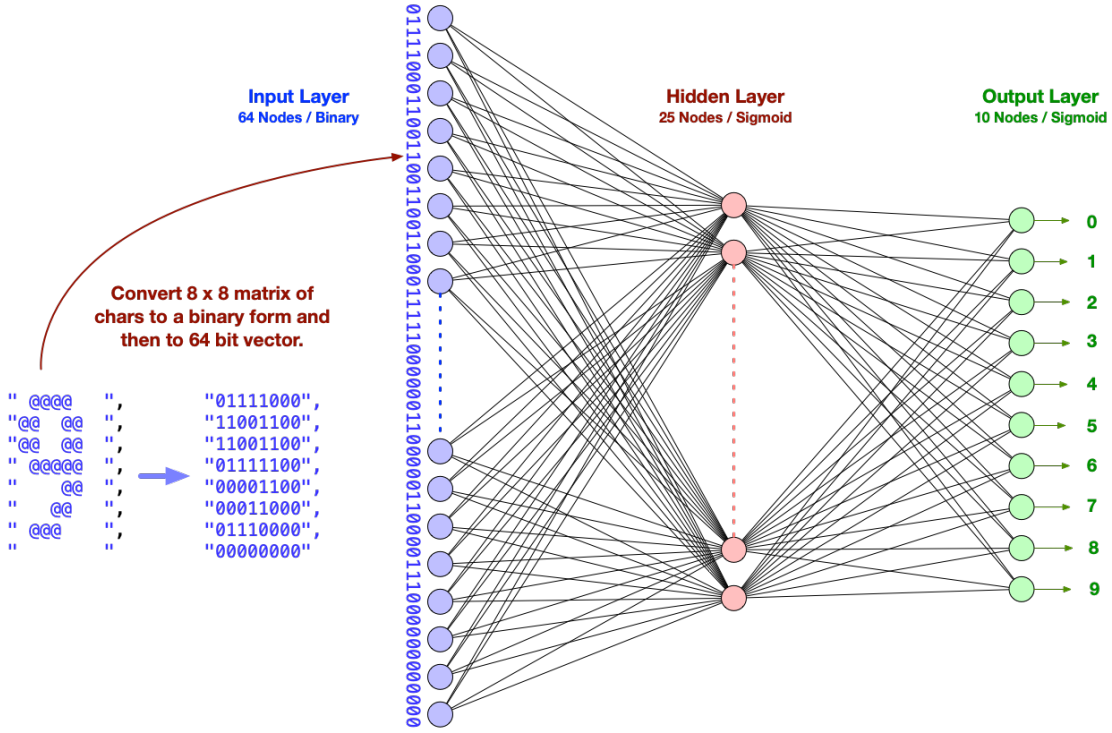
Department of Computer Science & Applied Physics

Classifying a Number Pattern

Neural networks excel at identifying hidden patterns in data and in using this information to make a prediction of a number (regression) or a type (classification). If you look at the number "9" in the image below, although it is composed of a set of "@" symbols, the neural network of our brain has no trouble interpreting the information and identifying it as a number. We can encode the symbols in the number as a 0 for a space and a 1 for an "@" symbol, flat-map the set of symbols into a vector, and a neural network will be able to interpret the resultant binary sequence.



In this lab, we will train a neural network to correctly identify the numbers 0 – 9 encoded as shown above. We will then test the stability and robustness of the trained model by changing some of the symbols in the encoded number and using this as test data. We could also directly encode the "@" symbols using the underlying Unicode number and then normalise the resultant input vector within a range suitable for the activation function that we have chosen for the first hidden layer. Note that the encoding above could be swapped with LED lights and the sequence of bits transmitted to the network through an encoded electrical signal.

Exercises

- Copy the class *PatternRunner.java* into the correct package of a project. Look through the source code and examine how the *String[][] numerals* encodes the numbers and the *String[][] year* encodes the test data. Note how the test data contains errors. The method *translate(String[][] s)* flat-maps the *String[][] numerals* into training data. ***Why does the***

data not need to be normalised?

- Use the class *NetworkBuilder* to create a neural network with the following configuration:

  Input Layer: 64
  Hidden Layer: 25 (TanH)
  Output Layer: 10 (TanH) with Softmax
  Alpha: 0.01
  Beta: 0.95
  Epochs: 100000
  Min Error: 0.000001
  Loss Function: Sum of Squares

- Use the statement `out.println(net)` to output the internal state of the neural network.

- Test the trained model with the *String[][] year* and output the result:

```
var sb = new StringBuffer();
var test = translate(year);
for (int i = 0; i < test.length; i++){
   sb.append((int) net.process(test[i], Output.LABEL_INDEX));
}
out.println("Result: " + sb.toString());
```

- Add / remove characters from the test data and see how long it takes before the network can no longer accurately identify the date.