Journal of Computer Science

Original Research Paper

An Enhanced Training- Based Arabic Sign Language Virtual Interpreter Using Parallel Recurrent Neural Networks

Mohamed A. Abdou

Informatics Research Institute, City for Scientific Research and Technology Applictions, Alexandria, Egypt

Article history
Received: 16-12-2017
Revised: 27-01-2018
Accepted: 16-02-2018

Email: doc_abdelrahman@yahoo.com

Abstract: Intelligent machine translation systems have a remarkable importance in integrating people with disabilities in community. Arabic to Arabic sign language systems are limited. Deep Learning (DL) was successfully applied to problems related to music information retrieval, image recognition and text recognition, but its use in sign language recognition is rare. This paper introduces an automatic virtual translation system from Arabic language into Arabic Sign Language (ASL) via a popular DL architecture: The Recurrent Neural Network (RNN). The proposed system uses a deep neural network training-based system for ASL that convolves RNN and Graphical Processing Unit (GPU) parallel processors. The system is evaluated using both objective and subjective measures. Obtained results are towards reducing errors, speeding up avatar and expressing signs and facial expressions in a well-received manner by Deaf. The signing avatar is highly encouraged as a simulator for natural human signs.

Keywords: Deep Learning, Recurrent Neural Network, GPU, Intelligent Arabic Sign Language, Signing Animations

Introduction

Every year in the US, more than 12,000 babies are born with hearing loss disability (http://www.parentcenterhub.org/wp-content/uploads/repo_items/fs3.pdf). Hearing loss could be classified as: slight, mild, moderate, severe, or profound; depending upon how well a person can hear either the intensity or the frequency of the acoustic signal. 'Deaf' are those who have profound hearing loss. In most developing countries, Deaf people rarely receive special education services and thus have high unemployment rate. Reasons are the lack of education tools supporting sign language, problems in communication with neighbors/colleagues and high cost for using sign language interpreters. Communication between Deaf and normal people is a major problem. Normal people usually lack sign language proficiency. Introducing software technology or virtual interpreters to help Deaf and embed them in education and training is a big challenge. Without appropriate interventions, children with hearing impairment could exhibit anxiety and may get mental health disorders (Azab *et al*., 2015). Signing videos could be a better alternative in education as such materials/lessons could be pre-prepared. However, videos are considered as half duplex communication, i.e., lacks interaction with audience. Furthermore, production of high-quality videos needs large storage space, wide bandwidth and outstanding internet speed. For these reasons, videos are time consuming, expensive and static education tools (Kennaway *et al*., 2007).

Signing avatars gained much more interest in the last decade and covered different applications. In (Jaballah and Jemni, 2013; Lincoln *et al*., 2001) TV programs and multimedia were translated into sign languages through signing avatars. The former lacked real English sign language; the latter had a remarkable delay between the voice and the signing avatar motion. TESSA (Bangham *et al*., 2000) which stands for 'text and sign support assistant' was developed to support Deaf people while communicating in service or governmental places. In a previous work, signing avatars were used to embed Deaf students with normal students in classrooms (Mohamed *et al*., 2016). The paper presented an automatic signing avatar in Arabic Sign Language (ASL) environment and presented three hybrid modules: Arabic speech recognizer, machine translator and signing avatar animator. Although the system presented in (Mohamed *et al*., 2016) achieved remarkable attention from researchers and Deaf community, the speech recognizer based on adaptation model to recognize Arabic language resulted limited

corpora. To better extend this work, training models will be introduced based on Deep Learning (DL). Advance in hardware design and architecture has significantly increased the efficiency of Deep Neural Networks (DNNs) for different applications. Evolutions include (but not limited to) development of Graphical Processing Units (GPUs), progress in distributed systems and High performance computing. The multi-level training used in DL makes it easier to learn complex functions that map input to output, without the need of dependence on handcrafted features (Zhang *et al*., 2014). Advantages of DL:

- Learning from the data itself
- Having state-of-the-art results and
- Outperforming humans and human-coded features

The motivation of this work is firstly to introduce DL in ASL recognition researches. Secondly, investigating DL in GPU environment and measure how it could solve complex problems with a fast learning rate. Finally, extend the adaptation-based Arabic sign language interpreter with limited corpora (Mohamed *et al*., 2016) into a better performance training-based system. The proposed system is evaluated using two performance metrics: Bilingual Evaluation Understudy (BLEU) and Sign Error Rate (SER). The paper is organized as follows: Section 2 describes the state of the art work. Section 3 shows the proposed Arabic data set preparation and the proposed DNN system. Section 4 shows obtained results and comparisons. Finally, section 5 concludes the proposed system.

State of The Art Work

Deep Learning (DL)

DL methods construct new features by transforming input data through multiple layers of nonlinear processing. This is accomplished by training large neural networks (NNET) with several hidden layers and data sets with very large sample sizes. Recently, there has been a trend to apply DL to data sets with limited sample sizes to fit with real world applications (LeCun *et al*., 2015; Strobl and Wisweswaran, 2013). DL is nowadays used by leading software companies to solve speech recognition, computer vision and natural language processing. It has many advantages when compared to traditional machine learning techniques. DL is able to detect complex interactions among features, capable to learn low-level features from minimally processed raw data and able to work with unlabeled data (Latha and Priya, 2016).

Recurrent Neural Network (RNN)

RNN plays an important role in pattern classification especially when expecting a sequence of data. A RNN, as shown in Fig. 1, is an artificial neural network that adds additional weights creating cycles to maintain an internal state. The layered topology of a multilayer perceptron is preserved, but every element has a weighted connection to another element in the architecture and has a single feedback connection to it. Not all connections are trained and the backpropagation through time approaches (non-linearity of the error derivatives) is employed (Elman *et al*., 1996). A RNN allows a sequence of inputs, where at every step the model will backpropagate the error, update its weights and save changes (a sort of memory). The model is perfect for capturing sequences of words (sentences or passages). Since one sentence spoken by the teacher is a sequence of words, it would be better to enter it to the RNN and wait for the decision; i.e., selection of the correct signing video. There exist several RNN models such as: Elman RNN (Elman *et al*., 1996), Hopfield RNN (Hopfield, 2008) and Jordan RNN (Jordan, 1986).

RNN Mathematical Model

To explain the mathematical model within a RNN, the neural network inputs and outputs are assumed $x(t)$ and $y(t)$. The three connection weight matrices are: W_{IH}, W_{HH} and W_{HO}; and the hidden and output unit activation functions are f_H and f_O, where 'H' stands for hidden, 'O' stands for output and 'I' stands for input. The performance action of the RNN could be explained in terms of a dynamic system using non-linear matrix equations given by:

$$h(t) = f_H(W_{IH} \cdot x(t) + W_{HH} \cdot h(t-1)) \quad (1)$$

$$y(t) = f_0(W_{HO} h(t)) \quad (2)$$

These are set of values that summarize the past behavior of the system, provide a unique description of its future behavior without considering the effect of external actions. Here, the situation could be defined by the set of hidden units $h(t)$. The order of the equivalent dynamic system is proportional to the dimensionality of the number of hidden layers. In Elman RNN, each set of weights appears only once, so it is possible to apply the gradient descent approach using the standard backpropagation algorithm. Here, the error signal will not be propagated to the network; this approximation is considered highly effective in many applications, one of them is the word pattern classification.

Hopfield (2008), the Hopfield Network or Hopfield Model is one good way to implement an associative memory. It is simply a fully connected RNN where activations are normally ±1 (defined using the signum function), rather than 0 and 1, so the neuron activation equation is:

$$x_i = \text{sgn}\left[\sum_j w_{ij} x_j - \theta_i\right] \quad (3)$$

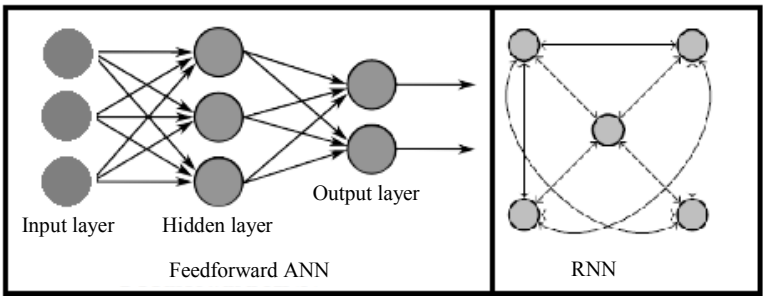

Fig. 1: ANN Vs. RNN diagrams

where, w are the weights and θ the threshold.

The activations depend on time; they keep changing till reaching a stable pattern. They could be updated in a synchronous or asynchronous way. The required associative memory can be achieved by setting the weights w_{ij} and thresholds θ_j according to the target outputs t_p using the following relations:

$$w_{ij} = \frac{1}{N}\sum_{p=1}^{P} t_i^p t_j^p \tag{4}$$

where, t are the stored patterns. Consider the neuron activation functions are unchanged, one stored pattern t_q could be:

$$\begin{aligned} t_i^q &= \text{sgn}\left[\sum_j w_{ij} t_j^q - \theta_i\right] = \text{sgn}\left[\sum_j \frac{1}{N}\sum_p t_i^p t_j^p t_j^q\right] \\ t_i^q &= \text{sgn}\left[t_i^q + \frac{1}{N}\sum_j \sum_{q\neq p} t_i^p t_j^p t_j^q\right] \end{aligned} \tag{5}$$

The second term in the last equation is zero or has a magnitude less than unity, it is clear that pattern number q is stable. In many practical problems this assumption is feasible whenever the number of stored patterns P is more or less small. Jordan-type RNNs are similar to Elman-type networks, except that the context nodes are fed from the output layer instead of from the hidden layer. The context nodes in a Jordan-type network are also referred to as the state layer. The difference between Elman and Jordan-type networks appears only in the hidden layer input (Jordan, 1986). Applications of RNNs include handwritten characters recognition, object recognition and detection in image, speech recognition and time series (Ball *et al*., 2017).

A RNN is defined by a set of time equations:

$$\begin{aligned} h_t &= Wf(h_{t-1}) + W^{(hx)} x_t \\ \hat{y} &= W^{(s)} f(h_t) \end{aligned} \tag{6}$$

where, h is the output transfer function, W are the weights, x is the input. All defined at one time step. Figure 2 illustrates a three time steps RNN.

Spoken Language to Animation

Sign languages are complete complex languages. They are represented visually with neither written letters nor characters. To compose a sentence or a sequence of signs, we use hands, arms, head and body. Many of us do not know that Deaf people could not read traditional languages; which means that they need much more attention to communicate with normal people. Generally, translation from spoken languages to animation goes through four steps:

- Text – Semantic representation
- Semantic – sign notation
- Sign notation – gesture notation
- Gesture – animation

Sign language linguistic depends on the application or the environment where sign language is needed. A pool of words or sentences used in traffic offices when renewing the driving license was the field of study in (Efthimiou *et al*., 2009); another related to bus transportation info was studied in (Efthimiou *et al*., 2012). Many challenges exist while converting spoken Arabic to ASL. One sign may refer to different written words, the lack of ASL linguistics (https://www-03.ibm.com/press/us/en/pressrelease/22316.wss) and lack of ASL bilingual corpora (Da Rocha Costa and Dimuro, 2001).

When focusing on sign language animations, researchers usually compared avatars to real signs (Wilbur, 1997). Since there does not exist standard benchmarks for performance evaluation (Hanke, 2004), it would be better to rely on Deaf as main users. When studying the development of spoken languages to sign languages translation prototypes, we can classify systems into: Example-based systems (Cooper and Bowden, 2009), rule-based systems (Courty and Gibet, 2010) and statistical approaches (San-Segundo *et al*., 2012). Signs must be transformed into gestures to be processed by computers. One of the well-known gesture notations is the **SiGML** (Sign Gesture Markup Language) which has been developed to be independent of any sign language (Elliott *et al*., 2010).

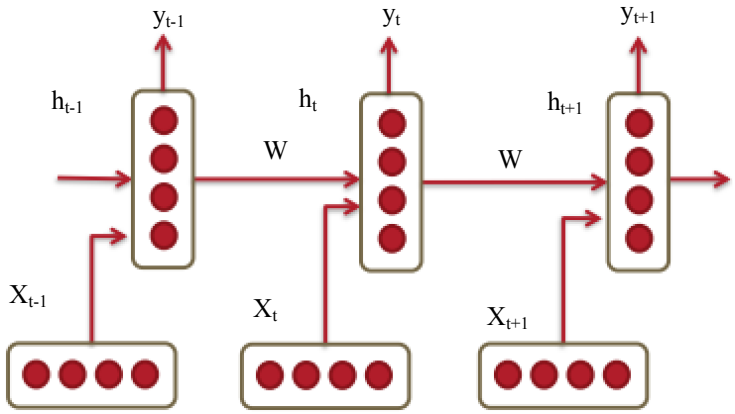

Fig. 2: Three Time Steps RNN

Two techniques could be used to animate signing avatars: The motion capture and the synthetic signing. The motion capture technique relies on a database of recorded signs with help of a human signer and using special equipment connected to the signer's body. The synthetic signing technique depends on transcribing signs by giving each component (hand shape, movement, orientation …) a specific value using a phonetic notation system. These values (in their gesture notation form) together with a description of the signing avatar geometry are passed to the signing avatar to animate it to show the required sign. The advantages of the synthetic signing technique are:

- No need for special equipment
- Recorded signs could be easily modifiable
- No need to download a database of signs to play the signing avatar

The next section approaches the proposed training-based DNN system. This uses a RNN that classifies recorded signing videos to their relevant text (from a speech recognizer). All target signing videos are then transformed into signing animations (avatar).

A Proposed Training-Based DNN Model

Overall System

The proposed system block diagram, as shown in Fig. 3, consists of three main parts: The Arabic Speech Recognizer, the RNN ASL Interpreter and the Signing Avatar Animator.

The speech recognizer receives the Arabic speech of the instructor and converts it to a written form (Arabic sentences). The module makes use of the CMUSphinx speech recognition toolkit (CMUSphinx website). This is based on Hidden Markov Model. The original system was implemented to support US English acoustic model by a group of researchers, however we modified this toolkit to recognize Arabic. The RNN ASL interpreter represents the interface between Arabic text and signing videos. This will be explained in full details in the incoming sections. Finally, the signing avatar module plays the sequence of signs to express an animated translation. The system uses the JASigning (Java Avatar Signing) software (Kipp *et al*., 2011) that has many advantages. Its flexibility, as it uses decentralized SiGML files, it can be used to develop desktop or web applications and it can be considered as a real time system.

Arabic Data Set Preparation for RNN

Prior moving to DL and RNN, we have to explain how the Arabic training data set is prepared. Firstly, Arabic characters will be represented by integer numbers relevant to their order in the Arabic alphabet (one character is given a number between 1 and 28). All words will be assumed having length of eight characters. We will use left zero padding for words less than eight characters. These in turns generate time series of 1×8 vectors. In Arabic language, one sentence could be written in different ways using the same words, order of words could be changed without affecting the meaning. Aided with a group of linguistic experts, a *sentence pool* is formed for each sentence. A *sentence pool* 'S_i^p' is a cell array containing all words' vectors that could be arranged in different possible forms maintaining the same meaning. The size of the pool is N. However this sentence may be formed in different manner with a variable number of words (K), where $k \leq N$:

$$S_i^p = \{W_{i1}, W_{i2}, W_{i3}, ...W_{iN}\} \tag{7}$$

where, S^p_i is sentence (i) pool of vectors (words), Ws are vectors representing words and the pool is of length N:

$$\begin{aligned} &S_i^k = [W_{i1}, W_{i2}, ...W_{in}] \\ &W_{in} \in S_i^p \end{aligned} \tag{8}$$

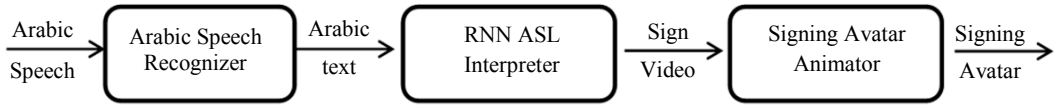

Fig. 3: The proposed ASL interpreter

where S_i^k is a sequence of vectors (1×8) for one possible pattern (k) for sentence (i). The reason for getting help from a language expert is that Arabic sentence structures are complex enough such that one sentence could have multiple of possible forms with similar meanings. To further explain the idea, let's assume the largest sentence in the corpora with a pool consisting of 9 words. S_i^p = {يحتوي، شريط، العنوان، على، اسم، البرنامج، و، المستند، الحالي}. S_i^p will be mapped as shown in Table 1 before being used as an input to the proposed RNN. From our language expert, we can have only four possible reference patterns for this sentence (i) in the following sequences:

$$\{W_i^1, W_i^2, W_i^3, W_i^4, W_i^5, W_i^6, W_i^7, W_i^8, W_i^9\},$$
$$\{W_i^1, W_i^2, W_i^3, W_i^4, W_i^8, W_i^9, W_i^7, W_i^5, W_i^6\},$$
$$\{W_i^2, W_i^3, W_i^1, W_i^4, W_i^5, W_i^6, W_i^7, W_i^8, W_i^9\},$$
$$\{W_i^2, W_i^3, W_i^1, W_i^4, W_i^8, W_i^9, W_i^7, W_i^5, W_i^6\}$$

All four sequences should result the same ASL target video. In our previous work (Mohamed *et al*., 2016), the total number of sentences was 54. Taking into considerations that the largest sentence has nine words and the average patterns for one sentence is 4, the corpora is extended to a 1×1944 cell array of 8×1 vectors; each vector represents one word.

Parallel DNN Training-Based Translator

Graphical Processing Unit (GPU) could play an important role in DL because of their parallel processing structure that speeds up both learning and inference (Ota *et al*., 2017). This application requires a host computer with an NVIDIA GPU card, where the proposed algorithms of the DNN are parallelized. This GPU device is a Single Instruction, Multiple Data (SIMD) device. Since DNN requires larger amount of neurons compared to conventional neural networks, it would be better to process DNN using GPU. In this study, MATLAB instructions are sent to a multicore structure; this takes cell arrays and converts them to GPU arrays before training the RNN. Furthermore, the GPU array result is converted back to cell arrays before simulation. This aims speeding up the RNN training and testing. Neural networks toolbox instructions are used for training and testing the RNN at each layer.

The Training-Based translation depends on the existence of a bilingual corpus: A set of Arabic sentences written in different forms and their corresponding ASL. In our previous work (Mohamed *et al*., 2016), each pair was called an example and the system was implemented in an example-based way. This encountered two challenges:

- Difficulties in Arabic sentences structures
- Changing SL interpreters led to different spoken language vocabulary

The RNN training- based translation system is shown in Fig. 4. The proposed RNN comprises 8/one neuron(s) in the input/output layers respectively. Weights are initialized randomly in the interval $\sqrt{\left|\frac{1}{\sqrt{n}}\right|}$ where n is the number of incoming connections from the previous layer. The maximum number of epochs is set to 1000; however the RNN should converge earlier, as will be shown in the next section. An RNN's ability to capture sequences of words definitely enables it to learn parts of speech, syntax and n-grams (Schaul *et al*., 2010). In Arabic, you can say the same sentence using different wordings, different word order, or even different phonetic. Our goal is to recognize the sentence and assign the relevant signing video even if the sentence is said in different ways or the speech recognizer adds or misses words. As stated in the previous section, training data set will take the form of a time series cell array. The output layer of the DNN will select the correct video number. The training data set comprises 70% of the total data set (1×1944 cell array), 20% for testing and 10% for validation. Results and comparison are shown in section 4.

Software Framework

The proposed system is developed as desktop application. The training-based model via DNN is implemented via MATLAB-R2014. CMUSphinx speech recognition toolkit is used. The existing US English acoustic model in the CMUSphinx has been modified to recognize Arabic as explained in full details in (Mohamed *et al*., 2016) and following the adaptation illustrated by the library tutorial. Finally, we use the JASigning (Java Avatar Signing) to represent the sequences of signs. This gets use of SiGML files to generate the motion data used to animate the avatar.

Hardware Structure

The proposed system is applied on a hardware platform that comprises a core i7 processor with 2.7 GHz and 8 GB memory. The H/W has an NVIDIA GeForce GT940 MX.

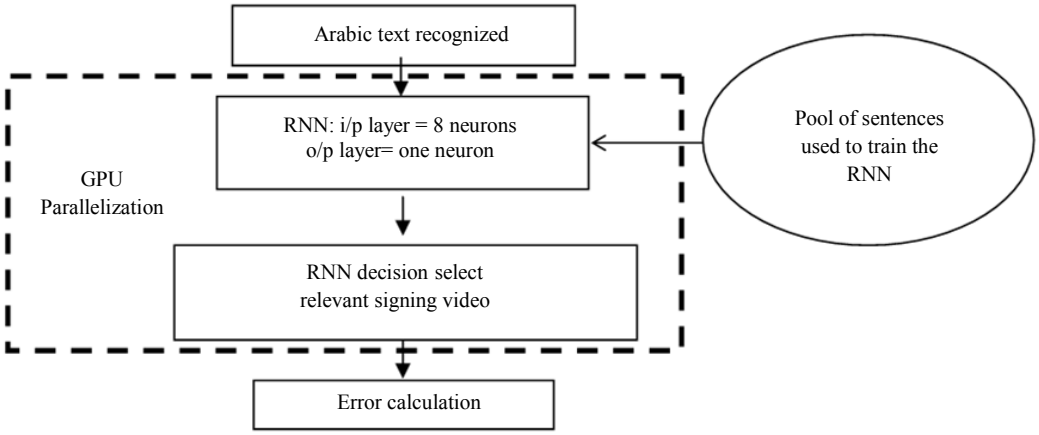

Fig. 4: The proposed RNN training- based block diagram over GPU

Table 1: Arabic sentences preparation for the RNN

	Word	Equivalent Map		Word	Equivalent Map
W_i^1	يحتوي	[0 0 0 28 27 3 6 28]	W_i^6	البرنامج	[5 24 1 25 10 2 23 1]
W_i^2	شريط	[0 0 0 0 16 28 10 13]	W_i^7	و	[0 0 0 0 0 0 0 27]
W_i^3	العنوان	[0 25 1 27 25 18 23 1]	W_i^8	المستند	[0 8 25 3 12 24 23 1]
W_i^4	على	[0 0 0 0 0 28 23 18]	W_i^9	الحالي	[0 0 28 23 1 6 23 1]
W_i^5	اسم	[0 0 0 0 0 24 12 1]			

Results

Proposed Criteria

Assuming the followings criteria:

- Microsoft word processing applications is selected as the education classroom
- 54 Arabic sentences are the benchmark, generating more than 1000 data set for training
- Evaluation of the training-based system is objective
- Evaluation of Avatar performance is subjective

Two *objective* performance metrics are used to evaluate system performance:

- Bilingual Evaluation Understudy (BLEU) and
- Sign Error Rate (SER)

The *subjective* evaluation of the signing process is carried out using a questionnaire directed to 30 Deaf students with different hearing ability and different level of using Microsoft applications.

Parallel DNN Results

This work generalizes the Elman network -that used only one layer- to have an arbitrary number of layers and '*tansig*' as the corresponding transfer functions. The proposed DNN (RNN) works as a parallel processing via GPU processors in order to speed up the training process. In this proposed RNN, weights and biases are updated in each epoch in the direction of the negative gradient. Figure 5 shows DNN training performance. From this figure, it could be seen that the network reaches the minimum error required after 582 epochs in 35 sec. Figure 6 shows the effective parameters, gradient value and MU when the network converges. The effective parameters (weights and biases) are those that remain approximately the same assuming that the network has been trained for a sufficient number of iterations to ensure convergence. The training may stop with maximum MU or maximum number of epochs is reached; or the Sum Squared Error (SSE) is relatively constant over several iterations.

Proposed System Evaluation

Objective Measures

Two objective measures will be adopted to evaluate the proposed system: the BLEU and the SER metrics. BLEU ranges between 0 (incorrect translation) and 1 (translation identical to the reference); while SER is defined as the total number of incorrect signs obtained from the language translator to the total number of signs. BLEU is used to evaluate the Arabic speech recognizer; on the other hand the SER is used to evaluate the proposed DNN.

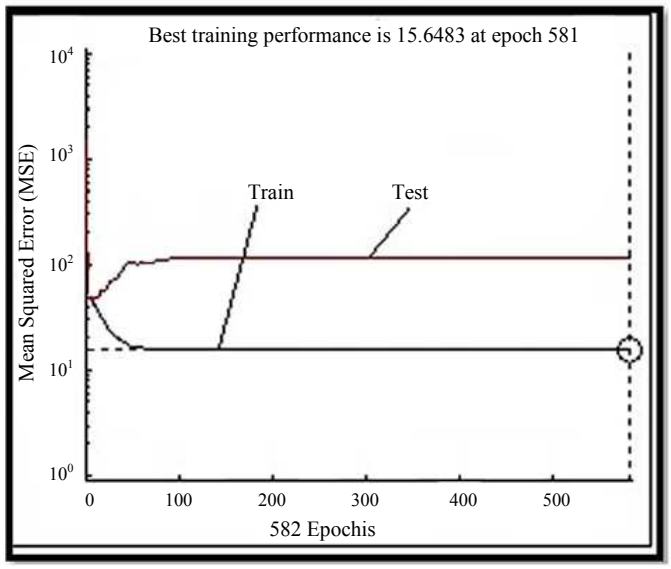

Fig. 5: DNN training performance

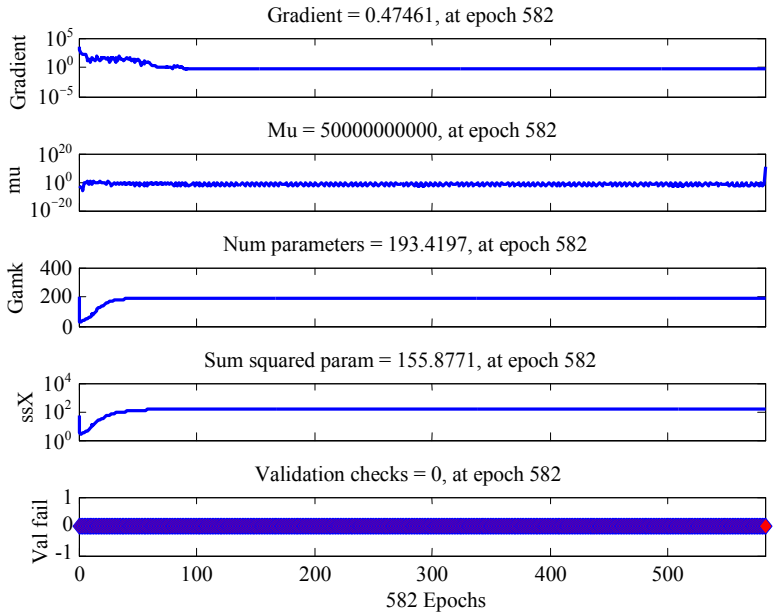

Fig. 6: DNN training state

Table 2 shows the effect of BLEU values on both the training-based DNN and the example-based method (Mohamed *et al*., 2016). From this table it could be observed that only the training based system succeeds with very low BLEU values. This means that the proposed training based system could overcome poor performance of speech recognizer stage. SER comparison is presented in Table (3); where the example based system reduces error from 0.034 to 0.0055, i.e., by about 84%.

Subjective Measure

According to World Health Organization (WHO) reports, the grades of hearing impairment are: *Slight* (26-40 db), *moderate* (41-60 db), *severe* (61-80 db) and *profound* (over 81 db) (http://www.who.int/pbd/deafness/hearing_impairment_grades/en/). The subjective evaluation of the signing avatar desktop application is carried out with the contribution of 30 (7 profound, 13 severe and 10 moderate) Deaf users who use the ASL as their mother language.

Table 2: Performance evaluation of the DNN example-based system Vs. training-based system

Sentence pool	Number of patterns (f)	BLEU		Example based (Mohamed *et al*., 2016)	Parallel DNN
1	5	0.34	Average	Correct	Correct
2	4	0.27	Low	Correct	Correct
5	7	0.77	High	Correct	Correct
17	9	0.91	High	Correct	Correct
18	8	0.81	High	Correct	Correct
30	4	0.25	Low	Correct	Correct
49	4	0.19	V. Low	Incorrect	Correct
53	3	0.18	V. Low	Incorrect	Correct

Table 3: SER Comparison for the total of 360 sentences

Method	No. of incorrect sentences	SER
Example based ((Mohamed *et al*., 2016))	12	0.0340
Parallel DNN	2	0.0055

Table 4: Subjective avatar measures

Category	Question	Strongly agree	Agree	Disagree	Strongly disagree	% Success
Overall	Do you want to use similar systems in the future?	27	3	-	-	100
	Do you find the system beneficial?	29	1	-	-	100
Interface	Do the avatar movements seem artificial?	4	1	12	13	83
	Do you find the tools in the interface useful?	20	9	-	1	97
	Is the interface simple and clear?	17	8	3	2	83
ASL Rating	Have you found unknown Arabic signs?	1	3	9	17	87
	Was the signs' speed as you used to see?	15	10	4	1	83
	Do you find the signs as you learnt before?	19	11	-	-	100

The target pool are equally distributed from *fair* to *professional* in word processing levels; i.e., 10 professionals, 10 intermediates and 10 beginners. The evaluation questionnaire is summarized in Table 4 and represents eight different questions covering three main areas: interface, ASL and overall performances. It could be seen that the overall performance of the training-based system succeeded 100%. Regarding the ASL rating, we achieved results from 90 to 100%. Regarding the interface, the system was accepted by 90 to 95%. Only one question is rated as 75% successful and from our opinion it is fair, as the avatar movement could not be considered natural like human signs that have emotions and face to face feelings.

GPU Performance

Hyper-threading technology is used to speed up computationally intensive applications. Mathematical operations and built-in MATLAB functions can be executed easily on GPU whenever presence of the Parallel Computing Toolbox. In this study, RNN moves from training, testing, to evaluation. When executing the algorithm on multiple threads during RNN evaluation, the parallel RNN shows a 3.5x speed up.

Discussions

The proposed work presents a training-based method that uses DNN assuming a RNN trained via GPU processors. Several observations could be discussed: Whenever the maximum reference patterns (*f*) increases for a sentence pool, higher BLEU values could be obtained. Moreover, low values of BLEU lead to wrong decision in case of example-based system (Mohamed *et al*., 2016), but reached a solid decision with DNN methods. The training based RNNs achieve higher BLEU and lower SER. While moving to the subjective measures, the speed of the signing avatar is adjusted per user; the degree of ASL knowledge differs from one person to another. Table 4 shows that the proposed system achieves high satisfactory ratios. Finally, during RNN training phase, the multicores technology exhibit a slow-down, as sending the data between the CPU and GPU takes larger time than running data on the GPU. However, during evaluation phase the GPU speeds up the proposed RNN by 3.5x.

Conclusion

An enhanced training-based interpreter for Arabic Sign Language has been presented. This has introduced a parallel RNN as an interface between the speech recognizer and the signing avatar. The proposed work has been evaluated objectively and subjectively. The obtained SER values are reduced by 84%, when compared to the example-based systems. Moreover, the proposed training-based RNN has reached a milestone even with low values of BLEU metric (0.18), reducing problems within the speech recognizer. In high BLEU metric values (0.91), both training-based and example-

based (Mohamed *et al*., 2016) systems has achieved correct matching. On the other hand, subjective evaluation has shown an 83% to 100% users' success ratios. Finally, using multi cores structure in GPU during RNN evaluation phase has speed up the algorithm by approximately 3.5x.

Acknowledgment

I am grateful to eSIGN consortium at the University of East Anglia to permit using the eSIGN and the JA Signing. Special thanks to Egyptian NGOs supporting Deaf for their help in evaluation process.

Ethics

The author confirm that this article content has no conflict of interest.

References

Azab, S.N., A. Kamel and S.S. Abdelrhman, 2015. Correlation between anxiety related emotional disorders and language development in hearing-impaired Egyptian Arabic speaking children. Commun. Disord Deaf Stud Hearing Aids.

Ball, J.E., D.T. Anderson and C.S. Chan, 2017. Comprehensive survey of deep learning in remote sensing: Theories, tools and challenges for the community. J. Applied Remote Sens.

Bangham, J.A., S.J. Cox, M. Lincoln, I. Marshall and M. Tutt *et al*., 2000. Signing for the deaf using virtual humans. Proceedings of the IEE Seminar Speech and Language Processing for Disabled and Elderly People, Apr. 4-4, IEEE Xplore Press, London. DOI: 10.1049/ic:20000134

CMUSphinx website: http://cmusphinx.sourceforge.net/

Cooper, H. and R. Bowden, 2009. Learning signs from subtitles: A weakly supervised approach to sign language recognition. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Jun. 20-25, IEEE Xplore Press, Miami, pp: 2568-2574. DOI: 10.1109/CVPR.2009.5206647

Courty, N. and S. Gibet, 2010. Why is the creation of a virtual signer challenging computer animation? MIG.

Da Rocha Costa, A.C. and G.P. Dimuro, 2001. A signwriting-based approach to sign language processing. Proc. GW Gesture Workshop.

Efthimiou, E., S.E. Fotinea and J. Segouat, 2009. Sign Language Recognition, Generation and Modelling: A Research Effort with Applications in Deaf Communication. Proceedings of the 5th International Conference, UAHCI, Jul. 19-24, San Diego, CA, USA, Springer.

Efthimiou, E., S.E. Fotinea, T. Hanke, J. Glauert and R. Bowden *et al*., 2012. Sign Language technologies and resources of the Dicta-Sign project. Proceedings of the Workshop to the 5th International Conference on Language Resources and Evaluation (LRE' 12), Istanbul, Turkey, pp: 37-45.

Elliott, R., J. Bueno, R. Kennaway and J. Glauert, 2010. Towards the integration of synthetic lamination with avatars into corpus annotation tools. Proceedings of the 4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies, (SLT' 10), Valletta, Malta.

Elman, J.L., E.A. Bates, M.H. Johnson, S.A. Karmiloff and D. Parisi *et al*., 1996. Rethinking Innateness: A Connectionist Perspective on Development. 1st Edn., MIT Press, Cambridge, ISBN-10: 026255030X, pp: 447.

Hanke, T., 2004. HamNoSys-representing sign language data in language resources and language processing contexts. Proceedings of the Workshop on the Representation and Processing of Sign Languages, (PSL' 04), ELRA, Paris pp: 1-6.

Hopfield, J.J., 2008. Searching for memories, sudoku, implicit check bits and the iterative use of not-always-correct rapid neural computation. J. Neuron Computation, 20: 1119-1164.

http://www.parentcenterhub.org/wp-content/uploads/repo_items/fs3.pdf

http://www.who.int/pbd/deafness/hearing_impairment_grades/en/

https://www-03.ibm.com/press/us/en/pressrelease/22316.wss

Jaballah, K. and M. Jemni, 2013. A review on 3D signing avatars: Benefits, uses and challenges. Int. J. Multimedia Data Eng. Manag., 4: 21-45. DOI: 10.4018/jmdem.2013010102

Jordan, I.M., 1986. Serial Order: A Parallel Distributed Processing Approach. University of California, Institute for Cognitive Science.

Kennaway, R., J. Glauert and I. Zwitserlood, 2007. Providing signed content on the internet by synthesized animation. ACM Trans. Comput. Human Interaction. DOI: 10.1145/1279700.1279705

Kipp, M., A. Heloir and Q. Nguyen, 2011. Sign language avatars: Animation and comprehensibility. Proceedings of the International Workshop on Intelligent Virtual Agents, (IVA' 11), pp: 113-126.

Latha, C.P. and M. Priya, 2016. A review on deep learning algorithms for speech and facial emotion recognition. J. Comput. Sci. Information Technol., 1: 88-104.

LeCun, Y., Y. Bengio and G. Hinton, 2015. Deep learning. Nature, 521: 436- 444. DOI: 10.1038/nature14539

Lincoln, M., S.J. Cox and M. Nakisa, 2001. The development and evaluation of a speech to sign translation system to assist transactions. Int. J. Human Comput. Stud.

Mohamed, S.S., M.A. Abdou and Y.F. Hassan, 2016. A cascaded speech to Arabic sign language machine translator using adaptation. Int. J. Comput. Applic.

Ota, K., M.S. Dao, V. Mezaris and F.G.B.D. Natale, 2017. Deep learning for mobile multimedia: A survey. ACM Trans. Multimedia Comput. Communications Applic. DOI: 10.1145/3092831

San-Segundo, R., J.M. Montero, R. Córdoba, V. Sama and F. Fernández *et al*., 2012. Design, development and field evaluation of a Spanish into sign language translation system. Pattern Anal. Applic., 15: 203-224.

Schaul, T., J. Bayer, D. Wierstra, Y. Sun and M. Felder *et al*., 2010. PyBrain. J. Mach. Learn. Res.

Strobl, E.V. and S. Wisweswaran, 2013. Deep multiple kernel learning. Proceedings of the 12th International Conference on Machine Learning and Applications, Dec. 4-7, IEEE Xplore Press, Miami, pp: 414- 417. DOI: 10.1109/ICMLA.2013.84

Wilbur, R., 1997. American Sign Language: Linguistic and Applied Dimensions. 2nd Edn., College-Hill, Boston.

Zhang, Q., M. Wu, Z. Luo and Y. Chen, 2014. Enhanced non-linear features for on-line handwriting recognition using deep learning. Proceedings of the International Conference on Neural Information Processing, (NAP' 14, pp: 358-365).