Proceedings of the International Workshop on Intelligent Information Systems
IIS2011, September 13-14, 2011, Chisinau, Republic of Moldova

Event-threading Algorithm on Modern Database Management Systems

Alexandr Chior

Abstract

Modern DBMS provide ample opportunities of developing various business intelligence systems. DB objects are represented by tables for data storage, stored procedures for data processing and various calculations, triggers and diagrams for description of the relations between tables. Implementation of event-threading algorithm is possible in DBMS that support triggers, event-threading model and table creation in trigger body.

Keywords: artificial neuronal network, event-driven model, database management system, database trigger, stored procedure.

1 Definitions

Algorithm *et*ANN handles input data using new methods based on parallel algorithms, event-driven model, database trigger model and synthesis of trained network. Algorithm *et*ANN allows decreasing the number of operations from *O(n)* (classical models) down to *O(ln n)* (*et*ANN mode) [1].

Connection between two artificial neurons is associated with *events* in case of event-driven model realization or with *triggers* when *et*ANN is realized in database [1].

Definitions of database management system, table, stored procedure and database trigger can be found in Encyclopedia [3].

Server starts trigger automatically when you try to change data in the table which the trigger is associated with. All data modifications are performed as transaction, which holds the action that caused the trigger to fire [3].

ANN model is presented in the form of n-partite directional graph K $(m_1, \ldots, m_n)$ with weighted connections and artificial neurons in its nodes, where m_i - size of layer and neurons from layer i and $i+1$ are interconnected, where $i \in [1, m]$ [2].

Logical entities of *et*ANN are layers $\{L_i\}$, neurons $\{N_{ij_i}\}$, connections $\{cN_{ij_i}N_{kl_k}\}$, where $i, k = \overline{1,n}, j_i = \overline{1,m_i}, l_k = \overline{1,m_k}, i \neq j$.

Special ANN parameters are training speed, error threshold, size of training and testing selection, size of model, precision of research, operating variables, casual variables, number of training epochs, domain of admissible solutions, etc. Training and testing time series are x_{uv}, where series length m_1 size of input layer.

2 Realization of *et*ANN in Database

Storage and functioning of the ANN in database requires preliminary stages implemented through stored procedures and repetitive stages designed by triggers [2].

DBMS that support stored procedures, triggers and parallel queries are Oracle, Microsoft SQL Server, DB2. In *et*ANN triggers are used only for ANN training/testing [4].

Table 1:Logical structure of database **tables** to store *et*ANN entities

Db Table	Fields / Table Structure
Parameters	id, name, value
Layers	id, name
Neurons	id, layerid, name, inner_potential, max_potential
Connections	id, neuronid_from, neuronid_to, weight
TimeSeries	series_id, neuron_id, value

Algorithm: *Collect and normalize data for table [TimeSeries]. Construct ANN topology by tables [Neurons] and [Connections]. Define*

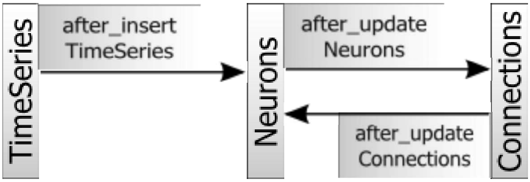

Figure 1:Trigger processing Diagram

characteristics in table [Parameters]. Train ANN filling [TimeSeries] with training data. Test ANN filling [TimeSeries] with testing data.

Database transactions depend on **triggers** which fire after insert data in [TimeSeries] table and after updating [Neurons] and [Connections].

Trigger [after_insert_TimeSeries] updates inner potential of input layer in [Neurons] table with data from training time series in :NEW variable. SQL script:

```
UPDATE [Neurons] nr
 SET nr.inner_potential +=
    :NEW.value * nr.inner_potential
WHERE nr.ID = :NEW.neuron_id;
```

Trigger [after_update_Neurons] checks inner potential of neuron in input layer, if inner potential is more or equal than maximum possible potential value of neuron, than neuron states in next layer are updated (chain reaction in *et*ANN). Trigger uses Gaussian threshold function and modulus operation. SQL script:

```
IF(:NEW.inner_potential >= :NEW.max_potential)
  UPDATE [Connections] cn
  SET cn.weight =
    Gaussian(:NEW.inner_potential,:NEW.max_potential)
WHERE cn.neuronid_from = :NEW.id;
 :NEW.inner_potential =
    MOD(:NEW.inner_potential,:NEW.max_potential);
```

```
END IF;
```

Trigger [after_update_Connections] changes inner potential of neurons which are connected with modified neurons from previous layer.

```
UPDATE [Neurons] nr
 SET nr.inner_potential +=
     :NEW.weight * nr.inner_potential
WHERE nr.ID = :NEW.neuronid_to;
```

3 Conclusion

Essential modification of classical ANN structure consists in elimination of all *unessential* connections after handling by *et*ANN calculation algorithm.

Performance increase is reached by parallel work of all database transactions/triggers and database architecture based on alternative model. In the modern DBMS parallel programming are realized with the aid of technologies of parallel queries.

References

[1] A. Chior *Event-threading calculation algorithm complexity estimation.* MITRE CECMI, Moldova State University, 2009, pp. 89–92.

[2] P. Dayan, L.F. Abbott, *Theoretical Neuroscience - Computational and Mathematical Modeling of Neural Systems.* MIT Press, 2000.

[3] L. Liu, M.T. Ozsu. *Encyclopedia of Database Systems.* Springer, 2009.

[4] Wikipedia. *Comparison of relational database management systems.* 2011.

Alexandr Chior,
State University of Moldova
Moldova, Chisinau
Phone: +373 68 895959
E–mail: alex.kior@gmail.com

Received May 20, 2011