Feature Selection Guided Auto-Encoder

Shuyang Wang,[1] **Zhengming Ding,**[1] **Yun Fu**[1,2]
[1]Department of Electrical & Computer Engineering,
[2]College of Computer & Information Science,
Northeastern University, Boston, MA, USA
{shuyangwang, allanding, yunfu}@ece.neu.edu

Abstract

Recently the auto-encoder and its variants have demonstrated their promising results in extracting effective features. Specifically, its basic idea of encouraging the output to be as similar as input, ensures the learned representation could faithfully reconstruct the input data. However, one problem arises that not all hidden units are useful to compress the discriminative information while lots of units mainly contribute to represent the task-irrelevant patterns. In this paper, we propose a novel algorithm, *Feature Selection Guided Auto-Encoder*, which is a unified generative model that integrates feature selection and auto-encoder together. To this end, our proposed algorithm can distinguish the task-relevant units from the task-irrelevant ones to obtain most effective features for future classification tasks. Our model not only performs feature selection on learned high-level features, but also dynamically endows the auto-encoder to produce more discriminative units. Experiments on several benchmarks demonstrate our method's superiority over state-of-the-art approaches.

Introduction

When dealing with high-dimensional data, the curse of dimensionality is a fundamental difficulty in many practical machine learning problems (Duda, Hart, and Stork 2001). For many real-world data (e.g., video analysis, bioinformatics), their dimensions are usually very high, which results in the significant increase of the computational time and space. In practice, not all features are equally important and discriminative, since most of them are often highly correlated or even redundant to each other (Guyon and Elisseeff 2003). The redundant features generally would make learning methods over-fitting and less interpretable. Consequently, it is necessary to reduce the data dimensionality and select the most important features.

Recently, the auto-encoder and its variants have drawn increasing attention as nonlinear dimensionality reduction methods (Hinton and Salakhutdinov 2006; Wang, Ding, and Fu 2016). The conventional auto-encoder tries to learn an approximation to the identity by encouraging the output to be as similar to the input as possible. The architecture forces the network to seek a compressed representation of the data while preserving the most important information. However, this scheme leads to one problem that the majority of the learned high-level features may be blindly used to represent the irrelevant patterns in the training data. Although the effort to incorporate supervision (Socher et al. 2011) has been deployed, it is still challenging to learn task-relevant hidden-layer representation since there must be some hidden units mainly used to faithfully reconstruct the irrelevant or noisy part of the input. It is unreasonable to endow the discriminablity to this kind of task-irrelevant units. Take object recognition for example, lots of hidden units are mainly used to reconstruct the background clutters, then its performance could be improved significantly if we can distinguish important hidden units (e.g., those encoding foreground) from large amounts of distracting hidden units (e.g., those encoding background).

To address this issue, a unified framework is proposed to integrate feature selection and auto-encoder (Fig.1). Intuitively, the feature selection is applied on learned hidden-layer to extract the discriminative features from the irrelevant ones. Simultaneously, the task-relevant hidden units can feed back to optimize the encoding layer to achieve more discriminability only on selected hidden units. Therefore, our model not only performs dynamic feature selection

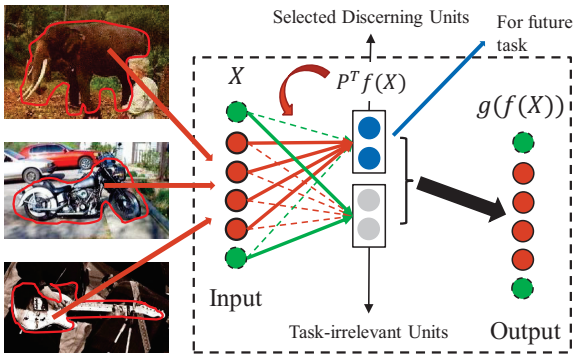


Figure 1: The feature selection is adopted in hidden layer to distinguish discerning units from task-irrelevant units, which in turn constrains the encoder to focus on compressing important patterns to selected units. All of the units are contributed to reconstruct the input, while only selected units are used for future tasks.

on high-level features, but also separates important and irrelevant information into different groups of hidden units separately through a joint learning mechanism with auto-encoder. We highlight our main contributions as follows:

- We propose the Feature Selection Guided Auto-Encoder (FSAE) that jointly performs feature selection and auto-encoder in a unified framework. The framework selects the discerning high-level features and simultaneously enhances the discriminability on the selected units.
- Our proposed method can be extended to different scenarios (e.g., classification, clustering), by shifting feature selection criterion (e.g., Fisher score, Laplacian score) on the hidden layer.
- The proposed FSAE can be adopted as a building block to form a stacked deep network. We deploy several experiments to demonstrate the effectiveness of our algorithm by comparing with state-of-the-art approaches.

Related work

Two lines of related works, feature selection and auto-encoder, are introduced in this section.

Feature selection The past decade has witnessed a number of proposed feature selection criterions, such as Fisher score (Gu, Li, and Han 2012), Relief (Liu and Motoda 2007), Laplacian score (He, Cai, and Niyogi 2005), and Trace Ratio criterion (Nie et al. 2008). In detail, suppose the original set of features denoted as $\mathbb{S}$, the goal is to find a subset $\mathbb{T}$ to maximize the above performance criterion $\mathcal{C}$,

$$\mathbb{T} = \arg\max_{\mathbb{T}\subseteq\mathbb{S}} \mathcal{C}(\mathbb{T}), \ \text{ s.t. } |\mathbb{T}| = m, m \ll d,$$

where m and d are the feature dimension of selected and original, respectively. It often requires prohibitively expensive computational cost in this combinatorial optimization problem. Therefore, instead of subset-level selection, one common traditional method first calculates the score of each feature independently and then select the top-m ranked features (feature-level selection). However, such features selected one by one are suboptimal, which neglects the subset-level score and results in discarding good combination of features or preserving redundant features. To address this problem, (Nie et al. 2008) and (Gu, Li, and Han 2012) proposed globally optimal solution based on Trace Ratio criterion and Fisher score respectively.

Auto-encoder is usually adopted as a basic building block to construct a deep structure (Hinton and Salakhutdinov 2006; Ding, Shao, and Fu 2016). To encourage structural feature learning, further constraints have been imposed on parameters during model training. Sparse Auto-Encoder (SAE) was proposed to constrain the average response of each hidden unit to a small value (Coates, Ng, and Lee 2011). Yu et al. proposed a graph regularized auto-encoder, aiming to adopt graph to guide the encoding and decoding (Yu et al. 2013). However, it is still challenging to learn with lots of irrelevant patterns in the data, and current auto-encoder variants have not yet considered the hidden units into two parts, one is task-relevant and the other is task-irrelevant. In our paper, we adopt feature selection on the hidden layer of auto-encoder, which aims at guiding the encoder to compress task-relevant and irrelevant information into two groups of hidden units.

The Proposed Algorithm

In this section, we first provide the preliminary and motivation of our proposed algorithm, followed by our detailed model by jointly selecting features and training auto-encoder. Then, we discuss two most relevant algorithms to our approach. Moreover, the deep architecture is described.

Preliminary and Motivations

The general idea of auto-encoder is to represent the data through a nonlinear encoder to a hidden layer and use the hidden units as the new feature representations:

$$h_i = \sigma(W_1 x_i + b_1); \quad \hat{x}_i = \sigma(W_2 h_i + b_2) \tag{1}$$

where $h_i \in \mathbb{R}^z$ is the hidden representation, and $\hat{x_i} \in \mathbb{R}^d$ is interpreted as a reconstruction of normalized input $x_i \in \mathbb{R}^d$. The parameter set includes weight matrices $W_1 \in \mathbb{R}^{z\times d}, W_2 \in \mathbb{R}^{d\times z}$, and offset vectors $b_1 \in \mathbb{R}^z$, $b_2 \in \mathbb{R}^d$ with dimensionality z and d. σ is a non-linear activation function (e.g., sigmoid).

The auto-encoder with single hidden layer is generally a neural network with identical input and target, namely,

$$\min_{W_1,W_2,b_1,b_2} \frac{1}{2n}\sum_{i=1}^{n} \|x_i - \hat{x}_i\|_2^2, \tag{2}$$

where n is sample size of the data, $\hat{x}_i$ is the reconstructed output and x_i is the target. A good representation thus can be obtained with the ability to well reconstruct the data.

As we mentioned before, all the high-level hidden units contribute to capture the intrinsic information of input data during data reconstruction, however, these units are not equally important in terms of our classification task. For example, some units play an essential role to reconstruct the background in an object image, but they have nothing to do with our final object classification task. We consider these units as task-irrelevant units, which are undesirable in our learned new features. On the other hand, since the traditional unsupervised model has limited capacity to model the marginal input distribution for the goal supervised task, some existing works exploited the label information on hidden units using a softmax layer (Socher et al. 2011). Considering the previous assumption about task-irrelevant units, it is inappropriate or even counterproductive to endow all the hidden units with discriminability.

Therefore, we have two conclusions: 1) feature selection is essential to distinguish discerning units out of task-irrelevant units, and 2) the discriminative information should be only applied on the selected task-relevant units. Based on above discussion, we propose our joint feature selection and auto-encoder model in a unified framework.

Feature Selection Guided Auto-Encoder

In this section, we propose our joint learning framework by integrating feature selection and auto-encoder together.

Specifically, we incorporate feature selection on the hidden layer units. Assume $X \in \mathbb{R}^{d\times n}$ is the training data, with d as the dimensionality of the visual descriptor and n as the number of data samples.

$$\min_{W_1,W_2,b_1,b_2,P} \frac{1}{2}\|X - g(f(X))\|_F^2 + \frac{\lambda}{2}\mathcal{C}(P, f(X)) \quad (3)$$

where $f(X) = \sigma(W_1X + B_1), g(f(X)) = \sigma(W_2f(X) + B_2)$, B_1, B_2 are the n-repeated column copy of b_1, b_2, respectively. $\mathcal{C}(P, f(X))$ is the feature selecting regularized term, with a learned feature selection matrix P performing on hidden units $f(X)$. Specifically, i-th column vector in P denoted by $p_i \in \mathbb{R}^z$ has the form,

$$p_i = [\underbrace{0, ..., 0}_{j-1}, 1, \underbrace{0, ..., 0}_{z-j}]^\top. \quad (4)$$

where z is the number of hidden units and j means this column vector selects the j-th units into the subset of new features $y \in \mathbb{R}^m$. Then the feature selection procedure can be expressed as given original feature $f(X)$, finding a matrix P to select a new feature set $Y = P^\top f(X)$ which optimizes an appropriate criterion $\mathcal{C}(P, f(X))$.

Generally, feature selection can be mainly split into three fashions: unsupervised, supervised and semi-supervised. That is, supervised models could preserve more discriminative information during feature selection, e.g., Fisher score (Gu, Li, and Han 2012), while unsupervised models aim to preserve more intrinsic data structures, e.g., Laplacian score (He, Cai, and Niyogi 2005). Aiming to deal with different cases in real world, we design the $\mathcal{C}(P, f(X))$ in a more general way. Specifically, we propose the following general feature selection regularizer as:

$$\mathcal{C}(P, f(X)) = \frac{\mathrm{tr}(P^\top f(X)L_w f^\top(X)P)}{\mathrm{tr}(P^\top f(X)L_b f^\top(X)P)}, \quad (5)$$

which provides a general model to fit in different scenarios by adapting L_w and L_b in different ways. In this paper, we adopt Fisher score (Gu, Li, and Han 2012; Nie et al. 2008) as our supervised feature selection method for classification. For Fisher score, two weighted undirected graphs G_w and G_b are constructed on given data (here, we use original input data X to preserve the geometric structure during selecting features), which respectively reflect the within-class and between-class affinity relationship (Yan et al. 2007). Correspondingly, two weighted matrices S_w and S_b are produced to characterize two graphs respectively. Therefore, we obtain the Laplacian matrices defined as $L_w = D_w - S_w$, where D_w is the diagonal matrix of S_w, similar for L_b and S_b.

By solving the following optimization problem, we can obtain the feature selection matrix P which produces the feature subset with the minimum criterion score:

$$P = \arg\min_P \frac{\mathrm{tr}(P^\top f(X)L_w f^\top(X)P)}{\mathrm{tr}(P^\top f(X)L_b f^\top(X)P)}, \quad (6)$$

Unfortunately, there has never been a straightforward issue to solve the above trace-ratio problem, due to the unavailable of closed-form solution. Thus, instead of directly dealing with trace-ratio problem, many works tend to transform it to an equivalent trace-difference problem to achieve a globally optimal solution (Nie et al. 2008).

Suppose the subset-level criterion score $\mathcal{C}(P, f(X))$ in Eq.(5) reaches the global minimum γ^* satisfying,

$$\gamma^* = \arg\min \frac{\mathrm{tr}(P^\top f(X)L_w f^\top(X)P)}{\mathrm{tr}(P^\top f(X)L_b f^\top(X)P)}, \quad (7)$$

that is to say,

$$\begin{aligned} &\frac{\mathrm{tr}(P^\top f(X)L_w f^\top(X)P)}{\mathrm{tr}(P^\top f(X)L_b f^\top(X)P)} \geq \gamma^*, \forall P \\ &\Rightarrow \mathrm{tr}(P^\top f(X)(L_w - \gamma^* L_b)f^\top(X)P) \geq 0, \forall P \\ &\Rightarrow \min_P \mathrm{tr}(P^\top f(X)(L_w - \gamma^* L_b)f^\top(X)P) = 0. \end{aligned}$$

To this end, we can define the function of γ when treating others as constant as:

$$r(\gamma) = \min_P \mathrm{tr}(P^\top f(X)(L_w - \gamma L_b)f^\top(X)P). \quad (8)$$

Therefore, finding the global optimal γ can be converted to finding the root of equation $r(\gamma) = 0$, which is a trace-difference problem. Note that $r(\gamma)$ is a monotonically increasing function (Nie et al. 2008). By introducing the above trace-difference optimization problem Eq.(8) into the hidden layer of auto-encoder updating, we reformulate our final objective function as:

$$\begin{aligned} \min_{W_1,W_2,b_1,b_2,P,\gamma} \mathcal{L} = &\frac{1}{2}\|X - g(f(X))\|_F^2 + \\ &\frac{\lambda}{2}\mathrm{tr}(P^\top f(X)(L_w - \gamma L_b)f^\top(X)P), \end{aligned} \quad (9)$$

where λ is the balance parameter between auto-encoder and feature selection term. γ is the optimized trace ratio score obtained with P in previous trace ratio optimization problem.

Optimization

Eq.(9) is hard to solve due to the complex non-linearity of the encoder and decoder, so the alternating optimization approach is employed to iteratively update the auto-encoder parameters W_1, W_2, b_1, b_2 and feature selection variable P as well as γ. Specifically, we solve the optimization with two sub-problems, one is feature selection score learning and the other is the regularized auto-encoder optimization.

Feature Selection Score Learning When the parameters of auto-encoder are fixed, we could optimize the feature selection score γ and feature selection matrix P in a traditional trace-ratio strategy. Specifically, we follow trace-difference equation

$$P = \min_P \mathrm{tr}(P^\top f(X)(L_w - \gamma L_b)f^\top(X)P), \quad (10)$$

Suppose P_t is the optimal result in t-th optimization iteration, thus γ_t is calculated by

$$\gamma_t = \frac{\mathrm{tr}(P_t^\top f(X)L_w f^\top(X)P_t)}{\mathrm{tr}(P_t^\top f(X)L_b f^\top(X)P_t)}, \quad (11)$$

Algorithm 1: Optimization for trace-ratio problem

Input: Learned hidden layer feature $f(X)$, selected feature number m, matrices L_w and L_b

Initialize: $P = I$, $I \in \mathbb{R}^{z\times m}$ is the identity matrix, γ with Eq.(11), $\epsilon = 10^{-9}$, $iter = 0$, $maxiter = 10^3$

while *not converge and* $iter \leq maxiter$ **do**

  Calculate the score of each j-th feature with Eq.(11) by setting $P = [\underbrace{0, ..., 0}_{j-1}, 1, \underbrace{0, ..., 0}_{z-j}]^\top$.

  Rank the features with the scores in ascending order

  Select the leading m features to update $P \in \mathbb{R}^{z\times m}$

  Calculate γ with Eq.(11)

  Check the convergence conditions: $\|\gamma_{old} - \gamma\| < \epsilon$

end

Output: feature selection matrix P, global optimal score γ

Therefore, we can obtain $r(\gamma_t)$ as

$$r(\gamma_t) = \mathrm{tr}(P_{t+1}^\top f(X)(L_w - \gamma_t L_b) f^\top(X) P_{t+1}), \quad (12)$$

where P_{t+1} can be efficiently calculated according to each single feature's score rank. The root of equation $r(\gamma) = 0$ and the optimal solution for Eq.(6) can be obtained through this iterative procedure. Note that γ is updated as the global optimal score for the feature selection criterion, and works as a parameter in next auto-encoder updating procedure. **Algorithm 1** summarizes the optimization solution. More details on globally optimal solution and its proof could be referred to (Nie et al. 2008).

Regularized Auto-encoder Learning When P and γ are fixed, we can employ the stochastic sub-gradient descent method to obtain the parameters W_1, b_1, W_2 and b_2. The gradients of the objective function $\mathcal{L}$ in Eq.(9) with respect to the decoding parameters are computed as follows:

$$\frac{\partial \mathcal{L}}{\partial W_2} = (X - g(f(X))) \odot \frac{\partial g(f(X))}{\partial W_2} f^\top(X),$$

$$\frac{\partial \mathcal{L}}{\partial B_2} = (X - g(f(X))) \odot \frac{\partial g(f(X))}{\partial W_2} = \mathcal{L}_2,$$

$$\frac{\partial \mathcal{L}}{\partial W_1} = (W_2^\top \mathcal{L}_2 + \lambda P P^\top f(X)(L_w - \gamma L_b)) \odot \frac{\partial f(X)}{\partial W_2} X^\top,$$

$$\frac{\partial \mathcal{L}}{\partial B_1} = (W_2^\top \mathcal{L}_2 + \lambda P P^\top f(X)(L_w - \gamma L_b)) \odot \frac{\partial f(X)}{\partial W_2}.$$

Then, W_1, W_2 and b_1, b_2 can be updated with the gradient descent algorithm as follows:

$$\begin{aligned} W_1 &= W_1 - \eta \frac{\partial \mathcal{L}}{\partial W_1}, \quad b_1 = b_1 - \eta \frac{\partial \mathcal{L}}{\partial b_1}, \\ W_2 &= W_2 - \eta \frac{\partial \mathcal{L}}{\partial W_1}, \quad b_2 = b_2 - \eta \frac{\partial \mathcal{L}}{\partial b_2}, \end{aligned} \quad (13)$$

where η is the learning rate. $\frac{\partial \mathcal{L}}{\partial b_1}$ and $\frac{\partial \mathcal{L}}{\partial b_2}$ are the column mean of $\frac{\partial \mathcal{L}}{\partial B_1}$ and $\frac{\partial \mathcal{L}}{\partial B_2}$, respectively. To sum up, the above two sub-problems could be updated iteratively. **Algorithm 2** summarizes the details of the optimization.

Algorithm 2: Solving Problem Eq.(9)

Input: Training data X, Parameters λ, layersize, select feature number $m < z$

Initialize: W_1, W_2, b_1 and b_2 are initialized with original auto-encoder, maxiter = 50, iter=0, $\epsilon = 10^{-7}$

while *not converged and* $\mathrm{iter} \leq \mathrm{maxiter}$ **do**

  Fix others and update P and γ using Eq. (10);

  Fix P and update W_1, W_2, b_1 and b_2 with Eq. (13);

  Check the convergence conditions: $\|\mathcal{L}_{new} - \mathcal{L}_{old}\|_\infty < \epsilon$

end

Output: $W_1, W_2,\ b_1,\ b_2,\ P$ ($Y = P^\top \sigma(W_1 X + B_1)$ could be used as input of next FSAE, to form stack architecture)

Testing: new feature represented with: $Y_{test} = P^\top \sigma(W_1 X_{test} + B_1)$

Relations to Existing Methods

Here we highlight our model's advantage by elaborating some connections with related methods.

Sparse AE: Ranzato et al. developed a Sparse Auto-Encoders (SAE) whose idea behind is to enforce activations of hidden units to be close to the zero during training (Coates, Ng, and Lee 2011). That is, SAE aims to seek a small set of hidden units to reconstruct the input, as one sample may only link to a small number of hidden units. However, SAE still adopts to minimize the reconstruction loss to seek a new representation. Differently, our proposed algorithm desires to select a small set of hidden units most useful for classification, while other hidden units would be still used for reconstruction. That is, we need to endow the discriminative ability only on the important discerning units.

Graph regularized AE: Yu et al. proposed a graph regularized AE, which utilized graph structure to guide the feature learning during the AE training (Yu et al. 2013). The idea behind is straightforward, that is to preserve more geometric structure of the data during non-linear dimensionality reduction. Our proposed algorithm also adopts graph regularizer, however, the idea of our work is to preserve the geometric structure only on the selected hidden units.

Deep Architecture of FSAE

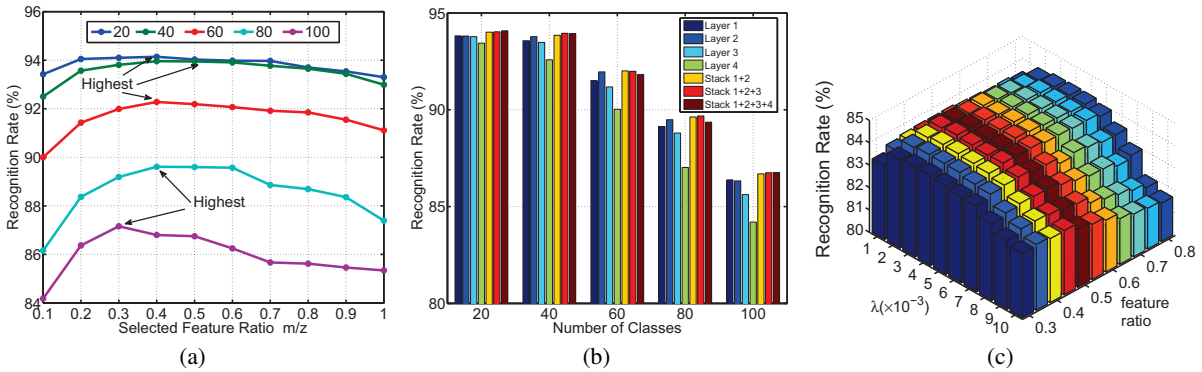
The proposed FSAE can be easily used as a building block to form a hierarchy. For example, we can first train a single-layer FSAE on the input images. Then the selected hidden units are worked as the input to feed in the next FSAE to obtain the stacked representation. Since the FSAE selects the task-relevant features with supervision, more discriminative information can be utilized in the higher-layer networks.

Experiments

We evaluate the proposed FSAE method through data classification on three benchmark datasets, including COIL100 (Nayar, Nene, and Murase 1996), Caltech101 (Fei-Fei, Fergus, and Perona 2007) and CMU-PIE (Sim, Baker, and Bsat 2002). LDA (Belhumeur, Hespanha, and Kriegman 1997), linear regression classification (LRC) (Naseem, Togneri, and Bennamoun 2010) and sparse auto-encoder (SAE) (Coates, Ng, and Lee 2011) are

Table 1: Average recognition rate(%) with standard deviations on COIL with different number of classes. Compared methods: subspace learning: NPE (He et al. 2005), LSDA (Cai et al. 2007), SRRS (Li and Fu 2015); dictionary learning: FDDL (Yang et al. 2011), DLRD (Ma et al. 2012), $D^2L^2R^2$ (Li, Li, and Fu 2014), DPL (Gu et al. 2014), LCLRD (Wang and Fu 2015).

Methods	20 objects	40 objects	60 objects	80 objects	100 objects	Average
LDA	81.94 ± 1.21	76.73 ± 0.30	66.16 ± 0.97	59.19 ± 0.73	52.48 ± 0.53	67.30
LRC	90.74 ± 0.71	89.00 ± 0.46	86.57 ± 0.37	85.09 ± 0.34	83.16 ± 0.64	86.91
SAE	91.28 ± 0.68	89.65 ± 0.77	87.26 ± 0.85	85.90 ± 0.61	84.46 ± 0.61	87.71
NPE	82.24 ± 2.25	76.01 ± 1.04	63.22 ± 1.36	52.18 ± 1.44	30.73 ± 1.31	60.88
LSDA	82.79 ± 1.70	75.01 ± 1.14	62.85 ± 1.41	51.69 ± 2.05	26.77 ± 1.05	59.82
FDDL	85.74 ± 0.77	82.05 ± 0.40	77.22 ± 0.74	74.81 ± 0.55	73.55 ± 0.63	78.67
DLRD	88.61 ± 0.95	86.39 ± 0.54	83.46 ± 0.15	81.50 ± 0.47	79.91 ± 0.59	83.97
$D^2L^2R^2$	90.98 ± 0.38	88.27 ± 0.38	86.36 ± 0.53	84.69 ± 0.45	83.06 ± 0.37	86.67
DPL	87.55 ± 1.32	85.05 ± 0.21	81.22 ± 0.21	78.78 ± 0.85	76.28 ± 0.94	81.77
LCLRD	92.15 ± 0.34	89.86 ± 0.49	87.23 ± 0.29	85.40 ± 0.61	84.15 ± 0.39	87.75
SRRS	92.03 ± 1.21	92.51 ± 0.65	90.82 ± 0.43	88.75 ± 0.71	85.12 ± 0.33	89.85
AE+FS [ours]	91.73 ± 0.75	90.24 ± 0.89	87.98 ± 0.81	86.42 ± 0.57	85.05 ± 0.60	88.28
FSAE [ours]	$\mathbf{94.12 \pm 0.45}$	$\mathbf{93.82 \pm 0.60}$	$\mathbf{91.97 \pm 0.94}$	$\mathbf{89.68 \pm 0.81}$	$\mathbf{86.75 \pm 0.79}$	**91.27**



(a) (b) (c)

Figure 2: (a) Recognition rates with different number of classes on COIL dataset in terms of feature selection ratio (m/z). The highest results appear mostly at 0.3∼0.5. (b) Recognition rates with different layersize setting on COIL dataset. (c) Effects of parameter selection of λ and selection ratio (m/z) on the classification accuracy on the CMU PIE database.

used as baseline algorithms on all datasets. What's more, for verifying the superiority of joint learning, we propose a simple combination framework as comparison, named as AE+FS, which first uses a traditional auto-encoder to learn a new representation, then the feature selection procedure is only applied on the obtained hidden layer to produce a subset of features. Besides above baseline methods, different state-of-the-art algorithms are compared in each dataset. Note that the layersize setting for those AE based methods are all the same and they only differ in the regularizer.

Parameter selection. In addition to the parameter of auto-encoder (layersize setting), there are two more parameters in our proposed objective function (Eq.(9)), which are balance parameter λ and feature selection ratio (m/z). Note that the parameter γ is automatically learned as global optimal score in feature selection step. The parameter λ balances the feature selection regularization and the loss function of AE, we empirically set it in our experiments and will give analysis in following sections. Specifically, λ is set as 2×10^{-3} for COIL100, 5×10^{-3} for Caltech101, 3×10^{-3} for CMU PIE. For selected feature size, we set it as 50% of the original hidden layer size for all the experiments, and we will analyze the impact of selection ratio.

COIL100 contains 7,200 color images of 100 objects (72 images per object) with different lighting conditions. The converted gray scale images with size 32×32 are used. 10 images per object are randomly selected to form the training set, the rest images for test. The random split is repeated 20 times, and the average results are reported with standard deviations. The experiments with different numbers of used objects (20, 40, 60, 80 and 100) are conducted to evaluate the scalability of our method. Each compared method is either tuned with parameters to achieve their best performance or directly copied from the original papers under same experimental setting. For all three AE based methods, we use three layers set as $[300, 200, 100]$, and report the results on the feature with three layer stack together.

Table.1 shows a large improvement on the recognition rates by our algorithm. Fig.2(a) and (b) show the analysis of FSAE with different values of selected feature ratio and layersize setting, respectively. We can observe from Fig.2(a)

Table 2: Average recognition rate(%) on Caltech101 with different number of training samples per class.

Methods	10Train	15Train	25Train	30Train
LDA	49.41	62.85	67.50	70.59
LRC	56.88	61.57	68.58	70.96
SAE	62.92	68.64	74.13	74.99
K-SVD	59.8	65.2	71.0	73.2
D-KSVD	59.5	65.1	71.1	73.0
SRC	60.1	64.9	69.2	70.7
LLC	59.77	65.43	70.16	73.44
LC-KSVD	63.1	67.8	72.3	73.6
SLRRC	-	66.1	-	73.6
LSAE	-	-	-	72.7
DPL	61.28	67.52	71.93	73.90
NILDFL	-	-	-	75.20
AE+FS [ours]	61.83	67.54	73.11	75.02
FSAE [ours]	**64.90**	**69.79**	**75.38**	**77.14**

that the highest recognition rate appears mostly when the selected feature number is 30%∼50% of original feature size. Therefore, we set the ratio to 0.5 for all experiments for simplicity. From Fig.2(b), we test the recognition rate using different layer of a stacked four layers FSAE with layersize $[300, 200, 100, 50]$. The figure indicates that using layer 2 or stacked together provide better and more robust results. Since the higher layer only uses the select important feature as input, the encoder could further focus on the task-relevant information, which results in better performance. The drops at layer 3 and 4 are probably due to the too few units to capture sufficient information. We also perform the convergence test of our proposed algorithm with 60 objects and show in Fig.3, which indicates the proposed method converges well.

Caltech101 is a widely used database for object recognition which contains a total of 9,144 images from 101 common object classes (animals, vehicles, trees, etc.) plus one background class (total 102 categories). Following the common experimental settings, we train on 10, 15, 25, and 30 random selected samples per category and test on the rest. We repeat the random spits 20 times to obtain reliable results. The final average recognition rates are reported. The spatial pyramid features are extracted following (Zhang, Jiang, and Davis 2013) and (Jiang, Lin, and Davis 2013) (three-level SIFT features with a codebook of size 1024 + PCA) with the final feature dimension of 1500. We compare with K-SVD (Aharon, Elad, and Bruckstein 2006), D-KSVD (Zhang and Li 2010), SRC (Wright et al. 2009), LLC (Wang et al. 2010), LC-KSVD (Jiang, Lin, and Davis 2013), SLRRC (Zhang, Jiang, and Davis 2013), LSAE (Luo et al. 2015), DPL (Gu et al. 2014) and NILDFL (Zhou, Lin, and Zhang 2016). The layersize setting for all AE based methods is $[500, 100]$.

The comparative results are shown in Table.2 and our approach consistently outperforms all the competitors. One observation is that, sometimes AE+FS performs worse than SAE. The basic reason is that, the Caltech dataset contains lots of background in each image, and simple feature selection on the final hidden units results in the lost of relevant information, while our joint learning framework could better distinguish the units and get improved results.

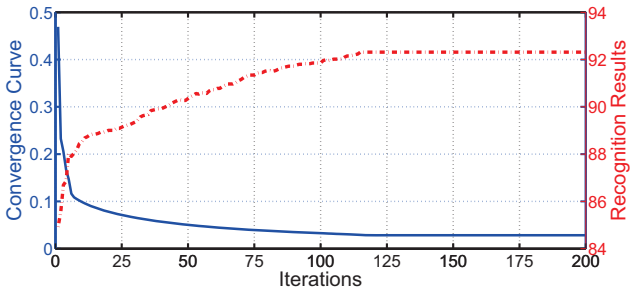


Figure 3: The optimization process of FSAE method on the COIL100 dataset with 60 objects.

CMU PIE dataset consists total of 41,368 face images from 68 identities, each with 13 different poses, 4 different expressions, and 43 different illumination conditions. We select five near frontal poses (C05, C07, C09, C27, C29) as a subset of PIE, and use all the images under different illuminations and expressions (totally 11,554 samples). Thus, there are about 170 images for each person and each image is normalized to the size of 32×32 pixels. We select different numbers of training samples per person to test these methods, and summarize the recognition rates in Table. 3. The experiments are repeated 20 times, and we use singly-layer with size 1500. Our method achieves good results and outperforms the compared methods. Also, the analysis of parameter λ and feature selection ration on 10 train CMU PIE are reported in Fig.2(c). The highest result is given when $\lambda = 3 \times 10^{-3}$ and selection ratio$= 0.5$.

Conclusion

In this work, we proposed a novel auto-encoder based framework, to joint training non-linear transformation and selecting informative features in a unified framework. Through these unified framework, the discerning hidden units were distinguished from the task-irrelevant units at hidden layer, and the regularizer on the selected features in turn enforces the encoder to focus on compress important patterns into selected units. As a general framework, different feature selection criterions could be fitted into our FSAE model depending on different tasks. What's more, a stacked architecture is also introduced using FSAE as building block. The supervised recognition results on three benchmark datasets indicated the effectiveness of our FSAE framework.

Acknowledgments

This work is supported in part by the NSF IIS award 1651902, NSF CNS award 1314484, ONR award N00014-12-1-1028, ONR Young Investigator Award N00014-14-1-0484, and U.S. Army Research Office Young Investigator Award W911NF-14-1-0218.

References

Aharon, M.; Elad, M.; and Bruckstein, A. 2006. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE TSP* 54(11):4311.

Table 3: Average recognition rate(%) with standard deviation on CMU PIE with different number of training samples per class.

Methods	5Train	10Train	20Train	30Train	40Train	50Train	60Train
LDA	57.18 ± 1.28	69.31 ± 0.82	78.51 ± 0.49	89.09 ± 0.31	92.19 ± 0.37	93.72 ± 0.14	94.57 ± 0.13
LRC	40.51 ± 1.04	68.81 ± 0.77	86.62 ± 0.63	91.85 ± 0.48	93.86 ± 0.46	95.09 ± 0.21	95.88 ± 0.18
SAE	69.33 ± 1.50	80.91 ± 1.10	88.99 ± 0.62	91.68 ± 0.36	93.02 ± 0.34	93.73 ± 0.31	94.08 ± 0.24
SRRS	60.02 ± 1.23	70.38 ± 1.31	80.17 ± 0.61	89.24 ± 0.32	92.38 ± 0.43	93.86 ± 0.31	94.93 ± 0.21
AE+FS [ours]	71.47 ± 1.42	83.69 ± 0.87	89.97 ± 0.56	93.19 ± 0.37	94.18 ± 0.29	94.78 ± 0.34	95.02 ± 0.27
FSAE [ours]	$\mathbf{72.43 \pm 1.32}$	$\mathbf{85.26 \pm 0.74}$	$\mathbf{91.93 \pm 0.57}$	$\mathbf{94.47 \pm 0.30}$	$\mathbf{95.73 \pm 0.24}$	$\mathbf{96.48 \pm 0.26}$	$\mathbf{96.79 \pm 0.18}$

Belhumeur, P. N.; Hespanha, J. P.; and Kriegman, D. J. 1997. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *TPAMI* 19(7):711–720.

Cai, D.; He, X.; Zhou, K.; Han, J.; and Bao, H. 2007. Locality sensitive discriminant analysis. In *IJCAI*, 708–713.

Coates, A.; Ng, A. Y.; and Lee, H. 2011. An analysis of single-layer networks in unsupervised feature learning. In *ICAIS*, 215–223.

Ding, Z.; Shao, M.; and Fu, Y. 2016. Deep robust encoder through locality preserving low-rank dictionary. In *ECCV*, 567–582. Springer.

Duda, R.; Hart, P.; and Stork, D. 2001. *Pattern Classification*. John Wiley & Sons, New York, 2 edition.

Fei-Fei, L.; Fergus, R.; and Perona, P. 2007. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *CVIU* 106(1):59–70.

Gu, S.; Zhang, L.; Zuo, W.; and Feng, X. 2014. Projective dictionary pair learning for pattern classification. In *NIPS*, 793–801.

Gu, Q.; Li, Z.; and Han, J. 2012. Generalized fisher score for feature selection. *arXiv preprint arXiv:1202.3725*.

Guyon, I., and Elisseeff, A. 2003. An introduction to variable and feature selection. *JMLR* 3:1157–1182.

He, X.; Cai, D.; Yan, S.; and Zhang, H.-J. 2005. Neighborhood preserving embedding. In *ICCV*, 1208–1213.

He, X.; Cai, D.; and Niyogi, P. 2005. Laplacian score for feature selection. In *NIPS*, 507–514.

Hinton, G. E., and Salakhutdinov, R. R. 2006. Reducing the dimensionality of data with neural networks. *Science* 313(5786):504–507.

Jiang, Z.; Lin, Z.; and Davis, L. S. 2013. Label consistent k-svd: Learning a discriminative dictionary for recognition. *TPAMI* 35(11):2651–2664.

Li, S., and Fu, Y. 2015. Learning robust and discriminative subspace with low-rank constraints. *TNNLS*.

Li, L.; Li, S.; and Fu, Y. 2014. Learning low-rank and discriminative dictionary for image classification. *IVC* 32(10):814–823.

Liu, H., and Motoda, H. 2007. *Computational methods of feature selection*. CRC Press.

Luo, W.; Yang, J.; Xu, W.; and Fu, T. 2015. Locality-constrained sparse auto-encoder for image classification. *Signal Processing Letters, IEEE* 22(8):1070–1073.

Ma, L.; Wang, C.; Xiao, B.; and Zhou, W. 2012. Sparse representation for face recognition based on discriminative low-rank dictionary learning. In *CVPR*, 2586–2593.

Naseem, I.; Togneri, R.; and Bennamoun, M. 2010. Linear regression for face recognition. *TPAMI* 32(11):2106–2112.

Nayar, S.; Nene, S. A.; and Murase, H. 1996. Columbia object image library (coil 100). *Department of Comp. Science, Columbia University, TR CUCS-006-96*.

Nie, F.; Xiang, S.; Jia, Y.; Zhang, C.; and Yan, S. 2008. Trace ratio criterion for feature selection. In *AAAI*, 671–676.

Sim, T.; Baker, S.; and Bsat, M. 2002. The cmu pose, illumination, and expression (pie) database. In *FG*, 46–51.

Socher, R.; Pennington, J.; Huang, E. H.; Ng, A. Y.; and Manning, C. D. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *EMNLP*, 151–161. Association for Computational Linguistics.

Wang, S., and Fu, Y. 2015. Locality-constrained discriminative learning and coding. In *CVPR*, 17–24.

Wang, J.; Yang, J.; Yu, K.; Lv, F.; Huang, T.; and Gong, Y. 2010. Locality-constrained linear coding for image classification. In *CVPR*, 3360–3367.

Wang, S.; Ding, Z.; and Fu, Y. 2016. Coupled marginalized auto-encoders for cross-domain multi-view learning. In *IJCAI*, 2125–2131.

Wright, J.; Yang, A. Y.; Ganesh, A.; Sastry, S. S.; and Ma, Y. 2009. Robust face recognition via sparse representation. *TPAMI* 31(2):210–227.

Yan, S.; Xu, D.; Zhang, B.; Zhang, H.-J.; Yang, Q.; and Lin, S. 2007. Graph embedding and extensions: a general framework for dimensionality reduction. *TPAMI* 29(1):40–51.

Yang, M.; Zhang, L.; Feng, X.; and Zhang, D. 2011. Fisher discrimination dictionary learning for sparse representation. In *ICCV*, 543–550.

Yu, W.; Zeng, G.; Luo, P.; Zhuang, F.; He, Q.; and Shi, Z. 2013. Embedding with autoencoder regularization. In *ECML PKDD*, 208–223. Springer.

Zhang, Q., and Li, B. 2010. Discriminative k-svd for dictionary learning in face recognition. In *CVPR*, 2691–2698.

Zhang, Y.; Jiang, Z.; and Davis, L. 2013. Learning structured low-rank representations for image classification. In *CVPR*, 676–683.

Zhou, P.; Lin, Z.; and Zhang, C. 2016. Integrated low-rank-based discriminative feature learning for recognition. *TNNLS* 27(5):1080–1093.