**Amended Claims - Marked Copy**

**WE CLAIM:**

1. (Currently Amended) A method **200** of solving a heat transport problem over an object characterized by a geometry, using a hardware multi-threading process, the hardware comprising: a processor configured to run a training model, a first number of storage process units configured to store input data, a second number of memory operation units configured to store output data, and a hardware switch configured to minimize idle time of the processor, the method comprising:

providing (**201**) a geometry and associated boundary conditions and discretizing the geometry into a grid, wherein the grid comprises a number of grid points;

specifying (**202**) temperature or heat flow conditions at the boundary surrounding the geometry and an initial condition at each grid point;

solving (**203**) a heat flow equation selected from one of conduction, convection or radiation for the geometry and the associated boundary conditions to obtain a temperature, or a heat flow rate, or both at each grid point at steady state;

storing (**204**) the solution for each grid point in a training database;

training (**205**) a model selected from a PPRNN, a DRNN or a DANN model using the training database, wherein

the ~~model is a~~ PPRNN model is given by:

$$PPRNN = \oint_{\Omega} \int_{j=1}^{M} tanh(h_{j_i} + b_{2_i}) dj d\Omega_i \quad \ldots\ldots\ldots\ldots\ldots (4)$$

where

$$h_{j_i} = W_{1_i} \cdot h_{j-1_i} + W_{2_i} \cdot x_{j-1_i} + b_{1_i} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots..(5)$$

*x* is the input, *h* is the hidden cell state and $W_1$, $b_1$ and $W_2$, $b_2$ are the weight and bias matrices for hidden-hidden and input-hidden

Formatted: Indent: Left: 0.5"

Formatted: Indent: Left: 0.83", First line: 0.17"

Formatted: Indent: Left: 1.33"

connections, $\Omega$ is the domain of interest, $M$ is the number of examples for training, tanh is an activation function;

the DRNN model is given by:

$$DRNN = \forall \oint_{\Omega} \int_{j=1}^{M} tanh\left(h_{j_i} + b_{2_i}\right) dj d\Omega_i \quad \ldots\ldots\ldots\ldots\ldots(7)$$

where,

$$h_j = W_1 \cdot h_{j-1} + W_2 \cdot x_{j-1} + b_1 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(8)$$

$x$ is the input, $h$ is the hidden cell state and $W_1$, $b_1$ and $W_2$, are the weight and bias matrices for hidden-hidden and input-hidden connections, Ω is the domain of interest, M is the number of examples for training, tanh is an activation function;

the DANN model is given by:

$$DANN = \forall \begin{cases} 0 if x \leq 0 \\ \int_{j=1}^{M} \left(h_j + b_2\right) dj if x > 0 \end{cases} \quad \ldots\ldots\ldots\ldots10)$$

where,

$$h_j = W_1 \cdot h_{j-1} + W_2 \cdot x_{j-1} + b_1 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(8)$$

$x$ is the input, $h$ is the hidden cell state and $W_1$, $b_1$ and $W_2$, are the weight and bias matrices for hidden-hidden and input-hidden connections, and M is the number of examples for training.

;

inputting (**206**) a modified boundary condition or initial condition or both associated with the geometry; and,

generating (**207**) a temperature, a heat flow rate ~~at~~ or both at each grid point corresponding to the modified boundary condition or initial condition.

**Formatted:** Indent: Left: 0.67", First line: 0.33"

**Formatted:** Indent: Left: 1.33"

**Formatted:** Indent: Left: 0.67", First line: 0.33"

**Formatted:** Indent: Left: 1.33"

**Formatted:** Normal, Indent: Left: 0.5"

2.(Original) The method as claimed in claim 1, wherein the training **400** comprises:

calling (**402**) a storage process unit for each input point i in an individual unit;

storing (**403**) all data points i=1, 2, 3… n in j=1, 2, 3… n storage units in a single logic call using AND logic enabled function;

calling (**404**) multi-threading/distributed RNN and executing the same;

calling (**405**) AND logic enabled machine code upon each point i=1, 2, 3… n in memory unit k=1, 2, 3… n with zero time lapse and zero k to k communication;

storing (**406**) output RNN data of each output point p, corresponding to each input point i=1, 2, 3… n in j=1, 2, 3… n storage units; and

repeating (**407**) the steps (i) to (v) for m samples and perform computations and storage for each m=1, 2, 3… sample in the geometry.

3. (Original) The method as claimed in claim 1, wherein the heat flow equation solved is a conduction equation wherein the temperature T is given by:

$$\frac{1}{a}\cdot\frac{\partial T}{\partial t}=\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}+\frac{\partial^2 T}{\partial z^2}+\frac{q}{k}\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

where x, y, and z represent Cartesian coordinates, $q$ is rate of heat generation, $k$ is thermal conductivity and $a$ is a product of density and specific heat capacity of the material.

4. (Original) The method as claimed in claim 1, wherein the heat flow equation solved is a convection equation given by:

$$\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}+h\cdot\frac{\partial T}{\partial y}=0\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2)$$

5. (Original) The method as claimed in claim 1, wherein the heat flow equation solved is a radiation equation given by:

$$d_z \rho c_p u \cdot \nabla T + \nabla \cdot q = d_z Q + q_0 + d_z Q_{ted} + d_z Q_R \ldots\ldots\ldots\ldots\ldots\ldots(3)$$

6. (Original) The method as claimed in claim 1, wherein the solving a heat flow equation comprises using an analytical solution, a finite element method solution (FEM) or finite difference method (FDM) solution to generate training data set for each input data i for each sample m in a geometry t.

7.(Canceled) The method as claimed in claim 1, wherein the model is a PPRNN model wherein:

$$PPRNN = \oint_{\Omega}^{-} \int_{j=1}^{M} tanh\left(h_{j_t} + b_{2_t}\right) dj d\Omega_t \ldots\ldots\ldots\ldots\ldots(4)$$

where

$$h_{j_t} = W_{1_t} \cdot h_{j-1_t} + W_{2_t} \cdot x_{j-1_t} + b_{1_t} \ldots\ldots\ldots\ldots\ldots\ldots\ldots(5)$$

*x* is the input, *h* is the hidden cell state and $W_1$, $b_1$ and $W_2$, $b_2$ are the weight and bias matrices for hidden-hidden and input-hidden connections, *Ω* is the domain of interest, *M* is the number of examples for training, tanh is an activation function.

87. The method as claimed in claim 71, comprising obtaining the weight and bias matrices by minimizing mean square error between the predicted and input temperatures for each mesh independent point i for an example m for a PPRNN model is given by:

$$MSE_i = \frac{1}{m} \sum_{n=1}^{m} \left| T_{Pred_i} - T_{Act_i} \right|^2 \ldots\ldots\ldots\ldots\ldots\ldots(6)$$

9.(Canceled) The method as claimed in claim 1, wherein the model is a DRNN model wherein:

$$DRNN = \forall \oint_{\Omega}^{-} \int_{j=1}^{M} tanh\left(h_{j_t} + b_{2_t}\right) dj d\Omega_t \ldots\ldots\ldots\ldots\ldots(7)$$

**Formatted:** Normal, Justified, Right: 0", Space Before: 0 pt, Line spacing: At least 24 pt, Tab stops: Not at 0" + 0.69"

**Formatted:** Normal, Justified, Indent: Left: 0", Right: 0", Space Before: 0 pt, Line spacing: At least 24 pt, Tab stops: Not at 0" + 0.69" where,

$h_j = W_1 \cdot h_{j-1} + W_2 \cdot x_{j-1} + b_1$……………………(8)

$x$ is the input, $h$ is the hidden cell state and $W_1$, $b_1$ and $W_2$, are the weight and bias matrices for hidden-hidden and input-hidden connections, Ω is the domain of interest, M is the number of examples for training, tanh is an activation function.

108. (Currently Amended) The method as claimed in claim 91, comprising obtaining the weight and bias matrices by minimizing mean square error between the predicted and input temperatures for each mesh independent point i for an example m for a DRNN model is given by:

$$MSE = \sum_{n=1}^{m} \|T_{Pred} - T_{Act}\|^2 \text{…………..(9)}$$

11.(Canceled) The method as claimed in claim 1, wherein the model is a DANN model wherein:

$$DANN = \forall \begin{cases} 0 if x \leq 0 \\ \int_{j=1}^{M} (h_j + b_2) dj if x > 0 \end{cases} \text{…………10)}$$

Formatted: Normal, Justified, Right: 0", Space Before: 0 pt, Line spacing: At least 24 pt, Tab stops: Not at 0" + 0.69"

where,

$h_j = W_1 \cdot h_{j-1} + W_2 \cdot x_{j-1} + b_1$……………………(8)

$x$ is the input, $h$ is the hidden cell state and $W_1$, $b_1$ and $W_2$, are the weight and bias matrices for hidden-hidden and input-hidden connections, and M is the number of examples for training.

129. (Currently Amended) The method as claimed in claim 11, comprising obtaining the weight and bias matrices by minimizing mean square error between the predicted and input temperatures for each mesh independent point i for an example m for a DANN model is given by:

$$MSE_i = \frac{1}{m}\sum_{n=1}^{m}\left|T_{Pred_i} - T_{Act_i}\right|^2 \ldots\ldots\ldots\ldots\ldots\ldots(6)$$

~~13~~10. (Currently Amended) A system for solving a heat transport problem over an object characterized by a geometry, the system comprising:

a hardware switch (**102**);

a processor (**101**) coupled to the hardware switch (**102**) to run a neural network engine, wherein the processor (**101**) is configured to:

receive a geometry and associated boundary conditions and discretize the geometry into a grid, wherein the grid comprises a number of grid points;

receive temperature or heat flow conditions at the boundary surrounding the geometry and an initial condition at each grid point;

solve a heat flow equation selected from one of conduction, convection or radiation for the geometry and the associated boundary conditions to obtain a temperature, or a heat flow rate, or both at each grid point at steady state;

store the solution for each grid point in a training database;

train a model selected from a PPRNN, a DRNN or a DANN model using the training database, wherein

the PPRNN model is given by:

$$PPRNN = \oint_{\Omega} \int_{j=1}^{M} tanh\left(h_{j_i} + b_{2_i}\right) dj d\Omega_i \ldots\ldots\ldots\ldots\ldots(4)$$

where

$$h_{j_i} = W_{1_i} \cdot h_{j-1_i} + W_{2_i} \cdot x_{j-1_i} + b_{1_i} \ldots\ldots\ldots\ldots\ldots\ldots\ldots..(5)$$

$x$ is the input, $h$ is the hidden cell state and $W_1$, $b_1$ and $W_2$, $b_2$ are the weight and bias matrices for hidden-hidden and input-hidden connections, $\Omega$ is the domain of interest, $M$ is the number of examples for training, tanh is an activation function;

the DRNN model is given by:

Formatted: Indent: Left: 0.79", First line: 0.05"

Formatted: Indent: Left: 0.83", First line: 0.17"

Formatted: Indent: Left: 1.33"

Formatted: Indent: Left: 0.67", First line: 0.33"

$$DRNN = \forall \oint_{\Omega} \int_{j=1}^{M} tanh\left(h_{j_i} + b_{2_i}\right) dj d\Omega_i \text{……………(7)}$$

where,

$$h_j = W_1 \cdot h_{j-1} + W_2 \cdot x_{j-1} + b_1 \text{…………………..(8)}$$

*x* is the input, *h* is the hidden cell state and $W_1$, $b_1$ and $W_2$, are the weight and bias matrices for hidden-hidden and input-hidden connections, Ω is the domain of interest, M is the number of examples for training, tanh is an activation function;

the DANN model is given by:

$$DANN = \forall \begin{cases} 0 if x \leq 0 \\ \int_{j=1}^{M} (h_j + b_2) dj if x > 0 \end{cases} \text{…………10)}$$

where,

$$h_j = W_1 \cdot h_{j-1} + W_2 \cdot x_{j-1} + b_1 \text{…………………..(8)}$$

*x* is the input, *h* is the hidden cell state and $W_1$, $b_1$ and $W_2$, are the weight and bias matrices for hidden-hidden and input-hidden connections, and M is the number of examples for training.

;

receive a modified boundary condition or initial condition or both associated with the geometry as input to the trained model; and,

generate a temperature, a heat flow rate at both at each grid point corresponding to the modified boundary condition or initial condition.

1411. (Original) The system as claimed in claim 1310, comprising a storage process unit (**110**) comprising a plurality of sub-storage process units (**111-1**, **111-2**,…, **111-X**) configured to store one or more data subsets of geometry and associated boundary conditions, and temperature and heat-flow conditions.

Formatted: Indent: Left: 0"

Formatted: Indent: Left: 1.33"

Formatted: Indent: Left: 0.67", First line: 0.33"

Formatted: Indent: Left: 1.33"

~~15~~12. (Original) The system as claimed in claim ~~13~~10, wherein the processor (**101**)is configured to divide the transport problem solution into ~~a plurality of~~ threads for concurrent execution.

~~16~~13. (Original) The system as claimed in claim ~~15~~12, wherein the hardware switch(**102**) is configured to execute the ~~one or more~~threads in parallel and allocatethe generatedtemperature and heat flow rate in memory operation units **120**.

~~17~~14. (Original) The system as claimed in claim ~~15~~12, wherein the memory operation units **120** comprises ~~a plurality of~~ memory operation units (**121-1**, **121-2**, **121-3**…., **121-X**)configured tostore generatedtemperature and heat flow rate.

**Amended Claims - Clean Copy**

**WE CLAIM:**

1. A method **200** of solving a heat transport problem over an object characterized by a geometry, using a hardware multi-threading process, the hardware comprising: a processor configured to run a training model, a first number of storage process units configured to store input data, a second number of memory operation units configured to store output data, and a hardware switch configured to minimize idle time of the processor, the method comprising:

providing (**201**) a geometry and associated boundary conditions and discretizing the geometry into a grid, wherein the grid comprises a number of grid points;

specifying (**202**) temperature or heat flow conditions at the boundary surrounding the geometry and an initial condition at each grid point;

solving (**203**) a heat flow equation selected from one of conduction, convection or radiation for the geometry and the associated boundary conditions to obtain a temperature, or a heat flow rate, or both at each grid point at steady state;

storing (**204**) the solution for each grid point in a training database;

training (**205**) a model selected from a PPRNN, a DRNN or a DANN model using the training database, wherein

the PPRNN model is given by:

$$PPRNN = \oint_{\Omega} \int_{j=1}^{M} tanh\left(h_{j_i} + b_{2_i}\right) dj d\Omega_i \ldots\ldots\ldots\ldots\ldots (4)$$

where

$$h_{j_i} = W_{1_i} \cdot h_{j-1_i} + W_{2_i} \cdot x_{j-1_i} + b_{1_i} \ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

$x$ is the input, $h$ is the hidden cell state and $W_1$, $b_1$ and $W_2$, $b_2$ are the weight and bias matrices for hidden-hidden and input-hidden connections, $\Omega$ is the domain of interest, $M$ is the number of examples for training, tanh is an activation function;

the DRNN model is given by:

$$DRNN = \forall \oint_{\Omega} \int_{j=1}^{M} tanh\left(h_{j_i} + b_{2_i}\right) dj d\Omega_i \text{……………(7)}$$

where,

$$h_j = W_1 \cdot h_{j-1} + W_2 \cdot x_{j-1} + b_1 \text{…………………..(8)}$$

$x$ is the input, $h$ is the hidden cell state and $W_1$, $b_1$and $W_2$, are the weight and bias matrices for hidden-hidden and input-hidden connections, Ω is the domain of interest, M is the number of examples for training, tanh is an activation function;

the DANN model is given by:

$$DANN = \forall \begin{cases} 0 if x \leq 0 \\ \int_{j=1}^{M} \left(h_j + b_2\right) dj if x > 0 \end{cases} \text{…………10)}$$

where,

$$h_j = W_1 \cdot h_{j-1} + W_2 \cdot x_{j-1} + b_1 \text{…………………..(8)}$$

$x$ is the input, $h$ is the hidden cell state and $W_1$, $b_1$and $W_2$, are the weight and bias matrices for hidden-hidden and input-hidden connections, and M is the number of examples for training;

inputting (**206**) a modified boundary condition or initial condition or both associated with the geometry; and,

generating (**207**) a temperature, a heat flow rate or both at each grid point corresponding to the modified boundary condition or initial condition.

2. The method as claimed in claim 1, wherein the training **400** comprises:

calling (**402**) a storage process unit for each input point i in an individual unit;

storing (**403**) all data points i=1, 2, 3… n in j=1, 2, 3… n storage units in a single logic call using AND logic enabled function;

calling (**404**) multi-threading/distributed RNN and executing the same;

calling (**405**) AND logic enabled machine code upon each point i=1, 2, 3… n in memory unit k=1, 2, 3… n with zero time lapse and zero k to k communication;

storing (**406**) output RNN data of each output point p, corresponding to each input point i=1, 2, 3… n in j=1, 2, 3… n storage units; and

repeating (**407**) the steps (i) to (v) for m samples and perform computations and storage for each m=1, 2, 3… sample in the geometry.

3. The method as claimed in claim 1, wherein the heat flow equation solved is a conduction equation wherein the temperature T is given by:

$$\frac{1}{a} \cdot \frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} + \frac{q}{k} \text{…………………}(1)$$

where x, y, and z represent Cartesian coordinates, $q$ is rate of heat generation, $k$ is thermal conductivity and $a$ is a product of density and specific heat capacity of the material.

4. The method as claimed in claim 1, wherein the heat flow equation solved is a convection equation given by:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + h \cdot \frac{\partial T}{\partial y} = 0 \text{ … … … … … … … . .}(2)$$

5. The method as claimed in claim 1, wherein the heat flow equation solved is a radiation equation given by:

$$d_z \rho c_p u \cdot \nabla T + \nabla \cdot q = d_z Q + q_0 + d_z Q_{ted} + d_z Q_R \dots\dots\dots\dots\dots\dots(3)$$

6. The method as claimed in claim 1, wherein the solving a heat flow equation comprises using an analytical solution, a finite element method solution (FEM) or finite difference method (FDM) solution to generate training data set for each input data i for each sample m in a geometry t.

$$PPRNN = \oint_{\Omega} \int_{j=1}^{M} tanh\left(h_{j_i} + b_{2_i}\right) dj d\Omega_i$$

$$h_{j_i} = W_{1_i} \cdot h_{j-1_i} + W_{2_i} \cdot x_{j-1_i} + b_{1_i}$$

7. The method as claimed in claim 1, comprising obtaining the weight and bias matrices by minimizing mean square error between the predicted and input temperatures for each mesh independent point i for an example m for a PPRNN model is given by:

$$MSE_i = \frac{1}{m} \sum_{n=1}^{m} \left| T_{Pred_i} - T_{Act_i} \right|^2 \dots\dots\dots\dots\dots\dots(6)$$

8. The method as claimed in claim 1, comprising obtaining the weight and bias matrices by minimizing mean square error between the predicted and input temperatures for each mesh independent point i for an example m for a DRNN model is given by:

$$MSE = \sum_{n=1}^{m} \| T_{Pred} - T_{Act} \|^2 \dots\dots\dots\dots(9)$$

9. The method as claimed in claim 1, comprising obtaining the weight and bias matrices by minimizing mean square error between the predicted and input temperatures for each mesh independent point i for an example m for a DANN model is given by:

$$MSE_i = \frac{1}{m} \sum_{n=1}^{m} \left| T_{Pred_i} - T_{Act_i} \right|^2 \dots\dots\dots\dots\dots\dots(6)$$

10. A system for solving a heat transport problem over an object characterized by a geometry, the system comprising:

a hardware switch (**102**);

a processor (**101**) coupled to the hardware switch (**102**) to run a neural network engine, wherein the processor (**101**) is configured to:

receive a geometry and associated boundary conditions and discretize the geometry into a grid, wherein the grid comprises a number of grid points;

receive temperature or heat flow conditions at the boundary surrounding the geometry and an initial condition at each grid point;

solve a heat flow equation selected from one of conduction, convection or radiation for the geometry and the associated boundary conditions to obtain a temperature, or a heat flow rate, or both at each grid point at steady state;

store the solution for each grid point in a training database;

train a model selected from a PPRNN, a DRNN or a DANN model using the training database, wherein

the PPRNN model is given by:

$$PPRNN = \oint_{\Omega} \int_{j=1}^{M} tanh\left(h_{j_i} + b_{2_i}\right) dj d\Omega_i \ldots\ldots\ldots\ldots\ldots (4)$$

where

$$h_{j_i} = W_{1_i} \cdot h_{j-1_i} + W_{2_i} \cdot x_{j-1_i} + b_{1_i} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

$x$ is the input, $h$ is the hidden cell state and $W_1$, $b_1$ and $W_2$, $b_2$ are the weight and bias matrices for hidden-hidden and input-hidden connections, $\Omega$ is the domain of interest, $M$ is the number of examples for training, tanh is an activation function;

the DRNN model is given by:

$$DRNN = \forall \oint_{\Omega} \int_{j=1}^{M} tanh\left(h_{j_i} + b_{2_i}\right) dj d\Omega_i \ldots\ldots\ldots\ldots\ldots (7)$$

where,

$h_j = W_1 \cdot h_{j-1} + W_2 \cdot x_{j-1} + b_1$……………….…..(8)

$x$ is the input, $h$ is the hidden cell state and $W_1$, $b_1$and $W_2$, are the weight and bias matrices for hidden-hidden and input-hidden connections, Ω is the domain of interest, M is the number of examples for training, tanh is an activation function;

the DANN model is given by:

$$DANN = \forall \begin{cases} 0 if x \leq 0 \\ \int_{j=1}^{M} (h_j + b_2) dj if x > 0 \end{cases} \text{…………10)}$$

where,

$h_j = W_1 \cdot h_{j-1} + W_2 \cdot x_{j-1} + b_1$……………….…..(8)

$x$ is the input, $h$ is the hidden cell state and $W_1$, $b_1$and $W_2$, are the weight and bias matrices for hidden-hidden and input-hidden connections, and M is the number of examples for training;

receive a modified boundary condition or initial condition or both associated with the geometry as input to the trained model; and,

generate a temperature, a heat flow rate at both at each grid point corresponding to the modified boundary condition or initial condition.

11. The system as claimed in claim 10, comprising a storage process unit (**110**) comprising sub-storage process units (**111-1**, **111-2**,…, **111-X**) configured to store data subsets of geometry and associated boundary conditions, and temperature and heat-flow conditions.

12. The system as claimed in claim 10, wherein the processor (**101**)is configured to divide the transport problem solution into threads for concurrent execution.

13. The system as claimed in claim 12, wherein the hardware switch (**102**) is configured to execute the threads in parallel and allocate the generated temperature and heat flow rate in memory operation units **120**.

14. The system as claimed in claim 12, wherein the memory operation units **120** comprises memory operation units (**121-1**, **121-2**, **121-3**…., **121-X**)configured to store generated temperature and heat flow rate.

Sd.- Dr V. SHANKAR IN/PA-1733
For and on behalf of the Applicants