Lecture 10b
Neural Networks Continued

Breitzman 8/8/2018

Previously…

- We built a couple of neural networks by hand and in Excel

- The last example computed square roots of numbers between 1 and 100

- The Model had an average error of 4% with the worst case being 11% for small numbers

- The last lecture had a lot of math and theory. This one we’ll do some R

R

- We will show how R implements a neural net
- Hopefully since R optimizes the learning rate, we will get a better result
- Go to R – SquareRootR.doc
- Go To Excel – SquareRootFinishedModels.xls (This shows we can build a model in R but embed it into anything {e.g. Excel, Java, an embedded system on a car or boat})

Census Data

- Recall over the last several weeks we’ve used the 32,000+ records from the 1994 census to predict people with income below or above $50k in 1994.
- We’ve used Decision Trees and Naïve Bayes classifiers and in both cases had an error rate of 16-17%
- Note only 24% of people had income above 50k, so if we built a predictor that always returned <50k, we would have an error rate of 24%

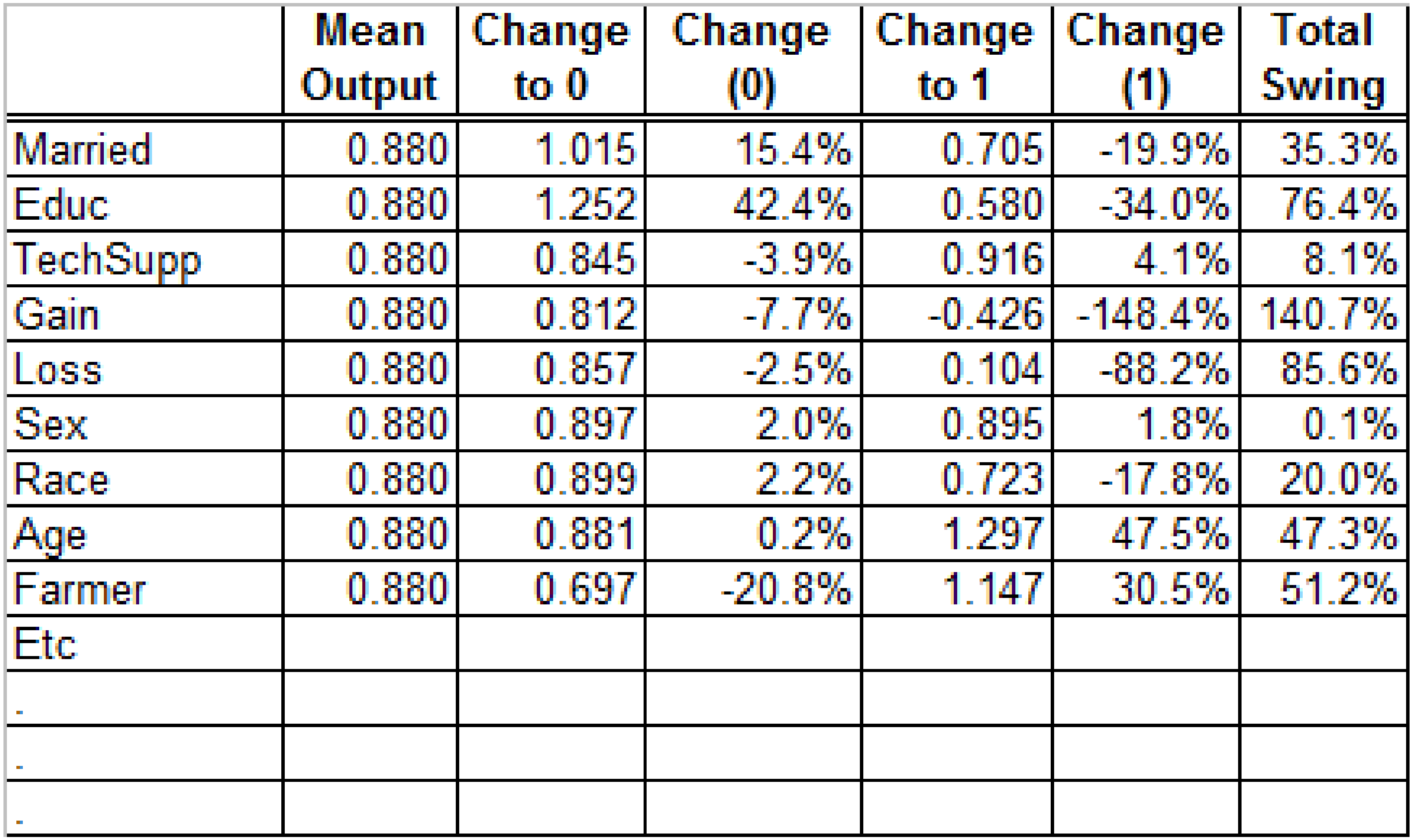
Census Data for Neural Nets

- Recall the census data is full of various class variables Relationship, Race, Job
- Recall also that Neural Nets can only deal with numeric values between 0 and 1.
- For education years, age, etc. we can use min-max normalization
- Since there are many job titles that have no relationship with each other we need to create multiple variables such as IsFarmer, IsFileClerk, etc. all with values of 0 or 1
- Go to R to finish

Sensitivity Analysis

- Compute average input vector
- Compute result of average vector
- Check results of average vector by varying each variable between 0 and 1
- Go to R

Partial Sensitivity Analysis for Census Data



	Mean Output	Change to 0	Change (0)	Change to 1	Change (1)	Total Swing
Married	0.880	1.015	15.4%	0.705	-19.9%	35.3%
Educ	0.880	1.252	42.4%	0.580	-34.0%	76.4%
TechSupp	0.880	0.845	-3.9%	0.916	4.1%	8.1%
Gain	0.880	0.812	-7.7%	-0.426	-148.4%	140.7%
Loss	0.880	0.857	-2.5%	0.104	-88.2%	85.6%
Sex	0.880	0.897	2.0%	0.895	1.8%	0.1%
Race	0.880	0.899	2.2%	0.723	-17.8%	20.0%
Age	0.880	0.881	0.2%	1.297	47.5%	47.3%
Farmer	0.880	0.697	-20.8%	1.147	30.5%	51.2%
Etc						
.						
.						
.						

Why do Sensitivity Analysis?

- Dimension is our enemy
- I killed a process didn’t finish after an hour with 21 variables, 20 hidden layers and one output
- It ran in 8 minutes with 21 variables and 15 hidden layers
- If I can remove a few variables things will run faster and I can optimize the number of hidden layers
- At some point we will talk about general strategies for reducing dimension, but sensitivity analysis is not a bad place to start

Random Meme that’s only funny after you’ve waited forever for a NN to finish