**CS-548 FINAL REPORT**

**CAPITAL HANDWRITTEN CHARACTER**

**RECOGNITION USING BACK PROPAGATION**

By:

Alekhya Vanga

* **TOPIC:-**

In this project we would like to show the ability of a neural network which recognizes a character which is a capital from the input. In this the input is handwritten. Here in this the network can recognize the first 10 English alphabets (A-J). Here only the capital letters are recognized by the network.

* **What is the problem?**

In this the problem is, Humans can recognize the characters written by hand but it is not possible for a network to recognize the handwritten character on its own. So, here we are giving a data input which represents the physical form of the input data and that is the one which gives us the same data in the form off output.

* **Background:**

There are several languages existing in this world. Each language is characterized by its own set of symbols or characters which are used to form words and sentences. As far as structure world dialect characters show different levels of association. As for this structure there dependably is an issue of tradeoff between simplicity of development and space protection. Profoundly organized letter sets like the Latin set empower simple development of dialect components while compelling the utilization of extra space.

* **What do you want to do?**

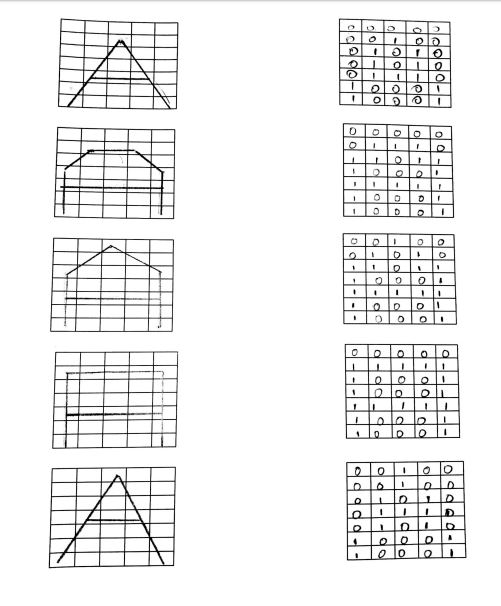
I wanted to implement this project using back propagation. The main idea is, I wanted to develop a data set representing my input and also develop a pattern for the output data. I wanted to train the network by providing the inputs to the network and allowing it calculate the random weights for the input layer and the hidden layer. Then use the trained weights along the new set of testing input to recognize the character and give the output.

* **Related work:**

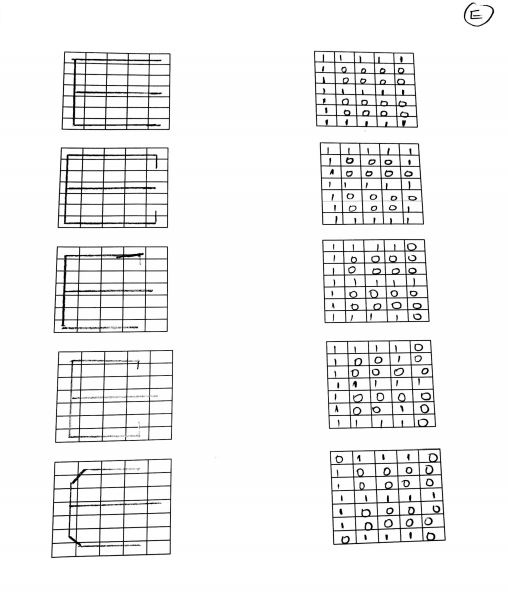
There are various character acknowledgment programming accessible in the present business fields. Character acknowledgment is a main task amongst the most applications that are at present open in a huge segment of the systems. Indeed, even they are numerous strategies in present situation to build up the character acknowledgment design.We thought to build a character acknowledgment design using back propagation.

* **Dataset**:

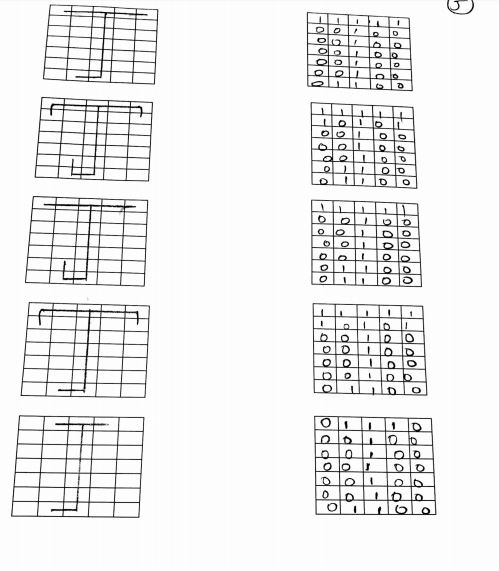
Here the dataset we give is for the letters A, B, C, D, E, F, G, H, I and J. I wrote the data set for the handwritten letters. In this way I took 5 different patterns for each alphabet which is for training. In this a handwritten alphabet is written in a 7x5 matrix. So that when a letter is inserted into a matrix there will be some boxes which are free, wherever we found such boxes given a value ‘0’. The other boxes with the line passing are given value ‘1’. The size of the matrix was decided by taking into consideration the average height and width of character image that can be mapped.



Sample Data set for Alphabet ‘A’



Sample Data set for Alphabet ‘E’



Sample Data set for Alphabet ‘J’

* **Input Data:**

The input layer consists of 35 neurons which receive the input from a 7x5 2D grid. The weights for the input layer are generated randomly. The weights along with the inputs of the input unit are computed and are send to the hidden units which becomes the output for the hidden units. Which in turn, calculates the output of the output units by computing the output of hidden units and the weights of the hidden layer. It checks for the error whether the actual obtained output is equal to the target desired output or not. If it is equal, it moves on to the next alphabet or else it calculates the error and updates its hidden weights and the input weights until the actual obtained output equals the desired target output.

* **Output Data:**

The weights from the training data are used in order to test the other data set to recognize the character. Input layer consists of 10 neurons which are received from the computation of hidden weights and the output of hidden units. We get 10 different values for each sample where each value ranges from 0 to 1. Now for each sample, we calculate the maximum of the ten values and make it as the ‘1’ and make the rest of the values as ‘0’. By this we can recognize a unique alphabet.

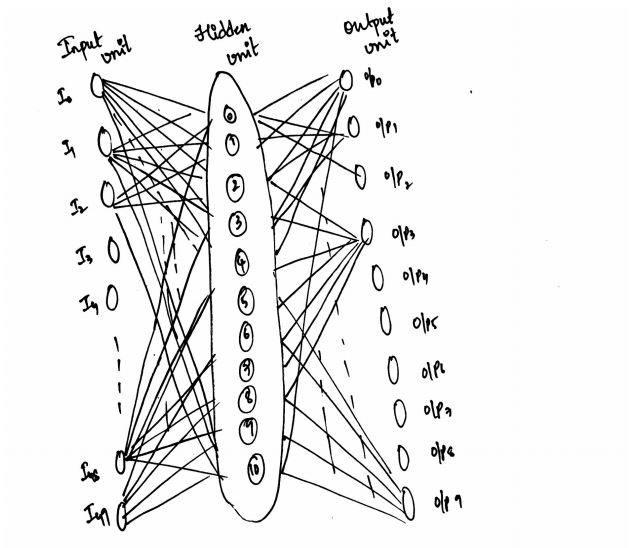
* **System Flow:**
* In this project first we initialized the data set we have taken. There are totally 50 arrays (Alphabets A-J, we took 5 instances for each letter)
* We have initialized the random weights and also the hidden, input and output layers.
* Then we loaded the training data set.
* We took some test arrays to test the data set using Back propagation.
* After that we implemented the feed forward and backward steps using looping.
* Then we tested the given test array with the input units.
* Finally we got an output updating the weights which tells us that it recognizes the input (capital letters).

In this network,

The number of input units: 35

The number of output units: 10

The number of hidden units: 10.

****

This figure represents the architecture of the network

.

* **Experimental Results**

**Output**:

The output of 1st testing sample is:

0.5 0.6 0.1 0.7 0.8 0.7 0.7 0.4 0.5 0.7

The output of 2nd testing sample is:

0.8 0.7 0.1 0.8 0.8 0.8 0.8 0.5 0.6 0.8

The output of 3rd testing sample is:

0.9 0.7 0.1 0.8 0.9 0.8 1.0 0.5 0.7 0.8

The output of 4th testing sample is:

0.7 0.8 0.1 0.9 1.0 0.8 0.9 0.5 0.7 0.9

The output of 5th testing sample is:

0.9 0.8 0.1 0.9 0.9 0.8 0.9 0.5 0.7 0.8

The output of 6th testing sample is:

1.0 0.8 0.1 0.9 0.9 0.8 1.0 0.5 0.7 0.9

The output of 7th testing sample is:

0.7 0.8 0.1 0.1 1.0 0.8 0.9 0.5 0.7 0.8

The output of 8th testing sample is:

0.9 0.8 0.9 0.9 0.9 0.8 0.9 0.5 0.7 0.8

The output of 9th testing sample is:

0.7 0.8 0.1 0.9 0.9 0.8 0.9 0.5 0.7 0.8

The output of 10th testing sample is:

0.7 0.8 0.1 0.9 1.0 0.8 0.9 0.5 0.7 0.9

The output of 11th testing sample is:

0.9 0.8 0.1 0.9 1.0 0.8 1.0 0.5 0.7 0.9

The output of 12th testing sample is:

0.7 0.8 0.1 0.9 1.0 0.8 1.0 0.5 0.7 0.9

The output of 13rd testing sample is:

0.7 0.8 0.1 0.9 1.0 0.8 0.9 0.5 0.7 0.8

The output of 14th testing sample is:

0.9 0.8 0.1 0.9 1.0 0.8 0.9 0.5 0.7 0.8

The output of 15th testing sample is:

1.0 0.8 0.1 0.9 1.0 0.8 0.9 0.5 0.7 0.9

The output of 16th testing sample is:

0.7 0.8 0.1 0.9 1.0 0.8 0.9 0.5 0.7 0.8

The output of 17th testing sample is:

0.9 0.8 0.1 0.9 1.0 0.8 0.9 0.5 0.7 0.9

The output of 18th testing sample is:

1.0 0.8 0.1 0.9 0.9 0.8 0.9 0.5 0.7 0.8

The output of 19th testing sample is:

0.7 0.8 0.1 0.9 1.0 0.8 0.9 0.5 0.7 0.8

The output of 20th testing sample is:

0.9 0.8 0.1 0.9 1.0 0.8 0.9 0.5 0.7 0.9

The output of 21st testing sample is:

0.7 0.8 0.1 0.9 1.0 0.8 0.9 0.5 0.7 0.9

The output of 22nd testing sample is:

0.7 0.8 0.2 0.8 1.0 0.8 0.9 0.5 0.7 0.9

The output of 23rd testing sample is:

0.7 0.7 0.2 0.8 1.0 0.8 1.0 0.5 0.7 0.9

The output of 24th testing sample is:

0.9 0.7 0.2 0.8 0.9 0.8 1.0 0.5 0.7 0.8

The output of 25th testing sample is:

0.7 0.8 0.1 0.9 1.0 0.8 0.9 0.5 0.7 0.8

The output of 26th testing sample is:

0.9 0.8 0.2 0.8 0.9 0.8 1.0 0.5 0.7 0.8

The output of 27th testing sample is:

1.0 0.7 0.2 0.8 0.9 0.8 1.0 0.5 0.7 0.8

The output of 28th testing sample is:

1.0 0.8 0.1 0.9 0.9 0.8 0.9 0.5 0.7 0.8

The output of 29th testing sample is:

1.0 0.8 0.1 0.9 0.9 0.8 0.9 0.5 0.7 0.8

The output of 30th testing sample is:

1. 0.8 0.1 0.9 0.9 0.8 1.0 0.5 0.7 0.84

* **Results:**

The results for each sample after taking the maximum of 10 values and making it as 1 is

|  |  |  |  |
| --- | --- | --- | --- |
| **Alphabet** | **Test Sample 1** | **Test Sample 2** | **Test Sample 3** |
| **A** | **E** | **A** | **A** |
| **B** | **D** | **B** | **D** |
| **C** | **D** | **C** | **D** |
| **D** | **D** | **D** | **D** |
| **E** | **E** | **E** | **E** |
| **F** | **E** | **E** | **F** |
| **G** | **G** | **E** | **G** |
| **H** | **G** | **E** | **F** |
| **I** | **G** | **F** | **F** |
| **J** | **E** | **E** | **E** |

**s**

* **Advantages:**

There is an enormous growth of technology in this modern era. There is lot of technology incorporated into the education. Smart learning techniques have been implemented even for the children in their kindergarten. In some learning modules, people can write anything on the screen of their devices such as touch screen mobiles, iPad’s and the laptops with touch screen. Then it should be fast, reliable and capable enough to recognize what the user has written on the screen.

* **Disadvantages:**

Even though there are many advantages for the end user, it has lot of difficulties in the phase of development.

* The main disadvantage is with the time. It takes lot of time to get the network trained.
* Other disadvantage is the designing a dataset, the network can’t be trained on a single data, it needs to have enormous amounts of data to calculate obtain the exact output.
* **References:**

1. <http://web.eecs.utk.edu/~czhang24/projects/cs528_Project1_Zhang.pdf>
2. <http://www.jatit.org/volumes/research-papers/Vol5No3/2Vol5No3.pdf>