Feature Extraction for Simple Classification

André Stuhlsatz, Jens Lippel and Thomas Zielke
Department of Mechanical and Process Engineering, University of Applied Sciences,
Josef-Gockeln-Str. 9, 40474 Düsseldorf, Germany
andre.stuhlsatz@fh-duesseldorf.de

Abstract

Constructing a recognition system based on raw measurements for different objects usually requires expert knowledge of domain specific data preprocessing, feature extraction, and classifier design. We seek to simplify this process in a way that can be applied without any knowledge about the data domain and the specific properties of different classification algorithms. That is, a recognition system should be simple to construct and simple to operate in practical applications. For this, we have developed a nonlinear feature extractor for high-dimensional complex patterns, using Deep Neural Networks (DNN). Trained partly supervised and unsupervised, the DNN effectively implements a nonlinear discriminant analysis based on a Fisher criterion in a feature space of very low dimensions. Our experiments show that the automatically extracted features work very well with simple linear discriminants, while the recognition rates improve only minimally if more sophisticated classification algorithms like Support Vector Machines (SVM) are used instead.

1. Introduction

A typical recognition system consists of data preprocessing for raw measurements, feature extraction and classification. For each processing stage, the design and implementation requires expert knowledge, as does the optimal composition of successive stages. For example, the use of any feature extractor that does not fit to a particular classifier, and vice versa, often yields a very poor performance. To simplify this process, we propose an easy to use framework that learns to extract low-dimensional features from raw measurements without assumptions about the underlying data domain. Our framework is based on DNNs which are Multilayer Neural Networks (MLNN) with two or more hidden layers and thousands, often millions of free parameters. Due to the high flexibility of DNNs, complex nonlinear projections of high-dimensional data to low-dimensional feature spaces can be learned. Unfortunately, conventional training of DNNs, i.e. random initialization and subsequent optimization of all parameters regarding some error function, most often gets stuck in a poor minimum or suffers from overfitting.

Recently, an efficient pre-optimization of DNNs for the nonlinear autoencoder task has been proposed [3]. Since our aim is a nonlinear feature extraction that improves simple linear classification in low dimensions, we enhance the pre-optimization of DNNs with respect to a subsequent fine-tuning using back-propagation with a Fisher discriminant criterion as objective function. We justify this approach by experiments on various real world datasets and different classifiers.

2. Deep Neural Networks for Discriminative Nonlinear Feature Extraction

Consider a target coding scheme $\boldsymbol{\Lambda} := (\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_c) \in \mathbb{R}^{c \times c}$ for a c-class problem where code vectors $\boldsymbol{\lambda}_i \in \mathbb{R}^c$ uniquely represent a category ω_i. Let be $\boldsymbol{t}_n \in \{\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_c\}$ the unique target vector associated to the known category $\omega(\boldsymbol{x}_n)$ of a measurement $\boldsymbol{x}_n \in \mathbb{R}^d$. We denote $\boldsymbol{v}^{out}(\boldsymbol{x}_n) \in \mathbb{R}^c$ the state of the output units of a MLNN with a measurement $\boldsymbol{x}_n$ clamped at its input units. As is well-known, the Mean Squared Error (MSE)

$$\hat{R}(\mathcal{O}_N) := \frac{1}{N} \sum_{n=1}^{N} \left\| \boldsymbol{t}_n - \boldsymbol{v}^{out}(\boldsymbol{x}_n) \right\|_2^2 \tag{1}$$

at the network outputs $\boldsymbol{v}^{out}$, using a training sample $\mathcal{O}_N := \{(\boldsymbol{t}_1, \boldsymbol{x}_1), \ldots, (\boldsymbol{t}_N, \boldsymbol{x}_N)\}$, asymptotically reaches with probability one the true risk

$$\int_{\mathbb{R}^d} \left\| \boldsymbol{r}(\boldsymbol{x}) - \boldsymbol{v}^{out}(\boldsymbol{x}) \right\|_2^2 \, p(\boldsymbol{x}) \, d\boldsymbol{x} + q = \lim_{\substack{N \to \infty \\ p}} \hat{R}(\mathcal{O}_N) \tag{2}$$

1051-4651/10 $26.00 © 2010 IEEE
DOI 10.1109/ICPR.2010.377

IEEE computer society

where $\boldsymbol{r}(\boldsymbol{x}) := \boldsymbol{\Lambda} \cdot (P(\omega_1|\boldsymbol{x}), \ldots, P(\omega_c|\boldsymbol{x}))^T$ and $q \in \mathbb{R}$ independently of the network parameters. In particular, if $\boldsymbol{\Lambda}$ equals the identity matrix and the network is flexible enough the outputs of an MSE optimized network approximate in the finite sample case the class posteriori probabilities $P(\omega_i|\boldsymbol{x})$.

In [5] it is proved, that minimizing the true risk (2) using a linear output network, i.e. $\boldsymbol{v}^{out}(\boldsymbol{x}) = \mathbf{W}\boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{b}$ with last hidden layer outputs $\boldsymbol{h}(\boldsymbol{x}) \in \mathbb{R}^m$, is equivalent to a maximization of a discriminant criterion $Q_{\boldsymbol{h}}$ evaluated in the m-dimensional space spanned by the last hidden layer outputs. For a finite number N of samples $\mathcal{O}_N$ the target coding scheme

$$\boldsymbol{\Lambda}_{j,i} := \begin{cases} \frac{\sqrt{N_i/N}}{P(\omega_i)} & \text{if } j = i \\ 0 & \text{otherwise} \end{cases}, \tag{3}$$

where N_i/N is the fraction of examples of class ω_i, results in a maximization of the well-known Fisher discriminant criterion used in Linear Discriminant Analysis (LDA):

$$Q_{\boldsymbol{h}} = \mathit{trace}\left\{\mathbf{S}_T^{-1}\mathbf{S}_B\right\}, \tag{4}$$

with the common total scatter-matrix $\mathbf{S}_T$ and between-class scatter-matrix $\mathbf{S}_B$.

This leads us to extend the classical LDA to a Generalized Discriminant Analysis (GerDA) using nonlinearities learned by a DNN.

2.1. Generalized Discriminant Analysis (GerDA)

In contrast to the classical LDA, GerDA aims to learn a nonlinear transformation into a low-dimensional feature space suitable for simple linear classification of arbitrarily distributed raw data. With the pre-optimization described in this section, DNNs are not initialized randomly but in a way that enables learning of complicated nonlinear functions.

Figure (1) depicts a multilayer stack of Restricted Boltzmann Machines (RBM) [1], composing a complete GerDA-DNN. Each layer consists of a RBM with biases $(\boldsymbol{b}^{v^i}, \boldsymbol{b}^{h^i})$ and weights $\mathbf{W}^i$ interconnecting binary *visual units* $\mathbf{v}^i$ and binary *hidden units* $\boldsymbol{h}^i$. Pre-optimization goes as follows: While clamping a sample to the visual units of a trained RBM, the activations of the RBM's hidden units are used as input samples for training a successor RBM. This procedure is repeated until all RBMs are trained. Training of a single RBM is performed by minimizing the Kullback-Leibler divergence

$$d(P^0||P^\infty; \boldsymbol{\Theta}) := \sum_{\boldsymbol{v}^i} P^0(\boldsymbol{v}^i) \log\left(\frac{P^0(\boldsymbol{v}^i)}{P^\infty(\boldsymbol{v}^i; \boldsymbol{\Theta})}\right) \tag{5}$$

with respect to the parameters $\boldsymbol{\Theta} = (\mathbf{W}^i, \boldsymbol{b}^{v^i}, \boldsymbol{b}^{h^i})$. That means the optimal parameters $\boldsymbol{\Theta}^*$ maximize the likelihood $\prod_{n=1}^N P^\infty(\boldsymbol{v}^i = \boldsymbol{x}_n; \boldsymbol{\Theta})$ with

$$P^\infty(\boldsymbol{v}^i; \boldsymbol{\Theta}) = \sum_{\boldsymbol{h}^i} P^\infty(\boldsymbol{v}^i, \boldsymbol{h}^i; \boldsymbol{\Theta}), \tag{6}$$

$$P^\infty(\boldsymbol{v}^i, \boldsymbol{h}^i; \boldsymbol{\Theta}) := \frac{\exp\left((\boldsymbol{v}^i)^T\mathbf{W}^i\boldsymbol{h}^i + \boldsymbol{b}^{v^i} + \boldsymbol{b}^{h^i}\right)}{Z(\boldsymbol{\Theta})}. \tag{7}$$

It is well-known that the computation of the gradient of (5) is intractable in general, because it involves the determination of empirical expectations w.r.t. samples drawn from the distribution (7) using a Markov chain Monte Carlo sampler. For making RBM learning practical, Contrastive Divergence (CD) [2] minimizes the difference of two Kullback-Leibler distances

$$CD_n(\boldsymbol{\Theta}) := d(P^0||P^\infty; \boldsymbol{\Theta}) - d(P^n||P^\infty; \boldsymbol{\Theta}) \tag{8}$$

instead of (5). Here, P^n denotes the distribution of states if the Markov chain for actually sampling from P^∞ is already stopped after a small number $n > 0$ of steps. The CD heuristic works well in practice for

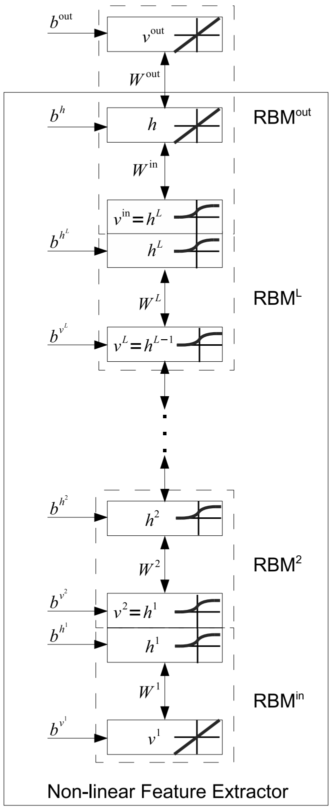

Figure 1. Multilayer RBM stack of GerDA.

$n = 1$ and thus reduces substantially the effort of determining the gradient, because only P^1 is involved. Sampling from P^1 is best done by a 1-step Gibbs sampler, i.e. first simulating all hidden states then simulating all visual states according to

$$P^\infty(h_j^i = 1|\boldsymbol{v}^i) = \frac{1}{1 + \exp\left(-(\mathbf{W}^i_{(\cdot)j})^T \boldsymbol{v}^i - b_j^{h^i}\right)}, \tag{9}$$

$$P^\infty(v_k^i = 1|\boldsymbol{h}^i) = \frac{1}{1 + \exp\left(-\mathbf{W}^i_{k(\cdot)} \boldsymbol{h}^i - b_k^{v^i}\right)}. \tag{10}$$

Note, $\mathbf{W}^i_{(\cdot)j}$ denotes the j-the column of the matrix $\mathbf{W}^i$, respectively $\mathbf{W}^i_{k(\cdot)}$ denotes the k-th row.

While an autoencoder [3] minimizes the reconstruction error between an input and output pattern, we aim at a DNN that is pre-optimized with respect to a discriminant criterion (4). For this purpose, we adapt the pre-optimization for a regression of real-valued target codes in (3): First, the input RBM's units $\boldsymbol{v}^1$ (RBM^{in} Fig. (1)) are modified to facilitate continuous measurements instead of binary inputs. Second, we extend the output RBM (RBM^{out} Fig. (1)) with extra continuous units $\boldsymbol{v}^{out}$ for learning a mapping from input states to output states. Likewise, the hidden units $\boldsymbol{h}$ are modeled continuously, because we consider the extraction of continuous features. However, the output RBM's input units $\boldsymbol{v}^{in}$ remained binary valued. As third modification, we adapt the CD heuristic (8) for the learning of input-output associations. That means, we want to maximize the likelihood w.r.t. a conditional density $p^0(\boldsymbol{v}^{out}|\boldsymbol{v}^{in})$.

In the end, the pre-optimized weights and biases up to the last hidden layer of a complete RBM stack (box in Fig. (1)) are used to initialize the non-linear feature extractor. A subsequent fine-tuning by a modified back-propagation maximizes the discriminant criterion (4) in the feature space spanned by the last hidden layer units.

2.2. Units for Real-valued Data

In order to adapt the units $\boldsymbol{v}^1$, $\boldsymbol{h}$ and $\boldsymbol{v}^{out}$ (Fig. (1)) to facilitate real-valued data, we extend the state distribution (7) such that the conditional distributions factorize into Gaussian distributions [6]. It follows for the input RBM

$$P^\infty(h_j^1 = 1|\boldsymbol{v}^1) = \frac{1}{1 + \exp\left(-(\mathbf{W}^1_{(\cdot)j})^T \boldsymbol{\Sigma}^1 \boldsymbol{v}^1 - b_j^{h^1}\right)} \tag{11}$$

with diagonal matrix $\boldsymbol{\Sigma}^1 := diag((1/\sigma_k^1)_k)$ and

$$P^\infty(v_k^1|\boldsymbol{h}^1) = \frac{1}{\sigma_k^1\sqrt{2\pi}} \exp\left(-\frac{(v_k^1 - \mu_k^1)^2}{2(\sigma_k^1)^2}\right) \tag{12}$$

with mean $\mu_k^1 := \sigma_k^1 \mathbf{W}^1_{k(\cdot)} \boldsymbol{h}^1 + b_k^{v^1}$. Similar, for the output RBM, we get

$$p^\infty(v_k^{out}|\boldsymbol{h}) = \frac{1}{\sigma_k^{out}\sqrt{2\pi}} \exp\left(-\frac{(v_k^{out} - \mu_k^{out})^2}{2(\sigma_k^{out})^2}\right) \tag{13}$$

with mean $\mu_k^{out} := b_k^{out} + \sigma_k^{out} \mathbf{W}^{out}_{k(\cdot)} \boldsymbol{h}$ and

$$p^\infty(h_j|\boldsymbol{v}^{in}, \boldsymbol{v}^{out}) = \frac{1}{\sigma_j^h\sqrt{2\pi}} \exp\left(-\frac{\left(h_j - \mu_j^h\right)^2}{2(\sigma_j^h)^2}\right) \tag{14}$$

with mean $\mu_j^h := b_j^h + \sigma_j^h (\boldsymbol{v}^{in})^T \mathbf{W}^{in}_{(\cdot)j} + (\sigma_j^h)^2 (\boldsymbol{v}^{out})^T \boldsymbol{\Sigma}^{out} \mathbf{W}^{out}_{(\cdot)j}$ and diagonal matrix $\boldsymbol{\Sigma}^{out} := diag((1/\sigma_k^{out})_k)$.

2.3. CD-learning of Input-Output Relations

For learning the real-valued target codes (3) at the output RBM (Fig. (1)), we modified the CD heuristic (8) by minimizing:

$$CD_n^{IO}(\boldsymbol{\Theta}) := d(p^0||p^\infty; \boldsymbol{\Theta}) - d(p^n||p^\infty; \boldsymbol{\Theta}) \tag{15}$$

with

$$d(p^n||p^\infty; \boldsymbol{\Theta}) := \sum_{\boldsymbol{v}^{in}} \int_{\mathbb{R}^{N_{\boldsymbol{v}^{out}}}} p^n(\boldsymbol{v}^{out}|\boldsymbol{v}^{in}; \boldsymbol{\Theta}) \cdot$$
$$P^0(\boldsymbol{v}^{in}) \log\left(\frac{p^n(\boldsymbol{v}^{out}|\boldsymbol{v}^{in}; \boldsymbol{\Theta})}{p^\infty(\boldsymbol{v}^{out}|\boldsymbol{v}^{in}; \boldsymbol{\Theta})}\right) d\boldsymbol{v}^{out} \tag{16}$$

and in particular $p^0(\boldsymbol{v}^{out}|\boldsymbol{v}^{in}; \boldsymbol{\Theta}) := p^0(\boldsymbol{v}^{out}|\boldsymbol{v}^{in})$. The benefit from our CD^{IO} heuristic is, that we can efficiently simulate samples drawn from p^1 using Gibbs-sampling and the distributions (13) and (14). Thus, the computation of the gradient of (15) for a stochastic gradient descent in the output RBM's parameter space is now tractable.

3. Experiments

For the first experiments with GerDA, we used the MNIST handwritten digit database [4] (28x28 pixels grayscale images) and the Statlog Satellite Image (Satimage) database (36-dimensional raw features). MNIST is an established benchmark database while Satimage is a very small dataset compared to MNIST. The MNIST database contains a total of 60,000 training samples and 10,000 test samples. The Satimage database consists of 4,435 training and 2,000 test samples of 6 different classes. All samples were normalized to zero mean and unity variance using empirical estimates from the training data.

Table 1. Classification Errors (MNIST)

	Test error in %	
Classifier	784dim. raw data	10dim. GerDA
linear	12.0 [4]	**1.58**
SVM lin.	—	1.46
KNN	2.83 [4]	**1.49**
SVM RBF	**1.4** [4]	1.47

Table 2. Classification Errors (Satimage)

	Test error in %		
Classifier	36dim. raw data	6dim. LDA	6dim. GerDA
linear	17.35	17.35	**9.80**
SVM lin.	16.85	19.20	**9.90**
KNN	**9.35**	11.85	11.70
SVM RBF	10.15	12.65	**10.05**

GerDA was combined with a linear classifier based on the Mahalanobis distance with equal covariance matrices as well as with a linear SVM. Additionally, we examined the performance of non-linear classifiers, namely the SVM with RBF kernel function and a KNN classifier. In case of Satimage, we also compared the GerDA features with LDA features. In all experiments, we reduced the dimensionality of the raw input features down to the number of classes.

For tuning the DNN topology and the classifiers, 10,000 samples from the original MNIST training set were used as validation set. The validation set for the Satimage experiments comprised 435 training samples. It turned out, that for MNIST a 784-1000-2000-1000-**10** topology, i.e. 3 hidden layers of 1,000, 2,000 and 1,000 units, and for Satimage a 36-40-40-**6** topology performed best on the validation set with respect to a linear classification.

Tab. (1) and Tab. (2) summarizes the results of our experiments. Both tables show the results on the raw data (784-dimensional and 36-dimensional resp.) in the first column. With GerDA features (10-dimensional and 6-dimensional resp.), simple linear classification in very low dimensions yields better or similar results than nonlinear classification of the raw data in high dimensions. Moreover, with GerDA features the accuracy of nonlinear classifiers is not considerably superior to the accuracy of linear ones. In Tab. (2) the results for GerDA features are also compared with repective LDA results. The comparison with LDA impressively shows that GerDA copes well with multimodal data distributions.

4. Conclusion

We describe a Generalized Discriminant Analysis (GerDA) for building pattern recognition systems with little expert knowledge and manual effort. At the core of the GerDA framework is a process for learning feature extractors based on DNNs. The use of GerDA features renders classification simple. It becomes simple in terms of the type of classifier required for excellent results and the computational resources required for the actual pattern matching.

In a two-stage training process, we first pre-optimize a DNN using stochastic learning. This stage involves layerwise training and stacking RBMs. The training of all but the topmost RBM is done using the well known CD algorithm. This part of the pre-optimization process is unsupervised. In practical terms, it can be interpreted as an adaptation of the network to the pattern world of the particular data domain. For the topmost RBM we use a novel modification of the CD heuristics. Together with an appropriate target coding, the pre-optimization is finalized using class information, i.e. with a supervised training scheme aiming to maximize a discriminant criterion. The pre-optimization of the DNN is a prerequisite for the success of the final training stage using a special back-propagation algorithm that maximizes the Fisher criterion in the feature space defined by the last hidden layer of the network.

Our results on the MNIST and Satimage databases are competitive with the best results reported so far. Although ongoing research work on GerDA may lead to even better recognition rates for these databases, our main focus is on the simplicity of using GerDA as a tool for building pattern recognition systems.

References

[1] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive Science*, 9:147–169, 1985.

[2] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

[3] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *SCIENCE*, 313:504–507, 2006.

[4] Y. LeCun and C. Cortes. The mnist database of handwritten digits; http://yann.lecun.com/exdb/mnist.

[5] H. Osman and M. M. Fahmy. On the discriminatory power of adaptive feed-forward layered networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16:837–842, 1994.

[6] M. Welling, M. Rosen-Zvi, and G. Hinton. Exponential family harmoniums with an application to information retrieval. *NIPS*, 17:1481–1488, 2005.