International Journal of Computer Trends and Technology
ISSN: 2231-2803 / https://doi.org/10.14445/22312803/IJCTT-V68I3P117

Volume 68 Issue 3, 84-88, March 2020


*Original Article*

# Computer Science Career Recommendation System Using Artificial Neural Network

Brijmohan Daga[1], Juhi Checker[2], Anne Rajan, Sayali Deo[3]

[1,2,3] *Department of Computer Engineering, Fr. Conceicao Rodrigues College of Engineering, Bandra West, Mumbai-400050, India*

Received Date: 18 February 2020
Revised Date: 02 April 2020
Accepted Date: 05 April 2020

***Abstract*** - *There is a trend amongst students to generally opt for career paths based on either the choices of their colleagues or the highest salary paying roles. They fail to know their strengths and choose their career randomly, which leads to frustration and demoralization. Moreover, while recruiting the candidates, recruiters need to assess them in all different aspects. Thus, there is a need for a system that helps students decide a job role that is best suited for him/her, which is based on /her skill-set and other evaluation metrics, which is now possible due to advancements in the field of deep learning. This paper proposes an automated system using an Artificial Neural Network which considers personality traits of the individual along with personal interests and academics to predict which computer science job role would be best suited for them.*

***Keywords*** - *Deep learning, Artificial neural networks, Multiclass classification, Backpropagation algorithm, Career recommendation, Computer Science.*

## I. INTRODUCTION

Artificial Neural Networks [1] is a data processing system consisting of a large number of simple, highly interconnected processing elements in architecture inspired by the structure of the cerebral cortex portion of the brain. Hence, neural networks are often capable of doing things that humans or animals do well but which conventional computers often do poorly. For career recommendation, various parameters (which are mentioned in section II. proposed system, subsection B. model) are considered, which becomes quite difficult to predict using traditional regression models. In recent years, recommendation systems [2] have been widely used in various commercial platforms to provide recommendations for users. Every field has various job roles, which makes it challenging for any undergraduate student and recruiter to decide on a well-suited job for students. Any student after graduation needs to decide which job role is best suited for him according to his profile. This is important for a long-term career plan. Similarly, for a recruiter, it is very crucial to recruit a candidate after assessing him/her in all different aspects. A career recommender system will help undergraduate students and recruiters in finding the right job based on their personality, academics, interests, etc. Roshani & Deshmukh (2014) [7] have proposed an ensembled incremental learning algorithm created by using the set of three classifiers, namely, Naive Bayes, K-Star and SVM. This was found to be a useful technique for offering the best career choice for the student. Furthermore, Arafath et al. (2018) [14] their research helped predict students' estimated careers, including students' strengths and weaknesses.

In this paper, we propose a career recommendation system using neural networks due to the high number of parameters for classification. These parameters include student performance in various subjects present in the undergraduate curriculum of computer science as well as student interests, interpersonal skills, talents, etc. [7,10 ,13]. This paper aims to implement the concept using an Artificial Neural Network (ANN) model. The model is trained and tested on 15,000 and 3,000 dataset entries, respectively. The model performs multiclass classification and is able to predict one of the 6 domains (i.e. Database Administrator, Project Manager, Software Developer, Business Intelligence Analyst, Security Administrator, Technical Support).

## II. PROPOSED SYSTEM

### *A. The Architecture of Neural Network*

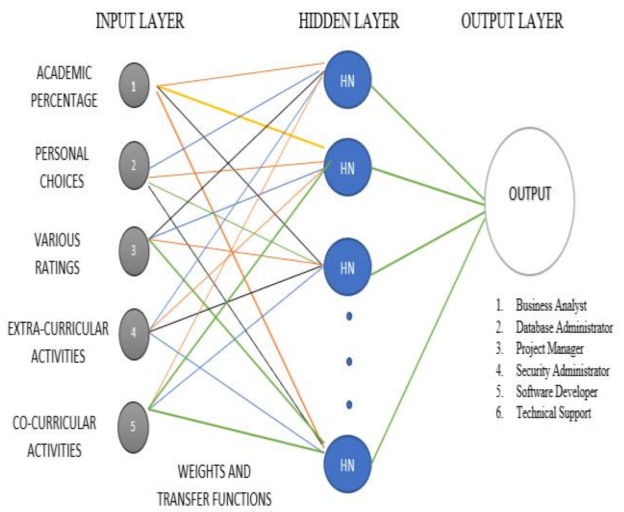


**Fig. 1 ANN Architecture**

An artificial neuron network (ANN) is a computational model based on the structure and functions of biological neural networks. The proposed ANN model consists of an input layer, 10 hidden layers and an output layer that uses a Sequential model.

The input layer is given inputs such as academic percentages, extra & co-curricular activities, various activities and personal choices, i.e. 15 input features. There are 20 nodes in total which take input and, using the activation function, give the output to the next layer. Here, the Rectified linear unit (ReLu) function was used instead of the sigmoid and tanh function since it gave better accuracy and was a less expensive operation than the other two. Further, an L1 regularizer was used instead of L2 as it was shrinking the less important features coefficient to zero.

The 2nd layer consists of 20 nodes that take input from the first layer. The ReLu function was used in this layer, just like the first layer. The next 9 layers are the same as the 2nd layer leaving the fact that it receives the input from its previous layer.

The last layer consists of 6 outputs because the model predicts one of the 6 domains, due to which the model uses multiclass classification in the last layer. Softmax activation function was used in the model for multiclass classification, which gave fairly good results. To select a particular domain, the model sets probability to each class so that it can select the class with maximum probability. It is done using categorical cross-entropy.

The flow of the model is shown in figure 2.

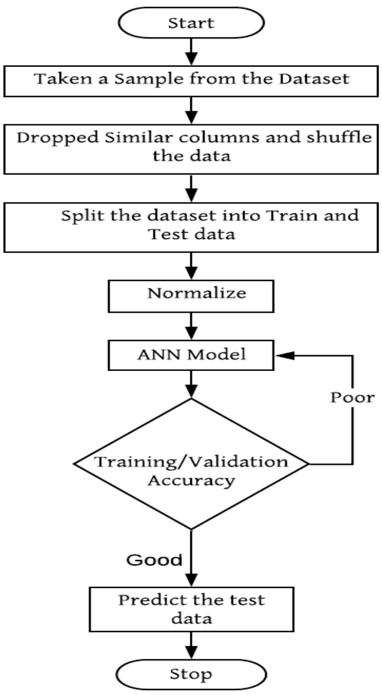


**Fig. 2 ANN Flowchart**

### *A. Model*

#### *a) Dataset*

Our dataset consists of 18,000 entries from students with their marks, personal interest extra & co-curricular and the job role they were suited for. For the model, we considered 15 parameters. The dataset was split into 70:20:10 ratio in training, testing and validation.

**Table 1. Input Parameters**

| No. | Name | Type |
|---|---|---|
| 1. | Percentage<br>1. Operating Systems<br>2. Algorithms<br>3. Programming Concepts<br>4. Software Engineering<br>5. Computer Networks<br>6. Mathematics<br>7. Communication Skills | Academics |
| 2. | Hackathons | Co-Curricular |
| 3. | Certifications<br>1. App Development<br>2. FullStack Development<br>3. Hadoop<br>4. Information Security<br>5. Machine Learning<br>6. Python<br>7. Shell Programming | Co-Curricular |
| 4. | Interests<br>1. Analyst<br>2. Management<br>3. Programming<br>4. Cloud Computing<br>5. Data Engineer<br>6. Software Engineering<br>7. System Developer<br>8. Hacking<br>9. Networks<br>10. Security<br>11. Web Development | Co-Curricular |
| 5. | Team-spirit (sports) | Extra-Curricular |
| 6. | Leadership Ability | Personality |
| 7. | Introvert/ Extrovert | Personality |
| 8. | Self Learning capability | Personality |
| 9. | Management/Technical | Personality |

### *b) Preprocessing*

First, the data was cleaned by dropping out a few features such as Salary Range Expected, Interested Type of Books, talent tests, Olympiads, Type of company you want to settle in? Interest subjects. Since they were repetitive and were not required for the prediction, the suggested job role field was categorical data, so OneHot encoding was performed to get into a binary form. There were parameters in the dataset that had different ranges, so the Standard Scaler python library was used on the dataset to normalize the data.

### *c) Algorithm*

To find the weight and bias in the model, a feedforward algorithm was used. Along with this, for the purpose of updating the corresponding weights according to respective layers, a backpropagation algorithm was used.

First, we need to multiply the input vector $\boldsymbol{X}$ and weights matrix $\boldsymbol{\theta^1}$ for the first layer $\boldsymbol{(X*\theta^1)}$ and then apply the activation function $\boldsymbol{g}$. What we get for the first layer is :

$$a_1^{(2)} = g(\Theta_{10}(1)x_0 + \Theta_{11}(1)x_1 + .... + \Theta_{115}(1)x_{15} + b_1)$$
$$a_1^{(3)} = g(\Theta_{10}(2)x_0 + \Theta_{11}(2)x_1 + .... + \Theta_{115}(2)x_{15} + b_1)$$
.
.
$$a_1^{(15)} = g(\Theta_{10}(14)x_0 + \Theta_{11}(14)x_1 + .... + \Theta_{115}(14)x_{15} + b_{15}) \quad ....(1)$$

Similarly, for the subsequent layers, we get the activation function as above.

By multiplying hidden layer vector with weights matrix $\boldsymbol{\theta}$ for the second layer($\boldsymbol{A*\theta}$) we get output for the hypothesis function:

$$y\Theta(x) = a_1^{(15)} = g(\Theta_{10}(2)a_0^{(2)} + \Theta_{11}(2)a_1^{(2)} + .... + \Theta_{115}(2)a_{14}^{(2)}) \quad ....(2)$$

The ReLu activation function is used because of its properties that give better accuracy and is a less expensive operation. It is given by the function:

$$g(z) = max(0,z) \quad ....(3)$$

The graphical representation of the ReLu function is shown in Figure 3:

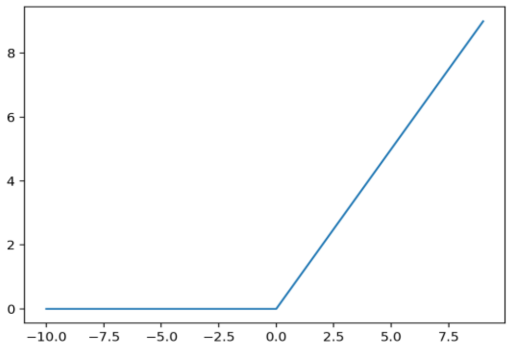


**Fig. 3 ReLu Function graph**

Second, the model calculates the loss using the categorical cross-entropy function. Here, a loss is calculated by taking the cross product of the actual output (y) and the log of the predicted value(y$\Theta$).

$$L(y,y\Theta) = - \sum_{j=0}^{M} \sum_{i=0}^{N} (y_{ij} * log(y\Theta_{ij})) \quad ....(4)$$

Using ( 4 as the cross-entropy loss function, the gradients of the weights($d\theta$(x)) and biases($db$(x)) are calculated as follows :

$$J(y,a) = \frac{1}{m}\sum_{j=1}^{m} L(y,y\Theta) \quad ....(5)$$

$$d\theta(x) = \frac{d}{d\theta(x)} * J(y,a) \qquad db(x) = \frac{d}{db(x)} * J(y,a) \quad ....(6)$$

Now, the model uses the Adam optimization technique for updating the weights and biases. For this calculation of variance(V$\theta(x)$) and standard deviation(S$\theta(x)$) of the weight, a matrix is calculated by a moment-wise update and RMS propagation-wise update, respectively. Initializing,

$$Vd\theta(x) = 0,\ S\theta(x) = 0,\ Vdb(x) = 0,\ Sdb(x) = 0 \quad ....(7)$$

$$Vd\theta(x) = \beta_1 * Vd\theta(x) + (1-\beta_1)*d\theta(x) \quad ....(8)$$

$$Vdb(x) = \beta_1 * Vdb(x) + (1-\beta_1)*db(x) \quad ....(9)$$

$$Sd\theta(x) = \beta_2 * Sd\theta(x) + (1-\beta_2)*d\theta(x) \quad ....(10)$$

$$Sdb(x) = \beta_2 * Sdb(x) + (1-\beta_2)*db(x) \quad ....(11)$$

Updating weights and biases as follows,

$$\theta(x) = \theta(x) - \alpha * \frac{Vd\theta(x)}{\sqrt{Sd\theta(x) + \epsilon}} \quad ....(12)$$

$$b(x) = b(x) - \alpha * \frac{Vdb(x)}{\sqrt{Sdb(x) + \epsilon}} \quad ....(13)$$

**Table 2. Training Parameters**

| Parameter | Symbol | Value |
|---|---|---|
| Learning Rate | $\alpha$ | 0.0001 |
| The exponential decay rate for a moment | $\beta_1$<br>$\beta_2$ | 0.9<br>0.999 |
| Epsilon | $\epsilon$ | $10^{-8}$ |
| Epochs | - | 1000 |
| Minibatch Size | - | 512 |

## III. RESULT

The proposed Recommender System is being trained on 18,000 datasets having 15 parameters which give an accuracy of 99%, whereas testing on 3,000 datasets with the same parameters gave an accuracy of 94.9%. In comparison with Roy, K. et al. (2018) [10] model, who used traditional machine learning algorithms like SVM, XGBoost and Decision, our model proved to give better accuracy than theirs.

The cost function shows an exponential decrease as follows:

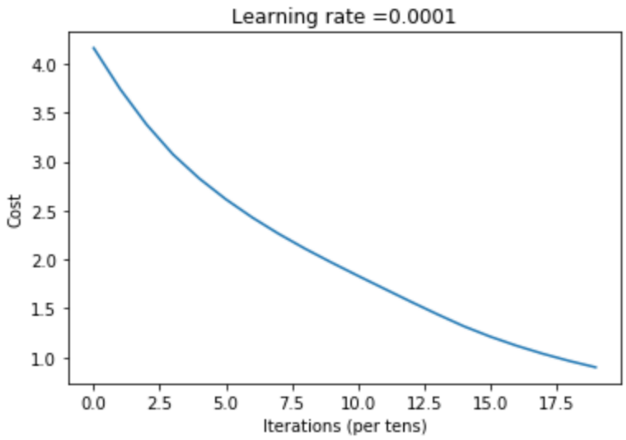


**Fig. 4 Cost function**

The result for our model is shown in figure 5.

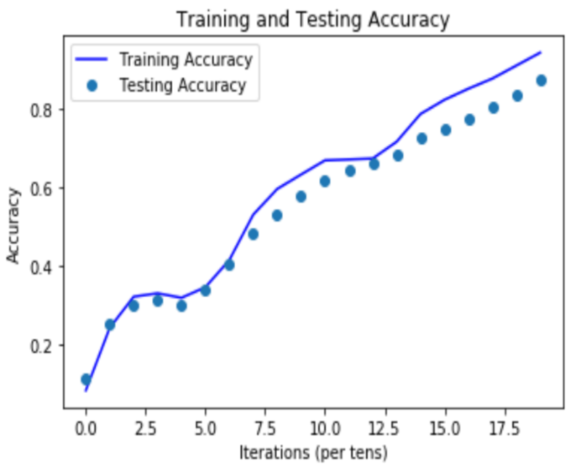


**Fig. 5 Training and testing accuracy**

## IV. CONCLUSION

This paper proposes an efficient ANN model for predicting a well-suited job role for the Computer Engineering student. The developed model is apt for the analysis of many objective factors for a person with qualified knowledge and skills. This recommender system can be used by any IT-based recruiter to hire a candidate appropriate for the job. Additionally, an individual as a Computer Engineering fresher can find out the domain that they are qualified for based on their profile and the ones who are unaware of their career. The proposed model used 15 parameters to predict one of the six job roles with an accuracy of 94.9%**.** Hence, the ANN model gives more accurate results than traditional machine learning models.

## V. FUTURE WORK

### *A. Sub-Domains*

The model can be further scaled to recognize the specialization within that particular domain. For instance, if the suggested role is in the domain of Security, there can be various subdomains within it like System Security Administrator, Network Engineer, Network Administrator, Information Security Analyst, etc.

### *B. Scalability*

The dataset has been trained in only computer engineering fields. The model can be scaled to perform for various other fields of education, e.g., Commerce, Arts and so on.

## ACKNOWLEDGEMENT

It is a privilege for us to have been associated with Dr Brijmohan Daga, Head of Department, Computer Science, our guide, during this project work. We have been greatly benefited by their valuable suggestions and ideas. It is with great pleasure that we express our deep sense of gratitude to them for their valuable guidance, constant encouragement and patience throughout this work.

## REFERENCE

[1] R. E. Uhrig, Introduction to artificial neural networks, Proceedings of IECON '95 - 21st Annual Conference on IEEE Industrial Electronics, Orlando, FL, USA, 1 (1995) 33-37 .

[2] S. Elayidom, S. M. Idikkula, J. Alexander and A. Ojha, Applying Data Mining Techniques for Placement Chance Prediction, 2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies, Trivandrum, Kerala, (2009) 669-671.

[3] Bao, Xuecai & Wang, Tangsheng & Hu, Jianfeng. ., Analysis of Individual Career Decision-Making Based on Partial Least-Squares Regression Model 6th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD, 2 (2009) 522-526. 10.1109/FSKD.2009.192.

[4] X. Bao, T. Wang and J. Hu, Analysis of Individual Career Decision-Making Based on Partial Least-Squares Regression Model, Sixth International Conference on Fuzzy Systems and Knowledge Discovery, Tianjin, (2009) 522-526.

[5] Guo, Bo & Zhang, Rui & Xu, Guang & Shi, Chuangming & Yang, Li., Predicting Students Performance in Educational Data Mining, (2015) 125-128. 10.1109/ISET.2015.33.

[6] A. Slim, G. L. Heileman, J. Kozlick and C. T. Abdallah, Employing Markov Networks on Curriculum Graphs to Predict Student Performance, 13th International Conference on Machine Learning and Applications, Detroit, MI, (2014) 415-41

[7] Ade, Roshani & Deshmukh, P.R., An incremental ensemble of classifiers as a technique for prediction of student's career choice. 1st International Conference on Networks and Soft Computing, ICNSC Proceedings. (2014) 384-387. 10.1109/CNSC.2014.6906655.

[8] L. S. Katore, B. S. Ratnaparkhi and J. S. Umale, Novel professional career prediction and recommendation method for individuals through analytics on personal traits using C4.5 algorithm, 2015 Global Conference on Communication Technologies (GCCT), Thuckalay, (2015) 503-506

[9] Naim, M. I. Tanveer, D. Gildea and M. E. Hoque, Automated prediction and analysis of job interview performance: The role of what you say and how you say it, 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), Ljubljana, (2015) 1-6.

[10] Uddin, Muhammad & Lee, Jeongkyu., Predicting good fit students by correlating relevant personality traits with academic/career data., (2016) 968-975. 10.1109/ASONAM.2016.7752357

[11] H. Mallafi and D. H. Widyantoro, Prediction modelling in career management, International Conference on Computational Intelligence and Cybernetics, Makassar, (2016) 17-21.
[12] R. H. Rangnekar, K. P. Suratwala, S. Krishna and S. Dhage ., Career Prediction Model Using Data Mining and Linear Classification, Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, (2018) 1-6.
[13] Roy, K. & Roopkanth, K. & Teja, V. & Bhavana, V. & Priyanka, J., Student Career Prediction Using Advanced Machine Learning Techniques, International Journal of Engineering and Technology (UAE). 7 (2018) 26-29. Doi:10.14419/ijet. v7i2.20.11738.
[14] Arafath, Yeasin & Saifuzzaman, Mohd & Ahmed, Sumaiya & Hossain, Syed., (2018). Predicting Career Using Data Mining. Doi:10.1109/GUCON.2018.8674995.
[15] L. Zhang, T. Luo, F. Zhang and Y. Wu, A Recommendation Model Based on Deep Neural Network, in IEEE Access, 6 (2018) 9454-9463.