Pattern Recognition Letters 80 (2016) 24–29

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

A feasibility study of an autoencoder meta-model for improving generalization capabilities on training sets of small sizes☆

Alexey Potapov[a,b], Vita Potapova[b], Maxim Peterson[a,*]

[a] Chair of Computer Photonics and Digital Video Processing, ITMO University, Kronverkskiy pr. 49, 197101, St. Petersburg Russia
[b] Faculty of Liberal Arts and Sciences, St. Petersburg State University, Universitetskaya nab. 7-9, 199034, St. Petersburg Russia

ARTICLE INFO

Article history:
Received 3 July 2015
Available online 24 May 2016

Keywords:
Autoencoders
Logistic regression
Overlearning
Generalization
Image recognition

ABSTRACT

The problem of training autoencoders (with logistic regression as the classification layer) on sets of small sizes is considered on the example of image classification and scene categorization tasks. Conventional autoencoders with uniform priors usually fail to learn useful features from few samples. A possibility to overcome this difficulty is considered on the example of a proposed meta-model generating autoencoders from a vector of meta-parameters of much smaller dimension than the number of autoencoder parameters. Handcrafted image features designed for the task of scene classification are used as the baseline for comparison. Unbiased autoencoders showed the worst results on tiny training sets and seem to require much larger sets to outperform other methods. On the other hand, the developed biased autoencoders work better on the training sets of small sizes, but have much less discriminative power, since the considered meta-model assigns strictly zero probabilities to a large subset of solutions. General possibility of increasing training speed (in term of consumed training patterns) of autoencoders is confirmed.

1. Introduction

Selecting appropriate features is the essential part of creating efficient image recognition methods. Previously, it was well acknowledged that discriminative features should be learned [1]. However, a lot of computer vision methods are still based on handcrafted image representations, which are usually task-dependent (e.g. compare [2,3]).

The most conventional approach to feature learning is based on neural networks, which have been applied in image recognition for a long time (e.g. [4]). Recent success in feature learning in particular for image recognition is connected to deep learning [5–8]. Deep learning exploits the fact that adding extra hidden layers in a multi-layer classifier helps to learn more complex features and to improve their representational power. The key problem here is to train such classifiers. Earlier, training of shallow feed-forward networks with the back-propagation algorithm yielded better results than training deep networks in the same manner, since bottom layers are difficult to train because of exponentially quick gradient vanishing, so these layers usually correspond to random (not useful or even harmful) nonlinear feature mappings. Successful training of such networks was achieved [9–11] with the help of consequent unsupervised pre-training of each hidden layer, and the use of supervised training afterwards.

Another component that is also considered as crucial for deep learning is the usage of large training sets [10]. Actually, even successful supervised training of multi-layer perceptrons (resulting in a very low 0.35% error rate on the MNIST benchmark) has been also achieved recently [12] using intensive training of perceptrons with patterns, relevantly transformed with affine and elastic image deformations. Even overlearning is avoided in spite of extremely large number of neurons (free parameters), because “the continual deformations of the training set generate a virtually infinite supply of training examples” [12].

Deep learning methods don’t simply benefit from using large datasets, but require them. They cannot outperform handcrafted representations being trained on small sets from scratch (e.g. [13]). The necessity of large training sets cannot be considered as an advantage, since only small sets can be available and invariants (necessary for generating deformed patterns) can be unknown in some applications. At the same time, humans have capabilities to learn complex invariant features just from few examples [14].

Thus, the problem of learning from training sets of small sizes is crucial. In this paper, we consider this problem on an example of autoencoders with logistic regression as the classification layer applied to the task of image classification. Pixel-level and handcrafted features are used for experiments.

☆ This paper has been recommended for acceptance by A. Petrosino.

* Corresponding author. Tel.: +7 8123157534.
E-mail addresses: pas.aicv@gmail.com (A. Potapov), maxim.peterson@gmail.com (M. Peterson).

http://dx.doi.org/10.1016/j.patrec.2016.05.018
0167-8655/

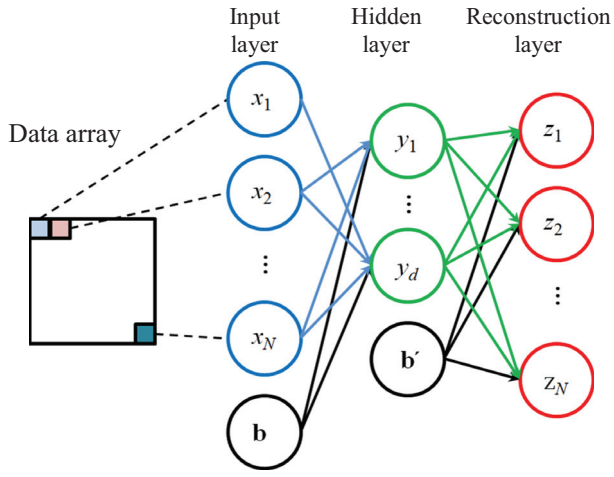

Fig. 1. Conventional autoencoder structure.

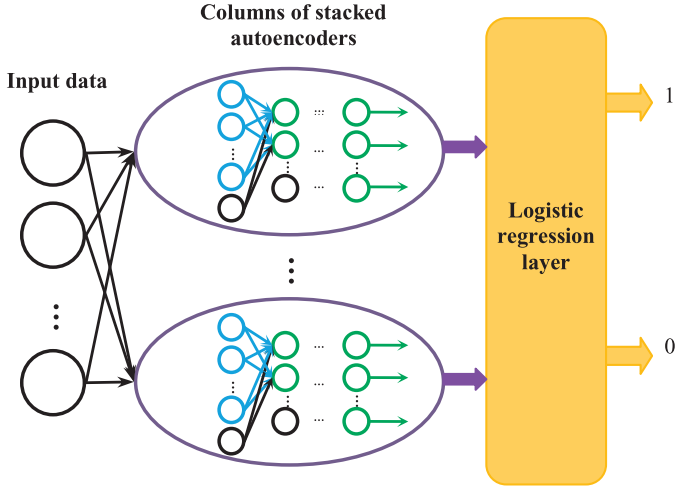

Fig. 2. Multi-column stacked autoencoder with logistic regression layer.

2. Autoencoders

2.1. Conventional autoencoders

Single autoencoder has input, hidden, and output reconstruction layers (Fig. 1). The input layer accepts a vector $\mathbf{x} \in [0, 1]^N$ of dimension N, which is transformed to activities of neurons of the hidden layer $\mathbf{y} = s(\mathbf{W}\mathbf{x} + \mathbf{b})$, $\mathbf{y} \in [0, 1]^d$, corresponding to the new features (hidden representation), where $\mathbf{W}$ is a $d \times N$ matrix of connection weights, $\mathbf{b}$ is a bias vector, and s is the activation (sigmoid) function. Activities of neurons of the last layer are calculated similarly as $\mathbf{z} = s(\mathbf{W}'\mathbf{y} + \mathbf{b}')$, $\mathbf{z} \in [0, 1]^N$. Autoencoders differ from other feed-forward networks in that they are trained to minimize difference between $\mathbf{x}$ and $\mathbf{z}$ (for patterns from a training set), that is $\mathbf{W}'$ is the matrix of the reverse mapping. $\mathbf{W}'$ is frequently taken as $\mathbf{W}^T$. One can calculate gradient of the reconstruction error relative to connection weights and bias vectors and to train the autoencoder using stochastic gradient descent.

In the case of stacked autoencoders, each autoencoder on the next level takes outputs of the hidden (not reconstruction) layer of the previous autoencoder as its input performing further nonlinear transformation of constructed latent representation of the previous level. Each next autoencoder is trained after training the preceding autoencoder. Outputs from the hidden layer of the last autoencoder are passed to the supervised feed-forward network.

Multinomial logistic regression computes probabilities of a pattern to belong to different classes based on softmax applied to linear combinations of features:

$$p(y = c|\mathbf{x}, \mathbf{W}) = \frac{\exp(\mathbf{w}_c^T\mathbf{x} + b_c)}{\sum_{c'=1}^{C} \exp(\mathbf{w}_{c'}^T\mathbf{x} + b_{c'})}, \quad (1)$$

where $\mathbf{x}$ is the input vector, c is the class index, C is the total number of classes, $\mathbf{W}$ is the weight matrix composed by C vectors $\mathbf{w}_c$, and b_c are biases. Parameters of the classifier are learned using stochastic gradient descent minimizing negative log-likelihood.

One common modification is denoising autoencoders [7], in which reconstruction during training is calculated using patterns with introduced noise, but reconstruction errors are calculated relative to initial patterns without this artificial noise. Thus, a denoising autoencoder learns to reconstruct a clean input from a corrupted one. This is also a way to virtually increase the size of the training set by deforming input patterns, but not in so problem-specific fashion.

We used a modified approach, which consists in constructing multi-column autoencoders. Multi-column deep neural networks have already been used (e.g. [5]), but each column in such networks is usually trained as a separate deep network, and predictions of all columns are then averaged. We use the modification [13], in which the number of columns corresponds to the number of classes. Each column is trained for its own class to produce some non-linear features. These features are then gathered and passed as the input to the logistic regression layer (Fig. 2).

We will refer to this type of autoencoders as unbiased autoencoders since uniform priors are used for them.

2.2. A meta-model for autoencoders

Poor generalization on training sets of small sizes follows from high prior uncertainty in model parameters (model information capacity). Autoencoders applied to the raw image features are characterized by the large parameter space with uniform priors.

Some non-uniform priors (e.g. in the form of a meta-model generating autoencoders) can be introduced to reduce this uncertainty. In this paper, we consider one such possible meta-model as an example. Of course, since this model doesn't implement universal priors [15], it has a limited applicability. However, it can show principal possibility of reducing required sizes of training sets.

In general, in order to introduce a meta-model one needs to specify a distribution over model parameters conditioned by meta-parameters, e.g. $p(\mathbf{W},\mathbf{W}',\mathbf{b},\mathbf{b}'|\boldsymbol{\chi})$, where $\boldsymbol{\chi}$ is a vector of meta-parameters. However, we consider a deterministic meta-model, which assigns zero probability to instantiations of autoencoders that can't be generated by it.

For example, suppose we estimate parameters of the normal distribution. Following Bayes's rule we should multiply the likelihood $P = (\{x\}|x_0, \sigma)$ by the prior $P = (x_0, \sigma)$, but how to define the prior? If we ignore priors, we'll get the maximum likelihood approach. Alternatively, one can define the prior probability $P = (x_0, \sigma)$ in terms of some other distribution. For instance, priors for the normal distribution are often defined in terms of the gamma distribution, which is the conjugate distribution for the normal one. In other words, the gamma distribution has its own parameters, which can be considered as meta-parameters for the normal distribution.

One can also consider different priors while training autoencoders. Minimization of the reconstruction MSE is equivalent to using the maximum likelihood approach with uniform priors. Different types of regularization (e.g. denoising [16] or sparse coding [17]) can be interpreted as implicit introduction of some priors. We propose to introduce priors by explicitly defining meta-models, and consider one example of highly non-uniform priors.