

Event-threading algorithm on modern database management systems

Alexandr CHIOR

State University of Moldova

alex.kior@gmail.com

Abstract – Modern DBMS provide ample opportunities of developing various business intelligence systems. Database objects are represented by tables for data storage, stored procedures for data processing and 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.

Index Terms – artificial neuronal network, event-driven model, database management system, database trigger, stored procedure.

INTRODUCTION

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

Connection between two artificial neurons is associated with *events* in case of event-driven model realization or with *triggers* when *ETann* 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 $K(m_1, \dots, m_n)$ 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 *ETann* are layers L_i , neurons N_{ij} , connections $cN_{ij}N_{kl}$, where i, k in $\{1, n\}$, j_i in $\{1, m_i\}$, l_k in $\{1, m_k\}$, $i < j$.

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_{iv} , where series length m_1 size of input layer.

REALIZATION OF ETANN IN DATABASE

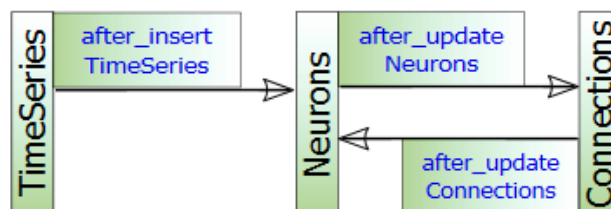
Storage and functioning of the ANN in database requires preliminary stages implemented through stored procedures and repetitive stages designed by triggers [2].

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

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

[Neurons] and [Connections].

FIG. I Trigger Flow Diagram



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 *ETann*). 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. SQL script:

```

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

TABLE I. Logical structure of database.

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

ALGORITHM

1. Collect and normalize data for table [TimeSeries].
2. Construct ANN (tables [Neurons] and [Connections]).
3. Define characteristics (table [Parameters]).
4. Train ANN filling [TimeSeries] with training data.
5. Test ANN filling [TimeSeries] with testing data.

I. CONCLUSION

Essential modification of classical ANN structure consists in elimination of all *unessential* connections after handling by *ETann* 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

- A. Chior, Event-threading calculation algorithm complexity estimation. MITRE CECMI, Moldova State University, 2009, pp. 89-92.
- P. Dayan, L.F. Abbott, Theoretical Neuroscience - Computational and Mathematical Modeling of Neural Systems. MIT Press, 2000.
- L. Liu, M.T. Ozsü., Encyclopedia of Database Systems. Springer, 2009.
- Wikipedia. Comparison of relational database management systems. 2011.