Soft Computing manuscript No.
(will be inserted by the editor)

Post–training discriminative pruning for RBMs

Máximo Sánchez-Gutiérrez · Enrique M. Albornoz · Hugo L. Rufiner · John Goddard Close

Received: date / Revised: date / Accepted: date

Abstract One of the major challenges in the area of artificial neural networks is the identification of a suitable architecture for a specific problem. Choosing an unsuitable topology can exponentially increase the training cost, and even hinder network convergence. On the other hand, recent research indicates that larger or deeper nets can map the problem features into a more appropriate space, and thereby improve the classification process, thus leading to an apparent dichotomy. In this regard, it is interesting to inquire whether independent measures, such as mutual information, could provide a clue to finding the most discriminative neurons in a network. In the present work we explore this question in the context of Restricted Boltzmann Machines, by employing different measures to realize post-training pruning. The neurons which are determined by each measure to be the most discriminative, are combined and a classifier is applied to the ensuing network to determine its usefulness. We find that two measures in particular seem to be good indicators of the most discriminative neurons, producing savings of generally more than 50% of the neurons, while maintaining an acceptable error rate. Further, it is borne out that starting with a larger network architecture and then pruning is more advantageous than using a smaller network to begin with. Finally, a quantitative index is introduced which can provide information on choosing a suitable pruned network.

Keywords Restricted Boltzmann Machines · Pruning · Discriminative Information · Phoneme Classification · Emotion Classification

1 Introduction

In the last few years, many large scale recognition challenges have been successfully addressed using recent advances in pattern recognition, machine learning and artificial neural networks (Le 2013; Taigman et al 2014; Simonyan and Zisserman 2014). In the case of artificial neural networks, there are several criteria used to evaluate a network's quality e.g. training time, scalability, and generalization ability, among others. However, one of the most relevant concerns in artificial neural networks is determining an appropriate network size for a specific task. Large networks can define complex decision regions, while smaller networks can reach superior generalization capacity (Du and Swamy 2014). One common approach to determining network size is by using heuristics and/or trial-and-error, usually looking for good performance and generalization ability on a validation set, especially if the problem size is large. Another approach considers ways of 'growing' an artificial neural network until satisfactory performance is achieved (Guo et al 2012; Stanley and Miikkulainen 2002). A different technique uses 'pruning' methods (Castellano et al 1997; Suzuki et al 2001; Hussain and Alili 2016). In general, these methods begin by training an artificial neural network, which is large

Máximo Sánchez-Gutiérrez · John Goddard Close
Departamento de Ingeniería Eléctrica, Universidad Autónoma Metropolitana - Iztapalapa (México).
E-mail: edmax86@gmail.com, uamijohn@gmail.com

Enrique M. Albornoz · Hugo L. Rufiner
Instituto de Investigacin en Señales, Sistemas e Inteligencia Computacional, sinc(i), FICH-UNL-CONICET, Ciudad Universitaria, Paraje El Pozo, (S3000) Santa Fe (Argentina).
E-mail: emalbornoz@sinc.unl.edu.ar,lrufiner@sinc.unl.edu.ar
http://fich.unl.edu.ar/sinc.
Laboratorio de Cibernética, Facultad de Ingeniería, UNER, Oro Verde, Entre Ríos (Argentina).
E-mail: lrufiner@bioingenieria.edu.ar

enough to ensure a satisfactory performance. Afterwards, neurons are removed from the trained net (for example, the ones with the smallest weights) and then the network is often fine-tuned or retrained. This procedure could also be repeated until some convergence criterion is achieved, otherwise the smallest network that performed adequately is assumed to have the most suitable topology for the given data set. This type of pruning was called post–training pruning (PTP) (Castellano et al 1997; Reed 1993). In this work we will consider PTP.

In particular, there are two well–known techniques that have been applied in this case: the Optimal Brain Surgeon (OBS) and Optimal Brain Damage (OBD) (Hassibi et al 1993, 1994), both of which remove unimportant weights from a trained neural network. OBD does this by approximating the change in the error function when pruning a certain weight using a Taylor series expansion (LeCun et al 1990). OBS continues on from OBD, and computes the full Hessian matrix iteratively, giving a more exact approximation of the error function. Both algorithms retrain the network after each weight removal. This process however, is carried out without considering the class information of each pattern, i.e. none of these approaches are supervised, and there is no easy way to evaluate the impact of the weight's pruning on the class classification.

In contrast, our proposed technique acts more like a feature selection, by ranking the neurons and eliminating the less discriminative ones, along with their weights. One advantage of our proposal over these techniques is that no further training is required after pruning.

The question of network size is especially relevant as recent works show that larger or deeper nets can map the problem tasks features into a more appropriate space (Lee et al 2009; Hinton and Salakhutdinov 2006; Huang et al 2006, 2011). Consequently, new complications associated with complex and computationally demanding training algorithms must be addressed (Sutskever and Hinton 2007; Huang et al 2007; Cao et al 2013; Lu et al 2013).

In this context, Restricted Boltzmann Machines (RBM) have received increasing attention. The idea behind the deep learning paradigm suggests that, in order to learn high–level representations of data, a hierarchy of intermediate representations is required (Bengio 2009). These intermediate representations in a deep architecture translate into a feed–forward artificial neural network that has several layers of hidden units between the input and output layers. However, usually these hidden layers are hard to optimize. The best results obtained on supervised learning tasks involve an unsupervised learning component, usually in an unsupervised greedy pre–training phase (Hinton et al 2012; Erhan et al 2010; Hinton 2012). This means that, if the network is allowed to discover representations at various levels of abstraction, it will obtain better results since in the lower layers the network will find basic features, while in the upper layers more complex concepts will be represented (Hinton et al 2006; Bengio 2009).

Restricted Boltzmann machines and deep belief networks (DBN) have been applied successfully to several tasks. For example, a system for understanding natural language using DBNs was proposed in (Sarikaya et al 2014), a neuro–image classifier based on RBMs was presented in (Hjelm et al 2014) and a model for forecasting time series was proposed in (Kuremoto et al 2014). RBMs and DBNs are also used for parametric voice synthesis by (Zen and Senior 2014) and modeling statistical and probabilistic networks (Atwood et al 2014) among others. In speech emotion recognition, RBM and DBN achieved a significative performance improvement in comparison with other machine learning techniques (Albornoz et al 2014; Sánchez-Gutiérrez et al 2014).

Even though all these results results are encouraging, there is still an area open to improvement concerned with selecting an adequate topology for the network. One way to attempt this, as we have mentioned above, is by selecting, according to some criterion, the neurons that contribute most to the network's objective.

In Berglund et al (2015), the authors propose the use of mutual information between all the visible neurons and each individual hidden neuron, as a way of measuring the usefulness of each hidden unit in the RBM. They do not use any information about classes since their objective is to measure how much of the input information is represented in each hidden neuron. By contrast, in the present work we address the problem of selecting the most discriminative neurons in the hidden layer of an RBM using the class information of each pattern. In order to find a suitable network size, while retaining an adequate classification performance, a PTP method is proposed based on the discriminative 'ability' of each neuron. This capacity is measured by firstly feeding the original input data into the trained RBM network to obtain its output activations. Then, using all of the output activations for a particular neuron, the discriminative measures are used to quantify how different the activations are between classes. Finally, the hidden neurons are ranked based on this information. In this way, the most discriminative units are the only ones employed to feed into the final classifier, while the rest are pruned. From another perspective, we can see that we are also performing a type of feature selection by pruning hidden units in the network.

Five different discriminative measures are used in this paper. The results show that two of these measures seem to be good indicators of the most discriminative neurons, producing savings of generally more than 50% of the neurons in the original network, while maintaining an acceptable or improved error rate. Further, it is borne out that starting with a larger network architecture and then pruning is advantageous compared to using a smaller network to begin with, in the sense of obtaining a better classification rate. In this work we design and test the method for the two-class problem only, however generalization to more classes is straightforward. Finally, a quantitative index is introduced which can provide information on choosing a suitable pruned network.

In the next two sections, the proposed method for pruning an RBM is given and the corresponding five discriminative measures are introduced. Section 4 describes both speech databases used in the experiments. Furthermore, it also explains which features were extracted and the experiments that were conducted on them. Section 5 presents the results obtained and discusses them. Finally, conclusions and future work are presented in Section 6.

2 Post-training approach to pruning RBMs

In this section, we describe our approach to selecting the most useful neurons from an RBM based on their activation and discriminative ability.

2.1 Restricted Boltzmann machines

In the last few years, RBMs (Smolensky 1986) have been used as the first stage in classification systems, either as feature extractors or as a way to initialize neural networks (Hinton and Salakhutdinov 2006; Erhan et al 2010; Albornoz et al 2014). Specifically, a RBM is an artificial neural network with two layers, one layer of visible input units, and the other containing hidden units. There are connections between the units of the two layers and with the bias unit, but no connections between units in the same layer. The RBM is a generative stochastic network, so it can learn the probability distribution over the data. To do this, the RBM defines an energy function, E, for every configuration of visible and hidden state vectors, denoted by v and h respectively, by:

$$E(v,h) = -a^\top v - b^\top h - v^\top Wh \tag{1}$$

where W is a symmetric matrix of the weights connecting the visible and hidden units, and a, b are bias vectors on the connections of a bias unit to the visible and hidden layer, respectively. The joint probability, $p(v,h)$, for the RBM mentioned above, assigns a probability to every configuration (v,h) of visible and hidden vectors using:

$$p(v,h) = \frac{e^{-E(v,h)}}{Z} \tag{2}$$

where Z, known as the partition function, is a normalization constant that makes the probability sum to one that is defined as:

$$Z = \sum_{v,h} e^{-E(v,h)} \tag{3}$$

The probability assigned by the network to a visible vector v is:

$$p(v) = \frac{1}{Z}\sum_{h} e^{-E(v,h)} \tag{4}$$

It turns out that the lack of connections in the same layer of an RBM contributes to the property that its visible variables are conditionally independent, given the hidden variables, and vice versa. This means that we can write these conditional probabilities as:

$$\begin{aligned} p(v_j = 1|h) &= \sigma(a_i + \sum_j h_j w_{i,j}) \\ p(h_j = 1|v) &= \sigma(b_j + \sum_i v_i w_{i,j}) \end{aligned} \tag{5}$$

where

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{6}$$

The contrastive divergence algorithm (Hinton 2002) is applied to find the parameters W, a, and b.

2.2 Discriminative evaluation

After training the RBM, the outputs from the hidden neurons are fed into the final classifier, which is simply a 1-nn classifier, although clearly other classifiers could be employed. We hypothesize that often not all the RBM hidden neurons provide useful discriminative information for the final classifier, however it is difficult to know which of these are the most important. We study the effect of using discriminative measures

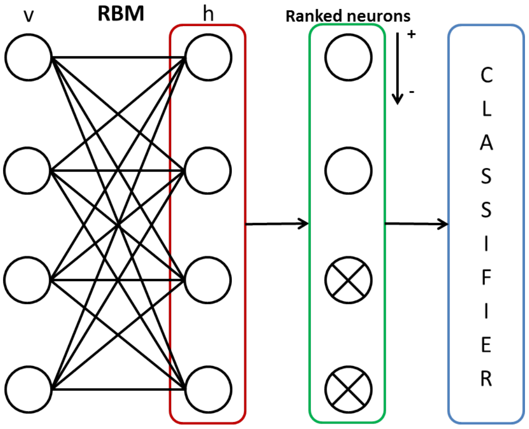


Fig. 1 Scheme for discriminative selection of neurons. It includes an unsupervised trained RBM, a method for ranking the most discriminative neurons and a final classifier.

to rank the hidden units with respect to their output values, and then prune them according to this ranking. We then analyze whether the resulting performance of the network and classifier is statistically acceptable. A graphical scheme of this process is shown in Figure 1.

As we mentioned previously, the work of (Berglund et al 2015) is of particular interest. They also propose a way to determine the importance of each hidden neuron. They note that while the variance of the output state of a hidden unit has been used previously, it is not applicable in the case of RBMs, given their stochastic nature. Their approach is to compute the mutual information between the input data (visible layer) and each hidden neuron. Both pruning and adding neurons are explored. The question arises as to whether mutual information is also useful in our approach, as a measure of a hidden unit's discriminative ability, and how pruning affects the resulting error rate. In this paper we also evaluate four other discriminative measures.

The general steps used in a multi–class approach are described in Algorithm 1, although in this paper we only use a binary–class version of the algorithm.

3 Discriminative Measures

Apart from mutual information, there are many other measures that can be used in our approach. In a general sense, they measure the *distance* between two discrete distributions p and q. The histograms of the output activations of each neuron are used to approximate the probability distributions of the two classes: p and q and, at the same time, to calculate the probabilities that are used in the computations of the following discriminative measures.

Algorithm 1 Discriminative evaluation

```
Input: An unsupervised trained RBM
for each class
    calculate the propagated value in the hidden layer for each training vector
end for

Input: The outputs of the RBM (the propagated vectors)
for each neuron $i$
    estimate separately the histograms of the output data for each class.
    calculate $i$'s discriminative value $D_i$ according to the selected measure.
end for
Rank the neurons according to their discriminative value in descending order.

Input: Ranked neurons
for $i \leftarrow 1$ to total number of neurons
    use the first $i$ neurons to classify the data using Knn.
end for
```

The first measure that we introduce is the *mutual information* (MI) computed for each hidden unit's activations for both classes.

3.1 Mutual information

The mutual information of two random variables is a measure of the dependence between them. More specifically, it quantifies the amount of information obtained about a random variable through another. This measure is closely related to the entropy H of a random variable X, with probability mass function p, that measures the randomness of the given variable i.e., the mean amount of information provided by an event is:

$$H(X) = -\sum_x p(x) \log_2 p(x) \tag{7}$$

The idea behind this definition is that, if one of the events is more probable than others, the observation of that event is less informative. Conversely, rarer events provide more information when observed. In this sense, it is possible to define the information for a particular event as $I(x) = -\log_2 p(x)$, so its expected value over all possible values of x leads to the Shannon's entropy (7).

From Shannon's entropy we can define the conditional entropy of a random variable X given the random variable Y by:

$$H(X|Y) = \sum_{x,y} p(x,y) \log_2 p(x|y) \tag{8}$$

where $p(x, y)$ is the joint probability that $X = x$ and $Y = y$.

Another definition we require in order to introduce the concept of mutual information is the joint entropy, which measures how much uncertainty there is in the two random variables X and Y taken together, and is defined by:

$$H(X, Y) = -\sum_{x,y} p(x, y) \log_2 p(x, y) \tag{9}$$

We have the following relationship:

$$H(X|Y) = H(X, Y) - H(Y) \tag{10}$$

In our implementation, we use the output activations of a hidden neuron for both classes as the joint distribution $p(x, y)$. The mutual information is then given by:

$$\begin{aligned} MI(X, Y) &= \sum_{x,y} p(x, y) \log_2 \frac{p(x, y)}{p(x)p(y)} \\ &= H(X) - H(X|Y) \\ &= H(X) + H(Y) - H(X, Y) \end{aligned} \tag{11}$$

3.2 Kullback–Leibler divergence

The Kullback-Leibler (KL) divergence is a discriminative measure between two probability distributions. Given two discrete random variables X and Y, described by probability distributions $p(x)$ and $q(x)$, the model defined by p is evaluated in terms of *closeness* to the distribution q. In other words, this divergence measures the ratio between the probability or uncertainty that a sample of p behaves like a sample of q.

Prior to the definition of KL divergence, we note that the *cross entropy* is defined by:

$$\begin{aligned} H(X; Y) &= E_X[\log_2 \frac{1}{q(y)}] \\ &= -\sum_{x} p(x) \log_2 q(y) \end{aligned} \tag{12}$$

where E_X represents the expectation regarding the probability distribution p. The KL information, or relative entropy of p with respect to q, can be defined as:

$$\begin{aligned} D_{KL}(p \parallel q) &= \sum_{x} p(x) \log_2 \frac{p(x)}{q(y)} \\ &= -\sum_{x} p(x) \log_2 q(y) \\ &\quad + \sum_{x} p(x) \log_2 p(x) \\ &= H(X; Y) - H(X) \end{aligned} \tag{13}$$

with $H(X; Y)$ being the cross entropy of X and Y and $H(X)$ the entropy of X.

However, the KL divergence is not symmetric and in our implementation we consider a symmetric measure, which we refer to as KLS and is also known as Jeffreys' divergence, defined by:

$$D_{KLS}(p \parallel q) = \frac{(D_{KL}(p \parallel q) + D_{KL}(q \parallel p))}{2} \tag{14}$$

3.3 Wasserstein distance

The Wasserstein, or *Earth Mover's Distance* (EMD) (Pele and Werman 2009), is based on the minimum cost to be paid, or work to be done, to transform one distribution into another. It is more robust than other techniques that use histograms since it operates with representations of variable length distributions, thus avoiding problems with intervals or bins that are typical when working with histograms.

Intuitively speaking, given two distributions, one can be considered as a mass of soil, while the other as holes to be filled with. This means that the EMD measures the work necessary to move or transform one distribution into another, where a unit of work corresponds to transport a unit of soil by one unit of distance. The distance measure between locations is known as the ground distance, and is introduced below in (15).

The EMD is defined for histograms of the form $(\mu, p(x))$, where μ is the mean of the histogram while $p(x)$ is the number of occurrences of x. The histograms may or may not be normalized, so that the total mass of two histograms may not be equal.

Given two histograms X and Y, the EMD is defined in terms of *optimal flow* as $F = (f_{ij})$ that minimizes the work W:

$$W(X, Y, F) = \sum_{i,j} f_{ij} \delta_{ij} \tag{15}$$

Where $\delta_{ij} = dist(\mu_i, \mu_j)$ is some distance between μ_i and μ_j, for example the Euclidean distance, while

$W(X, Y, F)$ is the work needed to move the soil from one histogram to another.

The flow (f_{ij}) must meet the following restrictions:

$$\begin{aligned} & f_{ij} \geq 0 \\ & \sum_j f_{ij} \leq \mu_i \\ & \sum_i f_{ij} \leq \mu_j \\ & \sum_{i,j} f_{ij} = min(\sum_i \mu_i, \sum_j \mu_j) \end{aligned} \tag{16}$$

The first constraint allows us to move soil from X to Y and not the other way, the second limits the amount of soil that can be sent from X, the third constraint limits the amount of maximum soil that Y can receive, and finally, the fourth restriction forces us to move the greatest possible quantity of soil *(total flow).*

Once the transport problem is solved and the optimal flow F is found, the EMD is defined as the work W normalized by the total flow defined in equation 16:

$$EMD(X, Y) = \frac{\sum_{i,j} f_{ij} \delta_{ij}}{\sum_{i,j} f_{ij}} \tag{17}$$

3.4 Difference of Conditional Activation Frequency

In (Rolon et al 2014), the authors describe a method to select the most discriminative atoms from a fixed dictionary in order to improve a neural network's classification performance on a sparse representation. A dictionary is defined as a matrix, $\Phi \in \mathbb{R}^{M \times N}$, whose columns $\boldsymbol{\phi}_j$ are called atoms. A particular signal can be synthesized by a linear combination of these atoms, also known as signal features.

The idea behind this particular method is to select the most discriminative atoms of the dictionary using the atom's 'activation' probability given the class. An atom is supposed to be active for a particular signal (of a given class) if the corresponding coefficient is different from zero in its representation. The candidates considered are those atoms with higher absolute difference between activation probabilities for each class. That is, an atom is more discriminative if it is active more times for signals belonging to one class, than for the signals belonging to the other class.

In our work we can apply a similar idea to neurons, instead of atoms, as follows and we refer to it as the *difference of conditional activation frequency* (DCAF):

Let $p_i \triangleq p(x_i \neq 0 | \mathbf{x} \in C_1)$ and $q_i \triangleq q(y_i \neq 0 | \mathbf{y} \in C_2)$ be the activation probabilities of the neuron i for Class 1 and Class 2, respectively. Our implementation of this criteria is as follow:

$$D_{DCAF}(X, Y) = |p_i - q_i| \tag{18}$$

where p_i and q_i are computed using the relative activation frequency for each neuron:

$$\begin{aligned} p_i &\approx \frac{\#\text{ activations of neuron } i \text{ for data} \in C_1}{\#\text{ all data} \in C_1} \\ q_i &\approx \frac{\#\text{ activations of neuron } i \text{ for data} \in C_2}{\#\text{ all data} \in C_2} \end{aligned} \tag{19}$$

In the particular case of equally represented classes (i.e. balanced dataset), both denominators are the same so the criteria can be simplified only computing the absolute difference between the number of activations per each class.

3.5 Welch's t–test

Welch's t–test is an adaptation of Student's t–test (Wilcox 1995; Keselman et al 2004) that compares the means of two groups. It is a good approach when the homogeneity of variances assumption is not met, especially with unequal sample sizes. The general idea here is that the means of the output activations can be used to estimate how far the distributions are from each other or, in a way, test if the statistical units underlying the two samples being compared are non-overlapping.

Let μ_i be the sample means of the output activations for one neuron and class *i=1,2*, σ_i the variance and n_i the group size, then our implementation of Welch's test is given by:

$$WT = \frac{(\mu_1 - \mu_2)^2}{\frac{\sigma_1}{n_1} + \frac{\sigma_2}{n_2}} \tag{20}$$

In the context of statistical pattern classification with a two-class problem with normal distributions, this test is related to the so called Fisher's ratio (Hegde et al 2015), since in equation 20 the numerator reflects the inter–class variance while the denominator considers the intra–class variance.

That is, μ_1 is the mean of the output activations of one neuron for Class 1 patterns and μ_2 is the mean of the output activations of one neuron for Class 2. Likewise, σ_1 is the variance of the output activations of one neuron for Class 1 and σ_2 is the variance of the output activations of one neuron for Class 2.

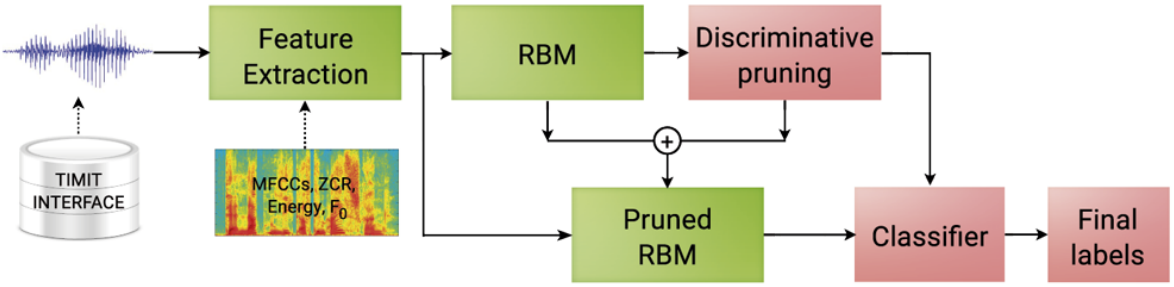


Fig. 2 Conceptual flowchart of the whole general process for the experiments.

4 Experiments

In this section we describe the experiments performed in the paper. The general procedure is presented in Figure 2, where the first step involves extracting features from the speech signal. In the second step, an RBM with N hidden units is trained in an unsupervised manner on a given training set. The values of N used in this paper are specified in section 5. Then, this training set is propagated through the RBM to obtain the activations of the hidden units and, according to Algorithm 1, each hidden unit is ranked using the discriminative measures. Once the hidden units are ranked, networks with $1, 2, \ldots, N$ best ranked units are systematically formed. Finally, the outputs of these pruned RBMs are classified using a minimum–distance criterion (with respect to a set of prototypes/centroids).

It is important to note that a repeated random validation method (Michie et al 1994) was implemented for the databases (see below for the description) in all the experiments conducted. This allows us to obtain more stable results and avoid the biased estimates of recognition error that are usually present in experiments with only one training and test partition.

In the next subsections we describe the speech databases used in the experiments, the feature extraction procedure, and, finally, comment on the classifier.

4.1 Speech Corpora

We evaluate our approach using two well-known speech corpora.

TIMIT

TIMIT is a corpus of read speech created for the development and evaluation of automatic speech recognition systems by the Massachusetts Institute of Technology, SRI International and Texas Instruments, Inc. (Garofolo et al 1993). The corpus has utterances of 630 speakers expressed in the eight major dialects of American English, which include time-aligned orthographic, phonetic and word transcriptions. In this work, the phonetic alignment is used to obtain individual files of every phoneme occurrence. Moreover, all the regional dialects, including both male and female speakers, are considered.

As is to be expected, certain sets of phonemes are more difficult to classify than others. For example, the set of English phonemes: /b/, /d/, /eh/, /ih/ and /jh/ are difficult to identify (Stevens 2000; Martínez et al 2012; Vignolo et al 2016). From these phonemes we consider the vowels /eh/ and /ih/ due to their closeness in formant space.

The test and training subsets defined in the TIMIT database are already balanced for phonetic and dialectal coverage. The training set contains 8904 samples while the test set has 3149 samples.

INTERFACE

This corpus was created to study emotional speech by the Center for Language and Speech Technologies and Applications (TALP) of the Polytechnic University of Catalonia (UPC) (Hozjan et al 2002). Although it has utterances in English, French, Slovene and Spanish, only the latter is employed in this work. The Spanish set was spoken by two professional actors, one male and one female. There are 184 utterances that are spoken using six emotions (joy, sadness, anger, fear, disgust and surprise) plus neutral. Thus, for each speaker, $1,288$ utterances were produced.

All utterances belonging to the same emotional class are labeled with the name of the class and their transcriptions are ignored. Each utterance is represented by one unique pattern in a data partition.

For the experiments, we use two classes (anger and neutral) and repeated random validation with 70% for training and 30% for testing. The partition is balanced with respect to the speakers and emotional classes.

4.2 Feature extraction

One of the most relevant parts in automatic classification systems is the pre-processing stage, where a set of significant features is produced. In automatic speech recognition (ASR) and emotion recognition systems, this process should be able to extract the key-features to exploit the capabilities of the classifier (Huang et al 2001). Many researchers have developed their own optimal feature sets for specific tasks (Vignolo et al 2016; Montefusco and Puccio 2014), however mel frequency cepstral coefficients (MFCC) are the most widely used features for speech recognition. The MFCCs are based on a linear model of voice production and a psycho-acoustic frequency mapping according to the mel scale (Huang et al 2001). The use of prosodic features in ASR and emotion recognition has already been studied and discussed extensively (Adell Mercado et al 2005; Borchert and Dusterhoft 2005; Milone and Rubio 2003).

As our principal concern is not with the selection of the best acoustic features, we simply chose a standard set of well-known features consisting of: energy, zero crossing rate and fundamental frequency (F_0). To this end, the first 12 mean MFCCs, the mean F_0, the average of the zero crossing rate and the mean of the energy, plus the means of first derivatives of each one were extracted using the *OpenSMILE* (Eyben et al 2010) toolbox. Hence, each utterance is represented by a 30–dimensional vector in all the experiments.

This can be seen in Figure 2, where these features serve as inputs to the RBM, and are then sent on from the RBM to the classifier. The exact outputs received by the classifier depend on the pruning applied to the hidden units which, in turn, depends on identifying the most discriminating neurons. Additionally, we can observe that the proposed process can also be seen as feature selection on pre–trained RBMs.

4.3 Classifier

The scheme presented in Figure 2 outlines our approach to network pruning. Here we apply a classifier to the pruned network in order to evaluate how effective this procedure has been. Several standard classifiers could be applied in this block: K-nearest neighbors (KNN), decision trees, multilayer perceptrons (MLP), and support vector machines (SVM), among others (cf. Haykin et al (2009)).

In this work we apply 1–NN classifiers to the binary classification tasks described above. The results reported in Section 5 are the averaged accuracy computed based on the outputs of the pruned networks taken into consideration. In order to determine if the pruning process is beneficial in attaining adequate classification results, a baseline was defined by using the initial unpruned RBM as input to the 1–NN classifier.

5 Results and Discussion

In this section, essentially three types of results are presented and discussed for each database, which consider different aspects of the pruning process according to Algorithm 1. The first type considers the classification error in relation to the number of pruned hidden units, for each of the discriminative measures described in Section 3 (Figures 3 and 4). The second is concerned with the percentage of hidden nodes required to obtain a reasonable classification error (Figures 5 and 6 for two of the measures). Finally, we introduce a quantitative index, analogous to one used in principal component analysis, to provide further information on the pruning process (Figure 7).

For all the figures and tables, the information given is the averaged outcomes of 10 randomly initialized experiments. Using this repeated random validation methodology, we can obtain more stable and reliable behaviors and tendencies in the error rates for the pruned units. For both databases, five different RBM configurations were initially implemented: the network's input always consisted of 30 visible units, while the number of hidden units were 15, 30, 60, 120 and 240 (that is, 0.5, 1, 2, 4 and 8 times the number of visible units). Since the results presented here are the averaged outcomes of 10 randomly initialized experiments, we use *confidence intervals (CI)*, defined by equation 21, and calculated with a 95% confidence level, as a way to determine if the errors in Figures 3–6 are suitably close to the baseline. The confidence intervals are calculated using:

$$CI = \left(\bar{x} - t\frac{\sigma}{\sqrt{n}}, \quad \bar{x} + t\frac{\sigma}{\sqrt{n}} \right) \tag{21}$$

where $\bar{x}$ is the sample mean, σ is the sample standard deviation, n is the number of samples and t is the 95% *t–statistic* with 9 degrees of freedom.

The baseline results are depicted in the figures by a continuous line, whereas the confidence intervals are represented by a dotted line.

Experiments using the TIMIT corpus are presented in Figure 3. The first experiment, presented in Figure 3a, shows that when the RBM configuration has less units than the dimensionality of the input vectors, none of the pruned networks really improves on the baseline result. The second experiment, presented in Figure 3b, shows that using 30 units is more beneficial than 15 since the baseline error is improved. At the same time, the first better-than-baseline error is around 15 units, intriguingly the same number of hidden units as in experiment (a). Here we see that the result obtained is better than the baseline and, in fact, better than all the pruned networks in the first experiment. In spite of this performance improvement, we can observe that there is still room for further improvement, as shown in Figures 3c,d,e.

The same experiments using the INTERFACE corpus are presented in Figure 4. The general behavior of these experiments is similar to those observed with the TIMIT dataset: the baseline error is higher when 15 hidden units are initially used and, all 15 units are required to attain it. In the same way as before, the experiments that use more units (Figure 4b,c,d,e) benefit the most. However, unlike the second TIMIT experiment (Figure 3b), it seems that 30 hidden units are not sufficient to improve the performance. This may be caused by several factors e.g. the variability between sentences, the fact that the features vectors are computed for the whole utterance and in general, the complexity of the INTERFACE corpus.

A trend can be observed in the experiments presented in Figures 3, 5 and Figures 4, 6 for the Welch and DCAF measures, and to a certain extent EMD, where around 20%-30% of the ranked units are sufficient to attain and even improve the error baseline, thus giving a possible reduction of at least 70% of the hidden units. Welch and DCAF measures therefore produce networks with fewer neurons that reach acceptable error rates faster.

Tables 1 and 2 provide further information on these two measures. The information they give refers to the smallest number of pruned units after which only acceptable classification errors are obtained. They show the error rates, the percentage of saved units, the number of units retained after pruning, and the error baseline for both the TIMIT and INTERFACE corpora. After inspecting the Tables, we can observe that by using more initial units in the network, the baseline error rates are generally improved.

The smallest number of pruned units, after which only acceptable classification errors are obtained, gives an adequate pruning of the network, without necessarily giving the pruned network with the smallest possible error rate, as we might expect. For initial networks with more than 15 units, we see from Figures 3 and 4 that this is the case for all the measures. With this in mind, we try to estimate a suitable pruning value using Equation 22. This attempts to provide a trade-off between the error rate and the number of pruned units, instead of solely focusing on the number of pruned neurons. This equation calculates the Relative Discriminative Cumulative Gain (RDCG) and is analogous to a similar equation used in Principal Component Analysis (Jolliffe 2002) for dimensionality reduction. The RDCG is calculated using:

$$RDCG_j = \frac{\sum_{i=1}^{j} d_i}{\sum_{i=1}^{30} d_i} \tag{22}$$

where d_i is the value of one of the measures on the ranked unit i.

Figure 7 illustrates RDCG for the five measures on both databases using 240 hidden unit RBMs. This provides useful information for determining a possible pruning point. We can explore this idea further by constructing Tables 3 and 4, which are similar to Tables 1 and 2, but are formed using three values of RDCG, .7, .8, and .9 for the Welch and DCAF measures. Here, .8 seems to be a reasonable compromise for both databases.

Table 1 Classification results on TIMIT using the best two measures.

		Welch		
RBM units	Retained	% Saved	% Error	% Baseline
15	7	53.33%	39.23%	37.87%
30	13	56.66%	35.18%	34.03%
60	15	75.00%	31.74%	30.98%
120	18	85.00%	29.64%	29.01%
240	12	95.00%	30.00%	29.45%
		DCAF		
15	5	66.66%	39.26%	37.87%
30	9	70.00%	35.23%	34.03%
60	10	83.33%	31.73%	30.98%
120	17	85.83%	29.68%	29.01%
240	13	94.58%	30.22%	29.45%

Table 2 Classification results on INTERFACE using the best two measures.

		Welch		
RBM units	Retained	% Saved	% Error	% Baseline
15	7	53.33%	14.63%	11.94%
30	17	43.33%	8.65%	6.84%
60	17	71.66%	4.84%	3.91%
120	35	70.83%	3.83%	3.23%
240	67	72.08%	2.63%	2.41%
		DCAF		
15	7	53.33%	15.01%	11.94%
30	13	56.66%	8.71%	6.84%
60	18	70.00%	5.04%	3.91%
120	23	80.83%	3.72%	3.23%
240	73	69.58%	2.63%	2.41%

6 Conclusions and future work

In this work, a post-training pruning method for restricted Boltzmann machines was proposed. The hidden units were ranked and then pruned using five discriminative measures: mutual information, Kullback-Leibler divergence, Wasserstein distance, Difference of Conditional Activation Frequency and Welch's t-test. In contrast to the work of Berglund et al (2015), which explored the application of mutual information to all the visible neurons and each individual hidden neuron without considering class information, that is, in an unsupervised fashion, our approach has included class information in the ranking algorithm (c.f. Algorithm 1). This can be considered as a method of feature extraction from the hidden units of a RBM. We employed two well-known speech corpora, given our previous interest in this area: a speech recognition corpus TIMIT, from which we used the phonemes /eh/ and /ih/ due to their closeness in formant space, and the emotional speech corpus INTERFACE, from which we took the Spanish utterances of the spoken emotions of anger and neutral.

We found that the adoption of the ranking approach in the pruning methodology presented in this work is very promising. The results indicate that once a suitable number of initial neurons has been chosen, pruned networks with less than 50% of the neurons produce better-than-baseline error results.

Results show that the two best measures in terms of achieving an acceptable error rate with fewer neurons, are Welch's t–test and the DCAF, with EMD coming closely behind. This is interesting, given that in previous work on pruning neural networks, MI has been the preferred measure.

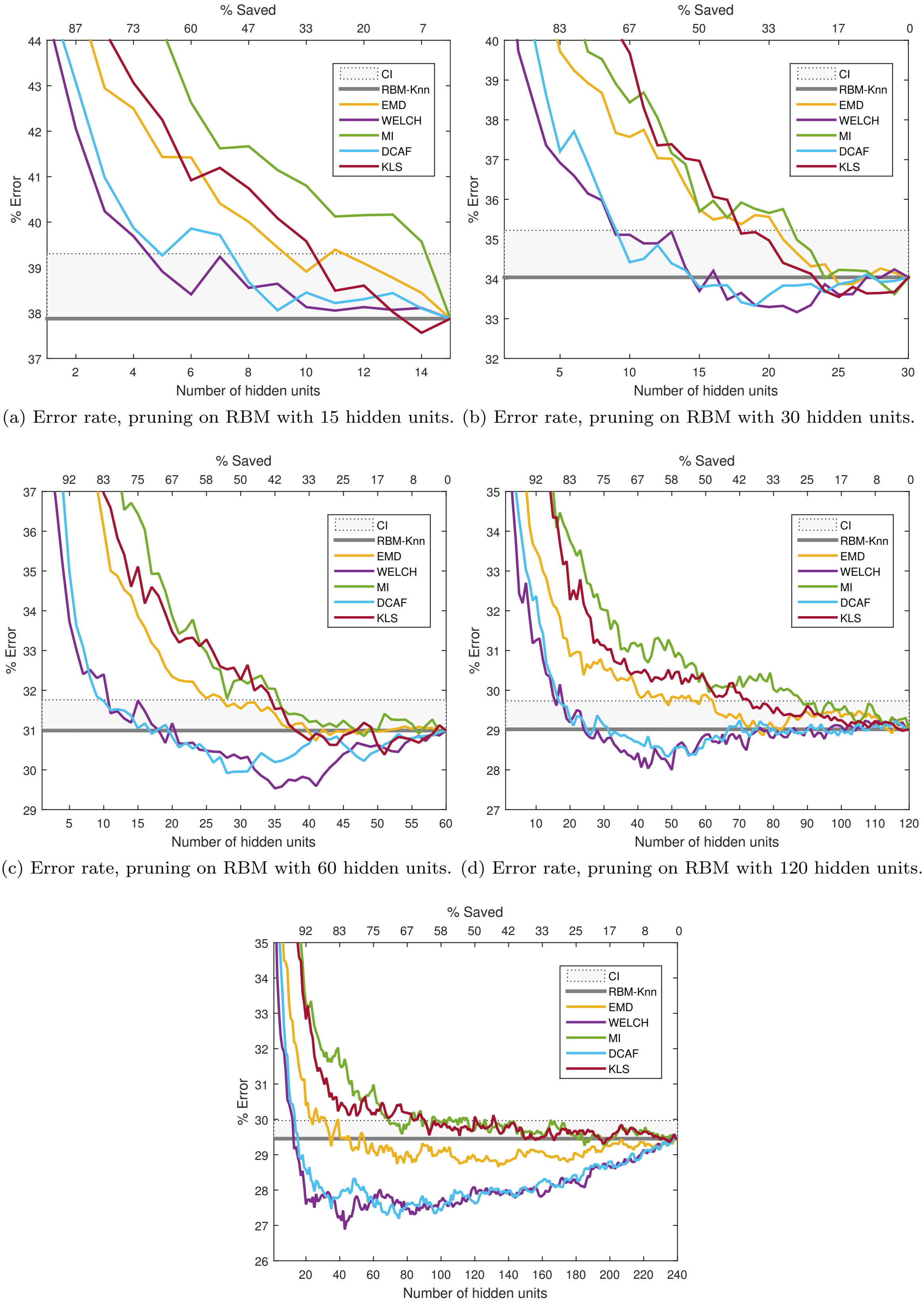
Almost all the pruned networks use less units than the full RBM in the previous row, and give better classification rates. This suggests the advantages of training a larger net and pruning, rather than trying to find an 'exact' architecture.

Finally, as the smallest number of pruned units doesn't necessarily give the pruned network with the smallest possible error rate, we introduced the RDCG index as an alternative way to find a suitable pruning value. The results suggest that an RDCG value of .8 for Welch or DCAF gives an acceptable error rate while still providing savings of at least 50% for each RBM architecture.

Acknowledgements The authors wish to thank to *SEP* and *CONACyT* (Program SEP-CONACyT CB-2012-01, No.182432) and the *Universidad Autónoma Metropolitana* from México; *Universidad Nacional de Litoral* (with PACT 2011 #58, CAI+D 2011 #58-511, PRODACT 2016 (FICH-UNL)), ANPCyT (with PICT 2015-0977) and *Consejo Nacional de Investigaciones Científicas y Técnicas* (CONICET) from Argentina, for their support. We also want to thank ELRA for supplying the, Emotional speech synthesis database, catalogue reference: ELRA-S0329.

Compliance with Ethical Standards

Conflict of Interest: The authors declare that they have no conflict of interest.
Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.



(a) Error rate, pruning on RBM with 15 hidden units.

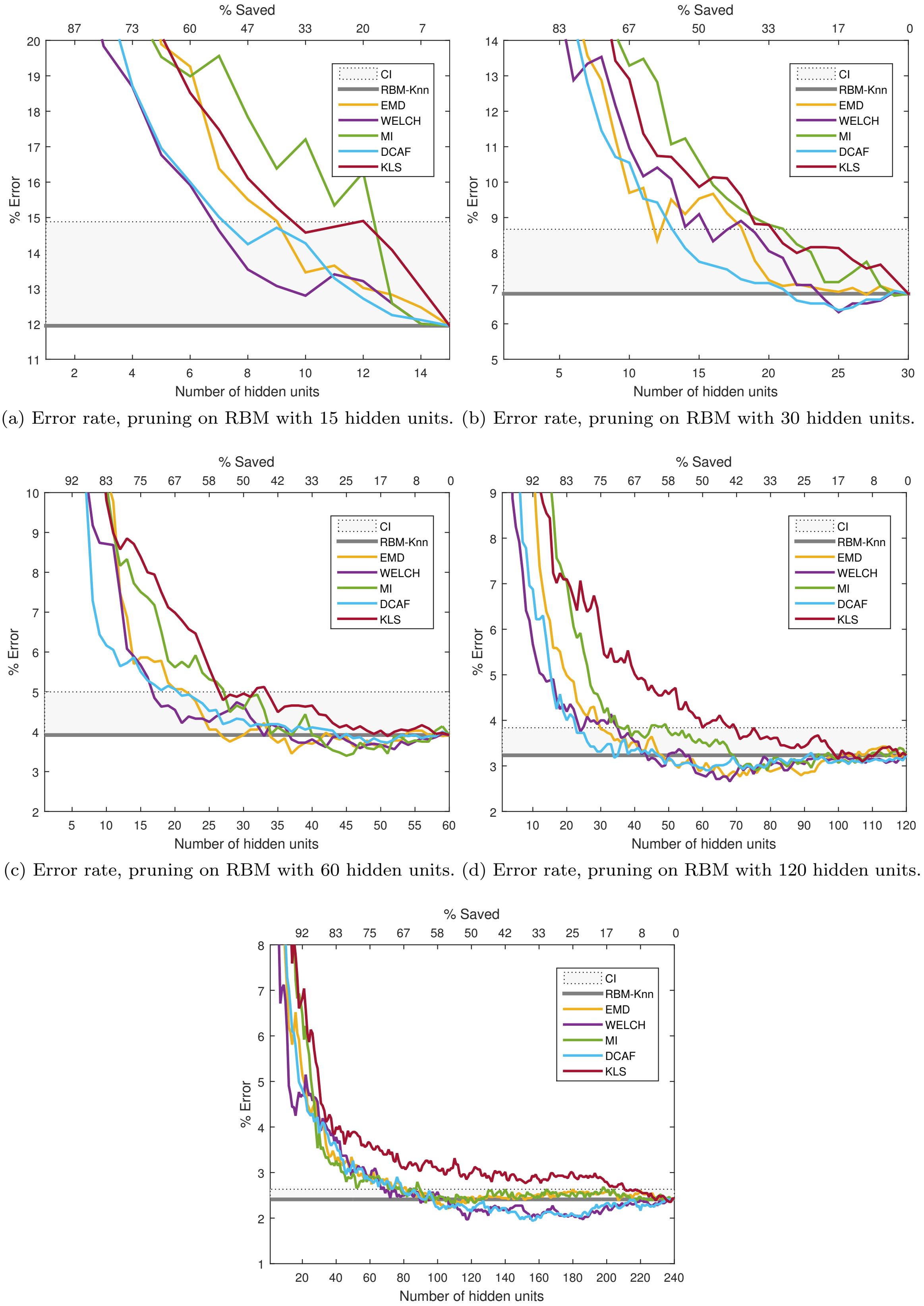
(b) Error rate, pruning on RBM with 30 hidden units.

(c) Error rate, pruning on RBM with 60 hidden units.

(d) Error rate, pruning on RBM with 120 hidden units.

(e) Error rate, pruning on RBM with 240 hidden units.

Fig. 3 Classification results using TIMIT corpus. Test were performed using 15, 30, 60, 120 and 240 neurons in the RBM.

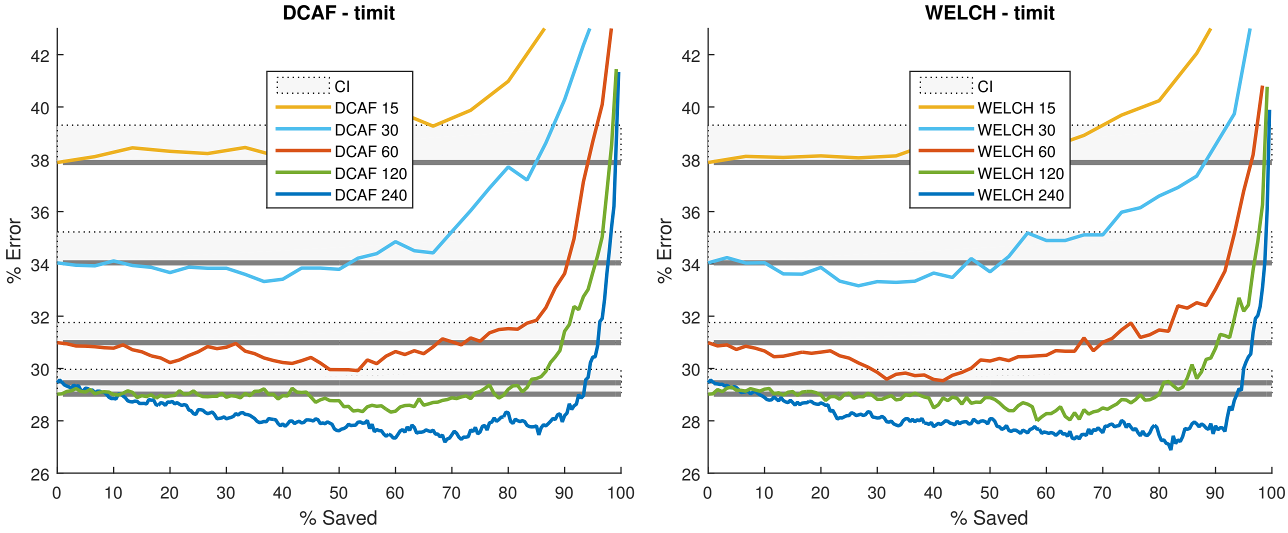


(a) Error rate, pruning on RBM with 15 hidden units. (b) Error rate, pruning on RBM with 30 hidden units.

(c) Error rate, pruning on RBM with 60 hidden units. (d) Error rate, pruning on RBM with 120 hidden units.

(e) Error rate, pruning on RBM with 240 hidden units.

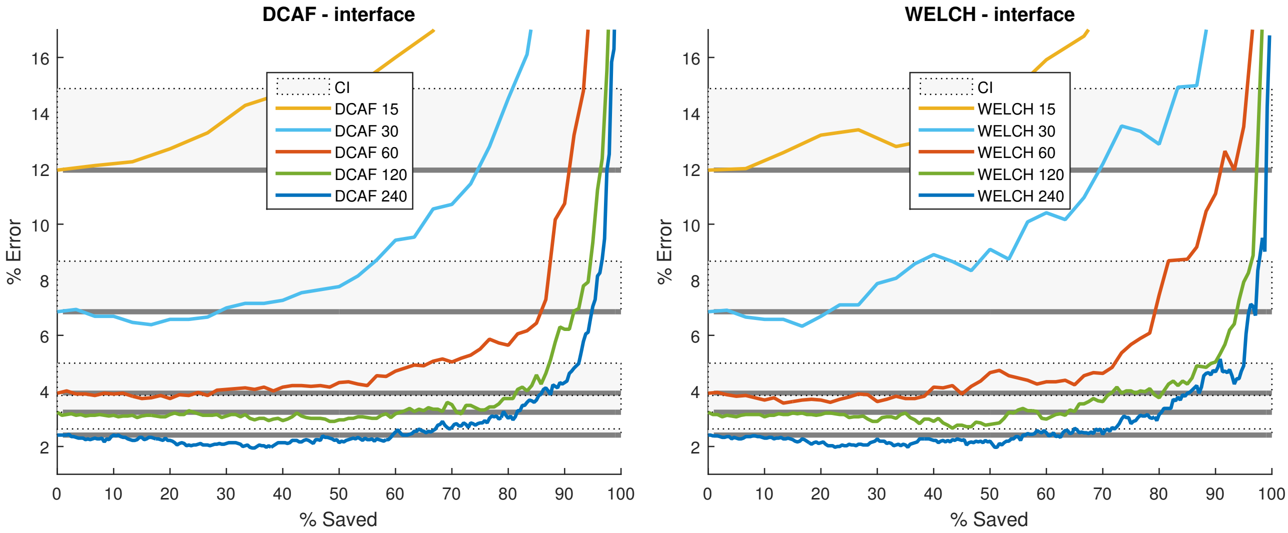
Fig. 4 Classification results using INTERFACE corpus. Test were performed using 15, 30, 60, 120 and 240 neurons in the RBM.



(a) DCAF-only error curves for 15, 30, 60, 120 and 240 hidden units.

(b) Welch-only error curves for 15, 30, 60, 120 and 240 hidden units.

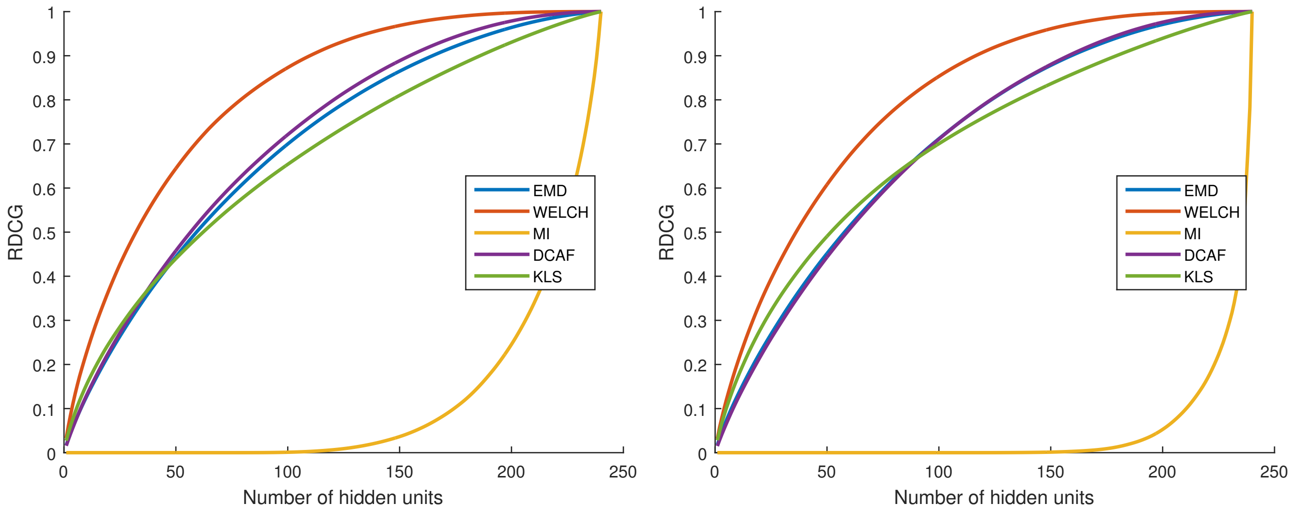
Fig. 5 Error curves for DCAF and Welch using TIMIT corpus. Test were performed using 15, 30, 60, 120 and 240 neurons in the RBM.



(a) DCAF-only error curves for 15, 30, 60, 120 and 240 hidden units.

(b) Welch-only error curves for 15, 30, 60, 120 and 240 hidden units.

Fig. 6 Error curves for DCAF and Welch using INTERFACE corpus. Test were performed using 15, 30, 60, 120 and 240 neurons in the RBM.



(a) Evaluation using TIMIT data.

(b) Evaluation using INTERFACE data.

Fig. 7 Relative Discriminative Cumulative Gain (RDCG) on RBMs with 240 hidden units.

Table 3 Classification results on TIMIT using the best two measures and RDCG as pruning point estimator.

Welch										
	0.7 RDCG			0.8 RDCG			0.9 RDCG			
RBM units	Retained	% Saved	% Error	Retained	% Saved	% Error	Retained	% Saved	% Error	% Baseline
15	4	73.33%	39.69%	5	66.67%	38.91%	7	53.33%	39.24%	37.87%
30	8	73.33%	35.98%	10	66.67%	35.11%	14	53.33%	34.28%	34.03%
60	15	75.00%	31.74%	20	66.67%	31.17%	28	53.33%	30.03%	30.98%
120	30	75.00%	28.69%	41	65.83%	28.06%	56	53.33%	28.39%	29.01%
240	59	75.42%	27.64%	79	67.08%	27.44%	109	54.58%	27.64%	29.45%
DCAF										
	0.7 RDCG			0.8 RDCG			0.9 RDCG			
RBM units	Retained	% Saved	% Error	Retained	% Saved	% Error	Retained	% Saved	% Error	% Baseline
15	6	60.00%	39.86%	7	53.33%	39.71%	9	40.00%	38.06%	37.87%
30	11	63.33%	34.50%	14	53.33%	34.22%	18	40.00%	33.41%	34.03%
60	24	60.00%	30.65%	30	50.00%	29.96%	39	35.00%	30.57%	30.98%
120	47	60.83%	28.53%	60	50.00%	28.77%	77	35.83%	29.13%	29.01%
240	96	60.00%	27.34%	121	49.58%	27.63%	155	35.42%	27.99%	29.45%

Table 4 Classification results on INTERFACE using the best two measures and RDCG as pruning point estimator.

Welch										
	0.7 RDCG			0.8 RDCG			0.9 RDCG			
RBM units	Retained	% Saved	% Error	Retained	% Saved	% Error	Retained	% Saved	% Error	% Baseline
15	5	66.67%	16.77%	7	53.33%	14.63%	9	40.00%	13.07%	11.94%
30	9	70.00%	12.16%	11	63.33%	10.16%	15	50.00%	9.09%	6.84%
60	17	71.67%	4.84%	22	63.33%	4.38%	30	50.00%	4.65%	3.91%
120	32	73.33%	3.91%	42	65.00%	3.26%	58	51.67%	2.84%	3.23%
240	65	72.91%	2.82%	86	64.16%	2.46%	117	51.25%	1.97%	2.41%
DCAF										
	0.7 RDCG			0.8 RDCG			0.9 RDCG			
RBM units	Retained	% Saved	% Error	Retained	% Saved	% Error	Retained	% Saved	% Error	% Baseline
15	6	60.00%	16.00 %	7	53.33%	14.01%	10	33.33%	14.27%	11.94%
30	11	63.33%	9.53%	14	53.33%	8.13%	18	40.00%	7.26%	6.84%
60	23	61.67%	4.82%	29	51.67%	4.32%	38	36.67%	4.16%	3.91%
120	47	60.83%	3.15%	60	50.00%	2.95%	76	36.67%	3.01%	3.23%
240	98	59.17%	2.32%	124	48.33%	2.30%	158	34.17%	1.97%	2.41%

References

Adell Mercado J, Bonafonte Cávez A, Escudero Mancebo D (2005) Analysis of prosodic features: towards modelling of emotional and pragmatic attributes of speech. Procesamiento de Lenguaje Natural 35:277–283

Albornoz EM, Sánchez-Gutiérrez M, Martinez-Licona F, Rufiner HL, Goddard J (2014) Spoken Emotion Recognition Using Deep Learning, Springer International Publishing, pp 104–111

Atwood J, Towsley D, Gile K, Jensen DD (2014) Learning to generate networks. In: Networks: From Graphs to Rich Data, NIPS 2014 Workshop, Montreal, QC, Canada

Bengio Y (2009) Learning deep architectures for AI. Foundations and Trends in Machine Learning 2(1):1–127

Berglund M, Raiko T, Cho K (2015) Measuring the usefulness of hidden units in boltzmann machines with mutual information. Neural Networks 64:12 – 18

Borchert M, Dusterhoft A (2005) Emotions in speech - experiments with prosody and quality features in speech for use in categorical and dimensional emotion recognition environments. In: Proceedings of IEEE International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE), pp 147–151

Cao F, Liu B, Park DS (2013) Image classification based on effective extreme learning machine. Neurocomputing 102:90 – 97

Castellano G, Fanelli AM, Pelillo M (1997) An iterative pruning algorithm for feedforward neural networks. IEEE Transactions on Neural Networks 8(3):519–531

Du KL, Swamy M (2014) Neural Networks and Statistical Learning. Springer London

Erhan D, Bengio Y, Courville A, Manzagol PA, Vincent P, Bengio S (2010) Why does unsupervised pre-training help deep learning? Journal of Machine Learning Research 11(Feb):625–660

Eyben F, Wöllmer M, Schuller B (2010) Opensmile: the munich versatile and fast open-source audio feature extractor. In: Proceedings of the international conference on Multimedia, ACM, New York, NY, USA, MM '10, pp 1459–1462

Garofolo JS, Lamel LF, Fisher WM, Fiscus JG, Pallett DS (1993) DARPA TIMIT acoustic-phonetic continous speech corpus CD-ROM. NIST speech disc 1-1.1. NASA STI/Recon technical report n 93

Guo XL, Wang HY, Glass DH (2012) A growing bayesian self-organizing map for data clustering. In: 2012 International Conference on Machine Learning and Cybernetics, vol 2, pp 708–713

Hassibi B, Stork DG, Wolff GJ (1993) Optimal brain surgeon and general network pruning. In: Neural Networks, 1993., IEEE International Conference on, IEEE, pp 293–299

Hassibi B, Stork DG, Wolff G, Watanabe T (1994) Optimal brain surgeon: Extensions and performance comparisons. Advances in neural information processing systems pp 263–263

Haykin SS, Haykin SS, Haykin SS, Haykin SS (2009) Neural networks and learning machines, vol 3. Pearson Upper Saddle River, NJ, USA:

Hegde S, Achary K, Shetty S (2015) Feature selection using fisher's ratio technique for automatic speech recognition. arXiv preprint arXiv:150503239

Hinton GE (2002) Training Products of Experts by Minimizing Contrastive Divergence. Neural Computation 14(8):1771–1800

Hinton GE (2012) A Practical Guide to Training Restricted Boltzmann Machines, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 599–619

Hinton GE, Salakhutdinov RR (2006) Reducing the dimensionality of data with neural networks. Science 313(5786):504–507

Hinton GE, Osindero S, Teh YW (2006) A fast learning algorithm for deep belief nets. Neural computation 18(7):1527–1554

Hinton GE, Deng L, Yu D, Dahl GE, r Mohamed A, Jaitly N, Senior A, Vanhoucke V, Nguyen P, Sainath TN, Kingsbury B (2012) Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal Processing Magazine 29(6):82–97

Hjelm RD, Calhoun VD, Salakhutdinov R, Allen EA, Adali T, Plis SM (2014) Restricted boltzmann machines for neuroimaging: An application in identifying intrinsic networks. NeuroImage 96:245 – 260

Hozjan V, Kacic Z, Moreno A, Bonafonte A, Nogueiras A (2002) Interface databases: Design and collection of a multilingual emotional speech database. In: Third Language Resources and Evaluation Conference LREC 2002, Las Palmas de Gran Canaria, Spain

Huang FJ, Boureau YL, LeCun Y, et al (2007) Unsupervised learning of invariant feature hierarchies with applications to object recognition. In: 2007 IEEE conference on computer vision and pattern recognition, IEEE, pp 1–8

Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70(1):489–501

Huang GB, Wang DH, Lan Y (2011) Extreme learning machines: a survey. International Journal of Machine Learning and Cybernetics 2(2):107–122

Huang X, Acero A, Hon HW, Foreword By-Reddy R (2001) Spoken language processing: A guide to theory, algorithm, and system development. Prentice hall PTR

Hussain S, Alili AA (2016) A pruning approach to optimize synaptic connections and select relevant input parameters for neural network modelling of solar radiation. Applied Soft Computing

Jolliffe I (2002) Principal component analysis. Wiley Online Library

Keselman H, Othman AR, Wilcox RR, Fradette K (2004) The new and improved two-sample t test. Psychological Science 15(1):47–51

Kuremoto T, Kimura S, Kobayashi K, Obayashi M (2014) Time series forecasting using a deep belief network with restricted boltzmann machines. Neurocomputing 137:47 – 56

Le QV (2013) Building high-level features using large scale unsupervised learning. In: 2013 IEEE international conference on acoustics, speech and signal processing, IEEE, pp 8595–8598

LeCun Y, Denker JS, Solla SA, Howard RE, Jackel LD (1990) Optimal brain damage. In: NIPs, vol 2, pp 598–605

Lee H, Grosse R, Ranganath R, Ng AY (2009) Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: Proceedings of the 26th Annual International Conference on Machine Learning, ACM, New York, NY, USA, ICML '09, pp 609–616

Lu B, Wang G, Yuan Y, Han D (2013) Semantic concept detection for video based on extreme learning machine. Neurocomputing 102:176 – 183

Martínez C, Goddard J, Milone D, Rufiner H (2012) Bioinspired sparse spectro-temporal representation of speech for robust classification. Computer Speech & Language

26(5):336–348

Michie D, Spiegelhalter D, Taylor C (1994) Machine Learning, Neural and Statistical Classification. Ellis Horwood, University College, London

Milone DH, Rubio AJ (2003) Prosodic and accentual information for automatic speech recognition. IEEE transactions on speech and audio processing 11(4):321–333

Montefusco L, Puccio L (2014) Wavelets: theory, algorithms, and applications, vol 5. Academic Press

Pele O, Werman M (2009) Fast and robust earth mover's distances. In: Computer vision, 2009 IEEE 12th international conference on, IEEE, pp 460–467

Reed R (1993) Pruning algorithms-a survey. IEEE transactions on Neural Networks 4(5):740–747

Rolon R, Di Persia L, Rufiner HL, Spies R (2014) Most discriminative atom selection for apnea-hypopnea events detection. In: Anales del VI Congreso Latinoamericano de Ingeniera Biomdica (CLAIB 2014), pp 709–712

Sánchez-Gutiérrez ME, Albornoz EM, Martinez-Licona F, Rufiner HL, Goddard J (2014) Pattern Recognition, Springer International Publishing, chap Deep Learning for Emotional Speech Recognition, pp 311–320

Sarikaya R, Hinton G, Deoras A (2014) Application of deep belief networks for natural language understanding. Audio, Speech, and Language Processing, IEEE/ACM Transactions on 22(4):778–784

Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:14091556

Smolensky P (1986) Information processing in dynamical systems: Foundations of harmony theory. Tech. rep., DTIC Document

Stanley KO, Miikkulainen R (2002) Evolving neural networks through augmenting topologies. Evolutionary Computation 10(2):99–127, URL `http://nn.cs.utexas.edu/?stanley:ec02`

Stevens KN (2000) Acoustic phonetics, vol 30. MIT press

Sutskever I, Hinton GE (2007) Learning multilevel distributed representations for high-dimensional sequences. In: AISTATS, vol 2, pp 548–555

Suzuki K, Horiba I, Sugie N (2001) A simple neural network pruning algorithm with application to filter synthesis. Neural Processing Letters 13(1):43–53

Taigman Y, Yang M, Ranzato M, Wolf L (2014) Deepface: Closing the gap to human-level performance in face verification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1701–1708

Vignolo LD, Rufiner HL, Milone DH (2016) Multi-objective optimisation of wavelet features for phoneme recognition. IET Signal Processing 10(6):685–691

Wilcox RR (1995) Anova a paradigm for low power and misleading measures of effect size. Review of Educational Research 65(1):51–77

Zen H, Senior A (2014) Deep mixture density networks for acoustic modeling in statistical parametric speech synthesis. In: Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on, pp 3844–3848