Towards Designing Linguistic Assessments Aggregation as a Distributed Neuroalgorithm

Alexander V. Demidovskij[1], Eduard A. Babkin[2]
National Research University
Higher School of Economics
Nizhny Novgorod, Russian Federation
[1]ademidovskij@hse.ru, [2]eababkin@hse.ru

***Abstract*— A challenging problem which arises in the domain of integrating symbolic and sub-symbolic computations within a massively parallel computational environments like Internet of Things is considered in application to the Linguistic Decision Making tasks. A novel theoretical idea is proposed on expressing linguistic operators in dynamics of an artificial neural network. The proposal consists of two consequent stages: expressing linguistic operators as structural manipulations and translating them in a neuroalgorithm. The theoretical foundation is Tensor Product Representation (TPR) that provides a generic framework of designing a neural network that does not require training and produces an exact result equivalent to the result of symbolic algorithms. This paper discusses viability of the proposed idea, demonstrates design of TPR-based arithmetic as a basic building block for construction of such a method and elaborates directions of further research.**

Keywords*— *artificial neural networks; linguistic decision making; tensor product representations

I. Introduction

For many years, symbolic and sub-symbolic computations were considered as conflicting and competing paradigms. At the same time, the question of building integrated solutions has received much attention over the last two decades. One of the main reasons for building hybrid solutions is to benefit from both paradigms' advantages: high interpretability of symbolic models and massive parallelism and robustness of the connectionist or sub-symbolic models. Artificial Neural Networks (ANN) are often considered as universal highly scalable execution units [1], [2] that are incorporated in a wide range of industrial applications. In the field of design of distributed neural models there is an important scientific task of constructing neural networks that perform significant intellectual tasks without preliminary training stage [3] in massively parallel computation environments like multi-agent systems or Internet Of Things (IoT), where each component plays a role of a single neuron or a small subnetwork [4].

The first step towards solving that task would be design of ANN capable of producing an exact solution for a selected motivating problem. The Multi-Attribute Decision Making based on fuzzy linguistic assessments provided by experts [5] can be considered as such motivating problem which includes different input types of symbolic structures [6]: semantic, syntax parse trees, morpheme analysis, as well as the alternatives, criteria and experts' assessments. Application of sub-symbolic paradigm to the Decision Making has been analyzed by many researchers. General rules of unification and logical output are presented in [3], however there is no proposal on the way to express arithmetic operations using the neural network. On the other hand, in [7] fuzzy assessments are considered, and researches made an attempt to build a neural model on top of pre-processed fuzzy numbers. However, this model requires training on a selected dataset, finding which is already a challenge in the field of decision making. Apart from that, the constructed neural network is specific for each task and should be retrained for each new Decision Making scenario. Therefore, the task of expressing Decision Making process with fuzzy assessments on the sub-symbolic level is still an actual, challenging and open task.

This paper investigates the foundations for the solution of assessments aggregation during Linguistic Decision Making on the neural level without initial training. In this paper the 2-tuple model and corresponding aggregation operators [8] are used for modeling linguistic assessments, while Tensor Product Representations approach (TPR) [9] is proposed for encoding and decoding of structures as well manipulating them on the tensor level. The aim of the current study is to determine vital preconditions of further work and analyze the applicability of TPR to expressing linguistic operators. The hypothesis can be formulated as follows: if 2-tuple manipulations can be expressed as structural operations, like joining or extraction of elements, then TPRs can be used to encode 2-tuples and perform their aggregation on the sub-symbolic level in a distributed fashion that neural networks provide.

This paper is organized as follows. Section II gives a brief overview of the Linguistic Decision Making methods and basic models. Section III examines the TPR, rules of encoding and decoding of arbitrary recursive structures. Design of TPR-based arithmetic as a foundation for 2-tuple aggregation is outlined in Section IV. Directions of further research and conclusions are drawn in Section V.

II. Fuzzy linguistic variables

Modern methods of multi-attribute multi-level linguistic decision making use a traditional 2-tuple model as a basic building block [8]. They are considered to be more effective than other ways of elaborating fuzzy judgments of experts [7].

The reported study was funded by RFBR, project number 19-37-90058

978-1-7281-9692-3/20/$31.00 ©2020 IEEE

With this model it is possible to consider not only qualitative, but also quantitative estimations given by experts for the given alternative solutions. The 2-tuple model is based on the concept of symbolic translation [5].

Definition 1. A 2-tuple structure includes a pair (s_i, α) where $s_i \in S = \{s_0, \dots, s_g\}$– is a linguistic term (concept), α– a numeric value or a symbolic translation that shows a result of execution of membership function. It shows the distance to the closest concept $s_i \in S = \{s_0, \dots, s_g\}$ if a membership function does not result in an exact value (s_i).

Definition 2. Translation rule. Let $S = \{s_0, \dots, s_g\}$ be a linguistic scale, where $g = \tau + 1$ denotes granularity level of S. If $\beta \in [0,1]$ is a result of symbolic aggregation, then there is a way to recover a corresponding 2-tuple element:

$$\Delta_g = [0,1] \to S \times [-0.5, 0.5)$$
$$\Delta_g(\beta) = (s_i, \alpha) = \begin{cases} s_i, & i = round(\beta\tau) \\ \alpha = \beta\tau - i, & \alpha \in [-0.5, 0.5) \end{cases} \quad (1)$$

Definition 3. Reverse translation rule. Let $S = \{s_0, \dots, s_g\}$ be a linguistic scale, where $g = \tau + 1$ denotes granularity level of S. Let (s_i, α) be a 2-tuple element on a linguistic scale S, where $\alpha \in [-0.5, 0.5)$. Then there is a way to transform 2-tuple element to a numeric form of $\beta \in [0,1]$:

$$\Delta_g^{-1} = S \times [-0.5, 0.5) \to [0,1]$$
$$\Delta_g^{-1}(s_i, \alpha) = \frac{i + \alpha}{\tau} \quad (2)$$

Recent developments in Linguistic Decision Making approaches have led to elaboration of multiple algorithms of aggregating linguistic data. They are usually referred to as Linguistic Operators: MTWA (Multigranularity 2-tuple Weighted Averaging), MHTWA (Multigranularity Hesitant 2-tuple Weighted Averaging) [10], P2TLWA (Pythagorean 2-tuple Linguistic Weighted Averaging) [11] etc. One of them, MTWA, performs calculation of weighted average across a set of 2-tuple elements.

Definition4. MTWA operator. Let (b_i, α_i) be a 2-tuple element on a linguistic scale S^{g_i}, $i = 1, 2, \dots, n$. Let $w = (w_1, w_2, \dots, w_n)$ be a given weighting vector where w_i denotes a weight for (b_i, α_i), $i = 1, 2, \dots, n$. Then the MTWA operator is defined by:

$$MTWA^{w}_{S^{g_k}}((b_1, \alpha_1), (b_2, \alpha_2), \dots, (b_n, \alpha_n)) = \Delta_{g_k}\left(\sum_{j=1}^{n} w_j \Delta_{g_j}^{-1}(b_j, \alpha_j)\right) \quad (3)$$

However, the 2-tuple model is a basic model, in recent years various methods that extend the original ideas of aggregating quantitative data were put forward: Hesitant Fuzzy Linguistic Term Sets (HFLTS) [12], Institutional 2-tuple [13], [14], hybrid models [15] etc. 2-tuple model and its modifications have received much attention in the past decade among traditional Decision Making methods that start actively integrating linguistic information aggregation. Several studies, for instance, have been carried out to integrate 2-tupel models into the TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution) framework [16], [17]. More recent evidence [18], [19] shows that there are emerging methods that consider linguistic assessments given by experts as well as a degree of their confidence in given estimations.

III. Tensor Product Variable Binding Framework

As it was stated before, representing knowledge in a form of a recursive structure has many applications in the field. At the same time, the structure is a symbolic element by definition. Therefore, it cannot be used directly in sub-symbolic computations. For many years there has been a proposal to consider any structure as a set of pairs, each pair containing a filler and a role. In [9] P. Smolensky states that considering a structure as a set of pairs is a viable solution and formulated it in a form of Tensor Product Variable Binding rules. Moreover, they proposed a method of translating symbolic structure to a distributed tensor representation as well as general rules of expressing symbolic operations on the tensor level [20].

Definition 4. Filler – a particular instance of the given structural type.

Definition 5. Role – a function that filler presents in a structure.

Definition 6. Tensor multiplication is an operation over two tensors a with rank x and b with rank y that produces a tensor z has rank $x + y$ and it consists of pair-wise multiplications of all elements from x and y.

Definition 7. Tensor product for a structure. A structure is perceived as a set of pairs of fillers $\{f_i\}$ and roles $\{r_i\}$ and its tensor product is found as (4).

$$\psi = \sum_i f_i \otimes r_i \quad (4)$$

Definition 8. Joining operation $cons(p, q)$ is an action over two structures (trees) so that the tree *p* is sliding as a whole "down to the left" so that its root is moved to the left-child-of-the-root position and tree *q* is sliding "down to the right".

Operation *cons* can be expressed for binary trees as:

$$\begin{aligned} cons(p, q) &= p \otimes r_0 + p \otimes r_1 \\ cons_0(p) &\equiv cons(p, \emptyset) \\ cons_1(q) &\equiv cons(\emptyset, q) \end{aligned} \quad (5)$$

where r_0 and r_1 are roles, $\emptyset$ is empty tree.

It was proved [9] that this operation can be expressed in matrix form given that it operates over the tensor representation of structures (6).

$$cons(p, q) = W_{cons0}\, p + W_{cons1}\, q \quad (6)$$

Definition 9. Extracting operation $ex_X(p)$ is an action over a single structure (tree) so that the X_{th} child of the root becomes an independent tree and the remaining part of the original tree is no longer used.

Operation ex_x can be expressed for binary trees as:

$$\begin{aligned} ex_0(s) &= p \\ ex_1(s) &= q \end{aligned} \tag{7}$$

where $s = cons(p, q)$.

It was proved [9] that this operation can be expressed in matrix form given that it operates over the tensor representation of a structure (8).

$$ex_x(s) = W_{ex_x} s \tag{8}$$

For the past five years there has been a rapid rise in the use of TPR in the applied tasks. In particular, modern neural architectures were proposed to encode a structure in a tensor form [21], recover a structure without any information loss from a distributed representation [22], as well as performing basic structural operation of joining two trees in one and extracting sub-tree by a given set of rules [23].

On the other hand, TPR ideas are widely used for Deep Neural Networks. For example, this approach was applied to a caption generation task [24]. TPR are attracting considerable attention due to their ability to examine distributed representations that a Deep Neural Network can learn during the training process. In [25], authors try to investigate whether a neural network was capable of learning structural relations between objects in the input data. To sum up, TPR provide a generic theoretical framework for translating structures to the distributed representation, performing symbolic operations on the tensor level and recovery of the symbolic structure from the resulting representation.

IV. Design of TPR-based Arithmetic as a Foundation for 2-Tuple Aggregation

Aggregation of linguistic information that is expressed in a form of 2-tuples implies translation to the numeric values according to (1). This reveals a hypothesis that expression of basic arithmetic rules in terms of structural manipulations with TPR would allow aggregation of 2-tuple on the neural level. To design TPR-based arithmetic the axiomatic should be defined, including both primitives and operators. It was decided that the best procedure would be to formulate arithmetic in a Peano arithmetic [26] manner, where three basic actions should be defined: inc – increase by one, $equal$ – check for equality and if – select between actions depending on equality of condition to zero.

In the framework of that axiomatic and following basic TPR principles we propose to consider any number as a tree with two positional roles: r_0 (left child) and r_1 (right child). A structure is defined recursively and has an arbitrary depth. As it was mentioned above, there are two primitive values. Let $\{0\}$ be a structure that consists of only zeros (Fig. 1,*a*). At the same time, the other primitive value $\{1\}$ (Fig. 1.b) consists of a filler with a role r_1. Those two primitive values would be used for construction of structures that stand for other numbers, for example {2} (Fig. 1,*c*).

Using the structures proposed all foundational arithmetic operations can be expressed in terms of TPR as follows.

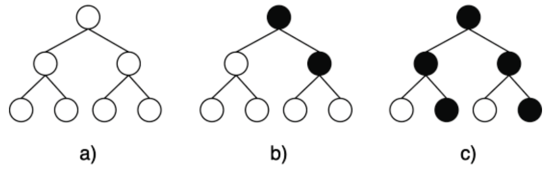

Fig. 1. Representing non-negative integers as structures. a) a structure representing {0} b) a structure representing {1} c) a structure representing {2}. Note: depth of the tree depends on the maximum value of integer that should be represented in distributed fashion

TPR-Inc. The operator for increasing a value by one. Let $\{1\}$ denote a structure, representing integer 1. Then incrementing operator receives a structure a, representing a number as an input. Result is a structure s that represents the number incremented by {1}. The operator can be defined by:

$$inc(a) = cons(a, \{1\}) \tag{9}$$

TPR-Dec. The operator for decreasing a value by one. Let $\{0\}$ denote a structure, representing 0. Then decrementing operator receives a structure a, representing a number as an input. Result is a structure s that represents the number decremented by {1}. The decrementing operator can be defined by:

$$dec(a) = \begin{cases} \{0\}, & if\ equal(a, \{0\}) \\ ex_0(a), & otherwise \end{cases} \tag{10}$$

TPR-Eq. The operator for checking two structures on equality. Let a, b denote two structures. Then the operator can be defined by:

$$equal(a,b) = \begin{cases} equal(dec(a), dec(b)), \\ \quad if\ a \neq \{0\}, b \neq \{0\} \\ \{0\}, \ if\ a \neq \{0\}, b = \{0\} \\ \{0\}, \ if\ a = \{0\}, b \neq \{0\} \\ \{1\}, \ if\ a = \{0\}, b = \{0\} \end{cases} \tag{11}$$

TPR-Sum. The operator for sum of two integers. Let a, b denote two structures. Then the operator can be defined by:

$$plus(a,b) = \begin{cases} plus(dec(a), inc(b)), & if\ a \neq \{0\} \\ b, & if\ a = \{0\} \end{cases} \tag{12}$$

TPR-Mult. The operator for multiplication of two integers. Let a, b denote two structures. Then the operator can be defined by:

$$times(a,b) = \begin{cases} plus(times(dec(a), b), b), \\ \quad if\ a \neq \{0\}, b \neq \{0\} \\ \quad \{0\}, \\ \quad if\ a \neq \{0\}, b = \{0\} \\ \quad \{0\}, \ if\ a = \{0\} \end{cases} \tag{13}$$

Examples of getting new numbers from primitive constants $\{0\}$ and $\{1\}$ are demonstrated on Fig. 1. For example, number 2 is a result of application inc operator to the primitive number 1, that is on the structural level equivalent to joining procedure of a structure that represents 1 with a role r_0 and a structure that represents 1 with a role r_1. Definition of basic operators allows defining more complicated operators like TPR-Sum (12) and TPR-Mult (13). The most basic building blocks, such as TPR-Inc (9), are designed in a neural form and available as an open-source project (https://github.com/demid5111/ldss-tensor-structures).

V. CONCLUSION

To sum, it was demonstrated that numbers can be encoded as a recursive TPR-based structure and integer arithmetic can be expressed in a form of structural TPR-based manipulations. These results confirm our initial hypothesis of feasibility of creation a method that allows aggregating fuzzy linguistic estimations on the neural level. Indeed, as it was demonstrated in Section II, any linguistic estimations can be translated to a numeric format and after aggregation back to a new 2-tuple element. Then an average weighting operator (3) that by definition consists of sums and multiplications could be expressed in a form of structural manipulations because each of these arithmetic operations is expressed in structural form (12, 13). In turn, there are known methods and instruments that allow performing such manipulations with a help of generated neural network [20], [21], [22], [23]. As our foundational hypothesis was confirmed the further research topic is integration of the blocks designed for building a complete solution capable of performing linguistics assessments aggregation on the neural level.

REFERENCES

[1] Rumelhart D. E., Hinton G.E., McClelland, J.L. A general framework for parallel distributed processing. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1, pp. 45-76.

[2] Pinkas G. Reasoning, nonmonotonicity and learning in connectionist networks that capture propositional knowledge. Artificial Intelligence, vol. 77, i. 2, pp. 203-247.

[3] Pinkas G., Lima P., Cohen S. Representing, binding, retrieving and unifying relational knowledge using pools of neural binders. Biologically Inspired Cognitive Architectures, vol. 6, pp. 87-95.

[4] Yousefpour A., Nguyen B.Q., Devic S., Wang G., Kreidieh A., Lobel H., Jue J. P. Failout: Achieving Failure-Resilient Inference in Distributed Neural Networks. 2020. arXiv preprint arXiv:2002.07386.

[5] Herrera F., Martínez L. A 2-tuple fuzzy linguistic representation model for computing with words. IEEE Transactions on fuzzy systems, vol. 8, i. 6, pp. 746-752.

[6] Cho P.W., Goldrick M., Smolensky P. Incremental parsing in a continuous dynamical system: Sentence processing in Gradient Symbolic Computation. Linguistics Vanguard, vol. 3, i. 1.

[7] Golmohammadi D. Neural network application for fuzzy multi-criteria decision making problems. International Journal of Production Economics, vol. 131, i. 2, pp. 490-504.

[8] Zhang H., Wu Y., Gao J., Xu C. A Method for Multi-Criteria Group Decision Making with 2-Tuple Linguistic Information Based on Cloud Model. Information, vol. 8, i. 2.

[9] Smolensky P. Tensor product variable binding and the representation of symbolic structures in connectionist systems. Artificial intelligence, vol. 46, i. 1-2, pp. 159-216.

[10] Wei C., Liao H. A multigranularity linguistic group decision- making method based on hesitant 2- tuple sets. International Journal of Intelligent Systems, vol. 31, i. 6, pp. 612-634.

[11] Wei G., Lu M., Alsaadi F.E., Hayat T., Alsaedi A. Pythagorean 2-tuple linguistic aggregation operators in multiple attribute decision making. Journal of Intelligent & Fuzzy Systems, vol. 33, i. 2, pp. 1129-1142.

[12] Rodriguez R.M., Martinez L., Herrera F. Hesitant fuzzy linguistic term sets for decision making. IEEE Transactions on fuzzy systems, vol. 20, i. 1, pp. 109-119.

[13] Liu P., Chen S. M. Multiattribute group decision making based on intuitionistic 2-tuple linguistic information. Information Sciences, vol. 430, pp. 599-619.

[14] Wei G., Alsaadi F.E., Hayat, T., Alsaedi A. Picture 2-tuple linguistic aggregation operators in multiple attribute decision making. Soft Computing, vol. 22, i. 3, pp. 989-1002.

[15] Wang J.H., Hao J. A new version of 2-tuple fuzzy linguistic representation model for computing with words. IEEE transactions on fuzzy systems, vol. 14, i. 3, pp. 435-445.

[16] Geng X., Gong X., Chu X. Component oriented remanufacturing decision-making for complex product using DEA and interval 2-tuple linguistic TOPSIS. International Journal of Computational Intelligence Systems, vol. 9, i. 5, pp. 984-1000.

[17] Cheng P., Zhou B., Chen Z., Tan J. The TOPSIS method for decision making with 2-tuple linguistic intuitionistic fuzzy sets. In 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 2017, pp. 1603-1607.

[18] WangJ., Wei G., Lu M. TODIM method for multiple attribute group decision making under 2-tuple linguistic neutrosophic environment. Symmetry, vol. 10, i. 10, pp. 486.

[19] Demidovskij A.V., Babkin E.A. Developing a distributed linguistic decision making system. *Biznes-informatika* [Business Informatics], vol. 13, i. 1. (In Russian)

[20] Legendre G., Miyata Y., Smolensky P. Distributed recursive structure processing. In Advances in Neural Information Processing Systems, 1991, pp. 591-597.

[21] Demidovskij A. Implementation Aspects of Tensor Product Variable Binding in Connectionist Systems. In Proceedings of SAI Intelligent Systems Conference, 2019, pp. 97-110.

[22] Demidovskij A., Babkin E., “Robust Encoding and Decoding of Recursive Structures in Connectionist Systems”, unpublished.

[23] Demidovskij A.V. Towards Automatic Manipulation of Arbitrary Structures in Connectivist Paradigm with Tensor Product Variable Binding. In International Conference on Neuroinformatics, 2019, pp. 375-383.

[24] Huang Q., Smolensky P., He X., Deng L., Wu D. Tensor product generation networks for deep NLP modeling. arXiv preprint arXiv:1709.09118.

[25] Soulos P., McCoy T., Linzen T., Smolensky P. Discovering the compositional structure of vector representations with role learning networks. arXiv preprint arXiv:1910.09113.

[26] Van Heijenoort, J. From Frege to Gödel: a source book in mathematical logic, 1879-1932. Harvard University Press. 1967.