A deep learning approach to sign language recognition using stacked autoencoders and neural networks

Pablo Rivas

Jan/24/17

1 Introduction

Language is an essential part of being human as it enables us to communicate with others. Some people have to develop additional means of communication for different reasons. A popular alternative language is through signs. This type of language is based on symbols or figures produced using the hands.

Sign language recognition is a task that has been studied in the last few years [13]. We believe this is an important problem to address because it directly affects the quality of life of people who need to communicate using sign language. The task is to develop algorithms that can recognize different signs in an alphabet and that can do so in an efficient manner. Ideally, the methodologies should be fast to the point of enabling the near real-time processing of signs [12].

In this research we deal with the 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 [5].

Typical approaches involve the using hidden Markov models [12], and a combination of them with other discriminative functions for feature extraction in multi-stage architectures [7]. Other major alternatives included the exploration of neural strategies combined with fuzzy systems [1]. For a more detaile review of alternatives for generic hand gesture recognition we can turn to the work in [9].

However, with the dramatic attention that deep learning has gained recently in the machine learning and the immage processing for pattern recognition communities, we are motivated to similarly explore this new algoritmic alternative to see if it offers better solutions to open problems. Most recently, the authors in [5] have explored a deep learning approach based on convolutional neural networks (CNNs) achieving outstanding results. The research presented here also explores a novel deep learning approach but based in autoencoders and neural networks. We will show that this alternative approach achieves great performance and efficiency.

The rest of the paper is organized as follows.

2 Background and Related Work

Recently, J. Nagi et.al. [10] studied a deep neural network approach for recognizing six different hand gestures given to a robot. Using CNNs the authors achieve a 96% accuracy rate. Then, M. Van den Bergh et.al. [3] explored the idea of using depth-sensor imagery to establish an algorithm for hand segmentation. The authors ultimately used this to recognize six different hand gestures in 3D. They achieved a 99.54% recognition rate. Some of the key commonents of the overall architecture were the usage of wavelets for image filtering and neural networks.

Moreover, B. Kang et.al. [5] took advantage of the recent success of CNNs for pattern recognition over images. They propose a system that makes use of depth-sensor images to study further the problem of hand gesture recognition. When recognizing signs within the same subject they averaged a 99.99% accuracy rate over five subjects. However, when recognizing signs across different subjects, their accuracy rate droped to 83.58%.
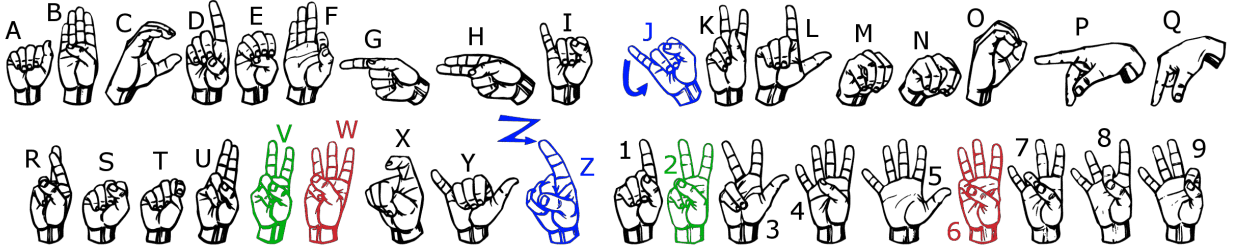
Figure 1: The American Sign Language (ASL) displaying 26 signs corresponding to the English alphabet (A to Z) and 9 signs for numbers (1 to 9).

Figure 2: Examples of ASL hand gestures corresponding to the number one and number two with variations in the horizontal axis and its inclination with respect to a depth-sensor camera.

Our research aims to further study the problem posed in [5] of increasing accuracy rates over different subjects. In the next section we discuss the dataset used and propose a deep learning approach using autoencoders and a neural network.

3 Methodology

3.1 Data

This research focuses in the ASL as a case study [2]. The ASL has 26 hand gestures corresponding to the letters in the English alphabet; it also contains 9 hand gestures for every single number. Figure 1 depicts the gestures pertaining to our study.

We use the data provided in [5] that was released in 2015. This dataset consists of images acquired with a three-dimensional depth-sensor camera. There is a total of $31,000$ images available with a resolution of 256×256. The images correspond to already segmented dept-images that contain the area of the hand. The authors use a rather simple algorithm to make the segmentation based on the fact that in all depth images the hand is the closest object with respect to the camera.

Not all the 35 signs (26 letters and 9 numbers) were considered in this study. Some signs were exluded and others considered jointly as follows.

The signs corresponding to J and Z clearly require a sequence of images rather than a single instance, as can be seen in Figure 1. This goes beyond the scope of this project; although the detection of signs from live-video sequences is part of a bigger project, we will not address it in this paper. For that reason these two signs were exluded.

The signs corresponding to V and 2, by observation, are nearly identical and their distinction requires contextual information. Similarly, the signs for letter W and number 6 require being interpreted in context. For this reason, these signs were considered as the same sign, leaving a total of 31 signs.

There is a total of five different subjects and each subject produced 200 images of each sign. This represents a total 1,000 images per sign and 31,000 images overall. The images collected capture each subject making a hand gesture moving across the horizontal axis and varying inclination of the hand gesture, as depicted in Figure 2.

3.2 Sparse Autoencoders

A sparse autoencoder is usually categorized as a neural network with unsupervised learning [6]. In a general sense an autoencoder is trained to output an approximation of the input provided a deep architecture of layered neurons that encode and decode based on the input stimuli. In our research we use a sparse autoencoder that specifically minimizes a modified loss function based on the mean squared error.

Let $\mathbf{x} \in \mathbb{R}^d$ be a d-dimensional input vector. Then the loss function of the common sparse autoencoder is defined as follows:

$$L = \frac{1}{N} \|\mathbf{x}_n - \hat{\mathbf{x}}_n\|_2^2 + \theta_w \frac{1}{2} \sum_{l=1}^{L} \|\mathbf{w}^l\|_2^2 + \theta_s \sum_{m=1}^{M} KL\left(\theta_\alpha \,\|\, \bar{\alpha}_m\right) \tag{1}$$

where N is the total number of training samples; $\hat{\mathbf{x}}_n$ is the learned (or encoded) approximation of the n-th input vector $\mathbf{x}_n$; θ_w controls the sparsenes of the weights of the network, $\mathbf{w} \in \mathbb{R}^d$; L denotes the number of layers in the deep network; θ_s regulates the sparsity of the activation functions' output, α, in every neuron in the network; M is the total number of neurons in the deep network; and $KL(\cdot)$ is the Kullback-Leibler divergence function [4] used to measure how much the observed average activation of the m-th neuron, $\bar{\alpha}_m$, actually deviates from the desired average output, θ_α.

The Kullback-Leibler divergence function can be defined as follows [11]:

$$\sum_{m=1}^{M} KL\left(\theta_\alpha \,\|\, \bar{\alpha}_m\right) = \sum_{m=1}^{M} \theta_\alpha \log\left(\frac{\theta_\alpha}{\bar{\alpha}_m}\right) + (1 - \theta_\alpha)\log\left(\frac{1 - \theta_\alpha}{1 - \bar{\alpha}_m}\right) \tag{2}$$

where the average output of the m-th neuron, $\bar{\alpha}_m$, at the l-th layer is given by

$$\bar{\alpha}_m = \frac{1}{N} \sum_{n=1}^{N} \psi\left(\mathbf{w}_m^{(l)T}\mathbf{x}_n + b_m^{(l)}\right) \tag{3}$$

where $\psi(\cdot)$ is the neuron's activation function, and $b_m^{(l)}$ is the bias term for the m-th neuron at the l-th layer. In this research we specifically use logistic (sigmoid) activation functions, $\psi(z) = 1/(1 - e^{-z})$, for any given value of z.

By observing the mathematical form of (1), it is feasible to apply an accelerated form of guided learning known as scaled conjugate gradient (SCG) descent [8]. SCG has proven to be a reliable and efficient form of minimizing loss functions such as the one for the sparse autoencoder, overcoming the shorcomings of a traditional conjugate gradient and a back-propagation with gradient descents [6].

3.3 Deep Learning Architecture

Two stacked autoencoders are combined with a feedforward neural network in a five-layer architecture, as shown in Figure 3. The first two layers are a set of unsupervised autoencoders that minimize the loss function in (1). 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 attibutes 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 estimulated $\tilde{\mathbf{x}}_n$ and is trained to predict the probability of the n-th sample belonging to a specific class $C \in \{1, 2, \ldots, 31\}$. The output of this layer for the n-th sample is the estimated probability of that sample belonging to all classes, formally denoted as $\hat{\mathbf{d}}_n \in \mathbb{R}^{31}$. The layer is trained to minimize the cross entropy function [14] given by:

$$E = \frac{1}{N} \sum_{n=1}^{N} \sum_{c \in C} \hat{d}_{cn} \ln d_{cn} + (1 - \hat{d}_{cn})\ln(1 - d_{cn}) \tag{4}$$

predicted class

31 neural units

neural network layer

100 neural units

decoding layer

50 neural units

encoding layer

65536 neural units

decoding layer

100 neural units
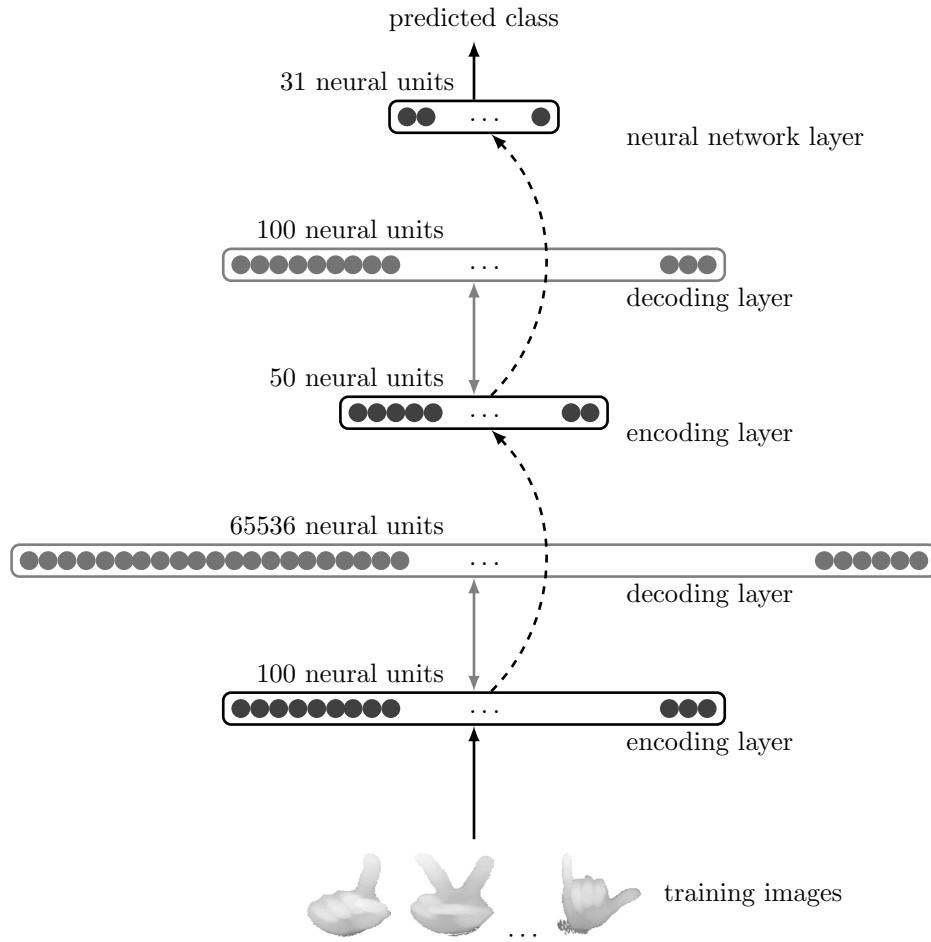
encoding layer

training images

Figure 3: Deep architecture during training. The training begins with the first two layers, encoder - decoder, and once the training is complete feature are encoded and propagated to the third layer in order to further encode - decode high-level features. Finally, the last neural layer retrieves the encoded features from the third layer and learns the target class. The training is performed using SCG descent [6].

output $\mathbf{d}_n \in \mathbb{R}_+^{31}$

softmax activations

$\tilde{\mathbf{x}}_n \in \mathbb{R}_+^{50}$

neural network layer

logistic activations

$\hat{\mathbf{x}}_n \in \mathbb{R}_+^{100}$

sparse encoding layer

logistic activations

$\mathbf{x}_n \in \mathbb{N}_+^{65536}$

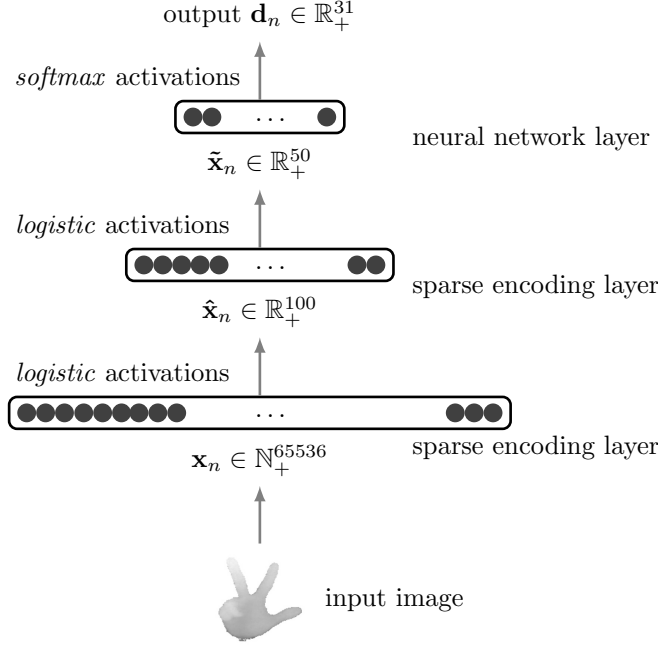sparse encoding layer

input image

Figure 4: The working architecture when testing the system. The input is any test image which is passed through the first sparse encoder in the stack, reducing the dimensionality of the problem to 100. In the second sparse encoder the feature space is further reduced to 50. The output are probabilistic outputs corresponding to each of the 31 classes.

where $\mathbf{d}_n \in \mathbb{R}^{31}$ is the true probability of the n-th sample belonging to a specific class.

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 trained simmulating a feed-forward neural network, as shown in Figure 4. The initial weights are those obtained during the encoding-decoding learning phase and fine tuned using SCG descent to minimize the cross entropy in (4).

4 Experiments

4.1 Setup

4.2 Results and Discussion

References

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

[2] Baker, C., Cokely, D.: American sign language. A Teacher's Resource Text on Grammar and Culture. Silver Spring, MD: TJ Publ (1980)

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

[4] Joyce, J.M.: Kullback-leibler divergence. In: International Encyclopedia of Statistical Science, pp. 720–722. Springer (2011)

[5] Kang, B., Tripathi, S., Nguyen, T.Q.: Real-time sign language fingerspelling recognition using convolutional neural networks from depth map. In: Pattern Recognition (ACPR), 2015 3rd IAPR Asian Conference on. pp. 136–140. IEEE (2015)

[6] Le, Q.V.: Building high-level features using large scale unsupervised learning. In: Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. pp. 8595–8598. IEEE (2013)

[7] Lichtenauer, J.F., Hendriks, E.A., Reinders, M.J.: Sign language recognition by combining statistical dtw and independent classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 30(11), 2040–2046 (2008)

[8] Møller, M.F.: A scaled conjugate gradient algorithm for fast supervised learning. Neural networks 6(4), 525–533 (1993)

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

[10] Nagi, J., Ducatelle, F., Di Caro, G.A., Cireşan, D., Meier, U., Giusti, A., Nagi, F., Schmidhuber, J., Gambardella, L.M.: Max-pooling convolutional neural networks for vision-based hand gesture recognition. In: Signal and Image Processing Applications (ICSIPA), 2011 IEEE International Conference on. pp. 342–347. IEEE (2011)

[11] Olshausen, B.A., Field, D.J.: Sparse coding with an overcomplete basis set: A strategy employed by v1? Vision research 37(23), 3311–3325 (1997)

[12] Starner, T., Pentland, A.: Real-time american sign language recognition from video using hidden markov models. In: Motion-Based Recognition, pp. 227–243. Springer (1997)

[13] Starner, T., Weaver, J., Pentland, A.: Real-time american sign language recognition using desk and wearable computer based video. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(12), 1371–1375 (1998)

[14] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2818–2826 (2016)