As an exploration of COS coding I have re-implemented a basic neural network system, originally in Java, and posted at https://medium.com/coinmonks/implementing-an-artificial-neural-network-in-pure-java-no-external-dependencies-975749a38114.

The Atelier project can be found at \\refiles\scratch\ryoung\SimpleNN.

The project can be run from the terminal in a couple of ways:

1. do ##class(ISC.Sample.NN).main()

   Displays statistics of learning process.

   e.g.

   ```
   =================
   Cost = .005619489458253496
   Predictions = [[0.01146,0.98732,0.98720,0.01448]]
   Iteration 4000. Elapsed time = 349.020259 seconds. Loop time = .08725506475 seconds.
   ```

2. do ##class(ISC.Sample.NN).main(1, 48, 132)

   Where the last two arguments are the rows and columns of your terminal window.

   This graphically shows the process of the network learning the prediction 0,1,1,0. It kicks in around iteration 3,800. E.g.

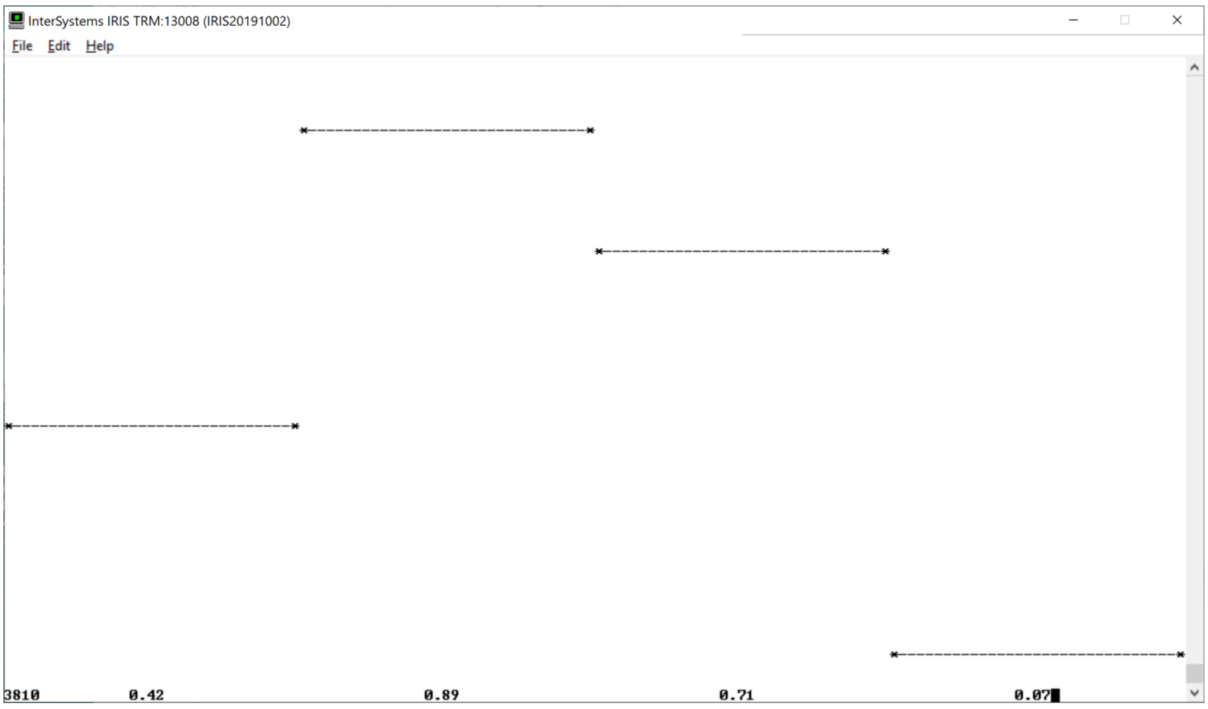

Figure 1 Dynamic plot of learning process

Array Handling

Apart from syntax the main difference in coding was the way arrays are handled. In COS the arrays were built dynamically with %Push(val), rather than specifying the whole array, and allocating memory, prior to use. E.g.

```
classMethod add(a, b) {
    set m = a.%Size()
    set n = a.%Get(0).%Size()
    set c = []

    for i = 0:1:m-1 {
             set arr = []
        for j = 0:1:n-1{
             do arr.%Push(a.%Get(i).%Get(j) + b.%Get(i).%Get(j))
        }
        do c.%Push(arr)
    }
    return c
}
```

Are there better ways to handle arrays?

Performance

There is a big difference in performance between the COS implementation and that of Java. E.g.

```
==============

Cost = 0.012759997825067041

Predictions = [[0.011011323053764899, 0.9874919378077374, 0.987347693544872,
0.014540865394245846]]

Iteration 4000. Elapsed time = 1.123 seconds. Loop time = 0.000281 seconds.

-----------------
```

That is, 349.020 seconds for COS against 1.123 seconds for Java. In other words, Java is about 300 times faster. Is this what you would expect? Are there ways that better performance can be achieved with the COS code?