SMART ORDER AND COURIER DELIVERY

USING AN AUTOMATED NEURAL NETWORK

ABSTRACT FOR PAPIER EVENT, CONVOLUTION-2018

Submitted by

Palash Sarkar (Student)
Dr. Sumit Gupta (Mentor)

Theme: E-Retail

INSTITUTE

BE (Computer Science & Engineering)

BURDWAN UNIVERSITY

**UNIVERSITY INSTITUTE OF TECHNOLOGY
THE UNIVERSITY OF BURDWAN
GOLAPBAG (NORTH), BURDWAN-713104**.

Abstract -

Ever since the advent of the internet-age, the e-retail industry is expanding at a geometric rate day by day. Hence, there is a need to provide customers with superior facilities for an elevated user-experience, along with an optimized delivery/servicing system which is both efficient and reliable.
As computer science under-graduates, the best possible solution would be to use the fundamentals of the subject, as it would automate the entire process and increase precision of such a scalable system. Developing such an ecosystem may not be easy, but it is worth trying.

For starters, we have developed a generic online-shopping portal with a simple front-end website for order-placement and easy item-navigation using indexed-TST (Ternary-Search Tree) searching and a fairly-complex backend abstraction which involves a database to manage all incoming information and a 10-layer deep-neural network for predicting optimal delivery routes for our imaginary delivery-agent, who would be using a mobile application for receiving assistance from the server.

When the user searches, for a specific item on the web-site, an auto-fill suggestion is provided by the 'TST' and it does this in linear time (Average Case - O(log n), Worst Case - O(n)), which is contradictory to database 'wildcard' matching, which is unpredictable and involves a lot of other overheads. For our ***'Optimal Path Calculation'*** module, we have considered using real-time addresses for distance calculation, for this purpose we have used 'Google's Distance Matrix' API. For every order address, a data-set is created wherein at first the shortest-path is first calculated using standard algorithms, namely 'Nearest Neighbor Algorithm' (Worst Case - $O(n^2*2^n)$) and 'MST Approximation Algorithm'. Then the corresponding results and the datasets are passed on to our neural network, which then trains itself by adjusting the input weights for learning. We have a separate module which is responsible for training this entity whenever a new data-set is generated and it also tracks the neuron's accuracy rate over the test-cases. Once, a standard accuracy rate is attained by the neural-network, the system will switch to its path predictions rather than using manual algorithms. Not only, that such information can be used for analytics also.

Since, we are dealing with a special case of 'Travelling-Salesman Problem', which is in fact an 'NP-Hard' problem without any polynomial-time solution, our 'Neural-Model' is still at a nascent stage of learning with a moderate prediction accuracy. Plus, we haven't even used any advanced 'AI-Library' like 'Google's Tensor-Flow' or 'IBM Watson'. The entire neural-network was implemented from scratch. So far, we have restricted our-model to 10-Layers and 10 City-Distances, with training based on 45-datasets, but hopefully we can expand on that later. Experimentations are ongoing.