CSC173 - Intelligent Systems
Binary Classifier Neural Network

Ramel Cary B. Jamen
ramelcary.jamen@g.msuiit.edu.ph

October 4, 2024

Binary Classifier Neural Network

Binary classification is a fundamental task in machine learning, where the goal is to categorize data into one of two classes or categories. Binary classification is used in a wide range of applications, such as spam email detection, medical diagnosis, sentiment analysis, fraud detection, and many more.

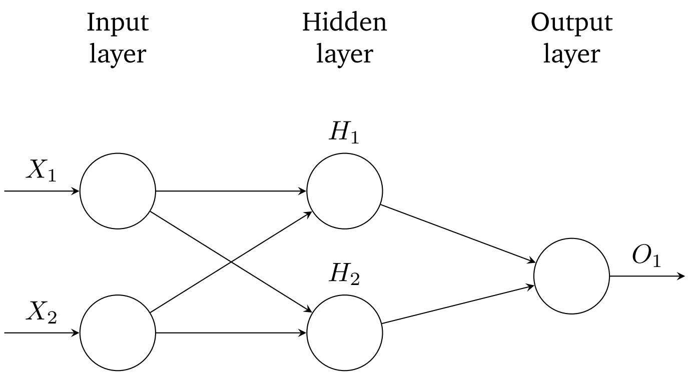
An example dataset from the website "A Visual Interactive Guide to Neural Network" found in this URL: (https://jalammar.github.io/visual-interactive-guide-basics-neural-networks) about house and lot.

Area (sq ft) (x1)	**Bathrooms (x2)**	**Classification (y)**
2,104	3	Good
1,600	3	Good
2,400	3	Good
1,416	2	Bad
3,000	4	Bad
1,985	4	Good
1,534	3	Bad
1,427	3	Good
1,380	3	Good
1,494	3	Good

Table 1: Property Classification Based on Area and Bathrooms

Do some experiment with Binary Classification Network and answer the following:

1. Design the neural network architecture that will model the classification of the house and lot based on the above dataset.



2. Identify the inputs and output variables in your model.

   Input and Output Variables: The input variables for the neural network are *Area (sq ft)* and *Number of bathrooms*. These features represent the size of the property and the number of bathrooms, respectively. The output of the neural network is a binary classification that predicts whether the property is considered *good* or *bad*.

3. Provide the initial values for weights and biases.

   np.random.normal() is a function from the NumPy library used to generate random numbers that follow a normal (Gaussian) distribution, In which it will be used for the inital values for weights and biases.

4. Identify your loss (or cost) and activation functions.

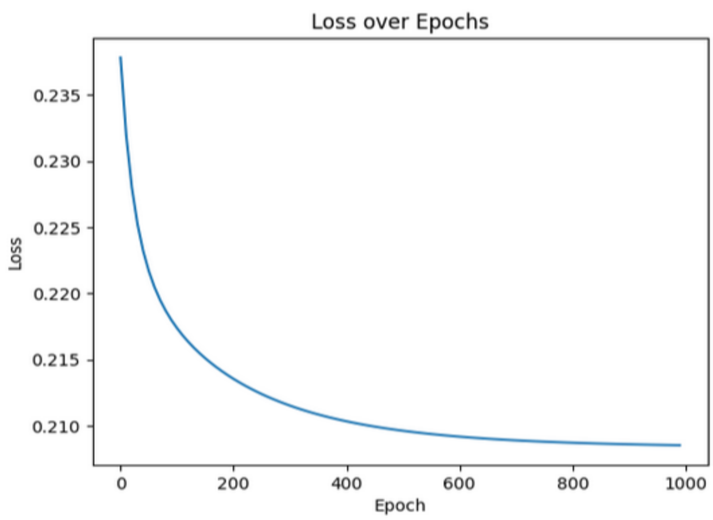
   Sigmoid Activation Function: The sigmoid function is used as the activation function for the hidden layer neurons (h1 and h2) and the output layer neuron (o1). It maps any real-valued number into the range (0, 1), making it suitable for binary classification tasks.

   Mean Squared Error (MSE) Loss Function: The MSE loss function measures the average squared difference between predicted and actual values. It is commonly used for regression problems and provides a straightforward measure of the model's prediction error.

5. Determine your learning rate and epochs.

   Learning Rate and Epochs A Learning rate of 0.01 and epochs of 1000, a smaller learning rate can lead to more precise results but might require more epochs to converge. A larger number of epochs gives the model more opportunities to learn but increases training time and might lead to overfitting if too large.

6. Collect all loss values during training and plot it to show the graph.



7. Collect the final value of the weights and biases and use it to classify at least two example property that is not found in the dataset.

Weights	**Biases**	**Loss**
$w_1 = -0.7151$ $w_2 = 1.0745$ $w_3 = -0.5823$ $w_4 = 0.6126$ $w_5 = 1.1394$ $w_6 = -1.4181$	$b_1 = 0.0660$ $b_2 = -0.8518$ $b_3 = 0.9979$	$\text{Loss} = 0.2085$

```
# Make some predictions
newData = np.array([1166.0, 0.9])
newData2 = np.array([-304, -.1])
print("newData: %.3f" % network.feedforward(newData))
print("newData2: %.3f" % network.feedforward(newData2))
```

```
newData: 0.748
newData2: 0.670
```

Summary

The implementation focuses on building a binary classifier neural network using a dataset related to property classification based on area and the number of bathrooms. Initially, the dataset is loaded using the `pandas` library, and the first few rows of the data are displayed using the `head()` method. To visualize the distribution of the features, histograms for the `area` and `bathrooms` are plotted. Subsequently, normalization is performed on both features. The mean of the `area` and `bathrooms` columns is computed and subtracted from the respective columns to create new normalized columns, `area_normalized` and `bathrooms_normalized`

After normalization, a scatter plot is generated to visualize the relationship between normalized area and bathrooms, with the data points colored according to their classification. This visualization helps in understanding how the features correlate with the target variable.

The neural network architecture is defined as a simple feedforward model with one hidden layer and one output layer. The sigmoid activation function is utilized in both the hidden and output layers to introduce non-linearity in the model.

The training process is initiated with a defined learning rate and a set number of epochs. For each training example, the model performs a feedforward calculation to compute the output based on the current weights and biases. The Mean Squared Error (MSE) is calculated to evaluate the difference between the predicted output and the true classification. Backpropagation is then applied, which involves calculating gradients with respect to the weights and biases and updating them accordingly to minimize the loss. The loss values are recorded at intervals during training to monitor the model's performance.

Finally, the loss values are plotted over the epochs, illustrating the convergence of the model as it learns to predict property classifications effectively. The entire process showcases the implementation of a neural network for binary classification using feature normalization, data visualization techniques, and a structured approach to training and optimization.

For the complete code implementation, visit: RCJamen GitHub Repository.