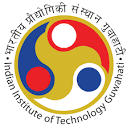
**INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI**

****

**DEPARTMENT OF MECHANICAL ENGINEERING**

**ME 674 (Soft Computing)**

**Coding Assignment 1**

**SUBJECT INSTRUCTOR: PROF. Sukhomay Pal**

**Assignment on Artificial Neural Network**

**Submitted by: KRUNAL GAUTAM**

**Roll No: 214103309**

**TABLE OF CONTENTS**

1. Introduction to the Problem
2. Brief introduction about theory of the Problem
3. Selecting dataset from the problem
4. Applying Artificial Neuron Network to train the model
5. Calculation part and output of result
6. Conclusion
7. Introduction to the Problem

The lattice Boltzmann technique may be used to correctly analyze fluid dynamics in complicated porous media such as gas diffusion layers (GDLs) in polymer electrolyte fuel cells (PEFCs) (LBM).

This document presents the results of a study that estimated diffusion parameters such as porosity, gas phase tortuosity, and diffusibility using simulated porous media.

When a water drop barrier is placed inside the GDL domain and the size of the water drop is adjusted, the data are calculated. Diagrams depicting the evolution of the flow velocity field are also included, as well as visuals depicting the change in local and bulk porosity for each barrier size.

Finally, there is a full method explanation of how the lattice Boltzmann technique is implemented, as well as a general description of computational codes for domain and obstacle creation, as well as simulation of boundary conditions.

1. Brief introduction about theory of the Problem

* **The Boltzmann technique on a lattice (LBM)**

As we are solving the fluid dynamics problem since there are large number of differential equations the traditional methods used in computational fluid dynamics are little bit slow so give faster and better solutions LBM is most efficient technique now a days.

The velocity ﬁeld inside the porous media needs to reach a steady state condition before it can be used to calculate diffusion parameters.

The LBM was used to acquire the calculated data on the fluid velocity. At the mesoscale scale, this approach works best on complicated structures. It starts with a digitally created porous material that represents a GDL and contains a water drop .

* **Steps involve in the experiment**
* Porous media and water drop generation
* Flow ﬁeld velocity calculation using the Lattice Boltzmann method
* Fluid ﬂow analysis and boundary conditions
* Solving the algorithm
* Diffusion parameters calculation

**Boundary conditions considered in the data obtention**.

|  |  |  |
| --- | --- | --- |
| Applied zone | Boundary Condition | Description |
| Inlet surface | Von Neumann | The condition is applied as the inlet velocity is known. This is a Velocity Boundary that applies to momentum conservation. |
| Outlet surface | Second Derivative Approximation | As the outlet velocities and pressure are unknown, the conditions are approximated using the velocities of the two previous lattice elements. |
| Parallel to the ﬂow surfaces | Periodic Boundary | In short, this consideration states that the velocities of the nodes outside the domain are the same as the velocities on the opposite boundary that enters the volume in the same direction. |
| Solid-Fluid Interface | Bounce Back Condition | When the ﬂuid node collides with a solid interface, the solid reﬂects the particle in the same action line but opposite direction |

C)Selecting dataset from the problem

|  |  |  |  |
| --- | --- | --- | --- |
| **Water Drop Radius [lu]** | **Diffusion Parameters** |  |  |
|  | **Bulk Porosity** | **Tortuosity** | **Diffusibility** |
| 15 | 0.785 | 1.0807 | 0.7264 |
|  | 0.7618 | 1.0875 | 0.7005 |
|  | 0.7665 | 1.086 | 0.7058 |
|  | 0.7857 | 1.0802 | 0.7274 |
|  | 0.7803 | 1.0792 | 0.723 |
|  | 0.769 | 1.0895 | 0.7058 |
|  | 0.7746 | 1.0756 | 0.7202 |
| 20 | 0.7568 | 1.0889 | 0.695 |
|  | 0.7493 | 1.0941 | 0.6849 |
|  | 0.7594 | 1.0766 | 0.7054 |
|  | 0.7559 | 1.0818 | 0.6987 |
|  | 0.76 | 1.0866 | 0.6994 |
|  | 0.758 | 1.0821 | 0.7005 |
|  | 0.7679 | 1.1038 | 0.6957 |
| 25 | 0.7445 | 1.1111 | 0.6701 |
|  | 0.7248 | 1.1015 | 0.658 |
|  | 0.7197 | 1.0838 | 0.6641 |
|  | 0.7343 | 1.0791 | 0.6805 |
|  | 0.7283 | 1.0895 | 0.6685 |
|  | 0.7211 | 1.0875 | 0.6631 |
|  | 0.7276 | 1.0968 | 0.6634 |
| 30 | 0.6873 | 1.107 | 0.6209 |
|  | 0.6915 | 1.0891 | 0.6349 |
|  | 0.6819 | 1.0992 | 0.6204 |
|  | 0.6992 | 1.0949 | 0.6386 |
|  | 0.6973 | 1.1059 | 0.6305 |
|  | 0.6975 | 1.1031 | 0.6323 |
|  | 0.7042 | 1.1033 | 0.6383 |
| 35 | 0.6354 | 1.1302 | 0.5622 |
|  | 0.6464 | 1.0909 | 0.5925 |
|  | 0.6441 | 1.0902 | 0.5908 |
|  | 0.6588 | 1.127 | 0.5846 |
|  | 0.6386 | 1.1067 | 0.577 |
|  | 0.6361 | 1.0967 | 0.58 |
|  | 0.6426 | 1.109 | 0.5794 |

D)Applying Artificial Neuron Network to train the model

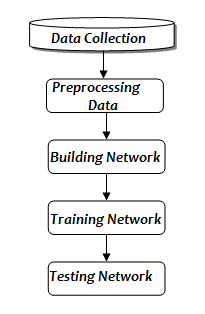
For given problem statement:-

NUMBER OF INPUT =3

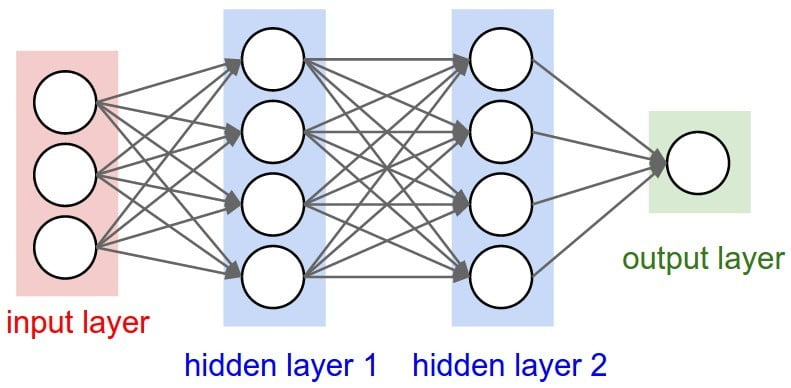
Number of Hidden Layer =5

Weights values (w,v) are assigned randomly at initial point.

* **Typical steps used in Artificial Neural network: -**

****

**Fig Artificial neural network diagram**

****

Some important formulas for the calculation purpose: -

1. Input to jth hidden neuron.
2. Output of the jth hidden neuron
3. Inputs of the kth output neuron
4. Output of the kth output neuron
5. Calculation of error

E)Calculation part and output of result

* **Output values at different value of learning rate η**

**Here**

IO[1][31]:Input of output layer

OO[1][31]: output of output layer

TO[1][31]: ]:Targeted value of that neuron .

Error=Absolute error at output neuron.

The output data is calculating by assuming error 10^-3 will calculating training pattern.

The below data is in normalized data

**1)η=0.3**

IO[1][31]:0.153493 OO[1][31]:0.152299 TO[1][31]:0.073123, Error=0.079175

IO[1][32]:0.124344 OO[1][32]:0.123707 TO[1][32]:0.035593, Error=0.088114

IO[1][33]:0.112528 OO[1][33]:0.112055 TO[1][33]:-0.010412, Error=0.122467

IO[1][34]:0.123084 OO[1][34]:0.122466 TO[1][34]:0.007748, Error=0.114718

IO[1][35]:0.117694 OO[1][35]:0.117153 TO[1][35]:0.004116, Error=0.113037

**2)η=0.4**

IO[1][31]:0.154919 OO[1][31]:0.153692 TO[1][31]:0.073123, Error=0.080568

IO[1][32]:0.107350 OO[1][32]:0.106939 TO[1][32]:0.035593, Error=0.071346

IO[1][33]:0.095891 OO[1][33]:0.095598 TO[1][33]:-0.010412, Error=0.106010

IO[1][34]:0.111774 OO[1][34]:0.111311 TO[1][34]:0.007748, Error=0.103563

IO[1][35]:0.102156 OO[1][35]:0.101802 TO[1][35]:0.004116, Error=0.097685

**3)η=0.5**

IO[1][31]:0.111450 OO[1][31]:0.110991 TO[1][31]:0.073123, Error=0.037867

IO[1][32]:0.129888 OO[1][32]:0.129162 TO[1][32]:0.035593, Error=0.093569

IO[1][33]:0.086160 OO[1][33]:0.085947 TO[1][33]:-0.010412, Error=0.096359

IO[1][34]:0.083902 OO[1][34]:0.083706 TO[1][34]:0.007748, Error=0.075958

IO[1][35]:0.096030 OO[1][35]:0.095735 TO[1][35]:0.004116, Error=0.091619

**4)η=0.6**

IO[1][31]:0.153891 OO[1][31]:0.152688 TO[1][31]:0.073123, Error=0.079564

IO[1][32]:0.070836 OO[1][32]:0.070718 TO[1][32]:0.035593, Error=0.035124

IO[1][33]:0.072181 OO[1][33]:0.072056 TO[1][33]:-0.010412, Error=0.082467

IO[1][34]:0.099755 OO[1][34]:0.099426 TO[1][34]:0.007748, Error=0.091678

IO[1][35]:0.077722 OO[1][35]:0.077566 TO[1][35]:0.004116, Error=0.073449

**5)η=0.7**

IO[1][31]:0.119620 OO[1][31]:0.119053 TO[1][31]:0.073123, Error=0.045930

IO[1][32]:0.131530 OO[1][32]:0.130777 TO[1][32]:0.035593, Error=0.095184

IO[1][33]:0.093383 OO[1][33]:0.093112 TO[1][33]:-0.010412, Error=0.103524

IO[1][34]:0.092224 OO[1][34]:0.091964 TO[1][34]:0.007748, Error=0.084215

IO[1][35]:0.102153 OO[1][35]:0.101799 TO[1][35]:0.004116, Error=0.097683

**6)η=0.8**

IO[1][31]:0.134347 OO[1][31]:0.133545 TO[1][31]:0.073123, Error=0.060421

IO[1][32]:-0.034744 OO[1][32]:-0.034730 TO[1][32]:0.035593, Error=0.070324

IO[1][33]:0.025119 OO[1][33]:0.025114 TO[1][33]:-0.010412, Error=0.035525

IO[1][34]:0.076086 OO[1][34]:0.075939 TO[1][34]:0.007748, Error=0.068191

IO[1][35]:0.022120 OO[1][35]:0.022116 TO[1][35]:0.004116, Error=0.018000

**7)η=0.9**

IO[1][31]:0.132988 OO[1][31]:0.132210 TO[1][31]:0.073123, Error=0.059086

IO[1][32]:0.139521 OO[1][32]:0.138622 TO[1][32]:0.035593, Error=0.103029

IO[1][33]:0.109315 OO[1][33]:0.108882 TO[1][33]:-0.010412, Error=0.119294

IO[1][34]:0.108630 OO[1][34]:0.108205 TO[1][34]:0.007748, Error=0.100457

IO[1][35]:0.116825 OO[1][35]:0.116297 TO[1][35]:0.004116, Error=0.112181

1. **Conclusion**

The calculations are performed using 36 input pattern has been given to the ANN model in which 31 patterns are used to train the pattern and finding the value of w and v and after that we are using 5 pattern we are testing the values .As we are trying to finding the error using different value of learning rate the optimum value of learning rate for required data is 0.8 .Since we have considering the error=10^-3 while calculating value of weights if we are increasing decimal point number of iteration will increase and time for compilation will increase .But our ANN will give output value very close to targeted value.