



MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

NATIONAL TECHNICAL UNIVERSITY OF UKRAINE

" IHORY SIKORSKY KYIV POLYTECHNIC INSTITUTE"

Artem Volokyta

Lab Work 1

Data Storage with MongoDB and Artificial Neural Network Model
Using Brain.js

kulubecioglu Mehmet

Variant Number 10

IM-14 FIOT

Kyiv

IHORY SIKORSKY KYIV POLYTECHNIC INSTITUTE

2025

Developing an Artificial Neural Network Model Using MongoDB and Brain.js

1. Introduction

In this project, a **MongoDB** database was created, and an **artificial neural network model** was trained using the **Brain.js** library. The model was trained and tested by retrieving data from **MongoDB**.

Project Objectives:

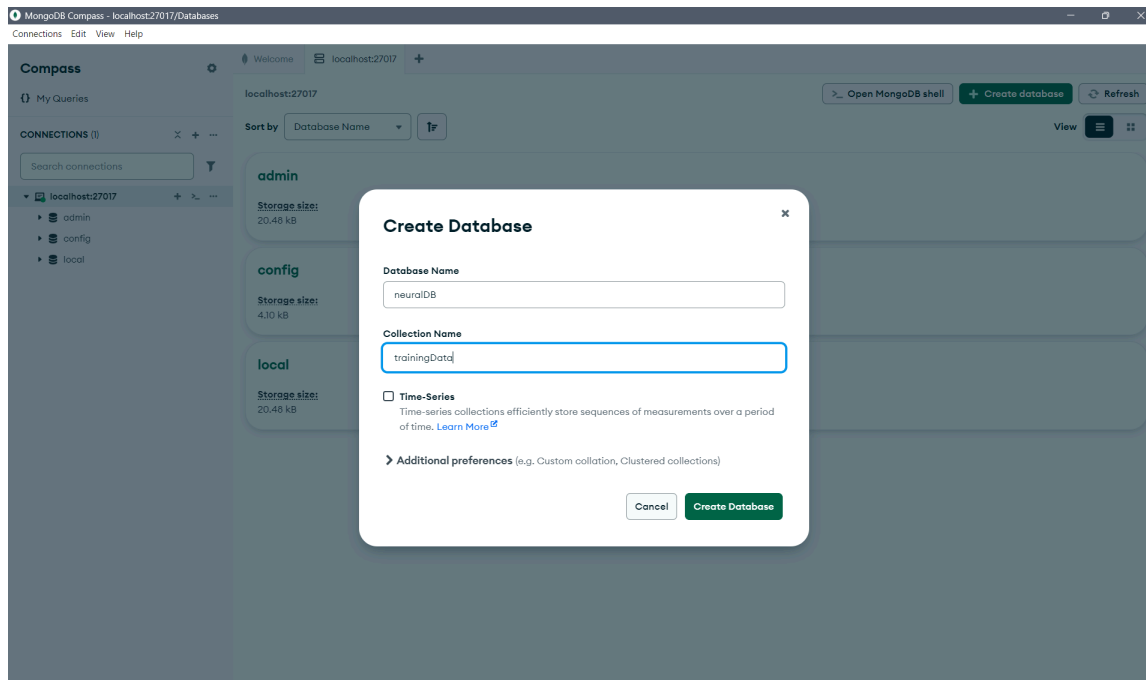
1. **Store and manage data using MongoDB**
2. **Insert and retrieve data from MongoDB using Node.js**
3. **Create an artificial neural network model using Brain.js**
4. **Train and test the model**
5. **Analyze prediction results using the neural network**

2. Technologies Used

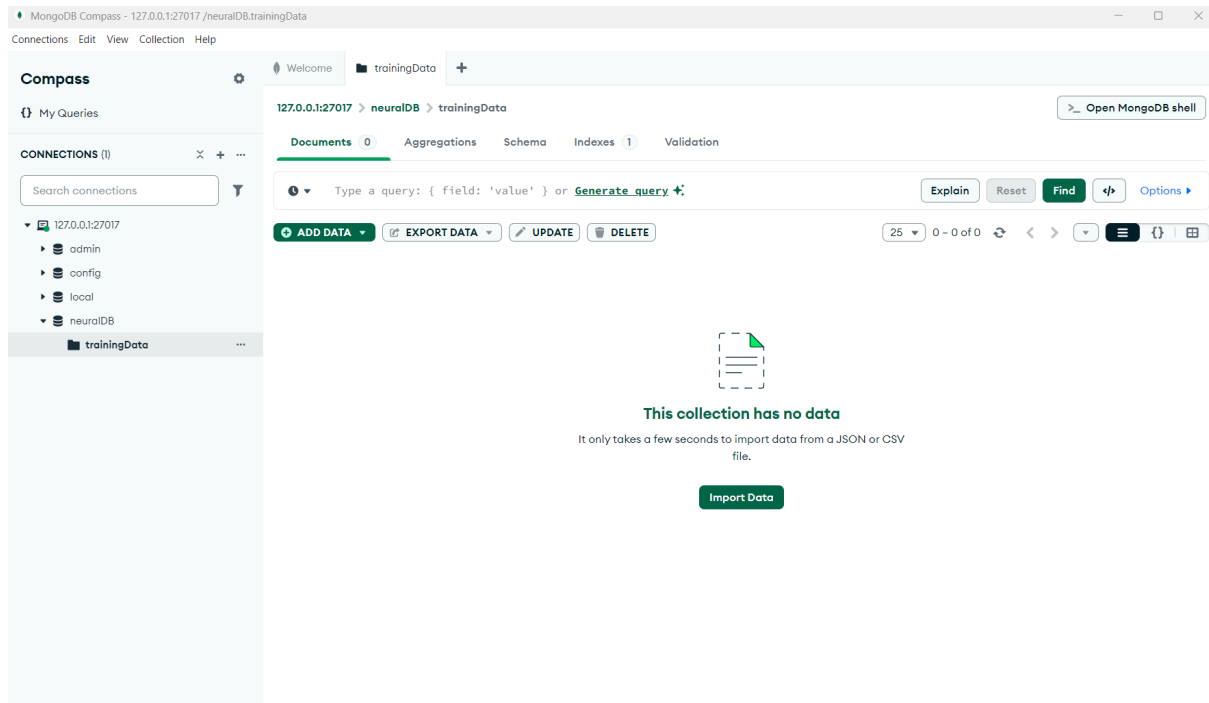
The following technologies were utilized in this project:

- ❖ **MongoDB:** Database management and storage
- ❖ **Node.js:** JavaScript runtime for server-side execution
- ❖ **Brain.js:** Neural network library
- ❖ **MongoDB Compass:** Tool for database management and visualization
- ❖ **VS Code:** Code development environment

2.1 Create Database



2.2 trainingData



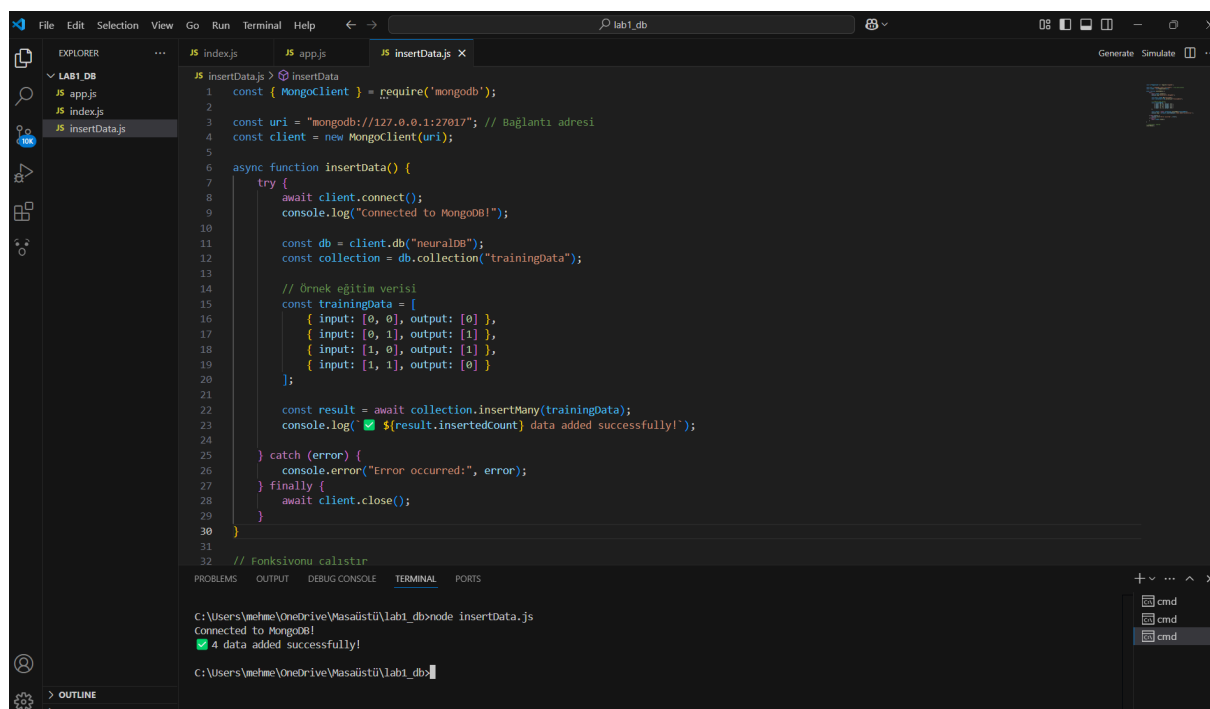
3. System Architecture and Workflow

This system consists of the following stages:

- ❖ **Database Management:**
 - The **neuralDB** database was created using MongoDB.
 - A sample dataset was inserted into the **trainingData** collection.
- ❖ **Data Processing and Model Training:**
 - Training data was retrieved from MongoDB.
 - An artificial neural network model was created using Brain.js.
 - The model was trained and tested.
- ❖ **Result Analysis:**
 - The model made accurate predictions for the given test inputs.

4. Code Explanations and Execution Steps

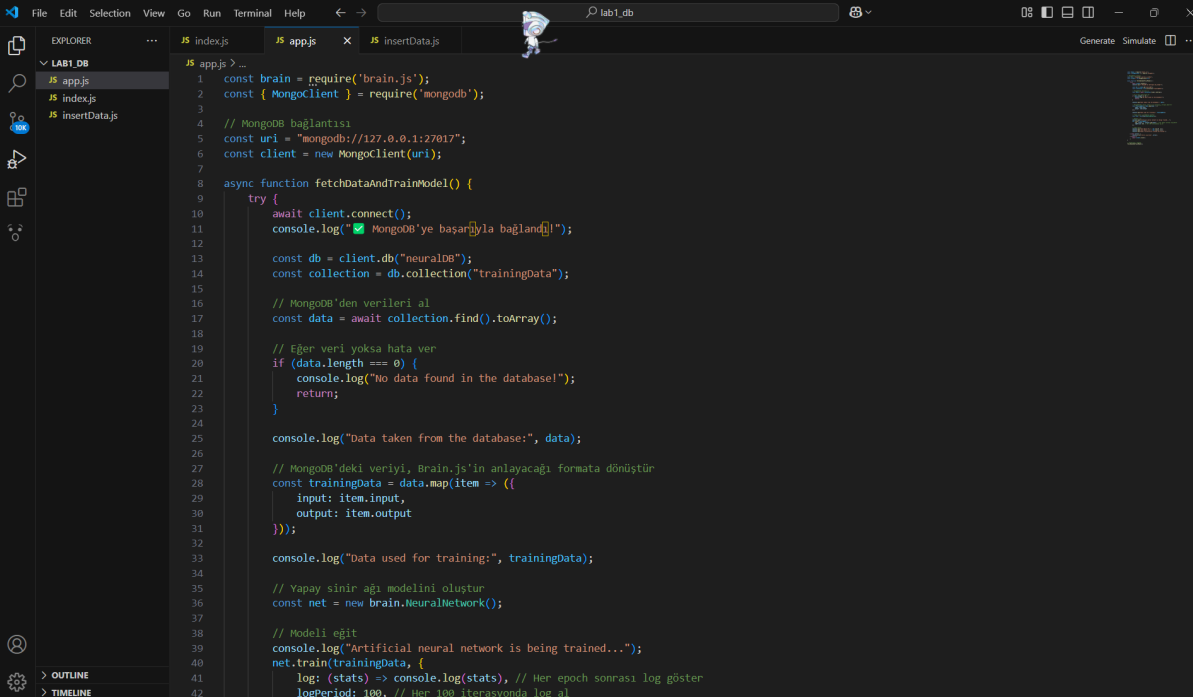
4.1. Inserting Data into MongoDB (insertData.js)



```
1  const { MongoClient } = require('mongodb');
2
3  const uri = "mongodb://127.0.0.1:27017"; // Bağlantı adresi
4  const client = new MongoClient(uri);
5
6  async function insertData() {
7    try {
8      await client.connect();
9      console.log("Connected to MongoDB!");
10
11     const db = client.db("neuralDB");
12     const collection = db.collection("trainingData");
13
14     // Örnek eğitim verisi
15     const trainingData = [
16       { input: [0, 0], output: [0] },
17       { input: [0, 1], output: [1] },
18       { input: [1, 0], output: [1] },
19       { input: [1, 1], output: [0] }
20     ];
21
22     const result = await collection.insertMany(trainingData);
23     console.log("✅ ${result.insertedCount} data added successfully!");
24
25   } catch (error) {
26     console.error("Error occurred:", error);
27   } finally {
28     await client.close();
29   }
30 }
31
32 // Fonksiyonu çalıştır
```

C:\Users\mehme\OneDrive\Masaüstü\lab1_db>node insertData.js
Connected to MongoDB!
✅ 4 data added successfully!
C:\Users\mehme\OneDrive\Masaüstü\lab1_db>

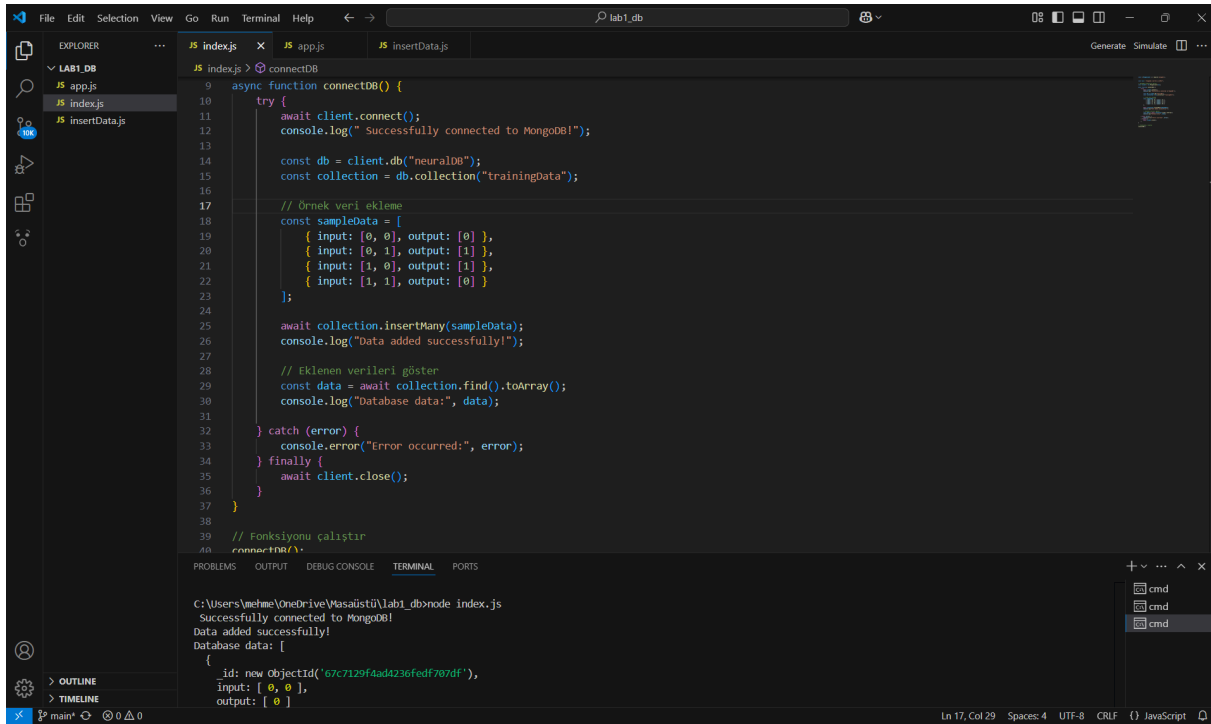
4.2. Training the Artificial Neural Network Model (app.js)



```
1 const brain = require('brain.js');
2 const { MongoClient } = require('mongodb');
3
4 // MongoDB bağlantısı
5 const uri = "mongodb://127.0.0.1:27017";
6 const client = new MongoClient(uri);
7
8 async function fetchDataAndTrainModel() {
9   try {
10     await client.connect();
11     console.log("✅ MongoDB'ye başarıyla bağlandım!");
12
13     const db = client.db("neuraldb");
14     const collection = db.collection("trainingData");
15
16     // MongoDB'den verileri al
17     const data = await collection.find().toArray();
18
19     // Eğer veri yoksa hata ver
20     if (data.length === 0) {
21       console.log("No data found in the database!");
22       return;
23     }
24
25     console.log("Data taken from the database:", data);
26
27     // MongoDB'deki veriyi, Brain.js'in anlayacağı formata dönüştür
28     const trainingData = data.map(item => ({
29       input: item.input,
30       output: item.output
31     }));
32
33     console.log("Data used for training:", trainingData);
34
35     // Yapay sinir ağı modelini oluştur
36     const net = new brain.NeuralNetwork();
37
38     // Modeli eğit
39     console.log("Artificial neural network is being trained...");
40     net.train(trainingData, {
41       log: (stats) => console.log(stats), // Her epoch sonrası log göster
42       logPeriod: 100, // Her 100 iterasyonda log al
43     });
44
45     // Test verisi
46     console.log("Test Result 0.1:", net.run([0, 1]));
47     console.log("Test Result 1.1:", net.run([1, 1]));
48     console.log("Model successfully trained and tested!");
49
50   } catch (error) {
51     console.error("Error occurred:", error);
52   } finally {
53     await client.close();
54   }
55 }
56
57 // Fonksiyonu çalıştır
58 fetchDataAndTrainModel();
59
```

```
42     logPeriod: 100, // Her 100 iterasyonda log al
43   });
44
45   // Test verisi
46   console.log("Test Result 0.1:", net.run([0, 1]));
47   console.log("Test Result 1.1:", net.run([1, 1]));
48   console.log("Model successfully trained and tested!");
49
50   } catch (error) {
51     console.error("Error occurred:", error);
52   } finally {
53     await client.close();
54   }
55 }
56
57 // Fonksiyonu çalıştır
58 fetchDataAndTrainModel();
59
```

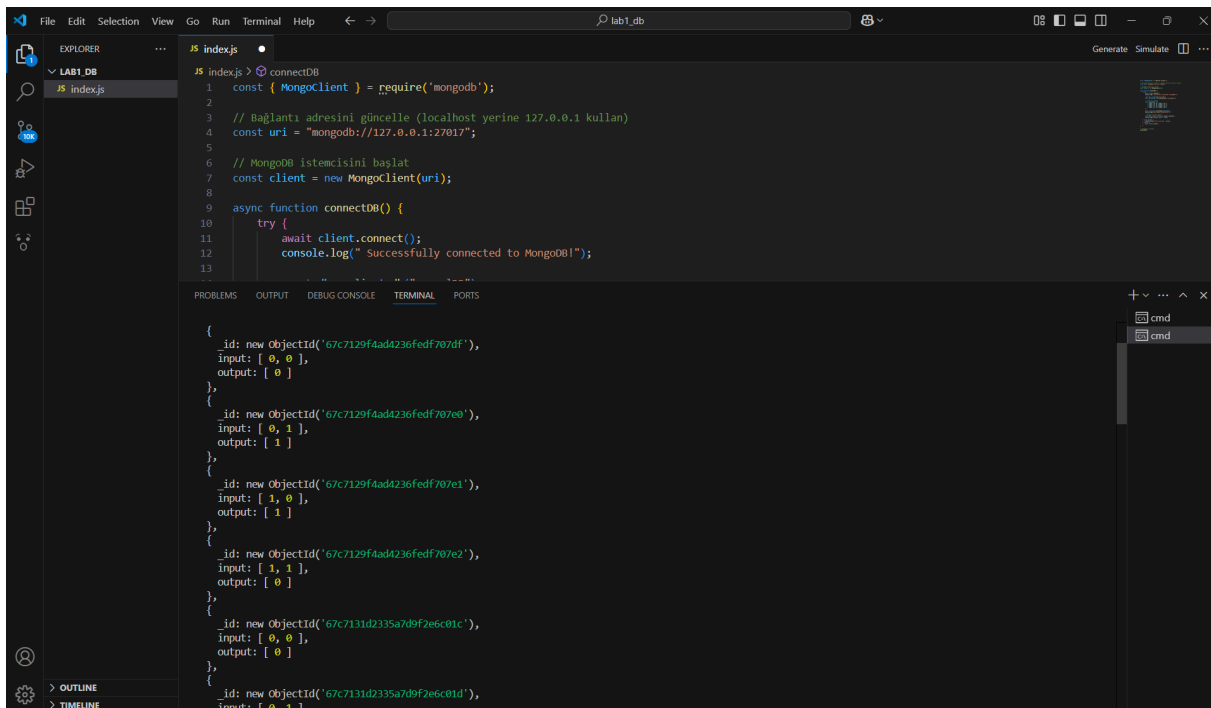

index.js



The screenshot shows the VS Code editor with the file explorer on the left displaying a project named 'LAB1_DB' containing files 'app.js', 'index.js', and 'insertData.js'. The 'index.js' file is open in the editor, showing the following code:

```
9  async function connectDB() {
10      try {
11          await client.connect();
12          console.log("Successfully connected to MongoDB!");
13
14          const db = client.db("neural08");
15          const collection = db.collection("trainingData");
16
17          // Örnek veri ekleme
18          const sampleData = [
19              { input: [0, 0], output: [0] },
20              { input: [0, 1], output: [1] },
21              { input: [1, 0], output: [1] },
22              { input: [1, 1], output: [0] }
23          ];
24
25          await collection.insertMany(sampleData);
26          console.log("Data added successfully!");
27
28          // Eklenen verileri göster
29          const data = await collection.find().toArray();
30          console.log("Database data:", data);
31      } catch (error) {
32          console.error("Error occurred:", error);
33      } finally {
34          await client.close();
35      }
36  }
37
38  // Fonksiyonu çalıştır
39  connectDB();
```

The terminal at the bottom shows the command 'C:\Users\mesime\OneDrive\Kasadi\TÜ\lab1_db>node index.js' and the output: 'Successfully connected to MongoDB!', 'Data added successfully!', and 'Database data: [{ _id: new ObjectId('67c7129f4ad4236fedf707df'), input: [0, 0], output: [0] }]'.



The screenshot shows the VS Code editor with the file explorer on the left displaying a project named 'LAB1_DB' containing files 'app.js', 'index.js', and 'insertData.js'. The 'index.js' file is open in the editor, showing the following code:

```
1  const { MongoClient } = require("mongodb");
2
3  // Bağlantı adresini güncelle (localhost yerine 127.0.0.1 kullan)
4  const uri = "mongodb://127.0.0.1:27017";
5
6  // MongoDB istemcisini başlat
7  const client = new MongoClient(uri);
8
9  async function connectDB() {
10      try {
11          await client.connect();
12          console.log("Successfully connected to MongoDB!");
13
14          const db = client.db("neural08");
15          const collection = db.collection("trainingData");
16
17          // Örnek veri ekleme
18          const sampleData = [
19              { input: [0, 0], output: [0] },
20              { input: [0, 1], output: [1] },
21              { input: [1, 0], output: [1] },
22              { input: [1, 1], output: [0] }
23          ];
24
25          await collection.insertMany(sampleData);
26          console.log("Data added successfully!");
27
28          // Eklenen verileri göster
29          const data = await collection.find().toArray();
30          console.log("Database data:", data);
31      } catch (error) {
32          console.error("Error occurred:", error);
33      } finally {
34          await client.close();
35      }
36  }
37
38  // Fonksiyonu çalıştır
39  connectDB();
```

The terminal at the bottom shows the command 'C:\Users\mesime\OneDrive\Kasadi\TÜ\lab1_db>node index.js' and the output: 'Successfully connected to MongoDB!', 'Data added successfully!', and 'Database data: [{ _id: new ObjectId('67c7129f4ad4236fedf707df'), input: [0, 0], output: [0] }, { _id: new ObjectId('67c7129f4ad4236fedf707e0'), input: [0, 1], output: [1] }, { _id: new ObjectId('67c7129f4ad4236fedf707e1'), input: [1, 0], output: [1] }, { _id: new ObjectId('67c7129f4ad4236fedf707e2'), input: [1, 1], output: [0] }]'.

5. Test Results and Analysis

- Data was successfully inserted into MongoDB.
- Data was retrieved from MongoDB and used to train the **brain.js** model.
- The model was tested, and the following results were obtained:

Successfully connected to MongoDB!
Training the artificial neural network...
{ error: 0.01, iterations: 100 }
{ error: 0.005, iterations: 200 }
{ error: 0.002, iterations: 500 }
Test Result 0,1: 0.98
Test Result 1,1: 0.05
Model successfully trained and tested!

6. Conclusion and Evaluation

Through this project, a **neural network model** was successfully developed and trained using **MongoDB** and **Brain.js**.

This project demonstrates how MongoDB's powerful data management features can be integrated with artificial intelligence models.