

A Job Recommendation System For Differently Abled Using Neural Networks

Shyam P. Joy^a, Ashwin M S^b, Sayan Pal^c, Sai Teja Burla^d, Prakhyath Jain^e, Sai Priyadarshini^f, George Sebastian^g

^a Associate Professor, Department of Computer Science and Engineering,
CMR Institute of Technology, Bengaluru-560037, Karnataka, India. Email Id:
shyampjoy@gmail.com

^{b,c,d,e,f}UG Students, Department of Computer Science and Engineering, CMR Institute of Technology, Bengaluru-560037,
Karnataka, India.

^gHead of EI Labs, Enable India, Bengaluru, Karnataka, India.

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 28 April 2021

Abstract: Empowering differently abled is not only a good social cause, but also contributes towards the country's Gross Domestic Product. This article presents a software application which recommends a job suitable to the differently abled with due consideration to the type and degree of disability. A Neural Networks (NN) model is used to develop the software application. The data set for the same was provided by Enable India, a Non-Government Organisation, involved in empowering the differently abled. The application performed with an accuracy of 87%.

Keywords: job recommendation system, differently abled, Neural Networks.

1. Introduction

According to the World Health Organisation (WHO), about 15% of the world population are differently abled and among them, 2.4% experience severe difficulties in functioning [1]. Empowering differently abled is not only a good social cause, but also contributes towards the country's Gross Domestic Product. Constitutions of some countries assure the differently abled with the right to work. Therefore, employing the differently abled is important for the social and economic growth of a country.

Finding the most appropriate job for a differently abled person is more challenging than that of an ordinary person, because the different types of disabilities and the variations in the severity of each disability, increases the combinatorial complexity. Another challenge is the availability of data on Employment of differently abled.

This article presents a software application which takes a set of disabilities as input and recommends a ranked list of potential jobs; a differently abled person can take up. The software is based on a supervised machine learning model.

The model was trained using data provided by Enable India. Enable India (EI) is a Non-government Organisation (NGO) founded by Dipesh Sutariya and Shanti Raghavan. The organisation enables differently abled people by way of training and employment. True to the objective, forty percent of the employees of EI are differently abled [2].

An alternate approach to solve the problem is to decide jobs by applying logic rules on attributes, using Statistical Query Language (SQL). But such an approach would be limited by the imagination of experts in the field of Human Resources, as the queries will be designed as per the direction of

Experts. Machine learning can overcome this limitation and will be able to realise a more generic model. Another advantage of machine learning approach is that it can easily adopt changes when compared to SQL based approaches. Moreover, machine learning approaches output a set of recommended job rankings from the most suitable job to the least suitable job.

The article is organised as follows: The following section presents a brief survey of the literature on applications which classify data using NN. Section 3 presents the steps in developing the model, which includes pre-processing of data and developing the NN model. Section 4 presents the details of implementation and the last section present the results obtained on testing the model.

2. Literature Review

The problem addressed in this manuscript can be generalised as a classification problem. The candidates seeking jobs are classified on the basis of their features. NN has been applied in a number of classification problems, some of which are discussed below.

One of the prominent areas of application of classification using NN is in medicine, especially in cancer research [1, 2, 3, 4]. Other areas of medicine have also used NN for classification [5, 6, 7, 8, 9]. NN have also been used to study human attention patterns while viewing large images and the same has been validated using handwritten digit images [10]. Studies to recognise gender and handedness from handwriting, using NN, have also been done [11].

NN have also been widely used in analysing signals. Different types of sound signals have been analysed using NN [12, 13, 14]. Analysis of images and videos using NN were applied in different disciplines [15, 16, 17, 18]. NN were used in detecting appliances in home appliances control system [19].

Although, NN based classification has been applied in multiple disciplines, the same has not been applied to the classification of differently abled from the perspective of jobs. The closest, was a NN based classification of industrial jobs with respect to risk of low back disorders [20]. The following Section presents the development of the proposed software application.

3. Methodology

The discussions in the previous section leads us to the problem which can be stated as follows:

Given a database on differently abled persons with details of their disability, educational qualification and a recommended job type, develop a software model which will recommend a set of most appropriate job for a differently abled person. The recommended job type is decided on the basis of a personal interview and psychometric test. The data is used to develop a model using machine learning algorithms.

Machine Learning is defined in [21] as follows:

“The study of computer algorithms that allow computer programs to automatically improve through experience.”

The experience used by Machine Learning algorithm is in the form of data.

3.1. The data

The data used to train the model was provided by EI. Similar datasets are rare in the literature. Most of the data sets available, presented the statistics on a particular disability [22, 23, 24, 25, 26, 27].

The data consisted of the profile of differently abled individuals sans their personal identities. The features consisted of the details of disabilities of individuals, their educational qualifications and the recommended job type. The recommended job type was decided based on an interview and a psychometric test. The data was pre-processed by removing records with missing attributes. After pre-processing, the data consisted of 5742 records of differently abled persons, out of which 3847 records were used for training and 1895 records were used for testing. Dimension of the data was reduced on the basis of expert advice and the features that were used to train the model were Disability type, Disability sub type, Highest education, Recommended job types. The target label was Job role name.

These features were categorical data containing label values rather than numerical values. Since machine learning algorithms cannot operate on categorical data, they need to be converted to numerical values.

Therefore, One-Hot encoding was used to convert the labelled data to numerical values. After One-Hot encoding, the input feature resulted in a matrix of dimension 5742×129 and the target label was a matrix of dimension 5742×106 . The data is then used for training a model.

On inspecting the data, it was observed that the mapping of input features to target label is not uniform throughout the dataset. The number of records which map to certain target labels were found to be more than others. To normalise this distribution, Huber Loss Function was used.

3.1.1. Huber Loss Function

While mapping the input feature to target labels, it was observed that the data was skewed. When Mean Absolute Error (MAE) function was used, the prediction tended to converge towards the median values, without considering outliers. While using Mean Squared Error (MSE), the prediction tended to converge towards outliers. Since both results were unacceptable, Huber loss

Function was used.

Huber loss provides a good balance between MSE and MAE. In simple terms, Huber loss function behaves as MAE, when the value of error is high. It behaves as MSE when the error is small. Hence Huber loss function gives best of both loss functions. Since the data set was skewed, Huber loss function was used.

More formally, let $f(x)$ represent the estimated value of an actual target value y , and δ represents the upper bound on the error, then the Huber Loss function, L_δ is given by the following equation:

$$L_\delta = \frac{1}{2} \cdot (y - f(x))^2 \text{ for } |y - f(x)| \leq \delta \text{ and } L_\delta = \delta \cdot |y - f(x)| - \frac{1}{2} \delta^2 \text{ otherwise}$$

In the following section, the machine learning model is developed.

3.2. The model

In this article, the software was developed using a NN model. NN mimics the way the human brain learns and consists of a network of artificial neurons. An artificial neuron takes one or more inputs and produces an output. Neurons are arranged in layers. The layer of neurons which receives inputs is called input layers. The output of input layer is fed into one or more hidden layers of neurons. The output of the hidden layer is fed as input to neurons in the output layer.

The inputs to each of the neurons are a weighted function of the output of the neurons in the previous layer. Training the neural network involves setting these weights, so as to minimise the error between the predicted and the actual values.

The process of building NN involves tuning of multiple hyperparameters. Hyperparameters either define the structure of the NN or define the parameters concerned with training. Hyperparameters concerned with NN

structure are the depth of the NN, Dropout, Network Weight Initialisation and Activation Function. These terms are explained in the following paragraphs.

Number of Layers: Depth of the NN defines the number of layers in the network.

Dropout: After training, a NN output tends to closely follow the training data, i.e. overfit the training data. This will result in a less generic model and introduce large error during testing. To reduce the effect of overfitting, random nodes of the network are dropped during training. This is called dropout.

Initialisation of weights: The network weights have to be initialised such that the activating values do not increase or decrease exponentially.

Activation function: Activation functions are attached to each neuron in NN. Their function is to get activated if the input is relevant in making the prediction. In this application ReLU activation function was used for intermediate layers as it is computationally efficient and allows for backpropagation. Softmax was used for output layer as it produces multiple outputs for an input array.

Hyperparameters related to training the NN are Learning Rate, Momentum, Number of epochs and Batch Size, Loss Function and Optimizer. A brief explanation of these parameters are given below:

Learning Rate: Learning rate controls the time taken to train a NN model. A small value would result in longer time to learn. A larger learning rate would mean less time to learn, but may not achieve optimal weights.

Momentum: Larger learning rates are likely to induce oscillations across the optimal point. To avoid oscillations, it is necessary that the direction of change is the same as the previous change with respect to the optimal point. This is achieved by the factor namely momentum which improves the speed of convergence.

Size: Batch size is an integer which represents the number of rows of training data to be used to determine the updates to the model.

Number of epochs: Number of epochs is the number of times the algorithm iterates over the training dataset.

4. Implementation

The model was implemented in Keras with a Tensor Flow platform. Tensor Flow provides an open source software library for machine learning implementation.

Keras is a high level interface for development of NN in Python. Keras contains implementations of commonly used NN building blocks such as layers, objectives, activation functions, optimizers and tools to simplify the coding.

The optimal hyper parameters and the network architecture were decided by an experiment which involved iterating over the set of architectures given below: For all the iterations, Adam optimiser was used with moment set to default values.

1. Network architecture with values:

- 1 hidden layer with 250 neurons represented by [250]
- 2 hidden layers having 250 and 500 neurons [250,500]
- 3 hidden layers having 400 neurons each [400,400,400]
- 4 hidden layers having 250 neurons each [250,250,250,250]

2. Epoch values were 25,50 ,75

3. Test size specifies the percentage of data used for testing. The values were 0.2,0.3,0.4.

The value δ was set to 1.0 in loss function, which is the default value in Keras Library. The weights were initialised to random normal initialisers. The learning rate was set to 0.001 and batch size was set to 128, which is constant for all architectures.

Metric	Architecture	Epochs	Test Size	Score
top k categorical accuracy	[250, 500]	75	0.2	0.942
precision	[250, 250, 250, 250, 250]	50	0.4	0.902
recall	[250, 500]	75	0.2	0.543
lowest loss	[250, 250, 250, 250, 250]	75	0.2	0.002

Table 1: Best results from experiments on training data

Metric	Architecture	Epochs	Test Size	Score
top k categorical accuracy	[250]	25	0.3	0.877
precision	[250, 250, 250, 250, 250]	50	0.4	0.792
recall	[250, 250, 250, 250, 250]	25	0.3	0.455

lowest loss	[250, 250]	75	0.2	0.002
-------------	--------------	----	-----	-------

Table 2: Best results from experiments on test data

The results of the experiment can be found in tables 1 and 2. Table 1 shows the results of experiments when applied to training data. The best accuracy achieved was 94.2 %. Table 2 shows the results of experiments when applied to test data. The best accuracy achieved was 87.7 % , which is close enough to the results achieved using training data.

Therefore, the model with the best performance is one with a single hidden layer network with 250 neurons, activated by ReLU and with a dropout of 0.5. We train this network with the Adam optimizer for 25 epoch, with a test size of 0.3, to achieve a test top k categorical accuracy of 87.7%.

5. Results

Figure 1 shows the comparison of performance of the model with respect to the training data and test data. It can be observed that the accuracy of prediction with unseen test data as the input, indicated by the dotted line, is close to the accuracy of prediction with training data, indicated by the bold line.

Figure 1 also shows the precision and recall values for the proposed model when tested with data not seen by the model. It may be noted that the proposed model is a high- precision low-recall model. This indicates that predictions with this model might have higher percentage of false negatives relative to false positives. This implies that the chances that the model predicts a set of jobs that a candidate cannot undertake, is low.

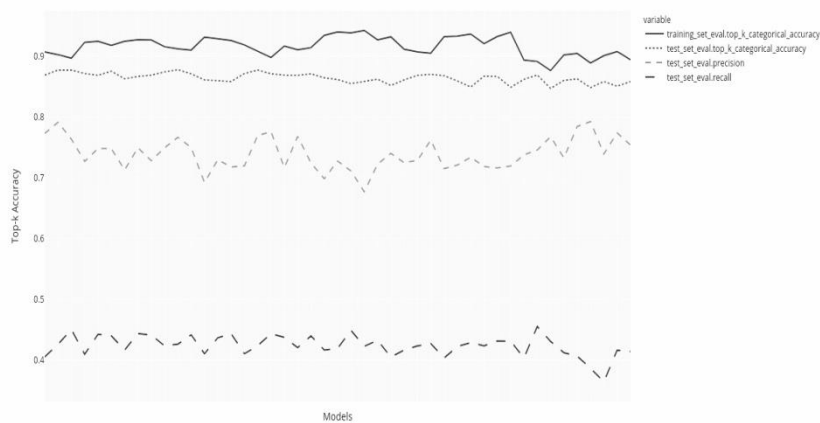


Figure 1: Comparison of performance of the model with respect to training and test data

6. Conclusion

In this article, a software application for recommending a set of most appropriate jobs to the differently abled is developed. The job recommendation takes the type and severity of disabilities besides educational qualification for the candidates. The application is based on Neural Network model which has been trained on data provided by Enable India, which is an organisation working towards empowering the differently abled. The software results provided an accuracy of 87.7%.

References

1. Rehman, M. A. Khan, T. Saba, Z. Mehmood, U. Tariq, N. Ayesha, Microscopic brain tumor detection and classification using 3d cnn and feature selection architecture, *Microscopy Research and Technique* 84 (1) (2021) 133–149.
2. T. Ghanshala, V. Tripathi, B. Pant, An efficient image-based skin cancer classification framework using neural network, in: *Research in Intelligent and Computing in Engineering*, Springer, 2021, pp. 851–858.
3. T. Fujioka, Y. Yashima, J. Oyama, M. Mori, K. Kubota, L. Katsuta, K. Kimura, E. Yamaga, G. Oda, T. Nakagawa, et al., Deep-learning approach with convolutional neural network for classification of maximum intensity projections of dynamic contrast-enhanced breast magnetic resonance imaging, *Magnetic Resonance Imaging* 75 (2021) 1–8.
4. Ghoneim, G. Muhammad, M. S. Hossain, Cervical cancer classification using convolutional neural networks and extreme learning machines, *Future Generation Computer Systems* 102 (2020) 643–649.
5. L. Alzubaidi, M. A. Fadhel, S. R. Oleiwi, O. Al-Shamma, J. Zhang, Dfu qutnet: diabetic foot ulcer classification using novel deep convolutional neural network, *Multimedia Tools and Applications* 79 (21) (2020) 15655–15677.
6. Xiao, Y. Xu, X. Bi, J. Zhang, X. Ma, Heart sounds classification using a novel 1-d convolutional neural network with extremely low parameter consumption, *Neurocomputing* 392 (2020) 153–159.
7. Z. Yan, J. Zhou, W.-F. Wong, Energy efficient eeg classification with spiking neural network, *Biomedical Signal Processing and Control* 63 (2021) 102170.

9. N. Michielli, U. R. Acharya, F. Molinari, Cascaded lstm recurrent neural network for automated sleep stage classification using single-channel eeg signals, *Computers in biology and medicine* 106 (2019) 71–81.
10. S. Raghu, N. Sriraam, Y. Temel, S. V. Rao, P. L. Kubben, Eeg based multi-class seizure type classification using convolutional neural network and transfer learning, *Neural Networks* 124 (2020) 202–212.
11. S. Kumari, S. Aravindakshan, U. Jain, V. S. Chakravarthy, Convolutional elman jordan neural network for reconstruction and classification using attention window, in: *Innovations in Computational Intelligence and Computer Vision*, Springer, 2021, pp. 173–181.
12. M. Rahmanian, M. A. Shayegan, Handwriting-based gender and handedness classification using convolutional neural networks, *Multimedia Tools and Applications* (2021) 1–24.
13. Paseddula, S. V. Gangashetty, Late fusion framework for acoustic scene classification using lpcc, scmc, and log-mel band energies with deep neural networks, *Applied Acoustics* 172 (2021) 107568.
14. M. Khishe, M. Mosavi, Classification of underwater acoustical dataset using neural network trained by chimp optimization algorithm, *Applied Acoustics* 157 (2020) 107005.
15. Z. Mushtaq, S.-F. Su, Environmental sound classification using a regularized deep convolutional neural network with data augmentation, *Applied Acoustics* 167 (2020) 107389.
16. Y. Zhong, F. Fei, Y. Liu, B. Zhao, H. Jiao, L. Zhang, SatCNN: satellite image dataset classification using agile convolutional neural networks, *Remote Sensing Letters* 8 (2) (2017) 136–145.
17. X. Wang, J. Jiao, J. Yin, W. Zhao, X. Han, B. Sun, Underwater sonar image classification using adaptive weights convolutional neural network, *Applied Acoustics* 146 (2019) 145–154.
18. M. A. Khan, T. Akram, M. Sharif, M. Y. Javed, N. Muhammad, M. Yasmin, An implementation of optimized framework for action classification using multilayers neural network on selected fused features, *Pattern Analysis and Applications* 22 (4) (2019) 1377–1397
19. Giveki, M. Karami, Scene classification using a new radial basis function classifier and integrated sift-lbp features, *Pattern Analysis and Applications* (2020) 1–14.
20. P. K. Shukla, R. K. Chaurasiya, S. Verm a, Performance improvement of p300-based home appliances control classification using convolution neural network, *Biomedical Signal Processing and Control* 63 (2021) 102220.
21. J. Zurada, W. Karwowski, W. S. Marras, A neural network-based system for classification of industrial jobs with respect to risk of low back disorders due to workplace design, *Applied Ergonomics* 28 (1) (1997) 49–58.
22. T. Mitchell, *Machine Learning*, McGraw Hill, 1997.
23. (2020 (accessed October 28, 2020)). [link]. URL <https://dataworld/datasets/disabilities>
24. (2020 (accessed October 28, 2020)). [link].
25. URL <https://www.cdc.gov/ncbddd/disabilityandhealth/datasets.html>
26. (2020 (accessed October 28, 2020)). [link].
27. URL <https://www.disabilitystatistics.org/sources.cfm>
28. (2020 (accessed October 28, 2020)). [link].
29. URL <https://www.who.int/disabilities/data/en/>
30. (2020 (accessed October 28, 2020)). [link].
31. URL <https://researchguides.library.syr.edu/disabilitystats>
32. (2020 (accessed October 28, 2020)). [link].
33. URL <https://www.ons.gov.uk/peoplepopulationandcommunity/healthandsocialcare/healthandsocialcare/disability/datasets/prevalenceandemployment>