L9

2023-05-02

Homework 2

Part 1

Controversy over SAT usage

- Vocab term “regatta” i.e. set of boats used by anti-SAT people
- Wouldn’t know the word unless you came from a rich family/favors the middle class

Problem 1: scale SAT score according to income

- Someone with a 70th percentile may have the 90th percentile score among low-income test takers

Quantile vs. Percentile

- 97th percentile == 0.97 quantile == 0.86 score (for example)

Categorical Variable:

- Using continuous var (i.e. SAT), break into n categories
- Ex. high income/middle income/low income categories

cut() Function

- cut() function will make n categories of var (i.e. wealth, score)
- Does not remember the cutoffs it makes
- Need to make our own to “remember” cutoffs for categorizing test inputs
- OR call levels() on output of cut() to get intervals (parse char >> numbers)
- New case outside of old ranges?
- Break data into groups with cut, then generate quantiles within these new groups

New Test Case

- Take income, convert to category
- Compare SAT to the quantiles of that category (i.e. if 6/10 of quantiles had lower score than this person, then they are in the 60th percentile of that category)
- Replace SAT column by scaled SAT column
- Apply ML algorithms to new SAT score
- Ideally, scaled SAT should be just as helpful in predicting success in college without risking wealth bias

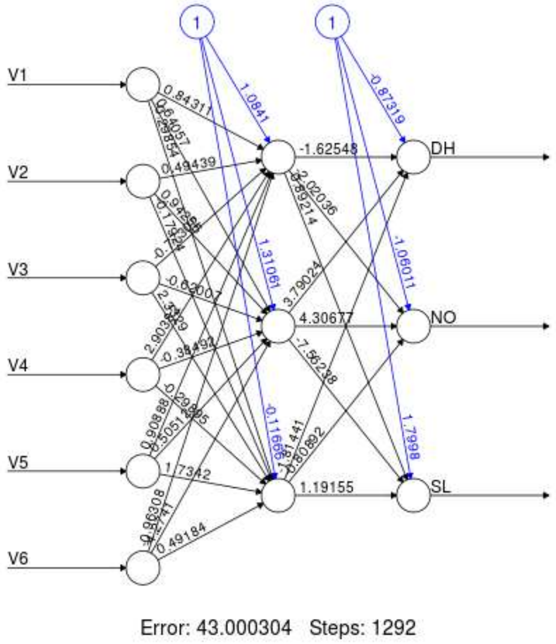
Measure Fairness and Utility

- Use qeML testacc to measure utility
- Use correlation between SAT and admission, correlation between scaled SAT and admission

Neural Networks

Structure:

- 6 input features
- 3 output nodes (spinal disease predictions)
    - Classification problem: the class with the largest value is used
    - Regression problem: the values are averaged
- 3 nodes in hidden layer



Learning Method:

- Each node receives a different linear combination of prior layer’s nodes
- Without activation, we would be generating linear combinations of linear combinations, which is still linear
- Activation Function: applied to linear combinations so that the next layer receives nonlinear transformations

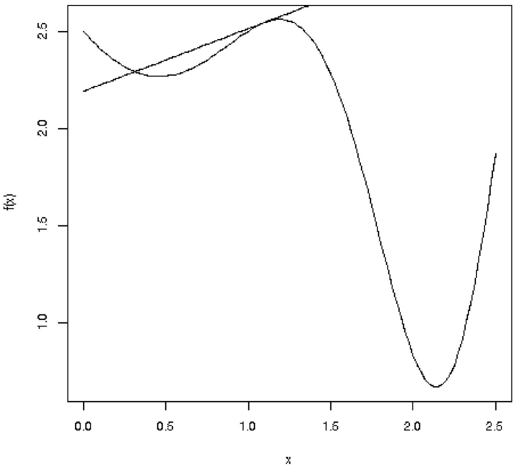
Hyperparameters:

- Activation function type
    - Hyperbolic tangent (tanh)
    - Logistic
    - Rectified linear unit (relu) - max(0, x)
- Layer Geometry
    - Number of layers
    - Type of layers
    - Number of nodes per layer
- Number of Iterations
- Learning Rate

Issues

- May not converge - scaling will reduce likelihood of nonconvergent iterations

Though far more complex than in the linear case, we are still in the calculus realm. We compute the partial derivatives of the sum of squares with respect to the n_w weights, and set the results to 0s. So, we are finding roots of a very complicated function in n_w dimensions, and we need to do so iteratively.


f(x)
2.5
2.0
1.5
1.0
0.0
0.5
1.0
1.5
2.0
2.5
x