels of the synthetic training samples. For example, a letter "I" could be deformed into "J" if too much local bending is allowed to the bottom-half. A digit "9" could be subject to excessive rotation and becomes a digit "6". Studies [30,31] have shown that such invalid labels in the training set are detrimental to the performance of the supervised classifiers. If too conservative, the deformations only results in a redundant set of synthetic samples with little variations that does not contribute much to the training.

An alternative to distortion is to interpolate existing training samples that belong to the same class, in the hope that such synthetic samples will share the same, valid labels as the original ones [16,32]. One major limitation is that if the original training

[1] If the input is a feature vector extracted from the input pattern, then the dimension of the input is the number of features. If the input is an image patch, then the dimension of the input is the number of pixels.

[2] That is, the sample size is smaller than or on par with the input dimension.