Brain Stroke Classification

Authors:
Taliya Shitreet 314855099
Renana Rimon 207616830

Abstract:

A brain stroke is a life-threatening medical disorder caused by the inadequate blood supply to the brain. After the stroke, the damaged area of the brain will not operate normally. As a result, early detection is crucial for more effective therapy.
Given a patient's medical and personal history, deep learning methods can be used to predict with high accuracy whether the patient will have a stroke.

This paper presents a process of improving results using different deep learning methods.
First, we used Machine Learning method: Logistic Regression with test accuracy of 75%.
Second, we used Multilayer Perceptron (MLP) with an accuracy of 86%.
At last, we used a recurrent neural network (RNN) model using long short-term memory (LSTM) layers for the classification of brain stroke, the model achieved good performance, with a test accuracy of 92%.
All models were trained and evaluated on a dataset containing 10 features and a binary label indicating the presence or absence of brain stroke.

Introduction:

Brain stroke is a serious and potentially life-threatening condition that occurs when the blood supply to the brain is disrupted. It is a leading cause of death and disability worldwide, and early diagnosis and treatment can improve patient outcomes and reduce the risk of long-term complications.
The symptoms of brain stroke can vary depending on the type and location of the stroke, but common symptoms include weakness or numbness on one side of the body, difficulty speaking or understanding speech, and vision problems.
Traditionally, the diagnosis and classification of brain stroke have been based on clinical examination, imaging studies, and laboratory tests. However, these methods can be time-consuming and may not always be accurate. In recent years, there has been growing interest in the use of machine learning techniques for the diagnosis and classification of brain stroke, as they have the potential to provide faster and more accurate results.
In this paper, we propose a machine learning approach for the classification of brain stroke based on a dataset containing 10 features and a binary label. Our approach uses a Logistic Regression model, Multilayer Perceptron (MLP) model and recurrent neural network (RNN) model with long short-term memory (LSTM) layers, which are well-suited for modeling time series data and capturing long-term dependencies. We evaluate the performance of the model on the dataset and discuss the implications of our findings.

Logistic regression is a type of supervised learning algorithm that is used for binary classification. It is a linear model that is used to predict the probability of a binary outcome (e.g., 0 or 1) based on one or more independent variables. In logistic regression, the output is a probability between 0 and 1, and a threshold is used to classify the input as either 0 or 1.
Multilayer perceptron (MLP) is a type of artificial neural network that consists of one or more hidden layers of artificial neurons (called "perceptrons") between the input and output layers. MLP is a supervised learning algorithm that is used for classification and regression tasks. It is capable of learning non-linear relationships between the input and output data.
Recurrent neural network (RNN) is a type of artificial neural network that is designed to process sequential data, such as time series or natural language. RNNs have a "memory" in the form of hidden states, which allows them to capture dependencies between the input data at different time steps. RNNs are used for tasks such as language translation, speech recognition, and time series prediction.
Long short-term memory (LSTM) is a type of recurrent neural network (RNN) that is specifically designed to capture long-term dependencies in sequential data. LSTM networks are composed of "memory cells" that can store and retrieve information over a long period of time, as well as input, output, and forget gates that control the flow of information into and out of the memory cells.

Related works:

There have been several previous studies that have explored the use of machine learning techniques for the diagnosis and classification of brain stroke.
A Deep Learning Approach for Detecting Stroke from Brain CT Images Using OzNet[1] aimed to classify brain stroke CT images using OzNet and hybrid algorithms.
In this paper they used OzNet-mRMR-NB that created the new structure achieved an accuracy of 98.42% to detect stroke from brain CT images.
Natural language processing and machine learning algorithm to identify brain MRI reports with acute ischemic stroke[2]
This article represent assessed performance of natural language processing (NLP) and machine learning (ML) algorithms for classification of brain MRI radiology reports into acute ischemic stroke (AIS) and non-AIS phenotypes.
During the process they used binary logistic regression, naïve Bayesian classification, single decision tree, and support vector machine for the binary classifiers. Of all ML algorithms, single decision tree had the highest F1-measure 93.2%, and accuracy 98%.
Prediction of stroke thrombolysis outcome using CT brain machine learning[3]
Linical records and CT brains of 116 acute ischemic stroke patients treated with intravenous thrombolysis were collected retrospectively (including 16 who developed SICH). The sample

[1] https://www.mdpi.com/2306-5354/9/12/783/pdf

[2] https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0212778

[3] https://www.sciencedirect.com/science/article/pii/S2213158214000217

was split into training (n =106) and test sets (n = 10), repeatedly for 1760 different combinations. CT brain images acted as inputs into a support vector machine (SVM) And get accuracy of 77% and identified 9 out of 16 SICHs.

Required background :

Unlike the above articles, our data are not images of the brain, but personal and medical history data. This allows faster detection, without the need of CT or MRI.
To understand the data well, prior knowledge of symptoms and connections to mental sabbath is necessary. We will have to understand the relationship of each feature to stroke.
Gender: Stroke has a **greater effect on women** than men because women have more events and are less likely to recover. Age-specific stroke rates are higher in men, but, because of their longer life expectancy and much higher incidence at older ages, women have more stroke events than men.
Age: Stroke occurs in all age groups. Studies show the risk of stroke doubles for each decade **between the ages of 55 and 85**. But strokes also can occur in childhood or adolescence.
Hypertension: can cause stroke through many mechanisms. **A high intraluminal pressure will lead to extensive alteration in endothelium and smooth muscle function in intracerebral arteries**. The increased stress on the endothelium can increase permeability over the blood-brain barrier and local or multifocal brain oedema.
Heart disease: heart disease and stroke are **closely related** because they are both caused by problems with the circulatory system. The circulatory system is the body's system for pumping blood to the body's tissues and organs. When the circulatory system is not working properly, it can lead to a range of health problems, including heart disease and stroke.
Marriage: Marriage has been linked to a **lower risk** of stroke, while not being married has been linked to a higher risk of stroke. married people tend to have better physical and mental health overall, which can reduce their risk of stroke. They also tend to have more social support, which can help them manage stress and other risk factors for stroke.
Work type: There is some evidence that **being unemployed or working in a low-paying or low-status job may be associated with an increased risk of stroke**, while working in a high-paying or high-status job may be associated with a decreased risk of stroke. This may be due to the stress and financial insecurity that can be associated with unemployment or low-paying work, which may increase the risk of stroke.
Residence type: There is some evidence that **living in a rural area may be associated with an increased risk of stroke**, while living in an urban area may be associated with a decreased risk of stroke. There are several potential reasons for this.
One reason may be that rural areas often have fewer resources and less access to healthcare, which can make it more difficult to manage risk factors for stroke and receive prompt treatment if a stroke does occur.
Glucose: Elevated levels of glucose in the blood, also known as hyperglycemia, can increase the risk of stroke. **High blood glucose levels** can damage blood vessels and **increase the risk** of complications such as heart disease and stroke.
BMI: Body mass index (BMI) is a measure of body fat based on height and weight. **A high BMI**, which is defined as a BMI of 25 or greater, has been linked to an **increased risk of stroke**. This is

because excess body fat, especially abdominal fat, can increase the risk of conditions such as high blood pressure, diabetes, and high cholesterol, which are all risk factors for stroke.
Smoking : **smoking is a major risk factor for stroke**. It damages blood vessels and increases the risk of blood clots, which can lead to a stroke. In addition, smoking increases the risk of other conditions that can increase the risk of stroke, such as high blood pressure and diabetes.

Project description:

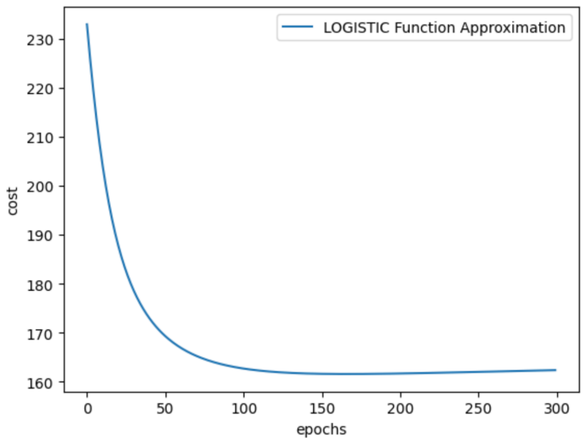
This data set has 4981 rows and 11 columns – 10 features and 1 label.
248 rows of class 1 and 4733 rows of class 0.
Due to the distribution of the '1' in relation to the '0', the model may learn biasedly, and even classify everything to 0.
In that's reason during our preprocessing we added 1's examples to the dataset such that y label divided like 4733 of class 1 and 4733 of class 0. Therefore, the data shape now is (9466, 11). In addition we changed the data to numeric, and normalize it with standard scaling.
The columns in the data set are:
Features - **gender**: "male" (0), "female" (1), **age**: age in years, **hypertension**: a binary variable indicating whether the individual has hypertension(1) or not(0), **heart_disease**: a binary variable indicating whether the individual has heart disease(1) or not(0), **ever_married**: a binary variable indicating whether the individual has ever been married(1) or not(0), **work_type**: "children"(0), "Govt_job"(1), "Never_worked"(2), "Private"(3) or "Self-employed"(4), **Residence_type**: the type of residence the individual lives in (rural(0) or urban(1)), **avg_glucose_level**: the average glucose level, **bmi**: body mass index, **smoking_status**: "formerly smoked"(0), "never smoked"(1), "smokes"(2) or "Unknown"(3).
Label: **stroke**: a binary variable indicating whether the individual has had a stroke(1) or not(0).

The purpose of the article is to notice the improvement in the learning of the different models. Starting with a machine learning model (logistic regression), simple neural networks (MLP) and ending with complex networks (RNN, LSTM).
We will describe how each model works on our data, and in addition the accuracy and the loss.

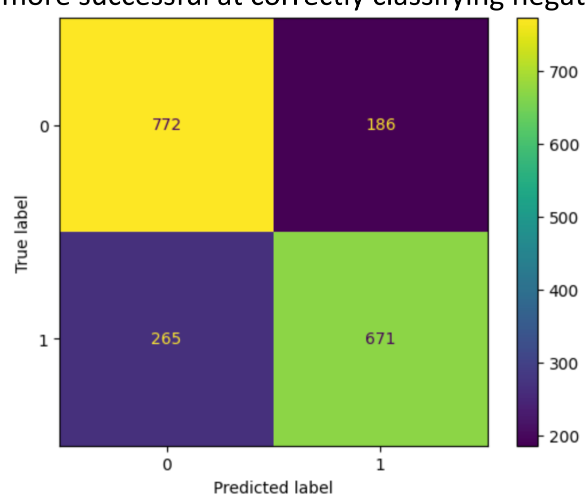
Logistic Regression Model

Our code defines a logistic regression model in TensorFlow using placeholders for the input data (x) and labels (y_), and variables for the model weights (W) and biases (b). The model produces a predicted probability of the positive class (pred) using a sigmoid activation function applied to the linear combination of the input data and model weights, plus the biases.
The model is trained using stochastic gradient descent with a learning rate of 0.01, using a cross-entropy loss function and the sigmoid_cross_entropy_with_logits function to calculate

the loss. The model's parameters (weights and biases) are updated in each iteration by minimizing the loss using the GradientDescentOptimizer.
The model is trained for 300 epochs (iterations over the entire training dataset) in mini-batches of size 300. The model's performance on the validation data is evaluated by calculating the L2 loss (mean squared error) between the predicted probabilities and the true labels, and the cost is printed out every 100 epochs.
Results:
The model was able to correctly classify about 75.5% of the test data.

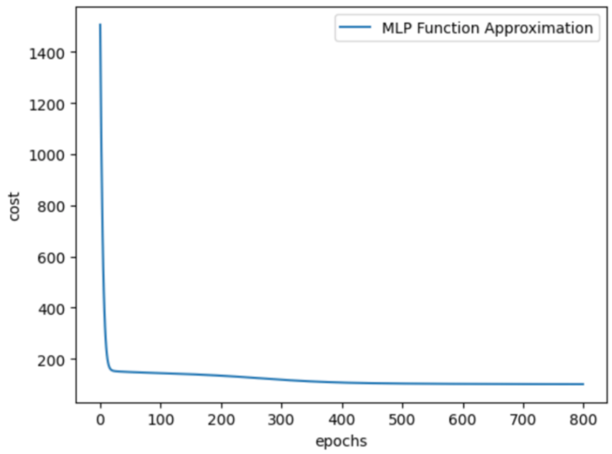


And according to the confusion matrix the model had 772 true positive cases, 186 false positive cases, 265 false negative cases, and 671 true negative cases. This means that the model was more successful at correctly classifying negative cases than positive cases.

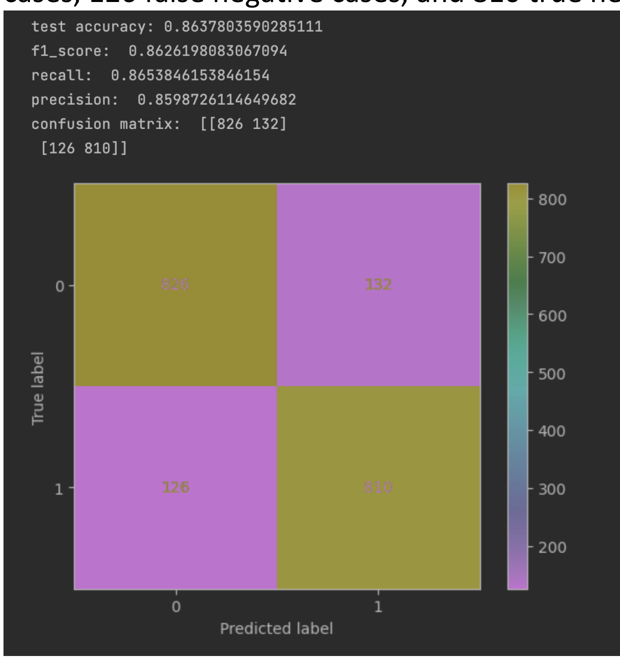


<u>MLP Model</u>
We wrote a code defines a Multilayer Perceptron (MLP) model in TensorFlow 1 for training on our dataset. The MLP has two layers: a hidden layer with 10 neurons and an output layer with 1 neuron. The input features are represented by the placeholder **X** and the labels by the placeholder **Y**. The model is trained using the Adam optimizer and the L2 loss function between the predicted output and the true labels. The model is trained using mini-batch gradient descent with a batch size of 300 for 800 epochs. At each epoch, the cost is computed on the validation set and stored in the **errors** list. The cost is printed every 100 epochs to track the progress of the training.

Results:
The model was able to correctly classify about 86% of the test data.



According to the confusion matrix the model had 826 true positive cases, 132 false positive cases, 126 false negative cases, and 810 true negative cases.



With this model we improved the accuracy percentage by 8.5%!

<u>RNN Model</u>
First, we reshaped the data since the LSTM model needs to receive 3D data as input in the following shape: (samples, time-series, features) and therefore we added a time-series dimension and set it to 1. This means that our model will actually treat the input data as a single snapshot of the data, instead of a sequence of data over time, it is sufficient because in this case because the input data is not continuous or time series data, and the order of the data points is not important.

We defined a model with the following architecture:

1. An LSTM layer with 32 units and input shape of (None, 10). The LSTM layer is a type of Recurrent Neural Network (RNN) layer, which is able to capture temporal dependencies in the data. The input shape of (None, 10) means that the input data has a time dimension with shape (batch_size, 1) and 10 features. The "None" value allows the model to accept any batch size.
2. A dense layer with 1 unit and a sigmoid activation function. This is the output layer, which will produce a single output value between 0 and 1, representing the probability that the input data belongs to the positive class (in this case, the probability of having a brain stroke).

The model is then compiled with a binary cross-entropy loss function and the Adam optimization algorithm, and metric of accuracy. This means that the model will try to minimize the difference between the predicted probability and the true label (0 or 1) by using the Adam optimizer to update the model's weights and biases. The accuracy metric will track the fraction of the time that the model predicts the correct class.

Finally, the model is fit to the training data using the fit() method. This will train the model on the input data (X_train_time) and label data (y_train) for 800 epochs (iterations over the entire dataset) with a batch size of 300. The validation data (X_val_time and y_val) is also provided, which the model will use to evaluate its performance during training and help prevent overfitting to the training data.

After training, the model's performance on the test data (X_test_time and y_test) is evaluated using the evaluate() method, which returns the loss and accuracy of the model on the test data. The loss is a measure of how well the model is doing on the test data, and the accuracy is the fraction of the test data that the model correctly classifies.

Results:

During training, the model achieved an accuracy of 97% on the training data and an accuracy of 94% on the validation data.

On the test data, the model achieved an accuracy of 92%.