CS-687 Assignment

Neural Networks

Anthony Sermania
Aakarshika Priydarshi

NAND and XOR

(net-build (convert-data *nand*) 3 1.0 5 20000 1000 t)
(net-build (convert-data *xor*) 3 1.0 5 20000 1000 t)

Both these functions converge on an average 2 out of 5 times.
The values used for various parameters are as follows:
Hidden neurons- 3
Alpha- 1.0
Initial bound- 5
Maximum iterations- 20000
Modulo- 1000

For the successfully converged run, the final iterations errors were in the range:
Worst error: 0.049394495 1.095704e-6
Mean error: 0.001236073 4.312458e-7

For unsuccessful runs:
Worst error: 0.080000695 0.005954131
Mean error: 0.043875602 0.004687714

Voting Records

Net-Build

(net-build (convert-data *voting-records*) 10 0.75 5 10000 500 t)

Worst error: 0.0049688066
Mean error: 0.4583382e-4

(net-build (convert-data *voting-records*) 15 0.75 5 2000 250 t)

Worst error: 0.025502264
Mean error: 9.190441e-4

(net-build (convert-data *voting-records*) 15 0.50 5 2000 250 t)

Worst error: 0.0064448463
Mean error: 5.4844434e-4

The optimum alpha for this data seems to be 1.0 with 10 hidden units. If the number of hidden neurons is increased, or the alpha decreased, the error increases.

Simple-Generalization

```
(simple-generalization (convert-data *voting-records*) 10 1.0 5 5000)
```

Mean error: 0.020959353

```
(simple-generalization (convert-data *voting-records*) 15 1.0 5 5000)
```

Mean error: 0.023795906

```
(simple-generalization (convert-data *voting-records*) 10 0.75 5 5000)
```

Mean error: 0.017280307

```
(simple-generalization (convert-data *voting-records*) 10 0.75 5 10000)
```

Mean error: 0.017316049

The mean error of simple generalization for voting records increases when hidden units are increased from 10 to 15.
If we keep hidden neurons constant to 10, the mean error decreased with a decrease in alpha, or an increase in number of max- iterations.

MPG

Net-Build

```
(net-build (convert-data *mpg*) 10 1.0 5 5000 250 t)
```

Worst error: 1039.6799
Mean error: 282.291

```
(net-build (convert-data *mpg*) 10 0.75 2 10000 500 nil)
```

Worst error: 1039.6799
Mean error: 282.291

This mean error and worst error remains nearly constant throughout the net-build run for this data.

Simple-Generalization

```
CL-USER¿ (simple-generalization (converted-data *mpg*) 10 1.0 2 5000)
373.0813
CL-USER¿ (simple-generalization (converted-data *mpg*) 10 1.0 2 10000)
373.0813
CL-USER¿ (simple-generalization (converted-data *mpg*) 15 1.0 2 2000)
373.0813
CL-USER¿ (simple-generalization (converted-data *mpg*) 8 1.0 2 2000)
373.0813
CL-USER¿ (simple-generalization (converted-data *mpg*) 8 0.5 2 2000)
373.0813
CL-USER¿ (simple-generalization (converted-data *mpg*) 8 0.75 5 2000)
373.0813
```

Simple generalization on ”mpg” records gives the exact same **mean error: 373.0183** for all these parameters.
This is irrespective of change in alpha, hidden neurons, or iterations.

Wine

Net-Build

```
(net-build (converted-data *wine*) 10 0.75 2 10000 500 nil)
```

```
Worst error: 0.3522382
Mean error: 0.25597098
```

```
(net-build (converted-data *wine*) 10 0.75 5 5000 250 nil)
```

```
Worst error: 0.31726447
Mean error: 0.23408356
```

```
(net-build (converted-data *wine*) 13 1.0 5 5000 250 nil)
```

```
Worst error: 0.3394108
Mean error: 0.24198712
```

Simple Generalization

```
CL-USER¿ (simple-generalization (converted-data *wine*) 10 1.0 5 5000)
0.43855825
```

```
CL-USER¿ (simple-generalization (converted-data *wine*) 10 0.75 5 5000)
0.42623
CL-USER¿ (simple-generalization (converted-data *wine*) 15 0.75 5 5000)
0.35706532
CL-USER¿ (simple-generalization (converted-data *wine*) 15 0.5 5 5000)
0.4000918
```

Mean error for wine data is the least with optimum value for alpha as 0.75.
Optimum number of hidden neurons is 15.