CS 461
Fall 2020
Final Program – Neural Networks
Due 11:59 PM Friday, Dec. 11

This project is using real-world data to try to identify health insurance customers who may also be interested in vehicle insurance.

You are given a data file (CSV) of just over 380,000 customers. For each you know their gender, age, whether they have a driving license, whether they already have vehicle insurance, the age of the vehicle, whether they have a history of damage to the vehicle, their region, a code for how they were reached (different agents, mailing, etc), the premium (amount the customer would pay) for the insurance, their number of days with the company, and whether they expressed interest in purchasing. The data dictionary from Kaggle.com is included at the end of this document. Note that this is from a company in India; premium amounts are in rupees, not dollars.

You should develop a neural network using TensorFlow to classify these cases. Because the number of input variables is small, you will not want a huge network; you probably won't need more than 1 hidden layer, and almost certainly won't need more than 2. (Remember, a network that's much larger than it needs is more vulnerable to overtraining.) Experiment with the number of layers and neurons per layer to see what gets the best result.

You will need to do some recoding of the data before feeding it to your network. Do this in whatever language you like. You do not need to submit your code for that part, and it doesn't need to be part of your TensorFlow code.

ID: not needed; can be omitted from input data.
Gender: 0/1
Age: Break into 5 year ranges; number the ranges, code them 1-hot. (The 1-hot coding can be done in TF). OR: Standardize into z-scores.
Driving license: Already 0/1
Region: 1-hot
Previously insured: already 0/1
Vehicle age: Already in categories, number them & code them 1-hot
Vehicle damage: Already 0/1
Annual premium: Standardize
Policy sales channel: 1-hot (given there are separate codes for each agent, this is likely to have a great many categories; but some categories do appear often enough to probably have some predictive value.)
Vintage: standardize
Response: This is your output, into 1 of 2 categories. Use 1-hot coding; for 2 cases, that means pick the one with the highest value as the classification.

Divide the data into training/test/validation data in 70/15/15 proportions. (So 15% validation data, the other 85% for training & test sets). Select a validation strategy; consider how you can optimize your network via regularization or other method. When you think you've developed the best neural network classifier you can, break the validation data out of the lockbox and run it to get your final results.

Prepare a short report summarizing your results. Discuss the configuration of your network, what validation strategy you used and why you chose it, and the quality of your results. No references are

required. Illustrations, charts, etc., may be helpful but aren't required. Submit your TensorFlow code along with your report.

If you want to see what others are doing with this (or dive into this as a real-world project), see: https://www.kaggle.com/anmolkumar/health-insurance-cross-sell-prediction.

Data dictionary:

Variable	Definition
id	Unique ID for the customer
Gender	Gender of the customer
Age	Age of the customer
Driving_License	0 : Customer does not have DL, 1 : Customer already has DL
Region_Code	Unique code for the region of the customer
Previously_Insured	1 : Customer already has Vehicle Insurance, 0 : Customer doesn't have Vehicle Insurance
Vehicle_Age	Age of the Vehicle
Vehicle_Damage	1 : Customer got his/her vehicle damaged in the past. 0 : Customer didn't get his/her vehicle damaged in the past.
Annual_Premium	The amount customer needs to pay as premium in the year
Policy*Sales*Channel	Anonymized Code for the channel of outreaching to the customer ie. Different Agents, Over Mail, Over Phone, In Person, etc.
Vintage	Number of Days, Customer has been associated with the company
Response	1 : Customer is interested, 0 : Customer is not interested