Stage-1

Source of Data:

https://www.kaggle.com/datasets/jillanisofttech/brain-stroke-dataset

No Copyrights. This data set represents records of people who had or had not had a brain-stroke before. It consists of 10 columns. The columns are gender, age, hypertension , heart_disease, ever_married, work_type, Residence_type, avg_glucose_level, bmi, smoking_status and stroke.

cleaning data:

All records were filled and there are no empty cells in this data. This data set met the requirements for this assignment. However, I decided to convert 3 string columns into numerical columns because the string in these columns are binary (0 or 1), and converting them from now would enhance the efficiency of the learning model that we want to build and would reduce the number of string inputs to our model. The columns that I changed are gender, ever_married, Residence_type.

Visualizations:

- Pie chart: represent the percentage of people who had or had not had a brain-stroke before.
- Histogram: counter for age groups in the data set.
- Par Chart: Divide the people in the data into groups based on smoking_status.

Stage-2

Columns:

The number of numerical inputs is 8. They are gender, age, hypertension, heart_disease, ever_married, Residence_type, avg_glucose_level, bmi. The output column is the stroke column and it is a binary value. The aim of this model is to provide a tool of prediction based on data that we have, and I want to see if we can predict if a group of data might lead us to a result such as a stroke or no stroke.

Splitting Data:

The data for training is 75%, and the data for testing is 25%. My decision is to provide more data for training to help the model in generalization. Moreover, we have a considerably good amount for training that should lead to a similar result as the training's result.

Model:

The model is the same exact model used in MINIST.ipynb, but I have changed the hyper-parameters. I used 4 layers. The first one is the input layer. The second and third layers are the hidden layers. Each of these layers consists of 128 neurons. I added a dropout layer before the output with 0.25 probability. The last layer is the output, and the number of neurons is 1 since it is a binary not a category classification.

Hyper-parameters:

I have changed the hyper-parameters; therefore it is not the same as MINIST. The reason for changing them is that my output is a binary data with 0 or 1. For the input and hidden layer, we use the RELU as an activation function, and the last layer uses the SIGMOID. The used optimizer is ADAM, and for loss function, my choice was binary_crossentropy. The metric is accuracy. The number of epochs is 10.

Result:

The train accuracy is 94.9% with loss at 0.1735 and the test is 95.3% with loss at 0.1759.

.