Associative Fine-Tuning of Biologically Inspired Active Neuro-Associative Knowledge Graphs

Adrian Horzyk
horzyk@agh.edu.pl
Google: Horzyk

AGH University of Science and Technology
Krakow, Poland

Janusz A. Starzyk
starzyk@ohio.edu
Google: Janusz Starzyk

Ohio University, Athens, Ohio, U.S.A., School of Electrical Engineering and Computer Science

University of Information Technology and Management, Rzeszow, Poland
Research inspired by brains and biological neurons

- Work in parallel and asynchronously
- Associate stimuli context-sensitively
- Use time approach for computations
- Represent various data and their relations
- Self-organize neurons developing a very complex structure
- Aggregate representation of similar data
- Integrate memory and the procedures
- Provide plasticity to develop a structure to represent data and object relations
ANAKG produces a complex graph structure of dynamic and reactive neurons and connections to represent a set of training sequences.

- Neurons aggregate all instances of the same elements that occur in all sequences.
Objectives and Contribution

- Construction of the fine-tuning algorithm for synaptic weights to achieve better recalling of associatively stored training sequences and better generalization.
- Avoid unintended activations to stop possible false-recalling of sequences.
- Construct a well-aggregative model for storing correlated training sequences.
- Reproduce functionality of the biological neural substance.
ASN Neurons

- Connect context-sensitively to emphasize training sequences and automatically develop an ANAKG network structure.
- Aggregate representations of the same elements of the training sentences - no duplicates!
- Work asynchronously in parallel because time influences the results of the ANAKG network.
- Integrate memory and associative processes

GOAL: Reproduce functionality of the biological neural substance!
Associative Spiking Neurons ASN

- Were developed to reproduce plasticity and associative properties of real neurons that work in time.

- They implement internal neuronal processes (IP) and efficiently manage their processing using internal process queues (IPQ) and a global event queue (GEQ).

- ASN neurons are updated only at the end of the internal processes (not continuously) to provide efficiency of data processing!
How ASN neurons work and how they are modeled?

Internal states of ASN neurons are updated only at the end of internal processes (IP) that are supervised by the Global Event Queue (GEQ).

IPQ represents a short sequence of internal changes of a neuronal state dependent on the external stimuli and previous internal states of the neuron.
Model and Adaptation of Associative Spiking Neurons

Synaptic efficacy defines the efficiency of the synapsis of the stimulations and spiking reactions of the postsynaptic neurons:

\[
\delta_{N_m,N_{m+r}} = \sum\{(s_m,s_{m+r}) \in S^n \in \mathbb{S}\} \frac{1}{\left(1 + \frac{\Delta t^A - \Delta t^C}{\theta_{N_{m+r}} \cdot \Delta t^R}\right)^\tau}
\]

It depends on:

- \(\Delta t^A\) - the period of time that lapsed between the stimulation of the synapse between the \(N_m\) and \(N_{m+r}\) neurons and the activation of the postsynaptic neuron \(N_{m+r}\) during training of the training sequence set \(\mathbb{S} = \{S^1, ..., S^N\}\);
- \(\Delta t^C\) - the period of time necessary to charge and activate the postsynaptic neuron \(N_{m+r}\) after stimulating the synapse between the \(N_m\) and \(N_{m+r}\) neurons (here \(\Delta t^C\));
- \(\Delta t^R = 200\text{ms}\) - the maximum period of time during which the postsynaptic neuron \(N_{m+r}\) recovers and returns to its resting state after its charging that was not strong enough to activate this neuron;
- \(\theta_{N_m} = 1\) - the activation threshold of the postsynaptic neuron \(N_{m+r}\);
- \(\tau = 4\) - the context influence factor changing the influence of the previously activated and connected neurons on the postsynaptic neuron \(N_{m+r}\).
Synaptic efficacy $\delta$ and the number $\eta$ of activations of the presynaptic neuron $N_m$ during training of the training sequence set $S$ is used to define synaptic permeability $p$:

$$p = \theta \cdot \frac{2 \cdot \delta}{\eta + \delta} \quad \text{OR} \quad p = \theta \cdot \frac{\eta \cdot \delta}{\eta \cdot \delta + \eta^2 - \delta^2}$$

Which is finally used to compute synaptic weights:

$$w = c \cdot p \cdot m$$

where $c$ is the synaptic influence: excitatory ($c = 1$) or inhibitory ($c = -1$), and $m$ is the multiplication factor modeling the number of synapses connecting the presynaptic and postsynaptic neurons.
Adaptation and Tuning of Associative Spiking Neurons

The weights computed in a presented way are good enough for the primary set of weights in the complex graph neural networks:

*I have a monkey. My monkey is very small. It is very lovely. It is also very clever.*

The introduced tuning process allows for the achievement of better recalling results thanks to the slight modification of the multiplication factors of the synapses.
Two repetitive steps of the **tuning process**:
1. All undesired and premature activations of neurons are avoided for all training sequences by using **weakening operations**.
2. Conflicts between correlated training sequences are fine-tuned using **strengthening operations**.

We define: $s_{\text{last}}^{\text{charge}}$ - the strength of the last stimulus,
$x$ – charge level at the moment when the last stimulus came
$x_{\text{all}}^{\text{max}}$ - the maximum dynamic charge level of each stimulated neuron

$$x_{\text{all}}^{\text{max}} = \begin{cases} x + s_{\text{last}}^{\text{charge}} & \text{if } x + s_{\text{last}}^{\text{charge}} > x_{\text{all}}^{\text{max}} \\ x_{\text{all}}^{\text{max}} & \text{otherwise} \end{cases}$$

$x_{\text{context}}^{\text{max}}$ - the previous maximum charge level establishing the context of the last stimulus that should activate the neuron:

$$x_{\text{context}}^{\text{max}} = x_{\text{all}}^{\text{max}} - s_{\text{last}}^{\text{charge}}$$

The correct activation of the neuron assumes that

$$x_{\text{context}}^{\text{max}} < \theta \leq x_{\text{context}}^{\text{max}} + s_{\text{last}}^{\text{charge}}$$

On this basis we can define **strengthening and weakening operations** for the tuning process.
Weakening Operation

The weakening operation defines how the multiplication factor $m$ decreases when a neuron is activated in the incorrect context or prematurely in the reduced context:

$$
\gamma = \begin{cases} 
\frac{\theta}{(x_{\text{all}}^{\text{max}} + \varepsilon)} & \text{for the undesired activations} \\
\frac{\theta}{(x_{\text{context}}^{\text{max}} + \varepsilon)} & \text{for the premature activations}
\end{cases}
$$

$$m = m \cdot \gamma$$

$$w = c \cdot p \cdot m$$

The multiplication factors of the incorrect activations must be deceased to operate on the right stimulation context of the next neurons of the recalled training sequence.

Weakening operations always start and finish the tuning process of the ANAKG network.
The strengthening operation defines how the multiplication factor $m$ increases when a neuron is not activated in the right context of all predecessor of the training sequence or too late:

$$\gamma = \frac{\theta}{x_{all}^{max} - \epsilon}$$

$$m = m \cdot \gamma$$

$$w = c \cdot p \cdot m$$

The strengthening operation always tries to achieve stimulation of the next sequence element. However, sometimes it is not beneficial if the initial context is not unique, e.g. there are few training sequences which start from the same subsequences of elements.
The achieved results confirm that the proposed tuning process is beneficial and produce better-adapted weights allowing to achieved better recalls from the ANAGK network.

<table>
<thead>
<tr>
<th>Input Stimulations</th>
<th>ANAKG Responses</th>
<th>Tuned ANAKG Responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>I also</td>
<td>I also have a</td>
<td>I also have a monkey</td>
</tr>
<tr>
<td>I have</td>
<td>I have a</td>
<td>I have a</td>
</tr>
<tr>
<td>I have an</td>
<td>I have a an old sister</td>
<td>I have an old sister</td>
</tr>
<tr>
<td>I have a young</td>
<td>I have a young brother</td>
<td>I have a young brother</td>
</tr>
<tr>
<td>I also have a big</td>
<td>I also have a big cat</td>
<td>I also have a big cat</td>
</tr>
<tr>
<td>You</td>
<td>You have a cat</td>
<td>You have a cat as well</td>
</tr>
<tr>
<td>My son</td>
<td>My son also has a monkey</td>
<td>My son also has a monkey</td>
</tr>
<tr>
<td>My brother</td>
<td>My brother</td>
<td>My brother is small</td>
</tr>
<tr>
<td>My monkey</td>
<td>My monkey</td>
<td>My monkey is</td>
</tr>
<tr>
<td>My monkey is very</td>
<td>My monkey is very small</td>
<td>My monkey is very small</td>
</tr>
<tr>
<td>It can</td>
<td>It can jump very quickly</td>
<td>It can jump very quickly</td>
</tr>
<tr>
<td>It is also</td>
<td>It is also very</td>
<td>It is also very clever</td>
</tr>
<tr>
<td>It is very</td>
<td>It is very</td>
<td>It is very lovely</td>
</tr>
<tr>
<td>It learns</td>
<td>It learns</td>
<td>It learns quickly</td>
</tr>
<tr>
<td>It likes to</td>
<td>It likes to sit on my his lamp monkey is small</td>
<td>It likes to sit on his lamp</td>
</tr>
<tr>
<td>She is very</td>
<td>She is</td>
<td>She is very lovely</td>
</tr>
<tr>
<td>She likes to</td>
<td>She likes to sit on my his lamp monkey is small</td>
<td>She likes to sit in the library and read books</td>
</tr>
<tr>
<td>He</td>
<td>He has a monkey</td>
<td>He has a monkey</td>
</tr>
<tr>
<td>He has</td>
<td>He has a monkey</td>
<td>He has a monkey and dogs</td>
</tr>
<tr>
<td>His monkey is</td>
<td>His monkey is</td>
<td>His monkey is small</td>
</tr>
<tr>
<td>His monkey is small</td>
<td>His monkey is small</td>
<td>His monkey is small as well</td>
</tr>
<tr>
<td>We have lovely</td>
<td>We have lovely</td>
<td>We have lovely dogs</td>
</tr>
</tbody>
</table>

TRAINING DATA SET:
I have a monkey.
My monkey is very small.
It is very lovely.
It can jump very quickly.
It is also very clever.
My monkey is lovely.
I also have a big cat.
My son also has a monkey.
It likes to sit on my head.
My sister has a small cat.
She is very lovely.
She likes to sit in the library and read books.
She quickly learns languages.
My sister has a cat.
It is very small.
You have a cat as well.
It is big.
I have a young brother.
My brother is small.
He has a monkey and dogs.
His monkey is small as well.
We have lovely dogs.
Conclusions

✓ The presented fine-tuning algorithm adapts weights of the associative pulsing neurons of the ANAKG neural network more accurately and allows to achieve better recalling of training sequences.

<table>
<thead>
<tr>
<th>ANAKG:</th>
<th>UNTUNED</th>
<th>FINE-TUNED</th>
</tr>
</thead>
<tbody>
<tr>
<td>Evaluation Results</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 sequences</td>
<td>85%</td>
<td>100%</td>
</tr>
<tr>
<td>25 sequences</td>
<td>76%</td>
<td>95%</td>
</tr>
<tr>
<td>hundreds of very correlated sequences</td>
<td>54%</td>
<td>91%</td>
</tr>
</tbody>
</table>
Questions or Remarks?


