



Performance Analysis of Deep Neural Networks for Classification of Gene-Expression Microarrays

A. Reyes-Nava¹(✉), J. S. Sánchez², R. Alejo³, A. A. Flores-Fuentes¹,
and E. Rendón-Lara³

¹ Universidad Autónoma del Estado de México,
Toluca-Atlacomulco KM. 60, 50000 Atlacomulco, Mexico
adriananava0@gmail.com

² Department Computer Languages and Systems,
Institute of New Imaging Technologies, Universitat Jaume I,
Av. Sos Baynat s/n, 12071 Castelló de la Plana, Spain

³ Division of Studies of Postgrade and Research, Instituto Tecnológico de Toluca,
Tecnológico Nacional de México, Av. Tecnológico s/n, Col. Agrícola Bellavista,
52140 Metepec, Mexico

Abstract. In recent years, researchers have increased their interest in deep learning for data mining and pattern recognition applications. This is mainly due to its high processing capability and good performance in feature selection, prediction and classification tasks. In general, deep learning algorithms have demonstrated their great potential in handling large scale data sets in image recognition and natural language processing applications, which are characterized by a very large number of samples coupled with a high dimensionality. In this work, we aim at analyzing the performance of deep neural networks for classification of gene-expression microarrays, in which the number of genes is of the order of thousands while the number of samples is typically less than a hundred. The experimental results show that in some of these challenging situations, the use of deep neural networks and traditional machine learning algorithms does not always lead to high performance results. This finding suggests that deep learning needs a very large number of both samples and features to achieve high performance.

Keywords: Deep learning · Gene-expression microarray
Curse of dimensionality

1 Introduction

Traditional neural networks generally consist of three layers: the first indicates the data entries, the second is the hidden layer, and the third corresponds to the output layer. When the architecture of the neural network has more than three layers, it is commonly referred to as deep neural network. The most representative example of this architecture is the multi-layer perceptron with many hidden

layers, where each layer trains a different set of features based on the output of the previous layer [1,2].

Deep learning algorithms have usually been applied to problems whose complexity is high due to the amount of data stored, that is, there is a large number of features and samples. They have been used extensively in various scientific areas to tackle very different problems [3,4]. The main advantages of this type of neural networks are three-fold: high performance, robustness to overfitting, and high processing capability.

In this work, we analyze the performance of several deep neural networks and other machine learning models in the classification of gene-expression microarrays, which are characterized by a very large number of features coupled with a small number of samples. This could represent a challenging situation because typical applications with deep neural networks refer to problems in which both the dimensionality and the number of samples are high. Therefore, the purpose of this paper is to investigate the efficiency of deep learning algorithms when applied to data sets with those especial characteristics, thus checking whether or not they perform as good as in those applications where they have demonstrated to behave significantly better than state-of-the-art algorithms.

2 Related Works

Nowadays, the use of deep learning to solve a variety of real-life problems has attracted the interest of many researchers because these algorithms allow to obtain generally better results than traditional machine learning methods [5]. As already mentioned, deep neural networks consist of a very large number of hidden layers, which lead to high computational cost when processing data of large size and high dimension.

The areas in which deep neural networks have been most applied are image recognition and natural language processing. For instance, Cho et al. [6] employed a recurrent neural network (RNN) encoder-decoder to detect semantic and syntactic representations of language when translating from English into French, thus obtaining a better translation in the analyzed sentences. The analysis of information to recognize translations, dialogues, text summaries and text produced in social networks was studied using techniques such as the convolutional neural network (CNN) and the RNN [7]. Nene [8] reviewed the developments and applications of deep neural networks in natural language processing.

In image processing, the use of deep neural networks makes tasks faster and allows to obtain better results. Dong et al. [9] proposed a CNN approach to learn an end-to-end mapping between low- and high-resolution images, performing better than the state-of-the-art methods. On the other hand, Wen et al. [10] combined a new loss function with the softmax loss to jointly supervise the learning of a CNN for robust face recognition. Gatys et al. [11] showed how the generic feature representations learned by high-performing CNNs can be used to independently process and manipulate the content and the style of natural images. A deep neural network based on bag-of-words for image retrieval tasks

was proposed by Bai et al. [12]. A novel maximum margin multimodal deep neural network was introduced to take advantage of the multiple local descriptors of an image [13].

Apart from image and natural language processing, deep neural networks have also been applied to some other practical domains. For instance, Langkvist et al. [14] reviewed the use of deep learning for time-series modeling and prediction. Hinton et al. [15] presented an overview of the application of deep neural networks to acoustic modeling in speech recognition. Noda et al. [16] utilized a deep denoising autoencoder for acquiring noise-robust audio features and a CNN to extract visual features from raw mouth area images. Wang and Shang [17] employed deep belief networks to extract features from raw physiological data. Kraus and Feuerriegel [18] studied the use of deep neural networks for predicting stock market movements subsequent to the disclosure of financial materials. Heaton et al. [19] introduced an autoencoder-based hierarchical decision model for problems in financial prediction and classification.

The biomedical domain is another scientific area where the use of deep learning is gaining much attention in last years. For instance, Maqlin et al. [20] proposed the application of the deep belief neural network to determine the nuclear pleomorphism score of breast cancer tissues. Danaee [21] used a stacked denoising autoencoder for the identification of genes critical for the diagnosis of breast cancer. Abdel-Zaher and Eldeib [22] presented an automatic diagnosis system for detecting breast cancer based on deep belief network unsupervised pre-training phase followed by a supervised back-propagation neural network phase. Hanson et al. [23] implemented deep bidirectional long short-term memory recurrent neural networks for protein intrinsic disorder prediction. Salaken et al. [24] designed an autoencoder for the classification of pathological types of lung cancers. Geman et al. [25] proposed the application of deep neural networks for the analysis of large amounts of data produced by the human microbiome. Chen et al. [26] developed an incremental RNN to discriminate between benign and malignant breast cancers.

3 Deep Learning Methods

In this section, the deep neural networks that will be further used in the experiments are briefly described.

3.1 Multilayer Perceptron

The multilayer perceptron (MLP) constitutes the most conventional neural network architectures. These are commonly based on three layers: input, output, and one hidden layer. Nevertheless, the MLPs can also be translated into deep neural networks by incorporating more than two hidden layers in its architecture; this allows to reduce the number of nodes per layer and use less parameters, but in turn this leads to a more complex optimization problem [1, 25].

In deep MLP networks, each layer trains with a different set of features, which are based on the output of the previous layer. It is possible to select features in a first layer and the outputs of this will be used in the training of the next layer.

3.2 Recurrent Neural Network

Recurrent neural networks are a type of network for sequential data processing, allowing to scale very long and variable length sequences [1]. In this type of network, a neuron is connected to the neurons of the next layer, to those of the previous layer and to it by means of the weights, values that change in each time step.

The recurrent neural networks can adopt different forms depending on the particular design:

- Networks that produce an output in each time step with recurring connections between the hidden units.
- Networks that produce an output and have recurring connections only from the output to the hidden unit of the next step.
- Networks with recurring connections between hidden units that read the complete sequence of data and produce a simple output.

A design that improves the use of recurrent neural networks is based on LSTM units, thus giving solution to the problem of the vanishing gradient that occurs in a conventional recurrent network. This means that the gradient changes the weights with respect to the change of the error. If the gradient is not known, then it is not possible to adjust the weights in the direction of decreasing the error, which causes the network to stop learning; this happens because the processed data go through many stages of multiplication.

Figure 1 shows the structure of the recurrent neural network working with LSTM cells, where x are the inputs, y are the outputs, and s consists of the values that the cells take. Unlike the bidirectional recurrent neural network, which works with both forward and backward propagation (see Fig. 2), the recurrent neural network works only with forward propagation.

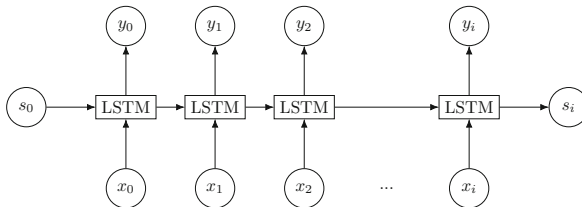


Fig. 1. Recurrent neural network with LSTM

An LSTM contains information in a closed cell independent of the flow of the neural network. This information can be stored, written or read, which helps

to preserve the error that can be propagated back to the passage of the layers. If the error remains constant, this allows the network to continue learning over time. The cell of the LSTM decides when to store, write or erase by means of gates that open and close analogically, which act by signals; this allows to adjust the weights by decreasing the gradient or to propagate the error again [27].

The basic idea of the LSTM is very simple: some of the units are called constant error carousels, which are used as an activation function (an identity function) and have a connection to itself with a fixed weight of 1.0 [2].

3.3 Bidirectional Recurrent Neural Network

Bidirectional recurrent neural networks are a type of network where a recurrent network is used with forward propagation and another with backward propagation. This type of network is used for input data sequences where it is known its beginning and end (e.g., spoken sentences and protein structures). To know the past and future of each sequence element, a recurrent network processes the sequence of data from the beginning to the end, and another processes backing up from the end to the beginning [2].

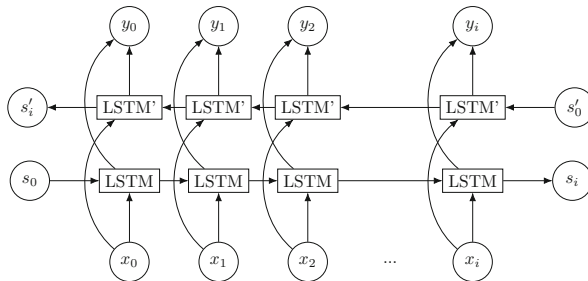


Fig. 2. Bidirectional recurrent neural network with LSTM

3.4 Autoencoder

An autoencoder is a type of neural network that copies the input to the output. It consists of an encoder that does the training task and a decoder that obtains the same inputs as outputs. In general, it can be used for feature selection, dimensionality reduction and classification [1].

There are different types of autoencoders, which can make different tasks depending on the structure of them:

- Incomplete autoencoder: wait for the results of the training, from where it takes useful features that result from restricting h less to x , where h are the nodes of the encoder and x are the inputs.
- Regularized autoencoder: this type uses a loss function that allows to have other properties in addition to copying the input to the output.

- Dispersed autoencoder: a training dispersion penalty is applied; it is used to learn functions used in classification tasks.
- Autoencoder for elimination of noise: it obtains useful characteristics minimizing the reconstruction error, this receives a damaged data set and is trained to predict the original data set not damaged as an output.

4 Experimental Set-Up

The purpose of the experiments in this work is to compare some state-of-the-art machine learning algorithms with deep learning for the classification of gene-expression microarrays. To this end, a collection of publicly available microarray cancer data sets taken from the Kent Ridge Biomedical Data Set Repository (<http://datam.i2r.a-star.edu.sg/datasets/krbd>) were used (see Table 1).

Table 1. Description of the data sets. The imbalance ratio (IR), which corresponds to the ratio of the majority class size to the minority class size is reported in the last column

Database	Features	Samples	Class 1	Class 2	IR
Lung-Michigan	7129	96	86 Tumor	Normal 10	8.60
Lung-Ontario	12533	182	150 ADCA	Mesothelioma 31	4.84
Ovarian	15154	253	162 Cancer	Normal 91	1.78
DLBCL	4026	47	24 Germinal	Activated 23	1.04
Colon	2000	62	22 Positive	Negative 40	1.82
Prostate	12600	136	77 Tumor	Normal 59	1.31
CNS	7129	60	21 Class1	Class0 39	1.86
Breast	24481	97	46 Class1	Class0 51	1.11

For the experimental design, we adopted the holdout method 10 times was adopted, with 70% of the samples for training and 30% for testing. The traditional machine learning methods used in these experiments were the radial basis function (RBF) neural network, the random forest (RNDF), the nearest neighbor (1NN) rule, the C4.5 decision tree, and a support vector machine (SVM) using a linear kernel function with the soft-margin constant $C = 1.0$ and a tolerance of 0.001. The deep learning models analyzed in this work were recurrent neural network (RNN), bidirectional recurrent neural network (BRNN) and autoencoder (AE). In addition, we included two versions of MLP: one with two hidden layers (MLP2) and one with three hidden layers (MLP3). The main parameters of the deep neural networks are listed in Table 2.

The state-of-the-art machine learning methods were applied using the default parameters as defined in the WEKA data mining toolkit [28].

Table 2. Parameters of the deep neural networks

Method	Parameters
MLP2	Sigmoidal activation function; learning rate = 0.3; Adam optimizer; hidden layers = 2
MLP3	Rectified linear unit (ReLU) activation function; learning rate = 0.1; Adagrad optimizer; hidden layers = 3
RNN	Hyperbolic tangent (tanh) activation function; learning rate = 0.1; gradient descent optimizer; hidden layers = 1
BRNN	tanh activation function; learning rate = 0.1; gradient descent optimizer; hidden layers = 1
AE	Autoencoders = 2; hidden size = 10; weight regularization = 0.1; sparsity proportion = 0.05; linear decoder transfer function; cross entropy loss function; softmax activation function; hidden layers = 1

5 Results

Table 3 reports the accuracy results and standard deviations for each classifier and each database. In addition, the Friedman’s average rankings are also included. Bold values indicate the best model for each data set.

From the Friedman’s rankings, one can see that the best algorithms were MLP2 and AE followed by the classical random forest, whereas the two versions of recurrent neural networks (RNN and BRNN) performed the worst in average. When focusing on the accuracy results on each particular database, it was found that the autoencoder was the best method in four out of the eight problems (Lung-Michigan, Lung-Ontario, Ovarian, and Colon), and the MLP2 model was the best performing algorithm in two cases (Prostate and Breast).

It is worth noting that Lung-Michigan, Lung-Ontario and Ovarian, which correspond to three of the databases where the AE method performed the best, are the cases with the highest imbalance ratio as reported in Table 1. On the other hand, the only problem where a state-of-the-art machine learning method achieved the best accuracy was CNS, which is one of the databases with the smallest number of samples and features.

To check the results of the classifiers and to determine whether or not there exist significant differences between each pair of algorithms, the Wilcoxon’s paired signed-rank test at a significance level of $\alpha = 0.05$ was employed. This statistic ranks the differences in performance of two algorithms for each data set, ignoring the signs, and compares the ranks for the positive and the negative differences. In Table 4, one can see the results of this test where the symbol “•” represents that the classifier in the column was significantly better than the

classifier in the row, whereas the symbol “o” indicates that the classifier in the row performed significantly better than the classifier in the column.

6 Conclusions

In this paper, we have carried out an empirical comparison between several deep neural networks and some traditional machine learning methods for the classification of gene-expression microarray data, which characterize by a large number of samples and a very small number of features. While deep learning has demonstrated to be a powerful tool in applications with a huge amount of both samples and features, there is no study in problems that suffer from the “curse of dimensionality” phenomenon, such as is the case of gene-expression microarray analysis.

The experimental results have shown that the autoencoder and an MLP with two hidden layers were the best performing deep neural networks. On the other hand, it has also observed that there is no single method with the highest accuracy on all databases, and even the SVM (a traditional machine learning algorithm) was superior to the remaining models on one problem. Another interesting finding is that the recurrent neural networks were the worst techniques in average.

Acknowledgment. This work has partially been supported by the Spanish Ministry of Education and Science and the Generalitat Valenciana under grants TIN2009–14205 and PROMETEO/2010/028, respectively.

References

1. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
2. Schmidhuber, J.: Deep learning in neural networks: an overview. *Neural Netw.* **61**, 85–117 (2015)
3. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**, 436–444 (2015)
4. Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., Alsaadi, F.E.: A survey of deep neural network architectures and their applications. *Neurocomputing* **234**, 11–26 (2017)
5. Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., Lew, M.S.: Deep learning for visual understanding: a review. *Neurocomputing* **187**, 27–48 (2016)
6. Cho, K., Merriënboer, B.V., Gulcehre, C., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. CoRR abs/1406.1078 (2014)
7. Young, T., Hazarika, D., Poria, S., Cambria, E.: Recent trends in deep learning based natural language processing. CoRR abs/1708.02709 (2017)
8. Nene, S.: Deep learning for natural language processing. *Int. Res. J. Eng. Technol.* **4**, 930–933 (2017)
9. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8692, pp. 184–199. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10593-2_13

10. Wen, Y., Zhang, K., Li, Z., Qiao, Y.: A discriminative feature learning approach for deep face recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9911, pp. 499–515. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46478-7_31
11. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, pp. 2414–2423 (2016)
12. Bai, Y., Yu, W., Xiao, T., Xu, C., Yang, K., Ma, W.Y., Zhao, T.: Bag-of-words based deep neural network for image retrieval. In: 22nd ACM International Conference on Multimedia, Orlando, FL, pp. 229–232 (2014)
13. Ren, Z., Deng, Y., Dai, Q.: Local visual feature fusion via maximum margin multimodal deep neural network. *Neurocomputing* **175**, 427–432 (2016)
14. Langkvist, M., Karlsson, L., Loutfi, A.: A review of unsupervised feature learning and deep learning for time-series modeling. *Pattern Recogn. Lett.* **42**, 11–24 (2014)
15. Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., Kingsbury, B.: Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups. *IEEE Sig. Process. Mag.* **29**(6), 82–97 (2012)
16. Noda, K., Yamaguchi, Y., Nakadai, K., Okuno, H.G., Ogata, T.: Audio-visual speech recognition using deep learning. *Appl. Intell.* **42**(4), 722–737 (2015)
17. Wang, D., Shang, Y.: Modeling physiological data with deep belief networks. *Int. J. Inf. Educ. Technol.* **3**(5), 505–511 (2013)
18. Kraus, M., Feuerriegel, S.: Decision support from financial disclosures with deep neural networks and transfer learning. *Decis. Support Syst.* **104**, 38–48 (2017)
19. Heaton, J.B., Polson, N.G., Witte, J.H.: Deep learning for finance: deep portfolios. *Appl. Stochast. Models Bus. Ind.* **33**(1), 3–12 (2017)
20. Maqlin, P., Thamburaj, R., Mammen, J.J., Manipadam, M.T.: Automated nuclear pleomorphism scoring in breast cancer histopathology images using deep neural networks. In: Prasath, R., Vuppala, A.K., Kathirvalavakumar, T. (eds.) MIKE 2015. LNCS (LNAI), vol. 9468, pp. 269–276. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26832-3_26
21. Danaee, P., Reza, G., Hendrix, D.A.: A deep learning approach for cancer detection and relevant gene identification. In: Pacific Symposium on Biocomputing, Honolulu, HI, pp. 219–229 (2016)
22. Abdel-Zaher, A.M., Eldeib, A.M.: Breast cancer classification using deep belief networks. *Expert Syst. Appl.* **46**, 139–144 (2016)
23. Hanson, J., Yang, Y., Paliwal, K., Zhou, Y.: Improving protein disorder prediction by deep bidirectional long short-term memory recurrent neural networks. *Bioinformatics* **33**, 685–692 (2016)
24. Salaken, S.M., Khosravi, A., Khatami, A., Nahavandi, S., Hosen, M.A.: Lung cancer classification using deep learned features on low population dataset. In: IEEE 30th Canadian Conference on Electrical and Computer Engineering, Windsor, Canada, pp. 1–5 (2017)
25. Geman, O., Chiuchisan, I., Covasa, M., Doloc, C., Milici, M.-R., Milici, L.-D.: Deep learning tools for human microbiome big data. In: Balas, V.E., Jain, L.C., Balas, M.M. (eds.) SOFA 2016. AISC, vol. 633, pp. 265–275. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-62521-8_21
26. Chen, D., Qian, G., Shi, C., Pan, Q.: Breast cancer malignancy prediction using incremental combination of multiple recurrent neural networks. In: Liu, D., Xie, S., Li, Y., Zhao, D., El-Alfy, E.S. (eds.) ICONIP 2017. LNCS, vol. 10635, pp. 43–52. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70096-0_5

27. Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R., Schmidhuber, J.: LSTM: a search space odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* **28**(10), 2222–2232 (2017)
28. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. *ACM SIGKDD Explor. Newsl.* **11**(1), 10–18 (2009)