Barrett Honors Enrichment Contract, Fall 2024
Course: Discrete Mathematical Structures (MAT 243)
Instructor: Adam Leighton, Student: _____

This project is adapted from one written by Dr. Rochus Boerner (Arizona State University).

Consider the task of predicting something about individuals coming from a certain population based on some number of their other features. For example, predicting for whom a randomly selected 2024 American voter would vote as president, based on their age and annual income. Here, one could formulate such a prediction by assuming these features are in a bivariate function, say $V: A \times W \rightarrow P$, where A is the set of all ages of 2024 American voters, W is the set of all annual incomes of 2024 American voters in 2024, and P is the set {0, 1}, where 1 represents one of the 2024 presidential candidates and 0 represents the others.

To estimate V's rule, one could use any rule that adequately relates these features for a random sample of the population's individuals. That is, if the age, annual income, and presidential candidate preference for n randomly selected 2024 American voters was collected (for example, in the table below), denoted $\{(a_i, w_i, p_i)\}^{n}_{i=1}$, any rule, say $\hat{V}$, that maps the subset of $A \times W$ corresponding to the sampled individuals to P could be used as an estimate of that for V. Then, predicting how an a*-year old American voter earning $w* annually would vote amounts to interpreting $\hat{p} = \hat{V}(a^*, w^*)$.

	Age	Annual Income	Voted For
Person 1	66	200000	Trump
Person 2	21	30000	Biden
Person 3	38	700000	Trump
Person 4	25	120000	Biden
Person 5	81	50000	Biden
Person 6	43	79000	Biden
Person 7	78	120000	Trump

Deciding which rule to use for $\hat{V}$ is the next subtask. One way is to by assuming it's of the form $\hat{V}(a, w) = [g(\beta_a a + \beta_w w + \beta_0)]$, where β_a is the marginal linear effect of an American voter's age on their voting preference, β_n is the marginal linear effect of an American voter's annual income on their voting preference, β_0 is a so-called intercept term that accounts for American voters' preferences as a whole, $g: \mathbb{R} \rightarrow [0, 1]$ is a so-called link function (specified by the user, commonly chosen to be defined by the rule $g(t) = \frac{1}{1+e^{-t}}$ in contexts like that in this example), and [.] represents the round-to-the-nearest-integer function.

At this point, the task of estimating $\hat{V}$'s rule amounts to determine the values of $(\beta_a, \beta_w, \beta_0)$ –called the model's parameters–that, in some way, make $\hat{V}$ adequately describe the data. Usually, these values are determined by picking some reasonable (or random) starting values, say $(\beta_a, \beta_w, \beta_0)_0$, for the parameters then algorithmically updating them through iteration in an effort to minimize some error (or loss) function. A common error function is $E(\beta_a, \beta_w, \beta_0) = \sum_{i=1}^{n} (g(\beta_a a + \beta_w w + \beta_0) - p_i)^2$. Since E is a differentiable function of the parameters, locating its minimum can be done by finding the roots of its derivatives and analyzing the geometry of its graph thereabout.

Although finding the derivatives of E with respect to each of its parameters is a calculus I exercise, finding their roots can be challenging. Often, this root-finding is done through an iterative procedure, the most common of which is the so-called gradient descent method defined by the recursive sequence $(\beta_a, \beta_w, \beta_0)_{k+1} = (\beta_a, \beta_w, \beta_0)_k - h\nabla E((\beta_a, \beta_w, \beta_0)_k)$, where $h > 0$ is some user-specified constant and $\nabla E = \left(\frac{\partial E}{\partial \beta_a}, \frac{\partial E}{\partial \beta_w}, \frac{\partial E}{\partial \beta_0}\right)$ is the gradient of E. The number of terms of this sequence that are computed when implementing the method of gradient descent depends on K, the maximum number of iterations allowed by the user, and $\epsilon > 0$, a small, user-specified constant called the convergence criterion. The algorithm stops if K iterations have occurred or if $h \cdot \left| \nabla E\left((\beta_a, \beta_w, \beta_0)_k\right) \right| < \epsilon$, where $|(x, y, z)| = \sqrt{x^2 + y^2 + z^2}$, whichever comes first, returning $(\beta_a, \beta_w, \beta_0)_K$ as the final answer.

(Some notes on jargon: In the machine-learning field, people like to give the elements of this approach special names. Namely, $\hat{V}$ is called an artificial neuron because it allegedly resembles how neurons work in the brain, $[\,.\,] \circ g$ is called an activation function, the root-finding process is called back-propagation, and h is called the learning rate. In practice, neural networks consist of compositions of perceptrons, each referred to as a layer, used to model highly complex relationships dictated by supposed non-linear models.)

To earn credit for this project, you must prepare a report addressing the following using only the information in this document, discussions with and resources approved by (ask!) your instructor, and your own prior knowledge. Your report shouldn't be an itemized list like the one below, but rather a narrative with graphics, tables, and so on. Indicate which parts of your report correspond to which of these steps by putting the step numbers thereafter, for example <1>.

1) Choose a population of interest for which you're interested in some sort of covariable relationship.
2) Find a data set of one quality and at least two quantities for at least 111 individuals from that population. Give these features one-letter variable names then write the function definition for their relationship.
3) Divide the data into a so-called training group of approximately 90% of the individuals and a so-called testing group of the remaining individuals.
4) Using the link and loss functions from above, express an artificial neuron for the quality in terms of the remaining features and newly-defined, appropriately named parameters.
5) Use calculus to determine the gradient of the loss function.
6) In terms of your training group size (say G) and maximum number of allowed iterations (say K), how many operations would be done in implementing the gradient descent method, assuming that it doesn't converge?
7) Choose initial values for the parameters and values for the learning rate, convergence criterion, and maximum number of allowed iterations.
8) Write a computer program or use a spreadsheet to implement the gradient descent method for locating the "best" artificial neuron, based on the individuals in the training group alone.
9) Evaluate the "best" artificial neuron for the individuals in the testing group to predict their value of the quality of interest. What percentage of these predictions are correct?

10) Repeat Steps 7-9 to try to get the greatest correct prediction percentage in Step 9. Discuss your observations and include your results for your “best” choices as well as those for at least two different values of each of the items in Step 7.