International Journal of Engineering & Technology, 7 (4.10) (2018) 15-18

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Critical Decision Making Using Neural Networks

Rajat Bhati, Shubham Saraff, Chhandak Bagchi, V. Vijayarajan*

School of Computer Science and Engineering, VIT University, Vellore, Tamil Nadu, India.
**Corresponding author E-mail: vijayarajan.v@vit.ac.in*

Abstract

Decision Making influenced by different scenarios is an important feature that needs to be integrated in the computing systems. In this paper, the system takes prompt decisions in emotionally motivated use-cases like in an unavoidable car accident. The system extracts the features from the available visual and processes it in the Neural network. In addition to that the facial recognition plays a key role in returning factors critical to the scenario and hence alter the final decision. Finally, each recognized subject is categorized into six distinct classes which is utilised by the system for intelligent decision-making. Such a system can form the basis of dynamic and intelligent decision-making systems of the future which include elements of emotional intelligence.

Keywords: *Artificial Intelligence, Neural Network, Decision Making, Emotional Intelligence, Facial Recognition, Computer Vision*

1. Introduction

This paper proposes a method for intelligent decision-making using neural networks where an abstract level of emotional intelligence has also been incorporated. A specific set-case is adopted where an emotionally influenced decision has to be taken where a car accident is bound to happen with 'n' different people standing on the road, in the path of the car. The system takes a quick decision and results with a priority list of whom to save first. This intelligent decision taken by the car system enables more effective decisions to be taken by the computer in critical times.

The AWS Rekognition is used to detect labels in the image seen by the car. The faces are segmented and features are extracted from the face. The segmented faces are matched with the faces of known people that are stored in the database. If there is a match a very high value is sent as a feature to the neural network. If there are no matches, then a value of zero is sent as a feature.

The neural network takes as input the age, gender, emotions of each person as well as the feature that is generated after comparing the faces. The neural network classifies each input into six classes ranging from 0 to 5 where 0 has the highest probability to be saved and 5 has the least probability of being saved. The high value of the compare faces feature reduces the probability of the person getting hit by the car as the class number is seen to decrease in each case.

2. Related Works

A feedforward neural network[1] has layers of fully connected neurons. Neurons are the building blocks of neural networks and each neuron has an activation function[2] associated with it. The output of the activation function of each neuron is given as an input to the next layer of neurons. Each layer is fully connected, i.e., all the neurons of one layer are connected to all the neurons of the immediate next layer. Feedforward networks are different from other neural networks as the connections do not form cycles in a feedforward network. Each connection between two neurons has a weight associated with it which determines the effect of the output of the previous layer of neurons on the next layer of neurons. Information flows in a feedforward network only in the forward direction, i.e., information flows from the input layer to the successive hidden layers and finally goes to the output layer.

The learning process in a neural network can be achieved using different optimization algorithms like backpropagation algorithm and particle swarm optimization[3-5]. The aim of any optimization algorithm is to minimize the error function. This is achieved through gradient descent in which the weights of the neural network are changed until we reach the minima of the error function.

The backpropagation algorithm was proposed in 1970 but it was not recognized until 1986 in a paper by Geoffrey Hinton, David Rumelhart, and Ronald Williams[6]. This paper dealt with applying backpropagation to neural networks and showed better results than previously known learning algorithms. Thus, it was possible to solve problems that were previously deemed as unsolvable. After that, it has been used to train neural networks to solve a vast number of problems.

The objective of the proposed problem is to minimize the cost function C. In other words we have to find the position in the vector space where the partial derivative of the cost function

We perform gradient descent using the backpropagation alg (1)

The activation function we use here is rectified linear unit (ReLU) [7]:

$$\sigma(x) = \mathrm{Max}(0, x) \tag{1}$$

The cost function C is defined by:

$$C = \frac{\Sigma(a_j^{(L)} - y_j)^2}{2m} \tag{2}$$

Where $a_j^{(L)}$ is the output of the final layer neural network, y_j is the target output and m is the number of samples in the dataset.

In the forward pass, we calculate:

$$x_j^l = \sigma(\sum_{k=0}^{n} w_{jk}^l \cdot x_k^{l-1}) \tag{3}$$

Where a_jl represents the j^{th} neuron in the l^{th} layer, $w_j$kl represents the weight going to the kth neuron from the j^{th} neuron in the l^{th} layer.
In the backward pass, we calculate the error δ:

$$\delta_j^L = (a_j^L - y_j).\, \sigma'(a_j^L) \tag{4}$$

Where,

$$\sigma'(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases} \tag{5}$$

$$\delta_i^{l-1} = \sigma'(a_i^{l-1}) \sum_{j=1}^{d(l)} (w_{ij}^l \cdot \delta_j^l) \tag{6}$$

Next, the weights are updated using the formula

$$w_{ij}^l := w_{ij}^l - \alpha\, a_i^{l-1}\, \delta_j^l \tag{7}$$

where α is the learning rate.
The forward and backward pass is repeated till a maximum number of epochs are reached or the error is lesser than a predefined minimum error value. Small (3Å~3) convolution filters based architecture reflect considerable progress compared with previous configurations and can be achieved by pushing depth to 16-19 weight layers [8]. The representation known as the ConvNet delivers state-of-the-art results in multiple datasets.
The generic framework for ontology based image retrieval discussed in [9] used the natural language to describe the image semantics and performed the retrieval based on automatic generation of SPARQL for the given query. From this we clearly understood that the general content based image retrieval methods lack in identifying and recognizing images with object semantics interpreted by human brain.
A major setback in the facial recognition benchmark datasets is selecting facial imagery using commodity face detector. This limits the capabilities of recognition of the faces in different postures and other compounding factors. The IARPA Janus Benchmark A (IJB-A) [10] with manually localized 500 images of faces has wider geographic variation while it can jointly use the face recognition and detection benchmarking in full pose variations. These features clubbed with other functions of the dataset lead to this accurate unconstrained benchmark.
Facial recognition from a single picture or from a set of face being tracked in a video footage using Convolutional Neural Networks (CNN) and large scale datasets for training, end to end learning has been achieved [11] . This mechanism also checks the tradeoff between the time and data purity while it improves it results compared to traditional benchmarks like LFW and YTF face benchmarks.
Image recognition software's like AWS Rekognition have led the research in this sector of Computer Vision. There are two groups in which the AWS Rekognition operates the computer vision API [12]:
1. Non-Storage API – Servers do not store any data from the API Operation. It simply provides the result to the input image without any storage on the server. The tasks that the API can conduct is as follows:
a. Detect labels or faces in an image.
b. Compare faces
c. Celebrity Face Detection
d. Analyse image sentiment/content
2. Storage API – In this operation the input facial feature vectors are stored a database on the server which can be used to search the database for any facial matches. However, the input image bytes are not stores in the database.

Along with Computer Vision, Emotional Intelligence also plays a crucial role in decision making. Emotional Intelligent can be defined as "the capacity to reason about emotions to enhance thinking. An array of noncognitive capabilities, competencies, and skills that influence one's ability to succeed in coping with environmental demands and pressures" [13] . According to studies, most recent works in the field of emotional intelligence have been based on "verbal, non-verbal, and textual thoughts, and also through speech and gesture".

3. Proposed Model

All In this paper, we are taking a specific example of who to save in case a car is going to have an unavoidable accident. We take a case where the car is going to have an accident and there are people on the road, so it has to take a spontaneous decision about which individual it wants to save and who has to be inevitably be hit by it.

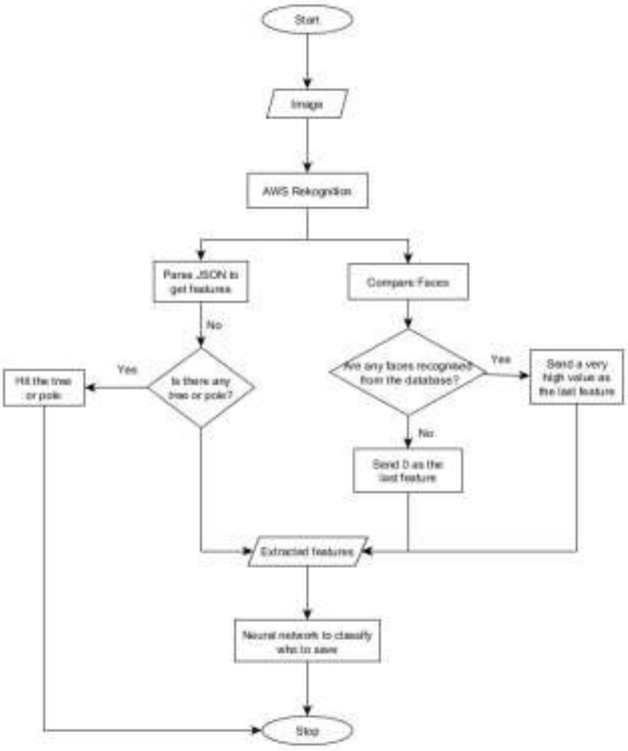

Start
Image
AWS Rekognition
Parse JSON to get features
Compare Faces
No
Yes
Hit the tree or pole
Is there any tree or pole?
Are any faces recognised from the database?
Yes
Send a very high value as the last feature
No
Send 0 as the last feature
Extracted features
Neural network to classify who to save
Stop

Fig.1: The proposed model

The dataset to train the neural network contains 15,000 tuples collected from individuals from an internet based survey. The first five features of the dataset are age, gender (0 for male, 1 for female), happy, sad and angry. The sixth feature is initially set to zero for all tuples.

Table 1: Dataset without Compare Faces feature

Age	Gender	Happy	Sad	Angry	Compare Faces	Category
86	1	1	1	0	0	5
7	1	1	0	1	0	0
42	0	0	1	1	0	3
75	0	0	0	1	0	5

Individuals were asked shown an image containing a diverse group of people of all ages and genders and asked who would they want to save in case of a car accident.. They classified each individual into any of the six categories ranging from 0 to 5, where 0 meant most likely and 5 meant least likely to save.

To prove the premise that a person is more willing to save someone he knows, irrespective of their age, gender and other features, over a stranger, data was collected using an internet based survey where people were asked in case they need to give a lift to someone, who would they prefer, a friend, a relative or a stranger. 98% of the people responded that they would save a friend or a stranger. Next, this data was used to augment our dataset where 5000 tuples were randomly selected and gave their sixth feature a very high value (of the order 104) and reduced their category number meaning that they would have a higher probability of getting saved. These tuples represented the friends or relatives who need to have the highest probability to be saved.

Table 2: Dataset with Compare Faces feature

Age	Gender	Happy	Sad	Angry	Compare Faces	Category
75	1	0	1	0	1000	2
50	0	0	0	1	0	1
16	0	1	1	0	1000	0
88	1	1	0	1	1000	2

In our proposed model, the image of the road taken from the camera attached to the front of the car is sent to the AWS Rekognition system. The API call to the AWS Rekognition system performs object detection and sends a JSON (JavaScript Object Notation) file back which contains labels containing data about the entire image, containing information about the environment, for eg., if there are trees and houses in the image, and also whether there are people on the scene. It sends another JSON that has information about the details derived from the faces of the individuals present in the image if there are any. It contains information like the age, gender and emotions (happy, sad and angry) of each face.

In our algorithm, if we see a tree or pole as a label, we take a rule-based decision to hit that and save any human on the scene. If there are no trees or poles, but there are people on the road, we extract the information like age, gender and emotions from their faces and send it to our neural network to classify whom to save.

The Compare Faces function from the AWS Rekognition Suite is used to match faces of the people on the road to the faces in the database. The database contains faces of people that we want to save in case of an accident, for e.g., friends and family of the owner of the car. If there is a match between the query image of the people on the road to any image in the database, a very high value (of the order 104) is sent as a feature to the neural network. If there are no matches, then a value of zero is sent as a feature.

The extracted features from the facial analysis function (age, gender, happy, sad, angry) are concatenated with the feature sent from the Compare Faces function (a high value or zero) to form the feature vector. This is sent to the neural network which classifies whom to save in such a situation.

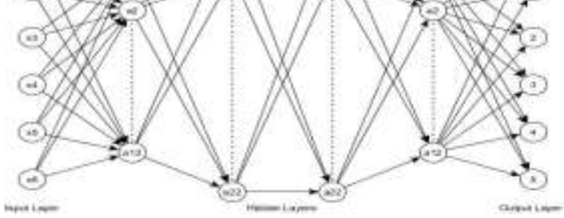


Fig. 2: The proposed model's neural network architecture

$$\sigma(x) = \frac{2}{1 + e^{-2x}} - 1$$

The neural network takes as input six features (age, gender, happy, sad, angry and compare faces factor). The neural network has four hidden layers containing 12, 22, 22 and 12 neurons respectively. Fig. 2 shows the architecture of the neural network. The activation function of choice is Rectified Linear Unit (ReLU). The activation function used in the output layer is hyperbolic tangent (tanh).

The neural network feeds the data forward through the layers and classifies each input into six categories. The person or people getting lower values will get saved and the one getting the highest value will have to inevitably be hit by the car. It is trained using backpropagation. Ten neural networks were created randomly sampling 5000 tuples from the dataset for each neural network. The results of the neural networks are then compared

4. Results and Discussion

Detailed Random sampling is done from the dataset and 5000 tuples are chosen to train the neural network. We use heavy L1 L2 regularization and Dropout to avoid overfitting on the network. This process is repeated ten times and ten neural networks are generated. The average accuracy is 96.37%.

Table 3: Accuracies of ten neural networks

Neural Network Number	Accuracy
1	95.6462
2	96.1031
3	94.9412
4	96.3772
5	97.6697
6	96.8221
7	95.731
8	95.8877
9	97.7415
10	96.8276

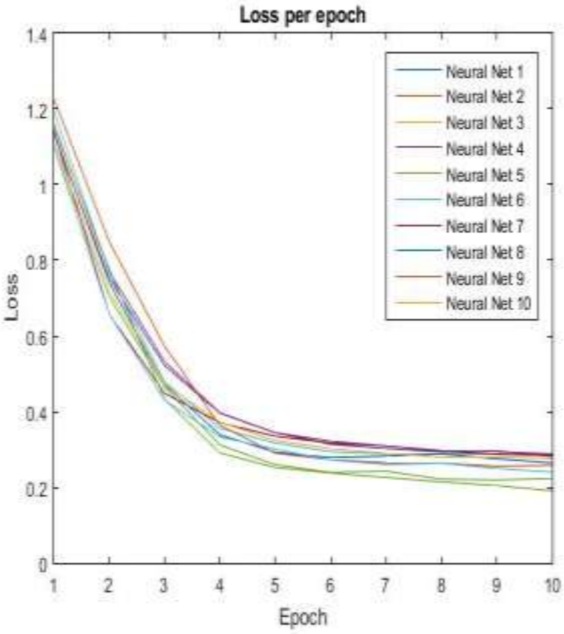


Fig. 3: Loss function curve for ten neural networks

Given an image, the algorithm extracts all the labels from the scene and gets all the facial features. The faces are compared to the known faces in the database. If there is a match, then a high value is sent as the last feature, else zero is sent. If a value of 1000 is sent as the last feature, the class of the output comes down, which means the probability of saving that person is the higher.

Table 4: The contrast when person is recognized

Age	Gender	Happy	Sad	Angry	Class	
					Compare Faces=0	Compare Faces=1000
5	1	1	0	1	0	0
24	1	1	0	1	1	0
75	0	0	0	0	5	1
42	0	1	0	0	3	0
60	1	0	1	1	4	0

We use this forest of neural nets and poll them against an input image and select the most selected answer. Now let us consider our scenario where a random group of people are in front of the car (Fig 4).

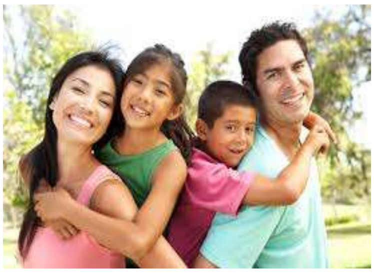


Fig. 4: Input image

Now we will analyze the number of faces and generate their respective tuples and later crop the faces (Fig 5).

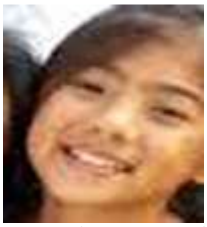
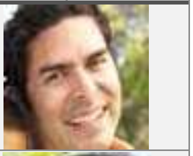
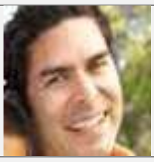
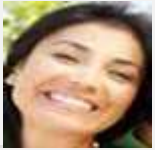
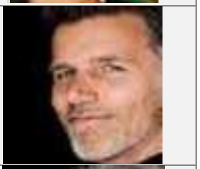
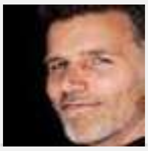
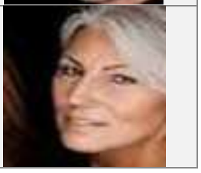


Fig. 5: One of the Cropped faces

We check whether the machine recognizes anyone of the given present faces and provides the compared faces variable value accordingly. Later we put the final tuples in the neural network to determine the order of priority of saving.

Table 5: Sample results

Input Image	**Recognized person**	**Least likely to be saved**
	None (no one recognized)	
		
	None (no one recognized)	
		

Hence, we were successfully able to replicate critical decision making in an emotionally influenced scenario.

5. Conclusion

Hence, we were successfully able to replicate critical decision making in an emotionally influenced scenario. This marks that a machine can replicate the emotionally influenced decisions if it is provided with the proper tools to have features which a human brain can use.

In our case we have provided our machine to influence its emotional decisions only on the basis of memory and have succeeded in mapping a simple scenario which proves that with appropriate tools machines can take complex decisions which involve an emotional aspect.

References

[1] Maind, Sonali B., and Priyanka Wankar. "Research paper on basic of artificial neural network." International Journal on Recent and Innovation Trends in Computing and Communication 2.1 (2014): 96-100.

[2] Leshno, Moshe, et al. "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function." Neural networks 6.6 (1993): 861-867.

[3] Shi, Yuhui. "Particle swarm optimization: developments, applications and resources." evolutionary computation, 2001. Proceedings of the 2001 Congress on. Vol. 1. IEEE, 2001.

[4] Kennedy, James. "Particle swarm optimization." Encyclopedia of machine learning. Springer US, 2011. 760-766.

[5] Poli, Riccardo, James Kennedy, and Tim Blackwell. "Particle swarm optimization." Swarm intelligence 1.1 (2007): 33-57.

[6] Rumelhart, David E., et al. "Backpropagation: The basic theory." Backpropagation: Theory, architectures and applications (1995): 1-34.

[7] Dahl, George E., Tara N. Sainath, and Geoffrey E. Hinton. "Improving deep neural networks for LVCSR using rectified linear units and dropout." Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013.

[8] Karen Simonyan and Andrew Zisserman. “Very Deep Convulation Netwroks for large-Scale Image Recognition.” ICLR 2015 Conference Paper (2015).

[9] Vijayarajan, V., M. Dinakaran, Priyam Tejaswin, and Mayank Lohani. "A generic framework for ontology-based information retrieval and image retrieval in web data." Human-centric Computing and Information Sciences 6, no. 1 (2016): 18.

[10] Brendan F. Klare, Ben Klein, Emma Taborsky, Austin Blanton, Jordan Cheney, Kristen Allen, Patrick Grother, Alan Mah and Anil K. Jain. “Pushing the Frontiers of Unconstrained Face Detection and Recognition: IARPA Janus Benchmark A.” Computer Vision Foundation 2015 Paper, IEEE Xplore (2015): 1931-1939.

[11] Omkar M. Parkhi, Andrea Vedaldi and Andrew Zisserman. “Deep Face Recognition.” Visual Geometry Group, Department of Engineering Science, University of Oxford (2015)

[12] Amazon Rekognition Developer Guide. “How it Works.” (2017): 2-4

[13] Syeda Erfana Zohora, A.M.Khan, A.K. Srivastava, Nhu Gia Nguyen and Nilanjan Dey. “A Study of the State of the Art in Synthetic Emotional Intelligence in Affective Computing.” Researchgate Article (2016).

[14] Images rights reserved by https://eadventistnews.com/tag/families/