Computer Science Career Recommendation System using Artificial Neural Network

Brijmohan Daga, Juhi Checker, Anne Rajan, Sayali Deo,
Department of Computer Engineering, Fr. Conceicao Rodrigues College of Engineering, Bandra West, Mumbai-400050, India

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 his/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 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 an architecture inspired by the structure of the cerebral cortex portion of the brain. Hence, neural networks are often capable of doing things which 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 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 student’s estimated careers including student’s 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. Architecture of Neural Network

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 which 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 activation function give the output to the next layer. Here, the Rectified linear unit (ReLU) function was used instead of sigmoid and tanh function since it gave better accuracy and was less expensive operation than the other two. Further, L1 regularizer was used instead of L2 as it was shrinking the less important features coefficient to zero.

The 2nd layer consisted of 20 nodes which 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 a categorical cross entropy.

The flow of the model is shown in figure 2.

B. Model

Dataset: Our dataset consists of 18,000 entries from students with their marks, personal interest and 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>
<thead>
<tr>
<th>No.</th>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Percentage</td>
<td>Academics</td>
</tr>
<tr>
<td>2.</td>
<td>Hackathons</td>
<td>Co-Curricular</td>
</tr>
<tr>
<td>3.</td>
<td>Certifications</td>
<td>Co-Curricular</td>
</tr>
<tr>
<td>4.</td>
<td>Interests</td>
<td>Co-Curricular</td>
</tr>
<tr>
<td>5.</td>
<td>Team-spirit (sports)</td>
<td>Extra-Curricular</td>
</tr>
<tr>
<td>6.</td>
<td>Leadership Ability</td>
<td>Personality</td>
</tr>
<tr>
<td>7.</td>
<td>Introvert/Extrovert</td>
<td>Personality</td>
</tr>
<tr>
<td>8.</td>
<td>Self Learning capability</td>
<td>Personality</td>
</tr>
<tr>
<td>9.</td>
<td>Management/Technical</td>
<td>Personality</td>
</tr>
</tbody>
</table>

Table I: Input Parameters
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 want to settle in?, Interested subjects. Since they were repetitive and were not required for the prediction. Suggested job role field was categorical data so OneHot encoding was performed to get into a binary form. There were parameters in the dataset who had different ranges, so Standard Scaler python library was used on the dataset to normalize the data.

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 $X$ and weights matrix $\theta'$ for the first layer ($X*\theta'$) and then apply the activation function $g$. What we get for the first layer is:

$\theta_0 x_0 + \theta_1 x_1 + \ldots + \theta_{15} x_{15} + b_1$  
\[ a^{(2)}_1 = g(\theta_0 x_0 + \theta_1 x_1 + \ldots + \theta_{15} x_{15} + b_1) \]

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

$L(y,y\theta) = -\frac{1}{m} \sum_{i=0}^{m} (y_i * \log(y_i\theta_i))$  
\[ J(y,a) = \frac{1}{m} \sum_{j=1}^{m} L(y_j,\theta) \]

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

$J(y,a) = \frac{1}{m} \sum_{j=1}^{m} L(y_j,\theta)$  
$\frac{d\theta(x)}{db} = \frac{\partial}{\partial b} J(y,a)$  
$\frac{d\theta(x)}{db} = \frac{\partial}{\partial b} J(y,a)$

Now, the model uses 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 matrix is calculated by a moment-wise update and RMS propagation-wise update respectively. Initializing,

$V\theta(0) = 0, S\theta(0) = 0, Vdb(x) = 0, Sdb(x) = 0$

... (7)

$V\theta(x) = \beta_1 * V\theta(x) + (1-\beta_1) * d\theta(x)$  
$V\theta(x) = \beta_1 * V\theta(x) + (1-\beta_1) * d\theta(x)$  

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

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

The graphical representation of the ReLu function is shown in Figure 3:
\[ b(x) = b(x) - \alpha^* \frac{\nabla b(x)}{\sqrt{\text{Sub}(x)} + \epsilon} \]  
...(12)

\[ ... \]  
...(13)

### TABLE II
**TRAINING PARAMETERS**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learning Rate</td>
<td>( \alpha )</td>
<td>0.0001</td>
</tr>
<tr>
<td>Exponential decay rate for moment</td>
<td>( \beta_1 )</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>( \beta_2 )</td>
<td>0.999</td>
</tr>
<tr>
<td>Epsilon</td>
<td>( \epsilon )</td>
<td>(10^{-8})</td>
</tr>
<tr>
<td>Epochs</td>
<td>-</td>
<td>1000</td>
</tr>
<tr>
<td>Minibatch Size</td>
<td>-</td>
<td>512</td>
</tr>
</tbody>
</table>

### III. RESULT

The proposed Recommender System is being trained on 18,000 datasets having 15 parameters which gives an accuracy of 99% whereas testing on 3,000 datasets with 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:

![Cost Function](image)

The result for our model is shown in figure 5.

![Training and Testing Accuracy](image)

### 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, ANN model gives more accurate results to 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 sub domains within it like System Security Administrator, network Engineer, Network Administrator, Information Security Analyst, etc.

B. **Scalability** - The dataset has been trained on only computer engineering fields. The model can be scaled to perform for various other fields of education eg: 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


