A deep learning approach to sign language recognition using stacked sparse autoencoders

Pablo Rivas[*], **Deep Dand**
Dept. of Computer Science
Marist College
Poughkeepsie, NY 12601

Ezequiel Rivas, Omar Velarde, Samuel Gonzalez
Division of Graduate Studies and Research
Nogales Institute of Technology
Nogales, Sonora, Mexico

1 Introduction

We investigated the problem of recognition of signs over the American sign language (ASL). The proposed approach uses depth images of subjects making different signs, building upon the work of B. Kang, et.al. in [1]. Typical approaches addressing similar problems involve the usage of hidden Markov models [2], and a combination of them with other discriminative functions for feature extraction in multi-stage architectures [3]. Other major alternatives included the exploration of neural strategies combined with fuzzy systems [4]. For a more detailed review of alternatives for generic hand gesture recognition one can turn to the work in [5]. Deep learning, on the other hand, has gained attention in the machine learning and the image processing for pattern recognition communities [6, 7], motivating us to similarly explore this alternative. Recently, the authors in [1] have explored a deep learning approach based on convolutional neural networks (CNNs) achieving outstanding results in solving the problem that we address here. Nonetheless, the training of a CNN and its deployment may be computationally expensive, inconsistent, and it may need a great deal of experimentation in order to find successful architectures [8]; furthermore, other simpler and less costly deep learning alternatives tend to be overlooked [9]. Our research aims to show that a simpler deep learning approach based on stacked autoencoders in a dense neural network architecture is capable of solving the same problem with comparable results. We claim that this simple alternative approach also achieves great performance and is naturally simpler [10].

2 Methodology and Results

We stack autoencoders and combine them with a feed-forward neural network in a five-layer architecture. The first two layers are a set of unsupervised autoencoders that minimize the loss function $L = \frac{1}{N} \|\mathbf{x}_n - \hat{\mathbf{x}}_n\|_2^2 + \theta_w \frac{1}{2} \sum_{l=1}^{L} \left\|\mathbf{w}^l\right\|_2^2 + \theta_s \sum_{m=1}^{M} KL\left(\theta_\alpha \,\|\bar{\alpha}_m\right)$ that minimizes the mean squared error, promotes sparsity of the weights, and also minimizes the Kullback-Leibler divergence. The first layer, i.e., an encoding layer, receives as input N images of 256×256 as row vectors, each denoted as $\mathbf{x}_n \in \mathbb{R}^{65536}$, where $n \in \{1, 2, \ldots, N\}$. The training phase encodes the attributes using 100 neural units to produce $\hat{\mathbf{x}}_n \in \mathbb{R}^{100}$, and decodes back to the feature space using, intuitively, 65536 neural units; all neural units use logistic activation functions. Similarly, the third and fourth layers are an encoder and decoder, respectively. The encoder in the third layer receives as input an encoded version of the input coming from the first layer, denoted as $\hat{\mathbf{x}}_n$, and encodes using 50 neural units producing a modified version of the feature vector denoted as $\tilde{\mathbf{x}}_i \in \mathbb{R}^{50}$. The decoder in the fourth layer decodes using 100 neural units. In the last layer of the model we use a network of 31 neural units with softmax activation functions. Each neuron is stimulated $\tilde{\mathbf{x}}_n$ and is trained to predict the probability of the n-th sample belonging to a specific class $C \in \{1, 2, \ldots, 31\}$. Once the process of training the autoencoders and the softmax layer, the network undergoes a last refined training phase. In this last process, only the first, third, and fifth layers are fully connected and

[*]Pablo.Rivas@Marist.edu. Partially supported by New York State Cloud Computing and Analytics Ctr

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

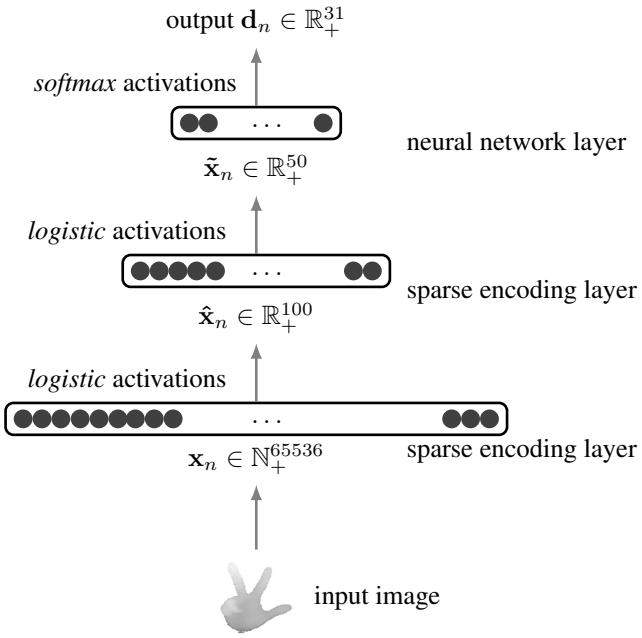

Figure 1: The working architecture when testing the system.

Table 1: Some experiments reported one subject left out of the training set and used for testing purposes, we refer here to that as leave-one-out (loo).

Ref.	Approach	Year	Class Type	C. Size	Input Type	ACC
[11]	FFNN	2004	Alphabets	24	Color Img	0.999
[12]	CNN	2011	Gestures	6	Color Img	0.9677
[13]	ANMM	2011	Gestures	6	Depth Img	0.9907
[14]	Gabor + RF	2011	Alphabets	24	Depth Img	0.69
[14]	Gabor + RF	2011	Alphabets	24	Color + Depth	0.75
[14]	Gabor + RF	2011	Alphabets	24	Color + Depth	0.49 (loo)
[15]	3D + MLRF	2013	Alphabets	24	Depth Img	0.87
[15]	3D + MLRF	2013	Alphabets	24	Depth Img	0.57 (loo)
[16]	Joint Info + RF	2015	Alphabets	24	Depth Img	0.90
[16]	Joint Info + RF	2015	Alphabets	24	Depth Img	0.70 (loo)
[1]	Deep CNN	2015	Alp.+Dig.	31	Depth Img	0.9999
[1]	Deep CNN	2015	Alp.+Dig.	31	Depth Img	0.855 (loo)
ours	Deep AE	2017	Alp.+Dig.	31	Depth Img	0.9889
ours	Deep AE	2018	Alp.+Dig.	31	Depth Img	0.8549 (loo)

trained simulating a feed-forward neural network, as shown in Figure 1. The initial weights are those obtained during the encoding-decoding learning phase and fine tuned using SCG descent to minimize the cross entropy.

The overall cross-validated accuracy is 98.9%. Table 1 shows the state of the art on methodologies that take on the general task of classifying hand gestures using different approaches. Our research indicates that deep autoencoders have the capability of matching the performance of a convolutional approach. Our main point is that a convolutional approach, while is adequate and performs well, is an *expensive* measure to a problem that may have a simpler deep learning solution, such as an autoencoder. By expensive we mean the amount of computations required to produce a solution using a convolutional neural network. It is known that CNN-based architectures suffer from having a massive amount of parameters to calculate during training [17], and often one sacrifices accuracy to gain efficiency, by using pooling, for example [18]. However, autoencoders offer a simple solution to the problem, as we have showed.

References

[1] Byeongkeun Kang, Subarna Tripathi, and Truong Q Nguyen. Real-time sign language fingerspelling recognition using convolutional neural networks from depth map. In *Pattern Recognition (ACPR), 2015 3rd IAPR Asian Conference on*, pages 136–140. IEEE, 2015.

[2] Thad Starner and Alex Pentland. Real-time american sign language recognition from video using hidden markov models. In *Motion-Based Recognition*, pages 227–243. Springer, 1997.

[3] Jeroen F Lichtenauer, Emile A Hendriks, and Marcel JT Reinders. Sign language recognition by combining statistical dtw and independent classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):2040–2046, 2008.

[4] Omar Al-Jarrah and Alaa Halawani. Recognition of gestures in arabic sign language using neuro-fuzzy systems. *Artificial Intelligence*, 133(1-2):117–138, 2001.

[5] GRS Murthy and RS Jadon. A review of vision based hand gestures recognition. *International Journal of Information Technology and Knowledge Management*, 2(2):405–410, 2009.

[6] Decebal Constantin Mocanu, Elena Mocanu, Phuong H Nguyen, Madeleine Gibescu, and Antonio Liotta. A topological insight into restricted boltzmann machines. *Machine Learning*, 104(2-3):243–270, 2016.

[7] Michele Donini and Fabio Aiolli. Learning deep kernels in the space of dot product polynomials. *Machine Learning*, pages 1–25, 2016.

[8] Michael C Burl and Philipp G Wetzler. Onboard object recognition for planetary exploration. *Machine learning*, 84(3):341–367, 2011.

[9] Elad Michael. Deep, deep trouble: Deep learning's impact on image processing, mathematics, and humanity. *SIAM News*, 50(04), 2017.

[10] Nathalie Japkowicz. Supervised versus unsupervised binary-learning by feedforward neural networks. *Machine Learning*, 42(1):97–122, 2001.

[11] Jason Isaacs and S Foo. Hand pose estimation for american sign language recognition. In *System Theory, 2004. Proceedings of the Thirty-Sixth Southeastern Symposium on*, pages 132–136. IEEE, 2004.

[12] Jawad Nagi, Frederick Ducatelle, Gianni A Di Caro, Dan Cireşan, Ueli Meier, Alessandro Giusti, Farrukh Nagi, Jürgen Schmidhuber, and Luca Maria Gambardella. Max-pooling convolutional neural networks for vision-based hand gesture recognition. In *Signal and Image Processing Applications (ICSIPA), 2011 IEEE International Conference on*, pages 342–347. IEEE, 2011.

[13] Michael Van den Bergh and Luc Van Gool. Combining rgb and tof cameras for real-time 3d hand gesture interaction. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 66–72. IEEE, 2011.

[14] Nicolas Pugeault and Richard Bowden. Spelling it out: Real-time asl fingerspelling recognition. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1114–1119. IEEE, 2011.

[15] Alina Kuznetsova, Laura Leal-Taixé, and Bodo Rosenhahn. Real-time sign language recognition using a consumer depth camera. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 83–90, 2013.

[16] Cao Dong, Ming C Leu, and Zhaozheng Yin. American sign language alphabet recognition using microsoft kinect. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 44–52, 2015.

[17] Pengjie Tang, Hanli Wang, and Sam Kwong. G-ms2f: Googlenet based multi-stage feature fusion of deep cnn for scene recognition. *Neurocomputing*, 225:188–197, 2017.

[18] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.