Contractive Rectifier Networks for Nonlinear Maximum Margin Classification

Senjian An[1], Munawar Hayat[1], Salman H. Khan[1]
Mohammed Bennamoun[1], Farid Boussaid[2], Ferdous Sohel[1,3]

[1] School of Computer Science and Software Engineering
The University of Western Australia, Crawley WA 6009, Australia
[2] School of Electrical, Electronic and Computer Engineering,
The University of Western Australia, Crawley WA 6009, Australia
[3] School of Engineering and Information Technology, Murdoch University, Australia

Abstract

To find the optimal nonlinear separating boundary with maximum margin in the input data space, this paper proposes Contractive Rectifier Networks (CRNs), wherein the hidden-layer transformations are restricted to be contraction mappings. The contractive constraints ensure that the achieved separating margin in the input space is larger than or equal to the separating margin in the output layer. The training of the proposed CRNs is formulated as a linear support vector machine (SVM) in the output layer, combined with two or more contractive hidden layers. Effective algorithms have been proposed to address the optimization challenges arising from contraction constraints. Experimental results on MNIST, CIFAR-10, CIFAR-100 and MIT-67 datasets demonstrate that the proposed contractive rectifier networks consistently outperform their conventional unconstrained rectifier network counterparts.

1. Introduction

Deep learning networks have achieved great successes in recent years and have become one of the most attractive research topics in machine learning, computer vision and speech recognition. Rectifier $\max\{0, x\}$ is one of the most popular nonlinear activation functions in modern deep learning networks. The advantages of deep rectifier networks are not only shown in their excellent empirical performance in object recognition [18, 20, 13], face verification [38, 37], speech recognition ([32, 14, 7] and handwritten digit recognition [5], but also justified by a number of theoretical works on their superior expressive powers [6, 28, 27] and their metric preserving properties [1] in transforming linearly inseparable pattern sets into linearly separable sets. It was shown that any disjoint pattern sets can be transformed to be linearly separable by two hidden layers while the distance distortions are controlled within a factor in the range $[0.5, 1]$ [1]. This nearly isometric property of the nonlinear transformation can be used to design nonlinear maximum margin classifiers through the applications of linear support vector machines (SVMs) in the output layer. However, in order to preserve metrics in hidden layer transforms, [1] requires the number of neurons in the first hidden layer to be at least twice the data dimension. In addition, it requires the number of neurons of the higher hidden layers to be at least twice the number of neurons of their preceding hidden layers. This requirement is impractical for deep rectifier networks with deep hidden layers and/or with high dimensional data. Motivated by the fact that the solution of an SVM is fully determined by their support vectors (usually a small subset of the training patterns), this paper proposes Contractive Rectifier Networks (CRNs) whose hidden layers are not designed to preserve the distances of any two arbitrary input vectors but are trained to best preserve only the distances of the support vectors. The proposed CRNs consists of a linear SVM in the output layer and two or more hidden layers each restricted to be a contraction mapping, that is, the distance between the outputs of the hidden layer for any two inputs (of this layer) is not enlarged. In this paper, we show that contraction constraints do not sacrifice the capacity of rectifier networks in achieving maximum margin classification (i.e., the optimal nonlinear separating margin achievable by unconstrained rectifier networks (URNs) can also be achieved by CRNs). Although they have equal theoretical capacity for nonlinear maximum classification, CRN is superior in practical training. The training of CRNs, with a linear SVM in the output layer, optimizes the weights to maximize the separating margin in the output layer while ensuring that the separating margin in the input space is larger than or equal to that in the output layer. On the other hand, the training of URNs, with a linear SVM in the output layer similarly, can result in a solution with an arbitrarily small margin in the input space even the separating margin in the output layer is infinitely large.

The major contributions of this paper include: 1) the first mathematical formulation of nonlinear maximum margin classification. Although maximum margin is a well-known property of linear SVM, nonlinear maximum-margin classification is yet to be addressed to the best of our knowledge; 2) the first deep learning network which can achieve guaranteed larger separating margin in the input space than that in the output layer; 3) a novel training method of rectifier networks under contraction constraints; and 4) the superior performance on a number of popular databases: CIFAR-10, CIFAR-100 for object classification, and MIT-67 for scene understanding.

Related Works: There are a large amount of successful rectifier networks upon which the proposed techniques have potentials to improve classification performance. Here, we only review the most relevant works. Specifically, this work is closely related to [1], which proves that any disjoint pattern sets can be transformed by two hidden layers to be linearly separable while preserving the distances within a factor ranging between 0.5 and 1. Although the orthogonal bidirectional rectified transform introduced in [1] can be used for nonlinear maximum margin classification, the rectifier network constructed in the proof of [1] is difficult to learn in practice due to the orthogonal constraints and the required large number of neurons in the hidden layers. The proposed CRN replaces these non-convex constraints by the convex contraction mapping constraints and the best CRN can be learnt with any given number of neurons in hidden layers. Another closely related work is the deep learning network using linear SVM [39], where the output layer uses a linear SVM instead of softmax regression in the traditional deep neural network. Without additional constraints on the weights, Section 4.1 will show that such rectifier network cannot guarantee any level of optimum for nonlinear maximum margin classification even if the separating margin is infinitely large in the output layer. The contraction constraint was also used to improve performance in [31] for auto-encoders, in [4] for invariant scattering transforms, and in [22] for structured labelling. Recently, a novel normalization technique [15] on the outputs of each neuron has been successfully applied in training deep neural networks to speed up training and improve accuracy. This technique can also address the scaling problem regarding the gap between the separating margin in the output layer and that in the original data space.

Notations: Throughout the paper, we use capital letters to denote matrices, lower case letters for scalar terms, and bold lower letters for vectors. For instance, we use $\mathbf{w}_i$ to denote the i^{th} column of a matrix W, and use b_i to denote the i^{th} element of a vector $\mathbf{b}$. For any integer m, we use $[m]$ to denote the integer set from 1 to m, i.e., $[m] \triangleq \{1, 2, \cdots, m\}$. We use I to denote the identity matrix with proper dimensions, $\mathbf{0}$ to denote a vector with all elements being 0. A positive semidefinite matrix X is denoted by $X \succeq 0$, and $X \preceq I$ is equivalent to $I - X \succeq 0$.

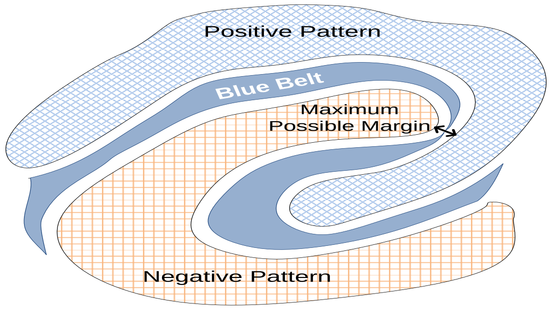

Figure 1. [Best Viewed in Color] Demonstration of Nonlinear Maximuum Margin Classification: Any separating boundary within the blue belt achives the maximum margin.

Organization: The rest of this paper is organized as follows. In Section 2, we introduce nonlinear maximum margin classification problem and review the solution of linear maximum margin classification through SVM. In Section 3, we address the formulation of contractive rectifier networks and show their equal capacity of achieving nonlinear maximum margin classification in comparison with their unconstrained rectifier networks counterparts. Section 4 addresses the training of CRNs, Section 5 presents experimental results while Section 6 concludes the paper.

2. Nonlinear Maximum Margin Classification

In this section, we define the margin of a nonlinear classifier as the minimum distance of the training patterns to the classifier's separating boundary in the input space, and introduce nonlinear max-margin classification aiming to find the nonlinear classifier with maximum margin in the nonlinear space. This is a natural extension of maximum margin linear classifiers such as linear support vector machines (SVMs) wherein the separating boundary is restricted be a hyper-plane in the data space. Figure 1 illustrates an example of nonlinear maximum margin classifiers for positive and negative patterns in 2D space. Clearly the ideal separating boundary is the middle line of the blue belt but it is hard to learn from data. The proposed nonlinear maximum margin classification aims to find a nonlinear classifier with a separating boundary within the blue belt.

For high dimensional patterns, the learning of maximum margin nonlinear classifiers from data is challenging due to the complexity of the nonlinear separating boundaries. Next, we will formulate the maximum margin classification problem, review the maximum margin properties of linear SVMs, and show the ways one can achieve nonlinear maximum margin classifications through linear classifiers based on nonlinearly transformed features.

Let $f(\mathbf{x};\mathfrak{p})$ denote a general pattern classifier, linear or nonlinear, where $\mathbf{x} \in \mathbb{R}^n$ is a vector representing an instance of the patterns and $\mathfrak{p}$ denotes the parameters of the classifier. If $\mathfrak{p} = \{\mathbf{w}, b\}$ and $f(\mathbf{x};\mathfrak{p}) = \mathbf{w}^T\mathbf{x} + b$, then $f(\mathbf{x};\mathfrak{p})$ is called a linear classifier and $\text{sign}\{\mathbf{w}^T\mathbf{x} + b\}$ predicts the label of an instance $\mathbf{x}$. Otherwise, if $f(\mathbf{x};\mathfrak{p})$ cannot be described as such a formulation, it is called a nonlinear classifier. Most nonlinear classifiers, such as kernel [33] and neural network classifiers [2], first conduct nonlinear transformations $\phi(\mathbf{x})$, explicitly or implicitly, and then apply linear classifiers on the transformed features.

In practical training of pattern classifiers, the parameters $\mathfrak{p}$, such as the weights of the output layer and hidden layers in neural networks, are allowed to vary within some ranges, that is, $\mathfrak{p} \in \mathcal{P}$ for some set $\mathcal{P}$. Each classification framework has its own parameter structure $\mathfrak{p}$ and its own distinct parameter set $\mathcal{P}$. In case of linear classification, $\mathfrak{p} = \{\mathbf{w}, b\}$ and $\mathcal{P} = \{\mathfrak{p} = \{\mathbf{w}, b\} : \mathbf{w} \in \mathbb{R}^n, b \in \mathbb{R}\}$.

The separating boundary margins of individual classifiers and the maximum margin of a group of classifiers are defined as follows. Suppose that a training set, namely $\{\mathbf{x}_i, y_i\}_{i=1}^N, \mathbf{x}_i \in \mathbb{R}^n, y_i \in \{-1, 1\}$, is separable by $f(\mathbf{x};\mathfrak{p})$ for some parameter $\mathfrak{p} \in \mathcal{P}$, that is, $f(\mathbf{x}_i;\mathfrak{p}) > 0$ for any positive patterns $\mathbf{x}_i$ and $f(\mathbf{x}_j;\mathfrak{p}) < 0$ for any negative patterns $\mathbf{x}_j$. Let

$$\mathcal{B}(\mathfrak{p}) \triangleq \{\mathbf{x} : f(\mathbf{x};\mathfrak{p}) = 0\} \tag{1}$$

denote the separating boundary. The separating margin of this classifier $f(\mathbf{x};\mathfrak{p})$ is then defined as the minimal distance from the training patterns to the separating boundary, i.e.,

$$\gamma(\mathfrak{p}) \triangleq \min_{i\in[N]} \inf_{\mathbf{x}\in\mathcal{B}(\mathfrak{p})} \|\mathbf{x}_i - \mathbf{x}\| \tag{2}$$

where $\|\cdot\|$ denotes the Euclidean norm and $\|\mathbf{x}_i - \mathbf{x}\|$ is the Euclidean distance between $\mathbf{x}$ and $\mathbf{x}_i$.

The maximum separating margin achievable by a group of classifiers $\{f(\mathbf{x};\mathfrak{p}) : \mathfrak{p} \in \mathcal{P}\}$ is defined as

$$\gamma_{\max}(\mathcal{P}) \triangleq \sup_{\mathfrak{p}\in\mathcal{P}} \gamma(\mathfrak{p}). \tag{3}$$

Although the maximization of $\gamma(\mathfrak{p})$ with respect to $\mathfrak{p} \in \mathcal{P}$ is extremely challenging for nonlinear classifiers, the maximum margin linear classifier can be obtained through the training of a linear SVM with hard constraints. In the linear case, the separating boundary $\mathcal{B}(\mathfrak{p})$ is a hyperplane $\{\mathbf{x} : \mathbf{w}^T\mathbf{x} + b = 0\}$ in n dimensional space, and the separating margin of a linear classifier is then

$$\gamma(\mathfrak{p}) = \min_{i\in[N]} \frac{y_i(\mathbf{w}^T\mathbf{x}_i + b)}{\|\mathbf{w}\|} \tag{4}$$

where $\|\mathbf{w}\| = \sqrt{\mathbf{w}^T\mathbf{w}}$ is the Euclidean norm of $\mathbf{w}$.

The maximum margin linear classifier can thus be obtained by solving the following optimization problem

$$\max_{\mathbf{w},b} \min_{i\in[N]} \frac{y_i(\mathbf{w}^T\mathbf{x}_i + b)}{\|\mathbf{w}\|} \tag{5}$$

or equivalently by a linear SVM with hard constraints given by

$$\begin{array}{ll} \min\limits_{\mathbf{w},b} & \frac{1}{2}\mathbf{w}^T\mathbf{w} \\ s.t. & y_i(\mathbf{w}^T\mathbf{x} + b) \geq 1,\ i \in [N]. \end{array} \tag{6}$$

Remarks: The kernel SVMs can only achieve the maximum margin property in the kernel induced feature space while the neural network methods with SVMs in the output layer can only obtain a maximum margin separating boundary in the feature space transformed by the hidden layers. However, the separating boundaries of these nonlinear classification methods do not have the maximum margin property in the input space due to distance distortions from the nonlinear transformations. To achieve certain levels of optimum for nonlinear maximum margin classifications by conducting linear SVM on a nonlinear transformed feature space, one needs to control the distance distortions of the associated nonlinear transforms so that the separating margin in the feature space and that in the input space are closely related. In this paper, we propose to use contraction constraints to control the distance distortions of rectifier hidden layer transformations and ensure that the maximization of the separating margin in the output layer leads to a larger separating margin in the input space.

3. Contractive Rectifier Networks

A rectifier network with single output and d hidden layers can be described as

$$\begin{array}{rcl} f(\mathbf{x};\mathfrak{p}) & = & \mathbf{w}^T\mathbf{z}_d + b \\ \mathbf{z}_k & = & \max\{\mathbf{0}, W_k^T\mathbf{z}_{k-1} + \mathbf{c}_k\}, k = 1, 2, \cdots, d \\ \mathbf{z}_0 & = & \mathbf{x} \end{array} \tag{7}$$

where $\mathbf{x}$ is the input, $\mathbf{z}_k$ is the output of the k^{th} hidden layer, and $\mathfrak{p} = \{\mathbf{w}, b, W_k, \mathbf{c}_k : k \in [d]\}$.

The parameter set $\mathcal{P}$ for all such rectifier networks with arbitrary numbers of neurons can be described as

$$\mathcal{P} \;\; = \;\; \bigcup_{\mathbf{l}\in\mathcal{L}} \mathcal{P}(\mathbf{l}) \tag{8}$$

where $\mathbf{l} = [l_1, l_2, \cdots, l_d]^T$ is an integer vector representing the numbers of neurons in d hidden layers,

$$\mathcal{L} = \{\mathbf{l} : l_k \in \mathbb{Z}_+, k \in [d]\} \tag{9}$$

and

$$\begin{array}{rcl} \mathcal{P}(\mathbf{l}) & \triangleq & \big\{\mathfrak{p} = \{\mathbf{w}, b, W_k, \mathbf{c}_k : k \in [d]\} : \mathbf{w} \in \mathbb{R}^{l_d}, \\ & & b \in \mathbb{R}, W_k \in \mathbb{R}^{l_{k-1}\times l_k}, \mathbf{c}_k \in \mathbb{R}^{l_k}, k \in [d]\big\} \end{array} \tag{10}$$

where $l_0 = n$.

A rectifier network $f(\mathbf{x}; \mathfrak{p})$, as defined in (7), is called a *contractive rectifier network* (CRN) if the transformation of each hidden layer is a contraction mapping, that is,

$$\|\mathbf{z}_1^{(2)} - \mathbf{z}_1^{(1)}\| \quad \leq \quad \|\mathbf{x}^{(2)} - \mathbf{x}^{(1)}\|, \; \forall\, \mathbf{x}^{(1)}, \mathbf{x}^{(2)} \in \mathbb{R}^n \tag{11}$$

and for each $k = 2, 3, \cdots, d$,

$$\|\mathbf{z}_k^{(2)} - \mathbf{z}_k^{(1)}\| \leq \|\mathbf{z}_{k-1}^{(2)} - \mathbf{z}_{k-1}^{(1)}\|, \; \forall\, \mathbf{z}_{k-1}^{(1)}, \mathbf{z}_{k-1}^{(2)} \in \mathbb{R}^{l_{k-1}} \tag{12}$$

where $\mathbf{z}_1^{(1)}, \mathbf{z}_1^{(2)}$ are the outputs of the first hidden layer from the inputs $\mathbf{x}_1^{(1)}, \mathbf{x}_1^{(2)}$ respectively, and $\mathbf{z}_k^{(1)}, \mathbf{z}_k^{(2)}$ are the outputs of the k^{th} hidden layer from the outputs $\mathbf{z}_{k-1}^{(1)}, \mathbf{z}_{k-1}^{(2)}$ of the preceding hidden layer respectively.

Note that

$$\begin{aligned} & \| \max\left\{\mathbf{0}, W^T\mathbf{x}^{(1)} + \mathbf{c}\right\} - \max\left\{\mathbf{0}, W^T\mathbf{x}^{(2)} + \mathbf{c}\right\} \|^2 \\ \leq \;& \|W^T\mathbf{x}^{(1)} - W^T\mathbf{x}^{(2)}\|^2 \\ = \;& (\mathbf{x}^{(1)} - \mathbf{x}^{(2)})^T W W^T (\mathbf{x}^{(1)} - \mathbf{x}^{(2)}). \end{aligned} \tag{13}$$

The constraints in (11,12) can be implemented by restricting the weight matrices W_k to satisfy the following conditions

$$W_k W_k^T \preceq I, \forall\, k \in [d] \tag{14}$$

or equivalently, by the well-known Schur Complement Lemma [3],

$$\begin{bmatrix} I & W_k^T \\ W_k & I \end{bmatrix} \succeq 0, \forall\, k \in [d] \tag{15}$$

which are convex constraints if the range of W_k is convex.

Correspondingly, the parameter space of all CRNs, denoted by $\mathcal{P}_c$, can be described as

$$\begin{aligned} \mathcal{P}_c \;&=\; \bigcup_{\mathbf{l}\in\mathcal{L}} \mathcal{P}_c(\mathbf{l}) \\ \mathcal{P}_c(\mathbf{l}) \;&\triangleq\; \{\mathfrak{p} = \{\mathbf{w}, b, W_k, \mathbf{c}_k : k \in [d]\} : \mathbf{w} \in \mathbb{R}^{l_d}, \\ &\quad\; b \in \mathbb{R}, W_k W_k^T \preceq I, \mathbf{c}_k \in \mathbb{R}^{l_k}, k \in [d]\} \end{aligned} \tag{16}$$

where $l_0 = n$.

Next, we show that the maximal separating margin achievable by a rectifier network, as defined in (7), can also be achieved by a CRN.

Proposition 1 *Let $\mathbf{l} = [l_1, l_2, \cdots, l_d]^T$ denote the numbers of hidden nodes in rectifier networks with d hidden layers, and $\mathcal{P}, \mathcal{P}(\mathbf{l}), \mathcal{P}_c, \mathcal{P}_c(\mathbf{l})$ be defined as in (7), (10) and (16) respectively. Then the maximum margin achievable by a rectifier network for a given training set $\{\mathbf{x}_i, y_i, i \in [N]\}$ can also be obtained by a contractive rectifier network. More precisely*

$$\begin{aligned} \gamma_{\max}\{\mathcal{P}(\mathbf{l})\} \;&=\; \gamma_{\max}\{\mathcal{P}_c(\mathbf{l})\}, \; \forall\, \mathbf{l} \in \mathcal{L}; \\ \gamma_{\max}\{\mathcal{P}\} \;&=\; \gamma_{\max}\{\mathcal{P}_c\} \end{aligned} \tag{17}$$

where $\gamma_{\max}(\cdot)$ is defined in (3).

Proof: Let $\mathfrak{p}^* \in \mathcal{P}(\mathbf{l})$ and $f(\mathbf{x}; \mathfrak{p}^*)$ be the rectifier network that achieves the largest margin $\gamma_{\max}\{\mathcal{P}(\mathbf{l})\}$ for a given training set $\{\mathbf{x}_i, y_i, i \in [N]\}$. By scaling the weights W_k to satisfy the contraction constraints (11,12), one can obtain a CRN, namely $f(\mathbf{x}; \hat{\mathfrak{p}}^*)$, which can also separate the training set. Furthermore, the separating boundaries of $f(\mathbf{x}; \hat{\mathfrak{p}}^*)$ and $f(\mathbf{x}; \mathfrak{p}^*)$ are identical. Hence, these two classifiers have the same separating margin and therefore $\gamma_{\max}\{\mathcal{P}(\mathbf{l})\} = \gamma_{\max}\{\mathcal{P}_c(\mathbf{l})\}$ for any $\mathbf{l} \in \mathcal{L}$. Similarly, one can prove $\gamma_{\max}\{\mathcal{P}\} = \gamma_{\max}\{\mathcal{P}_c\}$ to complete the proof.

□

4. Training of Contractive Rectifier Networks

Due to the complexity of the separating boundaries of RNs and CRNs, there is no efficient way to optimize the separating margin among all the possible RNs or CRNs. However, the maximum margin in the output layer can be achieved by a linear SVM, and we propose to optimise the nonlinear separating margin by training a linear SVM in the output layer of CRN. Next, we first show the necessity of enforcing contraction constraints on the rectifier hidden layers and then address the formulation and training of the proposed CRN with linear SVM in the output layer.

4.1. The Necessity of Contraction Constraints

Let $\{\mathbf{x}_i \in \mathbb{R}^n, y_i \in \{-1, 1\}, i \in [N]\}$ be a given training pattern set, and $\mathbf{l} = [l_1, l_2, \cdots, l_d]^T$ be an integer vector representing the numbers of neurons in d hidden layers, the training of RN, with hard margin linear SVM in the output layer but without contraction constraints on the hidden layers, can be formulated as

$$\begin{array}{ll} \min\limits_{\mathfrak{p}\in\mathcal{P}(\mathbf{l})} & \frac{1}{2}\mathbf{w}^T\mathbf{w} \\ s.t. & y_i(\mathbf{w}^T\mathbf{z}_d(i) + b) \geq 1, \; i \in [N] \\ & \mathbf{z}_k(i) = \max\{\mathbf{0}, W_k^T\mathbf{z}_{k-1}(i) + \mathbf{c}_k\}, \\ & \qquad\quad 2 \leq k \leq d, i \in [N] \\ & \mathbf{z}_1(i) = \max\{\mathbf{0}, W_1^T\mathbf{x}_i + \mathbf{c}_1\}, i \in [N] \end{array} . \tag{18}$$

However, this optimization may result in a rectifier network with an arbitrarily small separating margin in the input space even the separating margin is very large in the output layer. Let $\epsilon > 0$ be any small number and $f(\mathbf{x}; \mathfrak{p})$ be an error-free classifier of a training pattern set with a separating margin $\gamma(\mathfrak{p}) = \epsilon$ for some $\mathfrak{p} = \{\mathbf{w}, b, W_k, \mathbf{c}_k : k \in [d]\}$ satisfying the constraints of (18). Then $f(\mathbf{x}; \hat{\mathfrak{p}})$ is also an error-free classifier with $\hat{\mathfrak{p}} = \{\eta\mathbf{w}, \eta b, \eta^{-\frac{1}{d}} W_k, \eta^{-\frac{1}{d}}\mathbf{c}_k : k \in [d]\}$ satisfying the constraints of (18) as well. The cost $\frac{1}{2}\eta^2\mathbf{w}^T\mathbf{w}$ can be made arbitrarily small by choosing a sufficiently small η, and thus the cost tends to zero as η approaches to zero. Though the separating boundary margin in the output layer tends to be infinitely large when η approaches to zero, the margin in the original input space remains ϵ. Hence, the optimization problem of (18) is not well defined and some constraints on the weights W_k are in need. In the next subsection, we will show that the proposed contraction constraints on the hidden layers in CRN ensure that the separating margin in the input space is larger than or equal to that in the output layer.

4.2. Hard Margin Formulation of CRN

The training of CRN with linear SVM in the output layer is formulated as:

$$\begin{array}{ll} \min\limits_{\mathfrak{p}\in\mathcal{P}_c(\mathbf{1})} & \frac{1}{2}\mathbf{w}^T\mathbf{w} \\ s.t. & y_i(\mathbf{w}^T\mathbf{z}_d(i)+b)\geq 1,\ i\in[N] \\ & \mathbf{z}_k(i)=\max\{\mathbf{0}, W_k^T\mathbf{z}_{k-1}(i)+\mathbf{c}_k\}, \\ & \qquad 2\leq k\leq d, i\in[N] \\ & \mathbf{z}_1(i)=\max\{\mathbf{0}, W_1^T\mathbf{x}_i+\mathbf{c}_1\}, i\in[N] \\ & W_kW_k^T\preceq I, \forall\, k\in[d] \end{array} \tag{19}$$

where $\mathbf{z}_k(i)$ denotes the output of the k^{th} hidden layer for input $\mathbf{x}_i$.

Regarding the above optimization problem and its solution, we have the following result:

Proposition 2 *If the number of hidden layers is more than 1, i.e., $d\geq 2$, the constraints of (19) are feasible for any pattern set $\{(\mathbf{x}_i, y_i): y_i=\pm 1, i\in[N]\}$. Furthermore, let $\mathfrak{p}^*$ be the optimal solution of the hard margin optimization problem (19), then the separating margin of $f(\mathbf{x};\mathfrak{p}_{opt})$ in the input space is larger or equal to the separating margin in the output layer, i.e.,*

$$\gamma\{\mathfrak{p}^*\}\geq\frac{1}{\|\mathbf{w}^*\|} \tag{20}$$

where $\gamma(\cdot)$ is defined in (2).

Proof: In [1], it is shown that any two disjoint pattern sets can be transformed to be linear separable through two hidden layers of rectifier networks. By scaling, one can always restrict the weight matrix to satisfy the contraction constrains while the resulted contraction hidden layers can still transform the patterns to be linearly separable in the output layer. Hence the constraints of (19) are feasible for any training set.

Note that the transformation of each hidden layer is contractive, the distance of any training pattern to the separating boundary in the output layer must be smaller than the distance of this training pattern to the separating boundary in the input space. Hence the separating margin (i.e.,$\gamma\{\mathfrak{p}^*\}$) of $f(\mathbf{x};\mathfrak{p}^*)$ in the input space must be larger than or equal to the separating margin ((i.e. $\frac{1}{\|\mathbf{w}\|}$)) in the output layer.

□

4.3. Soft Margin Formulation of CRN

Similar to the formulation of linear SVMs, the soft margin version of the proposed maximum margin contractive rectifier network with hinge loss can be described as

$$\begin{array}{ll} \min\limits_{\mathfrak{p}\in\mathcal{P}_c(\mathbf{1})} & \frac{1}{2}\mathbf{w}^T\mathbf{w}+\frac{C}{N}\sum\limits_{i\in[N]}\xi_i \\ s.t. & y_i(\mathbf{w}^T\mathbf{z}_d(i)+b)\geq 1-\xi_i,\ i\in[N] \\ & \xi_i\geq 0; i\in[N] \\ & \mathbf{z}_k(i)=\max\{\mathbf{0}, W_k^T\mathbf{z}_{k-1}(i)+\mathbf{c}_k\}, \\ & \qquad 2\leq k\leq d, i\in[N] \\ & \mathbf{z}_1(i)=\max\{\mathbf{0}, W_1^T\mathbf{x}_i+\mathbf{c}_1\}, i\in[N] \\ & W_kW_k^T\preceq I, \forall\, k\in[d] \end{array} \tag{21}$$

Remarks: The soft version can also be formulated using squared hinge loss by replacing ξ_i with ξ_i^2, which gives more penalty on the larger instance training errors. The extension to multi-category classification is straightforward by summing up the costs and combining the constraints of the soft-margin formulations for all the one-versus-the-rest binary classifications.

4.4. The Updating Rule

For training of CRN, we need to ensure that the constraints are satisfied in each iteration. At iteration t, let $W_k(t)$ denote the value of W_k, $W_k(t)W_k(t)^T\preceq I$, and $\Delta_k(t)$ denote the gradient, $W_k(t+1)$ is updated as follows. Let

$$Q=W_k(t)+\eta\Delta_k(t), \tag{22}$$

and conduct singular value decomposition (SVD) on Q, i.e., $Q=U\Lambda V^T$. Then W_k is updated as

$$W_k(t+1)=U\min\{1,\Lambda\}V^T. \tag{23}$$

which implies $W_k(t+1)W_k(t+1)^T\preceq I$, and thus $W_k(t+1)$ satisfies the contraction constraints.

The operation of SVD to ensure contraction constraint is computationally expensive. The SVDs of 4096×2048 and 2048×1024 matrices on an Intel core i7 machine for CIFAR-10 dataset take 4.96 and 0.54 seconds respectively. Performing SVD for every forward pass of the training therefore makes the method computationally expensive. In our experiments, we however observed that performing SVD only at the initialization step and after every epoch generates 8.8 % error rate for CIFAR-10 dataset while performing SVD in every forward pass of the training achieves 8.75% error rate (a slight improvement of 0.05%). For the consideration of computational advantages, we therefore performed SVD at the initialization step and then after every epoch (instead of every forward pass) for all the experiments. This achieves a good trade-off between the performance and the required computational resources.

5. Experiments

The efficacy of the proposed method is demonstrated through extensive experiments on a number of datasets for numerous classification tasks. Specifically, these include MNIST dataset for handwritten digit recognition, CIFAR-10 and CIFAR-100 datasets for generic object recognition and MIT-67 dataset for indoor scene classification. In the followings, we first provide a brief description of each dataset and the experimental configurations (Sec. 5.1). We then describe the architectures of the baseline network (Sec. 5.2) and our proposed Contractive Rectifier Network (Sec. 5.3). We finally present our method for the selection of optimal hyper-parameters (Sec. 5.4) followed by a discussion and analysis of the achieved experimental performance (Sec. 5.5).

5.1. Datasets

MNIST[19]: This dataset consists of 28×28 grey scale images of handwritten digits. The total number of images in the dataset is $70,000$, of which $60,000$ are used for training while the remaining $10,000$ are used for testing. These images belong to 10 different classes (corresponding to digits from 0 to 9).

CIFAR-10 and CIFAR-100 [17]: CIFAR-10 dataset comprises $60,000$ color images of 10 different object categories. The resolution of the images in the dataset is 32×32. $50,000$ of these images are allocated for training while the remaining $10,000$ are used for testing. Similar to CIFAR-10, CIFAR-100 dataset also has a total of $60,000$ color images ($50,000$ for training and the other $10,000$ for testing) of resolution 32×32. However, the number of object categories in CIFAR-100 is much larger than CIFAR-10 (100 compared to 10) which makes the classification on CIFAR-100 a far more challenging task.

MIT-67 Dataset[29]: The MIT-67 Dataset contains $15,620$ color images of 67 indoor scene categories (e.g., kitchen, bedroom, dining room, library and bookstore). For performance evaluation and comparison, we followed the standard evaluation protocol [29] in which a subset of data is used (100 images per class) and the train-test split is defined to be $80\% - 20\%$ for each class.

5.2. Baseline Network Architecture

We train our contractive rectifier network (Sec 5.3) on top of the learned feature representations from a baseline Convolutional Neural Network (CNN). For MNIST dataset, our baseline CNN comprises of two alternating convolutional (filter size: 5×5, number of channels are 32 and 64 respectively) and max-pooling (size: 2×2, stride: 2) layers and one fully connected layer (number of neurons: 500) with a dropout rate of 0.5. Non-linear activations i.e., Rectified Linear Units (ReLUs) are used after each convolutional and fully connected layer.

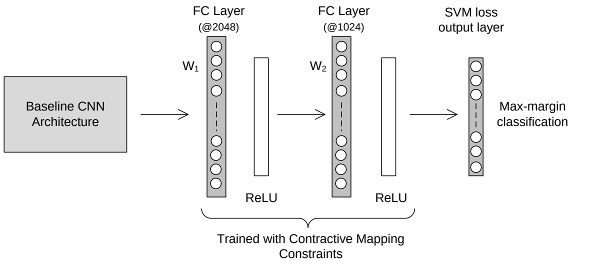

Figure 2. The architecture of the proposed Contractive Rectifier Network

For the other evaluated datasets, we trained our contractive rectifier network on top of the learned feature representations from a deep CNN architecture [34] which comprises 16 learnable weight layers (13 convolutional layers and 3 fully connected layers). The network takes a fixed size input (224×224) color image, subtracts a mean image (computed on the training set) and then feeds the resulting image to the layers of the network. All convolution filters in the network have a relatively smaller size of 3×3, which helps decreasing the number of parameters in the network and provides an effective larger receptive field due to the consecutive convolution layers. For spatial pooling of feature representations, the network contains five max-pooling layers (size: 2×2, stride: 2) after the 2^{nd}, 4^{th}, 7^{th}, 10^{th} and 13^{th} convolutional layers, respectively. ReLU activation function is applied after every layer in the deep CNN. Pairs of convolutional layers with 64 and 128 filters respectively appear at the start of the network. Afterwards, three triplets of convolutional layers with filters 256, 512 and 512 respectively appear before the final three fully connected layers. For our feature representations, we re-scale our input images to 224×224 and take 4096 dimensional output from the first fully connected layer of the trained deep network.

5.3. Proposed CRN Architecture

Our Contractive Rectifier Network (CRN) comprises of two hidden layers and one output layer. The training of the contractive rectifier network is performed through stochastic gradient descent in which the learned feature representations from the deep network are fed as an input while a binary vector with class label information is used as the output. With an annealed learning rate being used (initialized at 10^{-2}, and decreased by a factor of 10 after every 20 epoches), the network is trained for a total of 80 epoches. The network parameters including the total number of neurons in the hidden layers and the value of the regularization constant 'C' are selected after performing experiments on a held-out cross validation set (see Table. 1, Fig 3 and Sec. 5.5) on one of the evaluated datasets (CIFAR-

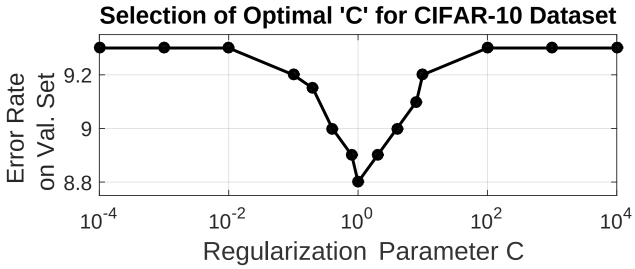

Figure 3. Selection of optimal value of Parameter ‘C’

10 dataset). They are then consistently used across other datasets. Our experimental results show that the performance of the proposed CRN is not very sensitive to the values of the hyper-parameters.

5.4. Optimal Hyper-parameters

The optimal value for the parameter ‘C’ is achieved by performing grid search over a range of values (from 10^{-4} to 10^{4}) and evaluating the performance of the method on a held out cross validation set on CIFAR-10 dataset. As shown in Fig 3, $C = 1$ achieves the best performance and was thus selected. In order to select the number of hidden nodes for the two layered network, we performed experiments for different combination of number of nodes. The experimental results on CIFAR-10 dataset for different number of hidden nodes are summarized in Table 1. Based upon these results, the number of neurons in the two hidden layers is selected as 2048 and 1024. Furthermore, two conclusions can be drawn from these experimental results: 1) having a significantly large number of hidden nodes only results in a slight performance drop, and 2) the best performance is achieved when the number of hidden nodes in the second layer is less than the first. Note that these hyper-parameters (the value of the regularization constant ‘C’ and the number of neurons in the hidden layers) are consistent across all datasets. Further, experiments performed for the selection of optimal hyper-parameters (see Table. 1 and Fig 3) show that the performance of the proposed method is quite robust and does not degrade much as the values of these hyper-parameters change.

Configuration	Performance
8192 – 8192	9.0%
8192 – 4096	8.9%
4096 – 4096	9.1%
4096 – 2048	9.0%
2048 – 2048	8.9%
2048 – 1024	**8.8**%
1024 – 1024	9.0%

Table 1. Cross validation performance for different numbers of hidden nodes.

Methods	**Error**(%)
Alex Net[18]	13.0%
Sum-product Networks [9]	16.31%
Multi-colomn Nets [5]	11.21%
Deeply Supervised Nets [20]	9.69%
Probabilistic Maxout Network [35]	11.35%
Maxout-Networks [11]	11.68%
Network in Network [25]	10.41%
Stochastic Pooling [41]	15.13%
Deep Learning + SVM [39]	11.9%
This paper (URN+Softmax)	10.8%
This paper (URN+SVM)	10.4%
This paper (CRN+SVM)	**8.8**%

Table 2. Performance comparisons on CIFAR-10 dataset.

5.5. Performance Analysis

The experimental results for the task of generic object recognition on CIFAR-10 and CIFAR-100 datasets are given in Table. 2 and 3. The results show that the proposed method achieves the lowest error rates with 8.8% and 34.4% on CIFAR-10 and CIFAR-100 datasets respectively.

The experimental performance for different methods in terms of average classification accuracy for the task of indoor scene classification on MIT-67 data is given in Table 4. The results suggest superior performance of the proposed method compared with existing techniques.

The experimental results for all datasets demonstrate that the learnt CRNs outperform their conventional unconstrained rectifier network (URN) counterparts, with linear SVM or softmax in the output layer.

Remarks: A recent work [12] has reported significant progresses on CIFAR-10 and CIFAR-100. The performance improvement of this work is due to the introduction of fractional max pooling to improve the quality of learnt CNN features, which can serve as a new baseline CNN architecture for the proposed CRN. The combination of fractional max-pooling techniques and CRN has potential to further improve the performance on CIFAR-10 and CIFAR-100.

6. Concluding Remarks

A novel rectifier neural network, termed contractive rectifier network, has been proposed by restricting the hidden layer transformations to be contraction mappings. Compared with unconstrained rectifier networks, the proposed network has the advantage, in practical training through linear SVM in the output layer, to achieve nonlinear maximum margin classification with guaranteed larger separating margin in the input space than that in the output

Methods	**Error**(%)
Deeply Supervised Nets [20]	34.57%
Network in Network [25]	35.68%
Tree based Priors [36]	36.85%
Probablistic Maxout Network [35]	38.14%
Maxout-Networks [11]	38.57%
Stochastic Pooling [41]	42.51%
Representation Learning [26]	39.2%
This paper (URN+Softmax)	35.8%
This paper (URN+SVM)	35.4%
This paper (CRN+SVM)	**34.4**%

Table 3. Performance comparisons on CIFAR-100 dataset.

Methods	**Accuracy**(%)
Spatial Pooling Regions [24]	50.1%
VC + VQ [23]	52.3%
CNN-SVM [30]	58.4%
Improved Fisher Vectors [16]	60.8%
Mid Level Representation [8]	64.0%
Multiscale Orderless Pooling [10]	68.9%
This paper (URN+Softmax)	68.6%
This paper (URN+Softmax)	68.5%
This paper (CRN+SVM)	**70.2**%

Table 4. Performance comparisons on MIT-67 dataset.

Methods	**Error**(%)
Deep learning via embedding [40]	1.5%
Convolutional Deep Belief Nets [21]	0.82%
Deep Learning + SVM [39]	0.87%
This paper (URN+Softmax)	0.86%
This paper (URN+SVM)	0.83%
This paper (CRN+SVM)	**0.73**%

Table 5. Performance comparisons on MNIST dataset.

layer. Experimental results demonstrate that the proposed contractive rectifier networks consistently outperform the conventional rectifier networks in a number of databases, namely, MNIST for handwritten digit recognition, CIFAR-10, CIFAR-100 for object classification, and MIT-67 for scene understanding.

Acknowledgements

This work was supported by the ARC grants DP150100294, DP150104251, DE120102960 and a UWA ECR Fellowship Support Grant.

References

[1] S. An, F. Boussaid, and M. Bennamoun. How can deep rectifier networks achieve linear separability and preserve distances? In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015. 1, 2, 5

[2] Y. Bengio, I. J. Goodfellow, and A. Courville. Deep learning. Book in preparation for MIT Press, 2014. 3

[3] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004. 4

[4] J. Bruna and S. Mallat. Invariant scattering convolution networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1872–1886, 2013. 2

[5] D. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 1, 7

[6] O. Delalleau and Y. Bengio. Shallow vs. deep sum-product networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2011. 1

[7] L. Deng, J. Li, J.-T. Huang, K. Yao, D. Yu, F. Seide, M. Seltzer, G. Zweig, X. He, J. Williams, et al. Recent advances in deep learning for speech research at microsoft. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013. 1

[8] C. Doersch, A. Gupta, and A. A. Efros. Mid-level visual element discovery as discriminative mode seeking. In *NIPS*, pages 494–502, 2013. 8

[9] R. Gens and P. Domingos. Discriminative learning of sum-product networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2012. 7

[10] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *European Conference on Computer Vision (ECCV)*, pages 392–407. 2014. 8

[11] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. In *Proceedings of The 30th International Conference on Machine Learning (ICML)*, 2013. 7, 8

[12] B. Graham. Fractional max-pooling. *arXiv preprint arXiv:1412.6071v4*, 2015. 7

[13] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *arXiv preprint arXiv:1502.01852*, 2015. 1

[14] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of

four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012. 1

[15] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 2015. 2

[16] M. Juneja, A. Vedaldi, C. Jawahar, and A. Zisserman. Blocks that shout: Distinctive parts for scene classification. In *CVPR*, pages 923–930. IEEE, 2013. 8

[17] A. Krizhevsky. Learning multiple layers of features from tiny images, 2009. 6

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems (NIPS)*, 2012. 1, 7

[19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 1998. 6

[20] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. *arXiv preprint arXiv:1409.5185*, 2014. 1, 7, 8

[21] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *International Conference on Machine Learning (ICML)*, 2009. 8

[22] Q. Li, J. Wang, Z. Tu, and D. P. Wipf. Fixed-point model for structured labeling. In *Proceedings of the 30th international conference on machine learning (ICML-13)*, 2013. 2

[23] Q. Li, J. Wu, and Z. Tu. Harvesting mid-level visual concepts from large-scale internet images. In *IEEE Conferecne on Computer Vision and Pattern Recognition (CVPR)*, 2013. 8

[24] D. Lin, C. Lu, R. Liao, and J. Jia. Learning important spatial pooling regions for scene classification. In *IEEE Conferecne on Computer Vision and Pattern Recognition (CVPR)*, 2014. 8

[25] M. Lin, Q. Chen, and S. Yan. Network in network. In *Statistical Language and Speech Processing*, 2013. 7, 8

[26] T.-H. Lin and H. Kung. Stable and efficient representation learning with nonnegativity constraints. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014. 8

[27] G. Montúfar, R. Pascanu, K. Cho, and Y. Bengio. On the number of linear regions of deep neural networks. *arXiv preprint arXiv:1402.1869*, 2014. 1

[28] R. Pascanu, G. Montufar, and Y. Bengio. On the number of inference regions of deep feed forward networks with piece-wise linear activations. In *International Conference on Learning Representations 2014(Conference Track)*, Apr. 2014. 1

[29] A. Quattoni and A. Torralba. Recognizing indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. 6

[30] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. *arXiv preprint arXiv:1403.6382*, 2014. 8

[31] S. Rifai, Y. Bengio, Y. Dauphin, and P. Vincent. A generative process for sampling contractive autoencoders. *arXiv preprint arXiv:1206.6434*, 2012. 2

[32] F. Seide, G. Li, and D. Yu. Conversational speech transcription using context-dependent deep neural networks. In *Interspeech*, pages 437–440, 2011. 1

[33] J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004. 3

[34] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*, 2015. 6

[35] J. T. Springenberg and M. Riedmiller. Improving deep neural networks with probabilistic maxout units. In *International Conference on Learning Representations(ICLR)*, 2014. 7, 8

[36] N. Srivastava and R. R. Salakhutdinov. Discriminative transfer learning with tree-based priors. In *Advances in Neural Information Processing Systems(NIPS)*, 2013. 8

[37] Y. Sun, Y. Chen, X. Wang, and X. Tang. Deep learning face representation by joint identification-verification. In *Advances in Neural Information Processing Systems (NIPS)*, 2014. 1

[38] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 1

[39] Y. Tang. Deep learning using linear support vector machines. *arXiv preprint arXiv:1306.0239*, 2013. 2, 7, 8

[40] J. Weston, F. Ratle, H. Mobahi, and R. Collobert. Deep learning via semi-supervised embedding. In *Neural Networks: Tricks of the Trade*, pages 639–655. Springer, 2012. 8

[41] M. D. Zeiler and R. Fergus. Stochastic pooling for regularization of deep convolutional neural networks. *arXiv preprint arXiv:1301.3557*, 2013. 7, 8