A First Approach to Face Dimensionality Reduction Through Denoising Autoencoders

Francisco J. Pulgar(✉), Francisco Charte, Antonio J. Rivera, and María J. del Jesus

Andalusian Research Institute on Data Science and Computational Intelligence (DaSCI), Computer Science Department, University of Jaén, 23071 Jaén, Spain
{fpulgar,fcharte,arivera,mjjesus}@ujaen.es

Abstract. The problem of high dimensionality is a challenge when facing machine learning tasks. A high dimensional space has a negative effect on the predictive performance of many methods, specifically, classification algorithms. There are different proposals that arise to mitigate the effects of this phenomenon. In this sense, models based on deep learning have emerged.

In this work, denoising autoencoders (DAEs) are used to reduce dimensionality. To verify its performance, an experimentation is carried out where the improvement obtained with different types of classifiers is verified. The classification method used are: kNN, SVM, C4.5 and MLP. The test for kNN and SVM show a better predictive performance for all datasets. The executions for C4.5 and MLP reflect improvements only in some cases. The execution time is lower for all tests. In addition, a comparison between DAEs and PCA, a classical method of dimensionality reduction, is performed, obtaining better results with DAEs in most cases. The conclusions reached open up new lines of future work.

Keywords: Classification · Deep learning · Autoencoders
Denoising autoencoders · Dimensionality reduction
High dimensionality

1 Introduction

Since the 20th century, different machine learning techniques have been developed. In particular, the classification task is one of the most well-known problems within automatic learning. A classifier aims to correctly categorize new data patterns. For this, a model is usually built from correctly labelled data. Normally, each instance has a single label associated with it [13].

These methods work with datasets with very different characteristics. Therefore, algorithms are affected in several ways by these features. One of them is the high dimensionality of the data [7]. Throughout history, many proposals that mitigate the effects of this fact have emerged. In this context, new proposals have

H. Yin et al. (Eds.): IDEAL 2018, LNCS 11314, pp. 439–447, 2018.
https://doi.org/10.1007/978-3-030-03493-1_46

appeared due to the rise of Deep Learning (DL) models. This type of algorithms have obtained very good results in different fields of application, such as automatic speech processing or computer vision [6,11]. In particular, autoencoders (AEs) are deep networks that offer good results in the selection of features due to their architecture and operation [8,18].

There are different models of AEs suitable to perform the feature fusion task. Fundamentally, AEs reproduce the input of the network at the exit, through a series of hidden layers. In this way, a coding of the input is achieved in the intermediate layers. One of the types of AEs are denoising AEs (DAEs). The objective of them is learning to generate robust characteristics from the partially altered input data. In this way, the features originated by the AE are much more resistant to corrupted inputs [8,26].

The objective of this study is to verify the improvement of the predictive performance of several classifiers after applying a DAE to reduce the dimensionality of the data. To see the effects on classifiers of different types, the following algorithms are going to be used: kNN [10], SVM [17], C4.5 [24] and MLP [19]. The experimentation will consist of a first phase where the reduction of dimensionality will be applied on 5 datasets and a second phase where classification will be carried out using the previous algorithms. Next, classification results from the data with reduced dimensionality and the original data will be compared. Finally, a comparison between the use of DAEs for dimensionality reduction with respect to other classical method, such as PCA [20,22].

2 Background

Classification is a contextualized task within machine learning. Fundamentally, it is a phase of data mining whose purpose is to predict or categorize a new instance, based on a series of correctly labeled data. Normally, the process is done using supervised learning methods.

Different methodologies have been developed in order to face this work: Instance-based learning (IBL) that does not build a model, but uses the information provided by the training instances directly [1]; Artificial Neural Networks (ANNs) that are inspired by the way the human brain works. ANNs learn to detect relationships and patterns in the data through their own experience [25]; Support Vector Machines (SVMs) that are supervised learning models used for classification and regression. These algorithms map the different instances as points in space, so that the examples of different categories are separated from each other [17]; Decision tree learning using predictive models based on trees. The different features of the data are represented in the branches of the tree while the leaves contain the objective values [23].

The methodologies indicated above are some of the most used to face the task of classification. Therefore, this work includes an algorithm of each type to validate the reduction of dimensionality through DAEs. In particular, the algorithms used are: kNN [10], MLP [19], SVM [17] and C4.5 [24].

The different proposals used to perform classification must take into account the features of the data. Among these factors is the dimensionality of the data.

A dataset with high dimensionality negatively affects the predictive performance of a large part of the classifiers. This is due to the curse of dimensionality [4,5].

This factor affects many of the classification models and different proposals have emerged to mitigate its effects [3,14,28]. The first solutions were manual, where experts decided which variables were the best for the process. However, this task was soon automated and the first methods of feature selection emerged. Some traditional approaches that face this problem are: LDA [27] and PCA [22]. Recently, new models have appeared based on DL that deal with this fact. In particular, AEs offer good results due to their operation and architecture [8,18].

An AE is an ANN whose goal is to reproduce the input to the output through unsupervised learning. These models present an adequate symmetrical architecture to achieve this purpose. The operation consists of learning to represent the input data in the output layer, but fulfilling a number of requirements to avoid copying the information throughout the network. This type of algorithms have recently gained momentum due to the rise of DL models.

The basic architecture of an AE is a feed-forward neural network where the information always goes in the same direction. These models are formed by a sequence of layers: an input layer, an output layer and a series of hidden layers. The elements of each layer are connected to all the units of the previous layers. A restriction associated with the operation of the AEs is that the number of neurons of the input layer must be the same as that of the output layer. In this way, the AE can learn to represent the input to the output of the network.

As has been said, the original purpose of the AEs is to look for useful representations of the data. For this, the model learns non-linear ways to combine the data features. In this sense, there are different variations of the most basic AE model that allows discovering new characteristics of the data. Thus, the different variants generate models that allow certain restrictions to be satisfied. In this regard, a method is DAE. This type of algorithm introduces noise into the input before beginning the training process. In this way, the generated characteristics are more robust since they are able to tolerate the introduced noise and correctly reconstruct the input [8,26].

In this study, DAEs are used to address the problem of dimensionality reduction. In Sect. 3, four classification algorithms are used with original high-dimensional data and low-dimensional data after applying DAE to fusion features. In this way, the performance to select characteristics of DAEs is analysed.

3 Experimentation

Once presented the main concepts involved in this study, the experimentation that is carried out is outlined. The fundamental objective of this stage is to analyse the behaviour of different classifiers with data sets where feature fusion is applied through DAEs.

Therefore, the tests performed have two fundamental phases. On the one hand, the fusion of features from the different datasets is done using DAE. On the other hand, the data is classified using 4 different classifiers: C4.5, kNN,

MLP, SVM. For each dataset, two classifications are made: one with the original data set and another the data with low dimensionality.

3.1 Experimental Framework

The conducted experimentation aims to determine the performance of DAEs for the task of dimensionality reduction. Therefore, the use of different types of datasets is necessary. Their traits are in Table 1. The source is shown in the last column. A 2×5 folds cross validation scheme is applied in all executions.

Table 1. Characteristics of the datasets used in the experimentation.

Dataset	Number of Samples	Number of Features	Number of Classes	Type	Ref.
Image	2310	19	7	Real	[2]
isolet	7797	617	26	Real	[9]
madelon	2000	500	2	Real	[16]
mfeat	2000	649	10	Real	[2]
nomao	1970	118	2	Real	[2]
semeion	1593	256	10	Integer	[2]

The parametrization of the different classification algorithms is the one given by default in each method. In the case of kNN, the value of k will be 5, since it is a usual value in the literature [12,15,21]. In addition, to evaluate the quality of the different models it is necessary to compute several measures. In this case, Accuracy, F-Score and runtime will be used.

3.2 AE Architecture

In this section, DAE architecture used to perform the fusion of features is presented. As indicated above, these models have a symmetric architecture where the number of neurons of the input layer is the same as that of the output layer. In this study, the objective is to reduce the dimensionality of the input data, therefore, the intermediate layer (or layers) must have a smaller number of units than the input and output.

To perform the evaluation of the model, a 50% reduction in the number of original attributes is proposed. Therefore, the model will have the following layers:

- Input layer: This part will have as many units as features has the data set.
- Hidden layer: The number of elements will be half of units of the input layer.
- Output layer: Due to the operation of the AE, it must have the same number of neurons as the input layer.

The architecture indicated above has a single hidden layer. The main reason for this is to conduct a baseline study on the most basic model of DAE, with the aim of obtaining a first approximation to the reduction of dimensionality through this type of AEs. Likewise, the number of elements in the hidden layer is established at 50% of the number of characteristics, since it is considered a significant reduction for observing the performance of DAEs in this task.

3.3 Results Analysis

The experimentation carried out consists of several executions where data sets are classified by 4 different algorithms: C4.5, kNN, MLP and SVM. Each method classifies the original data set and the reduced data set by DAE.

As described in Sect. 3.1, there are 6 data sets with different traits. Thus, Tables 2, 3, 4 show the results for Accuracy, F-Score and Runtime metrics, respectively.

Table 2. Accuracy classification results for test data.

Dataset	C4.5		kNN		MLP		SVM	
	Base	DAE	Base	DAE	Base	DAE	Base	DAE
Image	0.840	**0.861**	0.928	**0.951**	0.548	**0.865**	0.860	**0.879**
isolet	0.741	**0.748**	0.872	**0.899**	**0.371**	0.359	0.935	**0.949**
madelon	0.521	**0.538**	0.531	**0.598**	0.501	**0.618**	0.578	**0.588**
mfeat	0.890	**0.896**	0.438	**0.975**	0.086	**0.860**	0.976	**0.982**
nomao	**0.877**	0.872	0.891	**0.902**	**0.901**	0.563	0.914	**0.919**
semeion	**0.612**	0.608	0.908	**0.917**	**0.781**	0.453	0.890	**0.950**

Table 3. F-Score classification results for test data.

Dataset	C4.5		kNN		MLP		SVM	
	Base	DAE	Base	DAE	Base	DAE	Base	DAE
Image	0.833	**0.864**	0.927	**0.951**	0.494	**0.870**	0.850	**0.879**
isolet	0.743	**0.749**	0.874	**0.900**	**0.350**	0.329	0.935	**0.950**
madelon	0.521	**0.566**	0.551	**0.590**	0.512	**0.583**	0.571	**0.579**
mfeat	0.890	**0.898**	0.447	**0.975**	0.016	**0.849**	0.976	**0.982**
nomao	**0.877**	0.872	0.892	**0.902**	**0.897**	0.280	0.914	**0.919**
semeion	**0.614**	0.608	0.910	**0.917**	**0.784**	0.413	0.891	**0.952**

Table 4. Time classification results (in seconds) for test data.

Dataset	C4.5		kNN		MLP		SVM	
	Base	DAE	Base	DAE	Base	DAE	Base	DAE
Image	30.002	**20.424**	0.311	**0.173**	1451.797	**931.370**	1.248	**0.945**
isolet	8645.028	**5903.823**	64.465	**33.781**	71800.729	**38842.212**	410.516	**201.704**
madelon	250.329	**112.155**	3.967	**1.716**	6518.699	**3220.424**	18.843	**7.592**
mfeat	282.250	**183.543**	4.885	**2.221**	8618.690	**4287.535**	12.995	**6.746**
nomao	27.973	**15.148**	0.384	**0.223**	1347.476	**750.514**	1.396	**0.762**
semeion	91.609	**50.778**	1.023	**0.436**	2402.432	**1363.049**	5.417	**2.305**

C4.5 Analysis: The results of the C4.5 algorithm shown in Tables 2, 3, 4 reflect that the reduction of dimensionality with DAE improves the predictive performance in certain datasets. In particular, the executions with the reduced datasets obtain better results in 4 out of 6 tests for the Accuracy and F-Score metrics, while in the 2 remaining datasets better results are obtained with the original dataset. In terms of execution time, the results with the reduced dataset are better in all cases. The reason is obvious, since the time will be less when classifying a set of data with fewer features.

kNN Analysis: The tests show better predictive performance for the kNN algorithm by reducing the number of input characteristics trough DAE, since all datasets are improved. On the one hand, the tests performed after performing the fusion of characteristics obtain better results in all cases for the Accuracy and F-Score metrics. On the other hand, the results in terms of time are similar to the algorithm C4.5, since the classification with fewer features is faster.

MLP Analysis: The executions carried out indicate that the reduction of dimensionality does not have clear effects for MLP, since there are datasets with better performance and others with worse. In detail, there are 3 datasets whose results improve considering Accuracy and F-Score, while the performance decreases in the other 3. With respect to time, the results are similar to the two algorithms previously reported. Therefore, the performance applying dimensionality reduction is comparable to that obtained by not doing it. However, the time is less when the input space is less.

SVM Analysis: In the case of the SVM method, the analysis is very similar to the kNN algorithm, since the improvement of the predictive performance is clear, obtaining superior results for all the datasets. Tables show: on the one hand, the Accuracy and F-Score metrics show better results for the 6 datasets used; on the other hand, the execution time is less when reducing the dimensionality, in the same way as in the previous algorithms.

Results Discussion: The data presented in this study show improvements in predictive performance in the four families of algorithms analyzed. The use of DAE to reduce the dimensionality of the input space allows to achieve better results. The reason is different for each of the methodologies.

In particular, the reduction of dimensionality for kNN implies that the distances between individuals are more significant, therefore, the performance is improved. As for SVM, a smaller input space implies that the spatial representation is smaller and groupings of elements can be performed more precisely. In these two cases the improvements are significant, since these are observed for all datasets.

For the C4.5 and MLP methods the improvements are less significant, there are cases in which they are improved and others in which they are not. This gives a first sample of which the characteristics of this type of algorithms do that the improvements obtained when reducing the dimensionality by means of DAE are not as significant as for kNN and SVM. However, the execution time improves for all the methods analysed.

3.4 DAE vs PCA

In this Subsection, the objective is to assess the competitiveness of DAEs against traditional dimensionality reduction algorithm, such a PCA [20,22]. A 50% reduction in the number of original attributes is performed with these methods, since it is the same one that has been done with DAEs.

Table 5. Accuracy and F-Score classification results of DAE and PCA for test data

Dataset	Accuracy								F-Score							
	C4.5		kNN		MLP		SVM		C4.5		kNN		MLP		SVM	
	DAE	PCA	DAE	PCA	DAE	PCA	DAE	PCA	DAE	PCA	DAE	PCA	DAE	PCA	DAE	PCA
Image	**0.861**	0.835	**0.951**	0.862	**0.865**	0.801	**0.879**	0.843	**0.864**	0.836	**0.951**	0.862	**0.870**	0.805	**0.879**	0.844
isolet	**0.748**	0.701	**0.899**	0.589	**0.359**	0.333	**0.949**	0.943	**0.749**	0.703	**0.900**	0.638	**0.329**	0.301	**0.950**	0.944
madelon	**0.538**	0.513	**0.598**	0.511	**0.618**	0.590	**0.588**	0.562	**0.536**	0.515	**0.590**	0.503	**0.583**	0.582	**0.585**	0.569
mfeat	**0.896**	0.853	**0.975**	0.703	**0.860**	0.821	**0.982**	0.951	**0.898**	0.853	**0.975**	0.771	**0.849**	0.823	**0.982**	0.955
nomao	**0.872**	0.853	**0.902**	0.869	0.563	**0.934**	**0.919**	0.902	**0.872**	0.851	**0.902**	0.883	0.280	**0.853**	**0.919**	0.898
semeion	**0.608**	0.594	**0.917**	0.706	**0.453**	0.403	**0.950**	0.936	**0.608**	0.596	**0.917**	0.718	**0.453**	0.406	**0.952**	0.937

Table 5 shows the results obtained with different classification algorithms after applying dimensionality reduction with DAE and PCA. The best predictive performance is obtained with the dataset generated by DAEs for most of the dataset and classification algorithms.

4 Concluding Remarks

One of the problems that affect many classification algorithms is the high dimensionality of the data. This feature is present in many real datasets. Therefore, it

is important to propose solutions that mitigate the negative effects of this factor. In this work, DAE are used to reduce the dimensionality and a series of tests are performed to see their effects in classification algorithms corresponding to different methodologies.

On the one hand, the experimentation carried out has shown that the predictive performance for the kNN and SVM algorithms is significantly improved. On the other hand, the results for the C4.5 and MLP algorithms are improved for some of the datasets used. This indicates that the use of DAE to reduce the dimensionality offers better performance for the IBL and SVM algorithms. In addiction, the comparison between DAEs and PCA shows that DAEs offer better results in most of the cases analysed.

The results derived from this empirical verification open new lines of future work. The experimentation can be extended with new datasets. Likewise, other types of AEs can be used to face the task of dimensionality reduction. Another line of work is the creation of models that combine dimensionality reduction techniques based on AEs and traditional classification algorithms.

In conclusion, this study allows us to take the first step to address the problem of dimensionality reduction using DL-based models and generate hybrid models that improve predictive performance when classifying high-dimensional data.

Acknowledgment. The work of F. Pulgar was supported by the Spanish Ministry of Education under the FPU National Program (Ref. FPU16/00324). This work was partially supported by the Spanish Ministry of Science and Technology under project TIN2015-68454-R.

References

1. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithms. Mach. Learn. **6**(1), 37–66 (1991)
2. Bache, K., Lichman, M.: UCI Machine Learning Repository (2013)
3. Batista, G.E., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explor. Newsl. **6**(1), 20–29 (2004)
4. Bellman, R.: Dynamic Programming. Princeton University Press, Princeton (1957)
5. Bellman, R.: Adaptive Control Processes: A Guided Tour. Princeton University Press, Princeton (1961)
6. Bengio, Y.: Deep learning of representations: looking forward. In: Dediu, A.-H., Martín-Vide, C., Mitkov, R., Truthe, B. (eds.) SLSP 2013. LNCS (LNAI), vol. 7978, pp. 1–37. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39593-2_1
7. Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U.: When is "Nearest Neighbor" meaningful? In: Beeri, C., Buneman, P. (eds.) ICDT 1999. LNCS, vol. 1540, pp. 217–235. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-49257-7_15
8. Charte, D., Charte, F., García, S., del Jesus, M.J., Herrera, F.: A practical tutorial on autoencoders for nonlinear feature fusion: taxonomy, models, software and guidelines. Inf. Fusion **44**, 78–96 (2018)
9. Cole, R., Fanty, M.: Spoken letter recognition. In: Proceedings of the Workshop on Speech and Natural Language, pp. 385–390 (1990)

10. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theory **13**(1), 21–27 (1967)
11. Deng, L.: Deep learning: methods and applications. Found. Trends Signal Process. **7**(3–4), 197–387 (2014)
12. Derrac, J., Chiclana, F., García, S., Herrera, F.: Evolutionary fuzzy k-nearest neighbors algorithm using interval-valued fuzzy sets. Inf. Sci. **329**, 144–163 (2016)
13. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. Wiley, New York (1973)
14. Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., Herrera, F.: A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.) **42**(4), 463–484 (2012)
15. Ghosh, A.K.: On optimum choice of k in nearest neighbor classification. Comput. Stat. Data Anal. **50**(11), 3113–3123 (2006)
16. Guyon, I., Gunn, S., Ben-Hur, A., Dror, G.: Result analysis of the NIPS 2003 feature selection challenge. In: Proceedings of Neural Information Processing Systems, vol. 4, pp. 545–552 (2004)
17. Hearst, M.A., Dumais, S.T., Osuna, E., Platt, J., Scholkopf, B.: Support vector machines. IEEE Intell. Syst. Their Appl. **13**(4), 18–28 (1998)
18. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
19. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Netw. **2**(5), 359–366 (1989)
20. Hotelling, H.: Analysis of a complex of statistical variables into principal components. J. Educ. Psychol. **24**(6), 417–441 (1933)
21. Keller, J.M., Gray, M.R., Givens, J.A.: A fuzzy k-nearest neighbor algorithm. IEEE Trans. Syst. Man Cybern. SMC **15**(4), 580–585 (1985)
22. Pearson, K.: LIII. On lines and planes of closest fit to systems of points in space. Lond. Edinb. Dublin Philos. Mag. J. Sci. **2**(11), 559–572 (1901)
23. Quinlan, J.R.: Induction of decision trees. Mach. Learn. **1**(1), 81–106 (1986)
24. Quinlan, J.R.: C4. 5: Programs for Machine Learning. Elsevier, Amsterdam (2014)
25. Schalkoff, R.J.: Artificial Neural Networks, vol. 1. McGraw-Hill, New York (1997)
26. Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.A.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th International Conference on Machine Learning, pp. 1096–1103. ACM (2008)
27. Yu, H., Yang, J.: A direct LDA algorithm for high-dimensional data-with application to face recognition. Pattern Recognit. **34**(10), 2067–2070 (2001)
28. Zadrozny, B., Elkan, C.: Learning and making decisions when costs and probabilities are both unknown. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 204–213. ACM (2001)