Predicting the Number of Automobile-Related Fatalities Using A Densely-Connected Neural Network and Regression

By Spencer Goff

Abstract

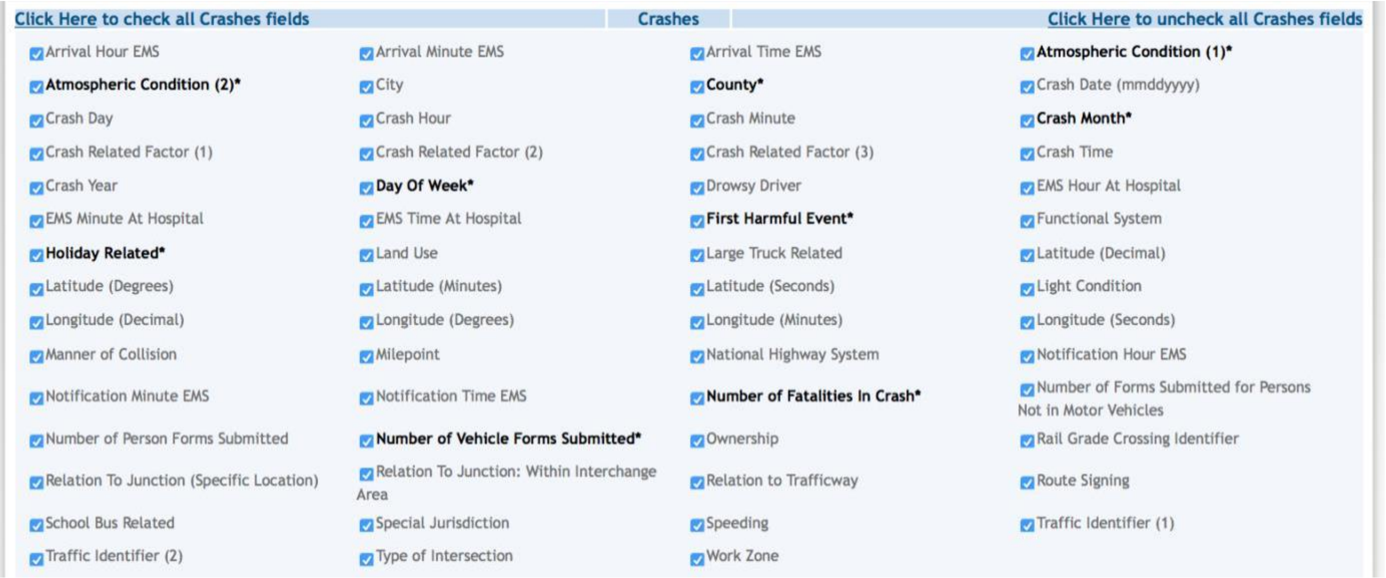
This paper will outline a project that had two major goals. The first of these goals was to uncover the factors that contribute most to automobile-related fatalities, analyzing patterns in this number that occurred nationwide in 2016 under given sets of conditions involving weather, time, and day of the week. Using the Pearson Coefficient, it was determined that the weather has the most significant impact on the number of automobile-related fatalities. The second of the project's goals was to optimize the parameters of a densely-connected neural network regressor to obtain the most accurate results. Bearing in mind that the data used contained 1680 data points with 3 features each, the optimal parameters were found to be six layers with three neurons each. The steps taken in this project and the results and analysis can be found below. The data and code for this project can be found on Github under the title automobile-fatality-prediction.

Table of Contents

Data Collection Process

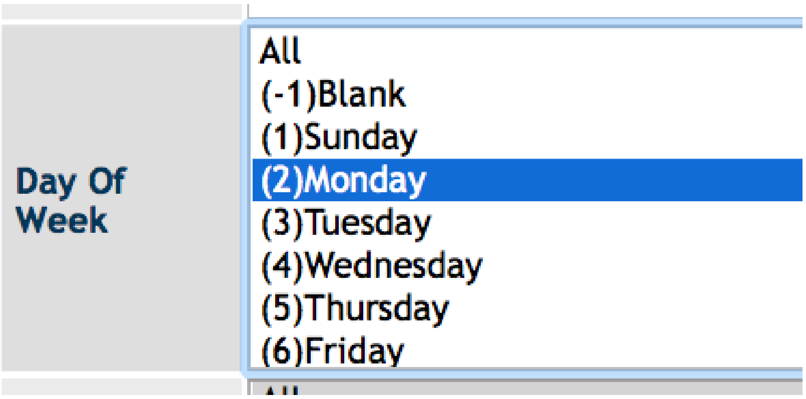
To repeat my process for data collection, follow these steps:

1. Go to https://www-fars.nhtsa.dot.gov/QueryTool/querysection/selectyear.aspx
2. Select the desired year in the top right corner and click submit.
3. Select all fields using the designated button and click submit.



Click Here to check all Crashes fields	Crashes		Click Here to uncheck all Crashes fields
☑ Arrival Hour EMS	☑ Arrival Minute EMS	☑ Arrival Time EMS	☑ **Atmospheric Condition (1)***
☑ **Atmospheric Condition (2)***	☑ City	☑ **County***	☑ Crash Date (mmddyyyy)
☑ Crash Day	☑ Crash Hour	☑ Crash Minute	☑ **Crash Month***
☑ Crash Related Factor (1)	☑ Crash Related Factor (2)	☑ Crash Related Factor (3)	☑ Crash Time
☑ Crash Year	☑ **Day Of Week***	☑ Drowsy Driver	☑ EMS Hour At Hospital
☑ EMS Minute At Hospital	☑ EMS Time At Hospital	☑ **First Harmful Event***	☑ Functional System
☑ **Holiday Related***	☑ Land Use	☑ Large Truck Related	☑ Latitude (Decimal)
☑ Latitude (Degrees)	☑ Latitude (Minutes)	☑ Latitude (Seconds)	☑ Light Condition
☑ Longitude (Decimal)	☑ Longitude (Degrees)	☑ Longitude (Minutes)	☑ Longitude (Seconds)
☑ Manner of Collision	☑ Milepoint	☑ National Highway System	☑ Notification Hour EMS
☑ Notification Minute EMS	☑ Notification Time EMS	☑ **Number of Fatalities In Crash***	☑ Number of Forms Submitted for Persons Not in Motor Vehicles
☑ Number of Person Forms Submitted	☑ **Number of Vehicle Forms Submitted***	☑ Ownership	☑ Rail Grade Crossing Identifier
☑ Relation To Junction (Specific Location)	☑ Relation To Junction: Within Interchange Area	☑ Relation to Trafficway	☑ Route Signing
☑ School Bus Related	☑ Special Jurisdiction	☑ Speeding	☑ Traffic Identifier (1)
☑ Traffic Identifier (2)	☑ Type of Intersection	☑ Work Zone	

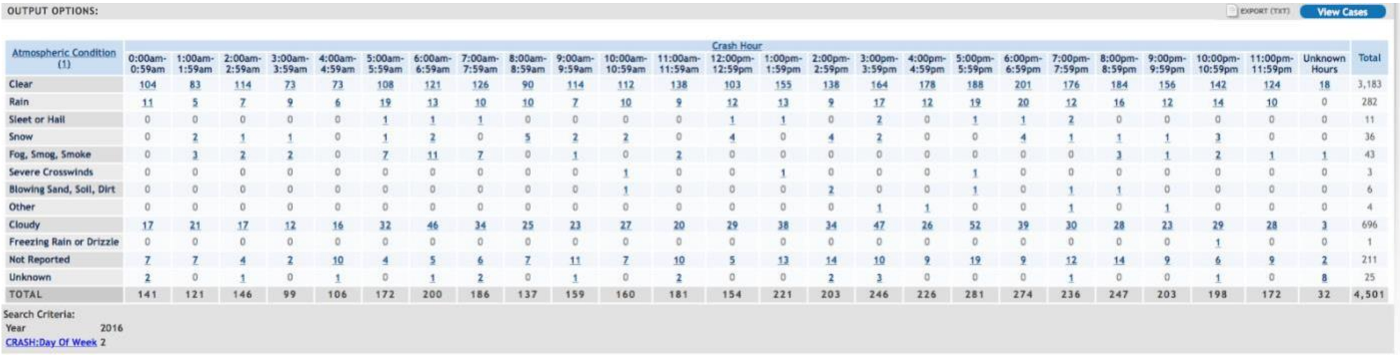
4. Select the day of week you desire.



Day Of Week

All
(-1)Blank
(1)Sunday
[✓ (2)Monday]
(3)Tuesday
(4)Wednesday
(5)Thursday
(6)Friday

5. Click "Cross Tab".
6. Select "Crash Hour" for the columns and "Atmospheric Condition (1)" for the rows, then click submit.



Atmospheric Condition (1)	Crash Hour																									
	0:00am-0:59am	1:00am-1:59am	2:00am-2:59am	3:00am-3:59am	4:00am-4:59am	5:00am-5:59am	6:00am-6:59am	7:00am-7:59am	8:00am-8:59am	9:00am-9:59am	10:00am-10:59am	11:00am-11:59am	12:00pm-12:59pm	1:00pm-1:59pm	2:00pm-2:59pm	3:00pm-3:59pm	4:00pm-4:59pm	5:00pm-5:59pm	6:00pm-6:59pm	7:00pm-7:59pm	8:00pm-8:59pm	9:00pm-9:59pm	10:00pm-10:59pm	11:00pm-11:59pm	Unknown Hours	Total
Clear	104	83	114	73	73	108	121	126	90	114	112	138	103	155	138	164	178	188	201	176	184	156	142	124	18	3,183
Rain	11	5	7	9	6	19	13	10	10	7	10	9	12	13	9	17	12	19	20	12	16	12	14	10	0	282
Sleet or Hail	0	0	0	0	0	1	1	1	0	0	0	0	1	1	0	2	0	1	1	2	0	0	0	0	0	11
Snow	0	2	1	1	0	1	2	0	5	2	2	0	4	0	4	2	0	0	4	1	1	1	3	0	0	36
Fog, Smog, Smoke	0	3	2	2	0	7	11	7	0	1	0	2	0	0	0	0	0	0	0	0	3	1	2	1	1	43
Severe Crosswinds	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	3
Blowing Sand, Soil, Dirt	0	0	0	0	0	0	0	0	0	0	1	0	0	0	2	0	0	1	0	1	1	0	0	0	0	6
Other	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	1	0	0	0	4
Cloudy	17	21	17	12	16	32	46	34	25	23	27	20	29	38	34	47	26	52	39	30	28	23	29	28	3	696
Freezing Rain or Drizzle	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
Not Reported	7	7	4	2	10	4	5	6	7	11	7	10	5	13	14	10	9	19	9	12	14	9	6	9	2	211
Unknown	2	0	1	0	1	0	1	2	0	1	0	2	0	0	2	3	0	0	0	1	0	0	1	0	8	25
TOTAL	141	121	146	99	106	172	200	186	137	159	160	181	154	221	203	246	226	281	274	236	247	203	198	172	32	4,501

Search Criteria:
Year 2016
CRASH:Day Of Week 2

7. If desired, click Export to download the data as a .txt file. This file can be opened in Excel using tab as the delimiter. I repeated all of these steps for each day of the week. Below is an example of a report for a single day of the week. I then copied the values of each column into a custom Excel spreadsheet I made, which had every possible combination of day of the week, atmospheric condition, and hour.

About the Data

The data set built using the above process resulted in the three feature columns of *day, hour,* and *weather,* in addition to the label column of the *number of fatalities* for the given conditions.

The data fields have the following possible values:

- day: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday
- hour: 0:00am-0:59am, 1:00am-1:59am, 2:00am-2:59am ... 11:00pm-11:59pm
- weather: Clear, Rain, Sleet or Hail, Snow, Fog or Smoke or Smog, Severe Crosswinds, Other, Cloudy, Blowing Snow, Freezing Rain or Drizzle
- num_fatalities: 0 or any positive integer

Here are the first few rows of data:

day	hour	weather	num_fatalities
Sunday	0:00am-0:59am	Clear	211
Sunday	0:00am-0:59am	Rain	17
Sunday	0:00am-0:59am	Sleet or Hail	1
Sunday	0:00am-0:59am	Snow	2

To work with the machine learning model, the non-numeric data fields (day, hour, and weather) had to be quantified. To do so, I assigned the following values:

Sunday: 0
Monday: 1
Tuesday: 2
Wednesday: 3
Thursday: 4
Friday: 5
Saturday: 6

0:00am-0:59am: 0
1:00am-1:59am: 1
2:00am-2:59am: 2
3:00am-3:59am: 3
4:00am-4:59am: 4
5:00am-5:59am: 5
6:00am-6:59am: 6
7:00am-7:59am: 7
8:00am-8:59am: 8
9:00am-9:59am: 9
10:00am-10:59am: 10
11:00am-11:59am: 11
12:00pm-12:59pm: 12
1:00pm-1:59pm: 13
2:00pm-2:59pm: 14
3:00pm-3:59pm: 15
4:00pm-4:59pm: 16
5:00pm-5:59pm: 17
6:00pm-6:59pm: 18
7:00pm-7:59pm: 19
8:00pm-8:59pm 20
9:00pm-9:59pm: 21
10:00pm-10:59pm: 22
11:00pm-11:59pm: 23

Clear: 0
Rain: 1
Sleet or Hail: 2
Snow: 3
Fog or Smoke or Smog: 4
Severe Crosswinds: 5
Other: 6
Cloudy: 7
Blowing Snow: 8
Freezing Rain or Drizzle: 9

The first few rows of the resulting data looks like this:

day	hour	weather	num_fatalities
0	0	0	211
0	0	1	17
0	0	2	1
0	0	3	2

To see my complete data set, go to the Github website and search for automobile-fatality-prediction. It will be associated with the user spencergoff.

Data Analysis

In this section, I'll do some basic data analysis using Excel to glean some interesting and potentially useful insights.

The highest number of fatalities occur on Mondays between 6:00pm and 6:59pm when the weather is clear. In fact, all of the top ten data points with the highest number of crashes occurred on Monday afternoons and evenings when the weather was clear.

day	hour	weather	num_fatalities
Monday	6:00pm-6:59pm	Clear	1429
Monday	7:00pm-7:59pm	Clear	1379
Monday	8:00pm-8:59pm	Clear	1375
Monday	4:00pm-4:59pm	Clear	1370
Monday	9:00pm-9:59pm	Clear	1345
Monday	5:00pm-5:59pm	Clear	1341
Monday	3:00pm-3:59pm	Clear	1253
Monday	10:00pm-10:59pm	Clear	1175
Monday	2:00pm-2:59pm	Clear	1114

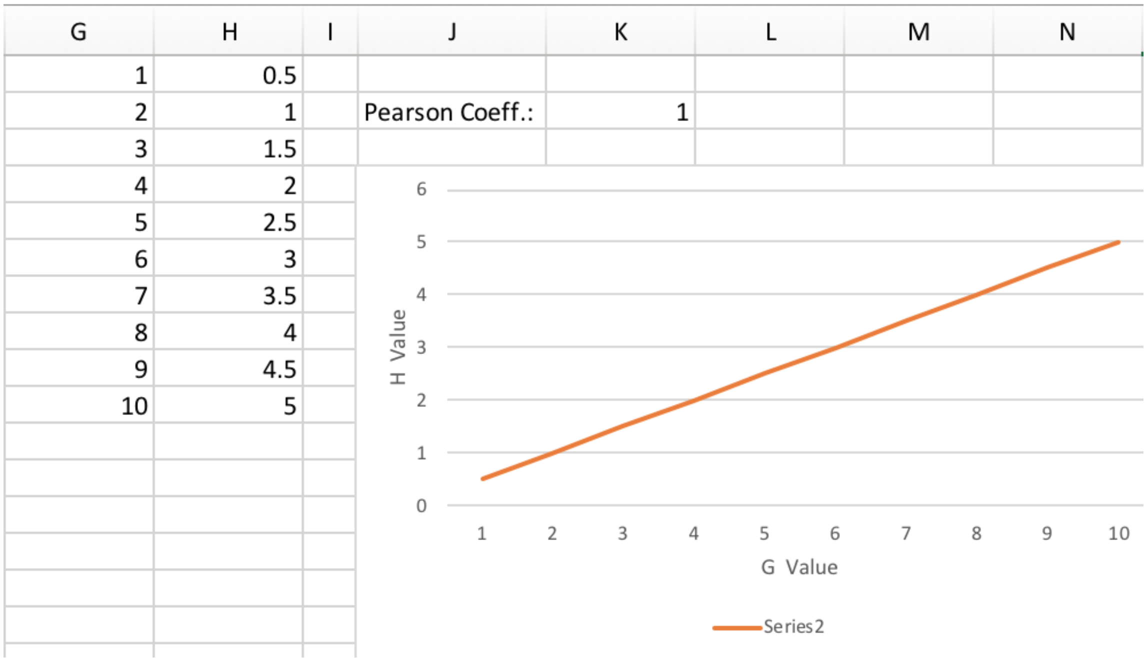
As for the conditions that result in the least number of fatalities, there were 697 (out of 1681) where no fatalities were reported. For example, no fatalities were reported on Sundays between 6:00am and 6:59am where there were severe crosswinds.

For perspective, the average number of fatalities considering all combinations of data is 36 (including all 697 data points where no fatalities were reported).

Introduction to the Pearson Coefficient

The Pearson Coefficient is a measure of the strength of the relationship between two variables[1].

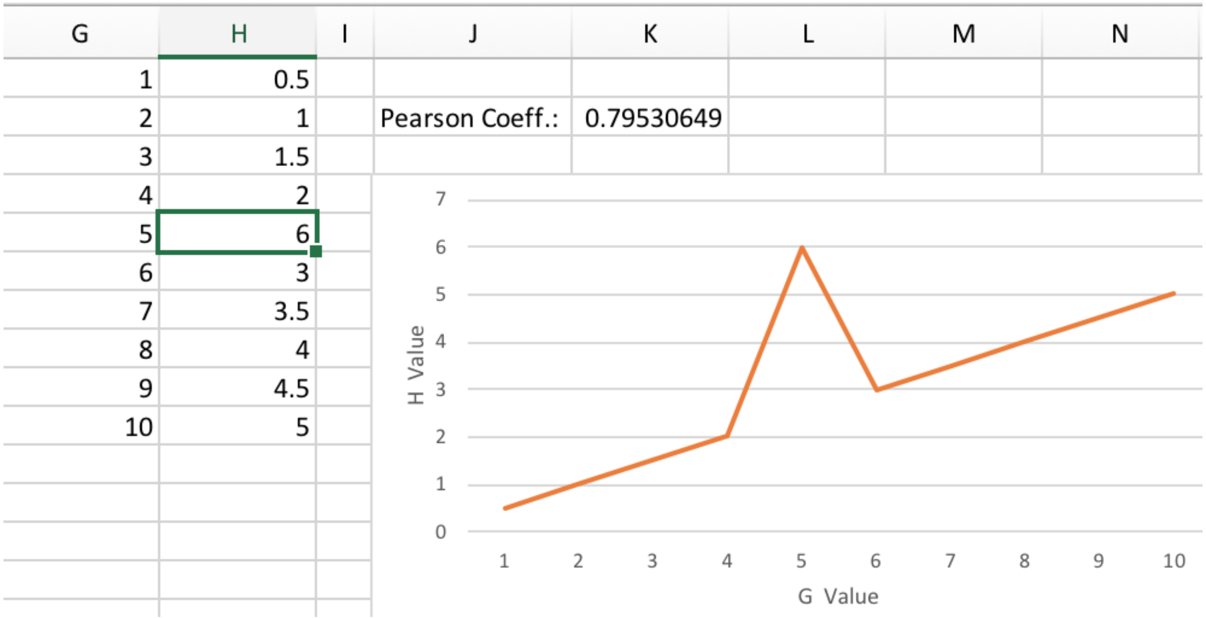
To learn more about the Pearson Coefficient, I created some simple test data in Excel. In column G, I placed consecutive integers 1 through 10. In column H, I placed 0.5, 1.0, 1.5, …, 5.0. I then created, a graph based off of these data points to visualize this relationship. In cell K2, I used Excel's Pearson function to calculate the correlation between the values of G and the values of H.



G	H	I	J	K	L	M	N
1	0.5						
2	1		Pearson Coeff.:	1			
3	1.5						
4	2						
5	2.5						
6	3						
7	3.5						
8	4						
9	4.5						
10	5						

In this case, Excel calculated the Pearson Coefficient to be 1. This means that the G value directly and proportionally determines the H value. Since the coefficient is positive, there is a positive correlation between G and H (i.e. as G increases, H also increases).

I then changed the 2.5 value in the H column to a 6 to see what would happen to the Pearson Coefficient. As seen below, it changed from 1 to approximately 0.79. This means that, while there is still a relatively strong correlation between the G and H values, the correlation isn't perfect and G does not precisely determine G for all values.



G	H	I	J	K	L	M	N
1	0.5						
2	1		Pearson Coeff.:	0.79530649			
3	1.5						
4	2						
5	6						
6	3						
7	3.5						
8	4						
9	4.5						
10	5						

Applying and Interpreting the Pearson Coefficient

Applying the Pearson to the real data for this project, the following correlations were obtained:

Attributes to Compare	*Pearson Coefficient*
day -> num_fatalities	-0.116913802
hour -> num_fatalities	0.047788412
weather -> num_fatalities	-0.308456356

This shows that, of these factors, weather has the highest impact on the number of fatalities, day of the week has the second highest impact, and the hour (time) has the lowest impact.

To further explore this phenomenon, I took the average of each variety of weather represented to see which ones result in the highest number of fatalities. These averages span all hours and days. Here are the results:

Weather		*Average num_fatalities*
Clear		272.375
Rain		22.98809524
Sleet or Hail		0.68452381
Snow		3.654761905
Fog or Smoke or Smog		3.571428571
Severe Crosswinds		0.482142857
Other		0.583333333
Cloudy		55.79166667
Blowing Snow		0.119047619
Freezing Rain or Drizzle		0.18452381

At first glance, it’s odd that clear weather has by far the highest average number of fatalities, but clear weather is probably the most common type of weather listed (based on personal experience). However, it is still notable that blowing snow and freezing rain are the two types of weather with the lowest average number of fatalities. I propose that this may be because of three factors: (1) it is likely that these types of weather are less common than the other types of weather listed; (2) it is likely that people try to limit the time they spend on the road when these conditions occur; and (3) it is likely that people drive more cautiously during these types of weather.

Prediction Accuracy Optimization

When designing a neural network/model, there are many factors that come in to play. The model needs enough experience (in the form of steps and epochs) with the training data so that it can adjust the weights of its neurons and give accurate predictions. It should be noted that densely connected networks will always have an input layer with the same number of neurons and the number of features (in our case, three), and an output layer with only one neuron (this yields the prediction, e.g. number of fatalities). There are four aspects of the neural network that I will vary to attempt to find the combination with the lowest average error: number of epochs, steps, number of hidden layers, and number of neurons per layer.

The number of epochs is how many times the network does a complete run through of all data, adjusting the weights of the neurons to more closely fit the training data. The number of steps is how many "smaller chunks" the model processes the data in. Hidden layers are those in between the input and output layers of a network. The neurons/layer is the number of neurons that each hidden layer contains.

Here are the results of my experiment. I altered one variable at a time, starting with epochs and working over to neurons per layer. Once I found what looked to be the optimal value of that variable, based on the lowest average error, I locked that value in and moved on to the next variable.

# epochs	# steps	# hidden layers	neurons/layer	Avg. Error	Optimal Number
1	1000	3	6	121.6443578	
2	1000	3	6	121.6481064	
5	1000	3	6	109.7531436	
20	1000	3	6	100.7449001	20 epochs
25	1000	3	6	106.0581492	
30	1000	3	6	110.5996539	
20	10000	3	6	100.9868205	
20	5000	3	6	94.20216374	
20	2500	3	6	94.18239015	2500 steps
20	1500	3	6	98.67548559	
20	2000	3	6	96.08997194	
20	2500	9	6	124.6691261	
20	2500	6	6	94.08466537	6 layers
20	2500	2	6	106.4522995	
20	2500	4	6	95.76352017	

20	2500	6	6	94.84665373	
20	2500	6	3	97.3171982	
20	2500	6	9	95.56610026	
20	2500	6	7	93.59007226	7 neurons per layer
20	2500	6	8	95.92002937	

So, the optimal combination (the one yielding the lowest error) is 20 epochs with 2500 steps, 6 layers, and 7 neurons per layer. Why didn’t the highest values of each of these fields yield the most accurate results? After all, it seems that more steps and epochs at least would increase accuracy since the model has more opportunities to adjust the weights of its neurons to fit the data. Herein lies the problem; since the training data and testing data are not the same, the model may *overfit* to the training data, and lose some of its accuracy when testing. This is why there is a “Goldilocks” value for each feature—not to high, not to low, just right. The purpose of this experiment was to find these optimal values for each aspect of the model.

It is of importance to note that this combination of neurons, layers, epochs, and steps will certainly not be optimal in every situation. This result is highly dependent on the number of data points being used (in this case, 1680) and the number of features that consist the data (in this case, 3).

Overview of the Neural Network

The complete code for this project can be found at https://github.com/spencergoff/automobile-fatality-prediction/blob/master/automobile-fatality-prediction-regression.ipynb. The code is well-commented and should be relatively easy to follow, especially after reading this document. Here I'll give a general overview of what that code does, assuming the reader has a basic understanding of neural networks.

The regression model using a densely-connected neural network works as follows. Quantified features are fed into neurons, one neuron per feature. Within each neuron, the values of each feature are altered using an activation function, which in this case is the Rectified Linear Unit function, y = max(0,x). The output of these neurons are fed into the next layer of neurons, until the single output neuron is reached and a prediction is made. Depending on how close the network's prediction is to the correct label, it adjusts the weights (x-values) of each neuron. It does this for every step, which is simply a small chunk of data, and every epoch, which is each complete passthrough of the data. The result is a model that should be able to predict, with some accuracy, the outcome (in this case, number of fatalities) based on a set of input values (in this case, weather, time, and day).

Conclusion

This paper outlined a project that had two major goals. The first of these goals was to uncover the factors that contribute most to automobile-related fatalities, analyzing patterns in this number that occurred nationwide in 2016 under given sets of conditions involving weather, time, and day of the week. Using the Pearson Coefficient, it was determined that the weather has the most significant impact on the number of automobile-related fatalities. The second of the project's goals was to optimize the parameters of a densely-connected neural network regressor to obtain the most accurate results. Bearing in mind that the data used contained 1680 data points with 3 features each, the optimal parameters were found to be six layers with three neurons each. The steps taken in this project and the results and analysis can be found below. The data and code for this project can be found on Github under the title automobile-fatality-prediction. Note that this project was for learning purposes only, and is not meant to provide or replace professional advice.

Sources

[1] An introduction to the Pearson Coefficient, by David M. Lane http://onlinestatbook.com/2/describing_bivariate_data/pearson.html

[2] The National Highway and Trafic Safety Administration’s data on automobile-related fatalities in the United States https://www-fars.nhtsa.dot.gov/QueryTool/querysection/selectyear.aspx

[3] José Portilla’s course on Udemy.com titled *Complete Guide to TensorFlow for Deep Learning https://www.udemy.com/complete-guide-to-tensorflow-for-deep-learning-with-python/learn/v4/t/lecture/8410106?start=870*