UPGRADING THE JANET NEURAL NETWORK BY INTRODUCING A NEW STORAGE BUFFER OF WORKING MEMORY

A. Tolic[*], *B.M. Boshkoska*[†], *S. Skansi*[‡]

Abstract: Recurrent neural networks (RNNs), along with long short-term memory networks (LSTMs), have been successfully used on a wide range of sequential data problems and have been entitled as extraordinarily powerful tools for learning and processing such data. However, the search for a new or derived architecture that would model very long-term dependencies is still an active area of research. In this paper, a relatively psychologically plausible architecture named event buffering JANET (EB-JANET) is proposed. The architecture is derived from the forget-gate-only version of the LSTM, which is also called just another network (JANET). The new architecture implements a new working memory mechanism that operates on information represented as dynamic events. The event buffer, as a container of events, is a reference to the state of the relevant pre-activation values on the basis of which historical candidate values were generated relative to the current timestep. The buffer is emptied as needed and depending on the context of information. The proposed architecture has achieved world-class results and it outperforms JANET on multiple benchmark datasets. Moreover, the new architecture is applicable to a wider class of problems and showed superior resilience when processing longer sequences, as opposed to JANET which experienced catastrophic failures on certain tasks.

Key words: *event buffer, recurrent neural network, memory modeling, long term memory, working memory*

Received: May 16, 2023
Revised and accepted: December 27, 2023

DOI: 10.14311/NNW.2023.33.024

1. Introduction

Artificial neural networks are mathematical models that leverage learning algorithms inspired by the brain to store information [1]. An RNN [2] is a type of artificial neural network designed for processing sequential data or time series data. To address the vanishing gradient problem and model long-term dependencies, Sepp

[*]Antonio Tolic – Corresponding author; Faculty of information studies, Ljubljanska cesta 31A, p.p. 603, 8000 Novo mesto, Slovenia E-mail: antonio.tolic@student.fis.unm.si

[†]Biljana Mileva Boshkoska; Department of Knowledge Technologies, Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia, E-mail: biljana.mileva@fis.unm.si

[‡]Sandro Skansi; Department of Philosophy and Cultural Studies, Faculty of Croatian Studies, Borongajska cesta 83d, 10000 Zagreb, Croatia, E-mail: sskansi@hrstud.hr

Hochreiter and Juergen Schmidhuber introduced a variant known as the (vanilla) LSTM network in 1997 [3]. Following this development, numerous studies have focused on analyzing the components of the LSTM and exploring alternative architectures. Significantly, simultaneous discoveries highlighted the forget gate as a crucial element of the LSTM [5, 7], an addition absent in the original 1997 architecture but introduced in 2000 [8]. Furthermore, a new neural network was proposed in 2018 as a transformation and simplification of the LSTM in that it uses only the forget gate and chrono-initialized biases [9] and it was named JANET, an abbreviation for just another network [10]. The JANET architecture, crucial for this research, consistently outperforms the LSTM on multiple benchmark datasets. Considering the achieved results, it is also remarkable that, even today, after several years of existence, it can still compete with the best state-of-the-art models (for example [11, 12]). However, the JANET architecture unfortunately also has certain shortcomings, which will be discussed further in the paper.

The LSTM architecture, upon psychological examination, is not inherently designed to be psychologically plausible, despite some correlations with cognitive concepts. Psychologically plausible models aim to reflect human cognition and brain functions. These models strive to incorporate more intricate mechanisms in order to accurately mirror human cognitive processes. While there are connections between certain LSTM architectures and certain aspects of human cognition, such as the notions of sensory memory (SM), short-term memory (STM), and long-term memory (LTM), the LSTM architectures do not directly implement or simulate specific cognitive mechanisms observed in human memory. However, while revising human memory theory and recent psychological research, an idea emerged to map specific findings, such as those in [19], related to working memory (WM) onto an existing neural network architecture. The intention was to create or adapt an artificial neural network, such as the LSTM architecture, as a psychologically plausible model aligned with the revised theory while maintaining functionality.

This research begins by examining the psychological aspects of LTM and WM. Subsequently, a basic overview of RNNs and the LSTM is provided. However, emphasis is placed on the JANET neural network as this architecture serves as a foundational element, possessing essential building components necessary for subsequent upgrades. The focus then shifts to the implementation, where, prior to the actual technical implementation, the gating mechanism of WM and the interoperability between LTM and WM are explored, also from a psychological perspective. Additionally, within the context of technical implementation, a new storage buffer for WM is introduced. Finally, the results of experiments involving real and publicly available datasets, along with synthetic datasets, are presented to further evaluate the performance of the proposed models and architectures. The implementation of this idea was anticipated to yield improvements in certain results from the conducted experiments, while also mitigating some of the drawbacks associated with the JANET architecture.

2. Related work

In the context of cognitive psychology, over the last few decades, various memory models have been proposed. In this study, particular emphasis is placed on

the Atkinson-Shiffrin model [13] and its variant [16] for several compelling reasons. Firstly, the Atkinson-Shiffrin model provides a comprehensive framework that elegantly captures the different stages of memory processing. Secondly, the Atkinson-Shiffrin model has substantial empirical support, with numerous studies validating its principles. Additionally, the variant proposed in [16] introduces a nuanced perspective or extends the original model, addressing emerging questions and incorporating additional factors. Although the Atkinson-Shiffrin model and its variant elegantly describe human memory processes, comprehending the components and their operations is a complex task. Nevertheless, extensive research has firmly established the foundational theory of these crucial components, including WM [17, 18], central to this paper. However, the specifics of WM operations and its interaction with information involve an additional theory. This theory proposes the existence of gate mechanisms [20, 21], directly controlling and interacting with information.

To transition successfully from the conceptualization to the implementation phase, it was necessary to revise the RNN architectures of artificial neural networks to achieve meaningful synergy. Just as in the field of cognitive psychology, diverse architectures and initialization methods have been formulated through extensive research, aiming to optimize sequential data processing. Upon reviewing the literature, it was concluded that the key reference points could be the building blocks of the LSTM architecture [8]. Nevertheless, as a simplification of the LSTM, the JANET neural network [10] quickly emerged as the foundation for constructing a new neural network. This is because of its simple architecture that incorporates an LTM mechanism, which already aligns well with the mechanisms of the Atkinson-Shiffrin model and its variants. With the aforementioned reasons, another significant factor drawing attention to the JANET architecture is a new technique [9] utilized by this architecture. This technique enables the finalization of the idea behind this research. Although various papers are implicitly linked to this research, the references in this section provide the primary source of information.

3. Memory

Memory encompasses the processes of collecting, storing, retaining, and retrieving information, existing in diverse forms and relying on distinct neural systems. Memory models are usually used to describe the way of organizing and defining how memory behaves. The model most people are familiar with is the Atkinson-Shiffrin model, also known as the multi-store model or the modal model, illustrated in Fig. 1 (disregarding the dashed lines A and B). Atkinson and Shiffrin developed the multi-store model of memory that describes the flow between the already mentioned storage systems of memory: SM, STM, and LTM [13]. Furthermore, when describing the information flow between the components, Atkinson and Shiffrin were not sure about one part of their model, that is, they left open the possibility that there may be a direct transfer of information from SM (only the dashed line A in Fig. 1) to LTM. According to [16], in addition to the many variants of the model, the Atkinson-Shiffrin model has also been modified by certain psychologists, who claim that information could directly enter LTM through "a back door", that is, without us consciously attending to it (automatic processing, as depicted in Fig. 1, while considering only the dashed line B).

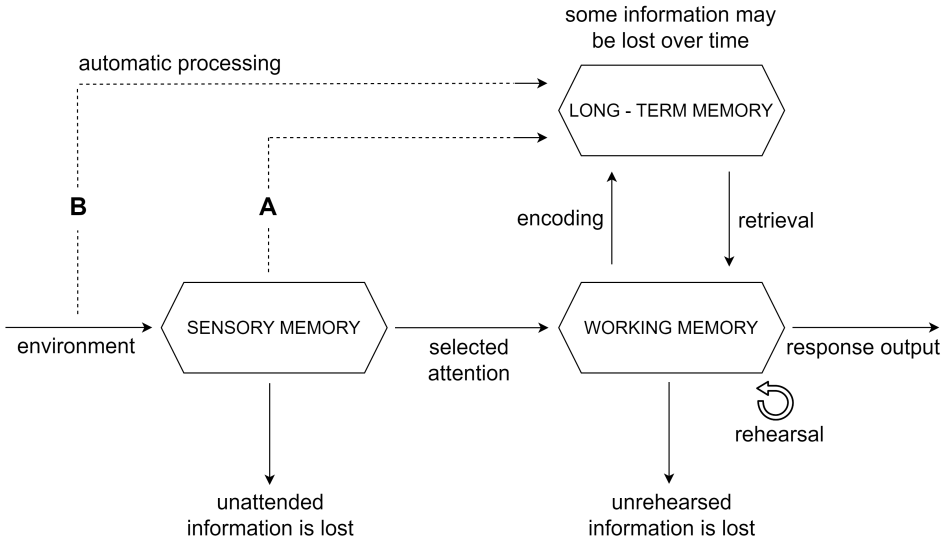


Fig. 1 *The Atkinson and Shiffrin's model of memory and its variants.*

3.1 Long-term memory

LTM represents the theoretical construct used in cognitive psychology and cognitive neuroscience which is described as an unlimited storage of information that is needed in the process of creating enduring memories [22, 23]. This storage of memories tends to be more durable and stable, lasting for a long time, in contrast to the relatively short-term nature of WM. In theory, the capacity of LTM could be unlimited, but a distinction should be made here in relation to the process of retrieving information from LTM. In other words, information can be written into LTM, but its retrieval can be hindered.

3.2 Working memory

WM is a theoretical construct in cognitive psychology, functioning as a limited capacity system for the temporary storage and manipulation of active (new) information necessary for complex cognitive tasks [17, 24], overlapping structurally and functionally with LTM [25, 26]. It is believed that the information moving through STM is encoded into LTM through a process called memory consolidation. The mentioned process either leads to the formation of a permanent change in the brain, which is defined as an engram [27], or the information decays or is replaced. It should also be noted that the term WM is sometimes associated with the term STM. Despite the conceptual differences between them, the use of the terms WM and STM in theory and literature is not always consistent. WM and STM mechanisms are supposedly different theoretical concepts that are assumed to reflect different cognitive functions. However, researchers have not been able to tangibly separate both constructs. Moreover, there is evidence for a potential overlap [26]. In this paper, the term "working memory" will be used and the potential

differences between WM and STM will not be dwelled upon. It is also interesting to note that from a psychological perspective, LTM can be highly accurate; however, it is often considered perishable and inaccurate when compared to WM. According to [28], there is a possibility that WM has better fidelity than LTM, and, moreover, a similar scenario could potentially exist in the world of artificial neural networks.

4. Recurrent neural networks

RNNs are a generalization of the feed-forward neural network that have an internal memory and are one of the most successful class of architectures for solving sequential problems. The simplest form of an RNN, at any timestep t, takes $\mathbf{x}_t$ as the input and updates its hidden state $\mathbf{h}_t$ according to the rule

$$\mathbf{h}_t = \phi(\mathbf{U}_\mathrm{h}\mathbf{h}_{t-1} + \mathbf{W}_\mathrm{h}\mathbf{x}_t + \mathbf{b}_\mathrm{h}). \tag{1}$$

The linear layer is commonly employed to make predictions or extract information through a linear transformation applied to the hidden state $\mathbf{h}_t$, as follows:

$$\mathbf{y}_t = \mathbf{W}_\mathrm{y} \cdot \mathbf{h}_t + \mathbf{b}_\mathrm{y}. \tag{2}$$

In the aforementioned equations, $\mathbf{U}$ and $\mathbf{W}$ represent the weight matrices with their respective layer denotations, while $\mathbf{b}$ represents the bias vector with its corresponding layer denotation. ϕ is the nonlinear activation function. However, RNNs suffer from the problem of vanishing gradients that highly affect the ability to learn. Therefore, several approaches have been developed to solve such a problem.

4.1 Long short-term memory

RNNs are hard to train due to the exploding and vanishing gradient problems [29, 30]. LSTM networks, as instances of a more general class of recurrent neural networks, aim to mitigate the aforementioned problems, especially the vanishing problem. Hence, the LSTMs are capable of modeling longer term dependencies by having memory cells which can maintain their state over time and the gates which control the information flow along with the memory cells. In other words, let $\mathbf{x}_t$ be the input at any time step t where $\mathbf{f}_t$, $\mathbf{i}_t$, $\mathbf{o}_t$, and $\mathbf{c}_t$ represent the forget gate, input gate, output gate, and the memory of the current timestep t. The forget gate is responsible for controlling what information to throw away (0 to forget entirely) or keep (1 to completely remember) in the memory. The LSTM decides what new information will be stored in the cell state. This is done in such a way that the input gate decides which values it should update or filter and the tanh layer $\widetilde{\mathbf{c}}_t$ creates a vector of new candidate values that can be added to the cell state. The equation of the candidate value is very similar to the simple RNN. The LSTM output gate controls how much of the current cell activity will be released. The hidden state $\mathbf{h}_t$ at each timestep learns which data to keep and which to discard. The operability of the standard LSTM can be summarized through the following equations (σ is the symbol for the sigmoid activation function, and $\odot$ is the symbol for the element-wise or pointwise product, also known as the Hadamard product)

$$\begin{aligned}
\mathbf{f}_t &= \sigma(\mathbf{U}_\mathrm{f}\mathbf{h}_{t-1} + \mathbf{W}_\mathrm{f}\mathbf{x}_t + \mathbf{b}_\mathrm{f}),\\
\mathbf{i}_t &= \sigma(\mathbf{U}_\mathrm{i}\mathbf{h}_{t-1} + \mathbf{W}_\mathrm{i}\mathbf{x}_t + \mathbf{b}_\mathrm{i}),\\
\widetilde{\mathbf{c}}_t &= \tanh(\mathbf{U}_{\widetilde{\mathrm{c}}}\mathbf{h}_{t-1} + \mathbf{W}_{\widetilde{\mathrm{c}}}\mathbf{x}_t + \mathbf{b}_{\widetilde{\mathrm{c}}}),\\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \widetilde{\mathbf{c}}_t,\\
\mathbf{o}_t &= \sigma(\mathbf{U}_\mathrm{o}\mathbf{h}_{t-1} + \mathbf{W}_\mathrm{o}\mathbf{x}_t + \mathbf{b}_\mathrm{o}),\\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t).
\end{aligned} \tag{3}$$

The linear layer is commonly employed to make predictions or extract information through a linear transformation applied to the hidden state $\mathbf{h}_t$, as follows:

$$\mathbf{y}_t = \mathbf{W}_\mathrm{y} \cdot \mathbf{h}_t + \mathbf{b}_\mathrm{y}. \tag{4}$$

In the aforementioned equations, **U** and **W** represent the weight matrices with their respective layer denotations, while **b** represents the bias vector with its corresponding layer denotation.

4.2 Just another network

In order to transform the LSTM architecture into the JANET architecture, the first step is to couple together the forget and input gate as in [5], that is $\mathbf{i}{=}1 - \mathbf{f}$. The tanh activation of $\mathbf{h}_t$ can worsen the vanishing gradient problem, whereas the weight matrix **U** could take values outside the range [-1,1]. Therefore, the specified non-linearity can be removed. Intuitively, if hypothetically more information is accumulated than forgotten, sequence analysis could be easier. This was empirically confirmed to be true by subtracting a pre-specified value $\boldsymbol{\beta}$ from the input control component. It was determined that the setting $\boldsymbol{\beta} = 1$ provides the best results for the data sets analyzed in [10]. The resulting architecture is described by the following equations:

$$\begin{aligned}
\mathbf{g}_t &= \mathbf{U}_\mathrm{g}\mathbf{h}_{t-1} + \mathbf{W}_\mathrm{g}\mathbf{x}_t + \mathbf{b}_\mathrm{g},\\
\widetilde{\mathbf{c}}_t &= \tanh(\mathbf{g}_t),\\
\mathbf{f}_t &= \sigma(\mathbf{U}_\mathrm{f}\mathbf{h}_{t-1} + \mathbf{W}_\mathrm{f}\mathbf{x}_t + \mathbf{b}_\mathrm{f} - \boldsymbol{\beta}),\\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + (1 - \mathbf{f}_t) \odot \widetilde{\mathbf{c}}_t,\\
\mathbf{h}_t &= \mathbf{c}_t.
\end{aligned} \tag{5}$$

The linear layer is commonly employed to make predictions or extract information through a linear transformation applied to the hidden state $\mathbf{h}_t$, as follows:

$$\mathbf{y}_t = \mathbf{W}_\mathrm{y} \cdot \mathbf{h}_t + \mathbf{b}_\mathrm{y}. \tag{6}$$

In the aforementioned equations, **U** and **W** represent the weight matrices with their respective layer denotations, while **b** represents the bias vector with its corresponding layer denotation.

Unfortunately, this architecture is not suitable for the standard LSTM initialization scheme, in which the weight matrices **U** and **W** are distributed as $\mathcal{U}\left[\frac{-\sqrt{6}}{\sqrt{n_l+n_{l+1}}}, \frac{\sqrt{6}}{\sqrt{n_l+n_{l+1}}}\right]$, where $\mathcal{U}$ denotes the uniform distribution, n_l is the size of each layer l [31,32], and all the biases are set to zero, except for the forget gate bias,

which is initialized to 1 [6]. Let us note the following: the free input regime [33] is a theoretical event where inputs stop after a certain time t_0, $\mathbf{x}_t = 0$ for $t > t_0$ and information omission through the hidden layer is neglected, assuming that all other associated weight matrices are set to 0. Therefore, assuming that the inputs stopped after a certain time and neglecting information leakage through the hidden layer (also known as the free input regime), JANET would not be able to retain the memory values $\mathbf{c}_t$ for more than a few steps in the case of such initialization. In other words, the pre-activation vector of the forget gate would have a value of 1, which means that for n zero-valued inputs, the memory values would tend to be centered around

$$\mathbf{c}_{t+n} = \sigma(1)^n \odot \mathbf{c}_t \approx 0.73^n \mathbf{c}_t. \tag{7}$$

In 2018, a new initialization scheme was proposed for the LSTM gate biases. It was named the chrono initialization [9] and it initializes the LSTM gate biases as

$$\begin{aligned} \mathbf{b}_\mathrm{f} &\sim \log(\mathcal{U}[1, T_\mathrm{max} - 1]), \\ \mathbf{b}_\mathrm{i} &= -\mathbf{b}_\mathrm{f}, \end{aligned} \tag{8}$$

with T_max as the expected range of long-term dependencies and $\mathcal{U}$ as the uniform distribution. As demonstrated in the example using the MNIST dataset [34] provided by [10], employing the technique Eq. (8) in the JANET architecture implies that JANET will be able to retain memory values $\mathbf{c}_t$ for more than a few steps if consecutive zero inputs occur. In other words, if the images of the MNIST dataset are transformed into sequences of $T_\mathrm{max} = 784$ individual pixels (raster of 28×28 pixels), with each pixel considered as a moment in time, and the technique Eq. (8) is applied, then for n zero-valued inputs, the memory values would tend to be centered around

$$\mathbf{c}_{t+n} = \sigma(\log(T_\mathrm{max} - 1))^n \odot \mathbf{c}_t = \sigma(\log(783))^n \odot \mathbf{c}_t \approx 0.99^n \mathbf{c}_t. \tag{9}$$

Furthermore, the value of the partial derivative $\frac{\partial \mathbf{c}_{t+1}}{\partial \mathbf{c}_t}$ will be approximately 1 [10]. This implies that the length of the sequence has minimal impact on the gradients of memory cells.

4.2.1 JANET'S drawbacks

This neural network is not suitable when anticipation occurs, for example, at each timestep. It has also been specified in [10] that it can be expected that the LSTM is a better choice in next-word prediction tasks, especially when inputs are discrete and non-zero. Furthermore, our experiments revealed instances of catastrophic failures in JANET, particularly in segments that were not initially mentioned here as shortcomings. These observations suggest that JANET's applicability is limited to a narrower range of problem classes. The experiments utilize precisely selected and constructed datasets, taking into account the corresponding statements. Additionally, specific adjustments and challenges were introduced, such as alterations in task types and the transition of input values from discrete to continuous formats, reinforcing previous statements.

5. Towards implementation

Examining the JANET neural network Eq. (5), reveals a direct interaction between the data and LTM. In abstract terms, this interaction is illustrated in Fig. 1, where data undergoes automatic processing and directly integrates into LTM. Additionally, Fig. 1 illustrates the WM mechanism, raising the question of how and which data should interact first with WM in the context of an artificial neural network. A scenario is created in which at the time t one piece of data would be obtained. When going back to Fig. 1, it could potentially be concluded that the data obtained at the time t might enter both LTM and WM, but under certain conditions. That is, one piece of data enters LTM in its encoded form, but according to Fig. 1, the same piece of data enters WM without being encoded, but by being carefully selected. In other words, the data enters the WM mechanism that selectively directs attention towards the target and reduces irrelevant distractors. Hence, the data should be carefully selected so that WM holds only the information relevant to the current task. The conclusion is that only the pre-activation vectors should be observed, based on which candidate values are generated. Looking again at Fig. 1, this makes sense, the selected pre-activation values enter WM and the same, but encoded data enters LTM.

Since the operability of LTM is already integrated into JANET, the following subsections will focus on the details of WM operability, its interaction with LTM, and its implementation. Given the complexity of processes in the human brain, the SM component has been excluded to simplify the implementation.

5.1 Working memory gating

According to [35], WM is thought to be strongly related to cognitive control. Cognitive control orchestrates thoughts and actions in harmony with goals and contexts. In order for cognitive control to be successful in such an orchestration, it selects relevant perceptual information which is to be updated into WM (input gating). There it maintains and protects the selected information so that the information does not get corrupted under the influence of distractions (maintenance). Furthermore, it then filters a subset of the maintained information that is ready for use, all the while executing a complex cognitive task (output gating) [20, 21, 35]. The feasibility of the aforementioned takes place with the help of additional mechanisms, which are sometimes metaphorically called "gates" by cognitive neuroscientists. The gates allow the relevant information to enter WM when opened, but they protect the WM contents from interference when closed; or in simpler words, these gates control the flow of information [20, 21].

The implementation of the aforementioned theory proceeds as follows. A new variable $\mathbf{e}$ has been introduced. The variable can be updated by the artificial neural network as needed, and the operation that allows the logical and expected updating is shown by the succeeding equation with the pre-activation vector $\mathbf{g}$

$$\begin{aligned} \mathbf{g}_t &= \mathbf{U}_{\mathrm{g}}\mathbf{c}_{t-1} + \mathbf{W}_{\mathrm{g}}\mathbf{x}_t + \mathbf{b}_{\mathrm{g}}, \\ \mathbf{e}_t &= \mathbf{r}_t \odot \mathbf{e}_{t-1} + (1 - \mathbf{r}_t) \odot \mathbf{g}_t, \\ \mathbf{h}_t &= \mathbf{e}_t. \end{aligned} \tag{10}$$

Here it can be imagined that the variable **e** contains a certain amount of information and that it represents WM. The equation represents the WM mechanism, which makes it possible to generate the WM value at the time t. The variable **r** represents the gate (or sometimes the reset gate), and it is defined as the sigmoid activation function. It controls how the WM **e** should be updated based on the recurrent information and the pre-activation vector **g**. Hence, the gate acts almost as the forget gate, as it decides how much of the past information needs to be neglected. To be more precise, this gate controls how the input and the previous state determine the current state in its case.

5.2 Memory harmonization

In addition to perceptual information, WM may also incorporate information from LTM [36]. The interaction between LTM and WM is highly complex. Until recently, it was unclear whether a gate controls the selection of LTM representations into WM and what kind of interconnection is created during the WM gating of the perceptual information and LTM information, and how that information relates to each other. Recent research now provides evidence for a gating mechanism controlling the selection of LTM content entering WM. The findings showed that the access to WM for both the perceptual and LTM sources of information is plausible when controlled by a gate and attentional selection mechanism [36]. It should also be noted that according to [37], the more strongly the items are associated in LTM, the more benefit will the WM performance have. In order to obtain a synergistic effect of the joint interaction of LTM and WM, the technical perspective of implementation had to be observed, while at the same time a meaningful manipulation of various pieces of incorporated information had to be maintained. It was concluded that the selection of the data in WM must take place with the help of the acquired knowledge stored in LTM. In other words, the intention is to control the gate with the help of LTM. It was deduced and later empirically confirmed that, for controlling the WM state with LTM support, the most effective mathematical formulation for the gate function is as follows

$$\mathbf{r}_t = \sigma(\mathbf{U}_\mathrm{r}\mathbf{c}_t + \mathbf{W}_\mathrm{r}\mathbf{x}_t + \mathbf{b}_\mathrm{r}), \tag{11}$$

where **U** and **W** represent the weight matrices with their respective layer denotations, while **b** represents the bias vector with its corresponding layer denotation. It should be noted that the LTM values for the current timestep t have already been generated. In terms of the interoperability between LTM and WM, it has just been mentioned how LTM can impact WM. The question arises as to whether there is an opposite direction. According to the theory and our empirical tests, there is. It was deduced that if the forget gate is impacted by WM (analogous to the earlier case where it was shown how LTM impacts WM), the performance of the architecture will increase. In other words, the forget gate equation is now set as follows

$$\mathbf{f}_t = \sigma(\mathbf{U}_\mathrm{f}\mathbf{e}_{t-1} + \mathbf{W}_\mathrm{f}\mathbf{x}_t + \mathbf{b}_\mathrm{f}). \tag{12}$$

To conclude, it can be said that the control of retroactive memory interference depends on the adjacent type of memory.

5.3 Event buffering JANET

In the previous section and its corresponding subsections, the mathematical equations were systematically developed and integrated into the JANET neural network architecture, aligning with theories of cognitive processes related to WM. The constructed architecture can be visually observed in Fig. 2. If SM is neglected in the analysis due to the complexity of the entire model, it becomes evident that the architecture of the artificial neural network in Fig. 2 closely corresponds to and aligns with the Atkinson–Shiffrin memory model depicted in Fig. 1 (taking into account the variant with the dashed line B). At this point, it is presumed that the artificial neural network is also functional.

The JANET architecture, with additional WM, underwent testing and examinations in a manner similar to that described in the Experiments section. The results were compared with the same architecture but without WM. Although some improvements were present, they were not entirely satisfactory, and therefore, WM was not deemed acceptable (the aforementioned results were not included into the research of this paper). However, a paper that came out recently in relation to the

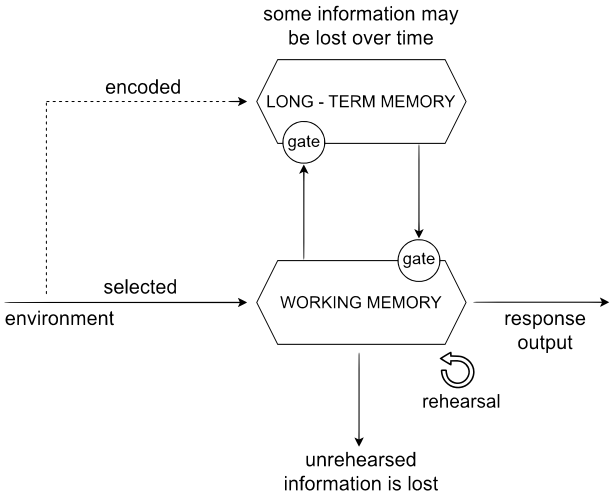


Fig. 2 *The JANET architecture leveraging working memory.*

time of writing this paper suggested changes in the way WM operates. In essence, the theory describing WM and WM gate mechanisms remains intact. However, recent advancements in cognitive psychology suggest an enhanced mechanism that refines the existing one. Namely, in [19], a new model of information processing in WM was proposed. The storage buffer of event information in WM was researched, defining the event as a segment of time at a given location perceived by the observer, having both a starting and ending point. Analyzing the WM mechanism through the Eq. (10) and considering the initial weight initializations, it can be inferred that there are deviations from the recently presented refinement of the WM theory. The Eq. (10) indicates a preference for retaining historical information, which might come into conflict with the principles of the new event buffer theory. The latter advocates for a structured and controlled approach to information re-

tention. Mathematically, this can be represented by resetting the WM container's value to a baseline level as required.

In order to use and technically implement the previously described event buffer theory, it was concluded that the key is to to access sequentially historical information, but only up to the point that breaks relevance for the current timestep. The mitigating factor is that there is no need for extensive mathematical manipulation with equations to carry out the implementation of the event buffer theory. In essence, the mechanism necessitates a subtly different implementation, incorporating a subtle yet pivotal adjustment, signifying that the following approach is the correct one

$$\mathbf{e}_t = \mathbf{r}_t \odot \mathbf{g}_t + (1 - \mathbf{r}_t) \odot \mathbf{e}_{t-1}. \tag{13}$$

The variables $\mathbf{e}_{t-1}$ and $\mathbf{g}_t$ have been interchanged, but the same initialization scheme of the $\mathbf{r}$ gate, as it was the case earlier, has been kept. It would now be best to reintroduce the theoretical situation of the free input regime. If there was no external input to the neural network after the time step t_0, that is, $\mathbf{x}_t = 0$ for $t > t_0$ with $\mathbf{U}_\mathrm{r} = 0$, the $\mathbf{r}_t$ gate would be activated and its values would go to 1. In that case, the WM buffer reference would be set to the value of $\mathbf{g}_t$. If it is taken into account that $\mathbf{g}_t$ is defined as

$$\mathbf{g}_t = \mathbf{U}_\mathrm{g}\mathbf{c}_{t-1} + \mathbf{W}_\mathrm{g}\mathbf{x}_t + \mathbf{b}_\mathrm{g}, \tag{14}$$

and the assumption of the free input regime with $\mathbf{U}_\mathrm{g} = 0$, the following applies:

$$\mathbf{g}_t = \mathbf{b}_\mathrm{g}, \tag{15}$$

that is,

$$\mathbf{e}_t = \mathbf{b}_\mathrm{g}. \tag{16}$$

This mechanism suggests that it is possible to define the beginning and the end of the event with a zero input situation. Namely, with the start of the free input regime, the event is to be stopped, and with the end of the free input regime, the event is to be started. In this way, information is created and eliminated from WM, that is, the buffer of WM is completely emptied when needed and set to the bias value of the variable $\mathbf{g}$ at the moment when the event ends. It is important to note that the application of the new architecture is not influenced by the sequence's number of events or by any specific class of problems. For example, sequences composed entirely of non-null values indicate the processing of only one event. Theoretically, in such cases, the new WM storage buffer will not be completely emptied through all time steps. Given the mentioned initialization Eq. (8), this leads to a difference compared to the Eq. (10), particularly in terms of information emphasis. In other words, the Eq. (10) prioritizes the retention of historical information, while the WM mechanism in the EB-JANET architecture, as described by Eq. (13), emphasizes the incorporation of new information.

Different bias values of the $\mathbf{g}$ variable were experimented with, and it turned out that the bias with the value of 1 gave slightly better results than the bias with the value set to 0.

What can now be seen is that LTM possesses the entire knowledge of processed information, while WM possess only a small amount of relevant information from the event that is goal-oriented. Consequently, the forget gate is influenced by information in WM, and the functioning of WM is influenced by the overall knowledge

from LTM. Due to modifications in the WM mechanism, it was decided to term that buffer the "event buffer" (symbolically represented by the letter e) and the new neural network architecture as the event buffering JANET (EB-JANET).

Assuming the context of the free input regime, the updating of the weights outside the event is not allowed. The partial derivative $\frac{\partial \mathbf{e}_t}{\partial \mathbf{e}_k} = 0$ if the timestep k is outside the event spanning time t.

The finalized overview of the architecture is given by the following equations

$$\begin{aligned}
\mathbf{g}_t &= \mathbf{U}_{\mathrm{g}}\mathbf{c}_{t-1} + \mathbf{W}_{\mathrm{g}}\mathbf{x}_t + \mathbf{b}_{\mathrm{g}},\\
\widetilde{\mathbf{c}}_t &= \tanh(\mathbf{g}_t),\\
\mathbf{f}_t &= \sigma(\mathbf{U}_{\mathrm{f}}\mathbf{e}_{t-1} + \mathbf{W}_{\mathrm{f}}\mathbf{x}_t + \mathbf{b}_{\mathrm{f}}),\\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + (1 - \mathbf{f}_t) \odot \widetilde{\mathbf{c}}_t,\\
\mathbf{r}_t &= \sigma(\mathbf{U}_{\mathrm{r}}\mathbf{c}_t + \mathbf{W}_{\mathrm{r}}\mathbf{x}_t + \mathbf{b}_{\mathrm{r}}),\\
\mathbf{e}_t &= \mathbf{r}_t \odot \mathbf{g}_t + (1 - \mathbf{r}_t) \odot \mathbf{e}_{t-1},\\
\mathbf{h}_t &= \mathbf{e}_t.
\end{aligned} \tag{17}$$

The linear layer is used to make predictions or extract information through a linear transformation applied to the hidden state $\mathbf{h}_t$, as follows:

$$\mathbf{y}_t = \mathbf{W}_{\mathrm{y}} \cdot \mathbf{h}_t + \mathbf{b}_{\mathrm{y}}. \tag{18}$$

In the aforementioned equations, $\mathbf{U}$ and $\mathbf{W}$ represent the weight matrices with their respective layer denotations, while $\mathbf{b}$ represents the bias vector with its corresponding layer denotation.

6. Experiments

In this section, the performance of the newly developed EB-JANET architecture is compared with several other RNN architectures, including the LSTM, LSTM-chrono, and JANET models. The LSTM-chrono model represents the LSTM model initialized using the chrono initialization technique Eq. (8). In the case of the JANET model, parameters $\boldsymbol{\beta} = 0$ and $\boldsymbol{\beta} = 1$ were used to control a certain amount of information that will be forgotten Eq. (5).

The goal was to assess and compare the convergence of the models towards the optimal minimum, while also evaluating the effectiveness of these models in capturing long-term dependencies as observed through performance metrics. It is important to note that a key aspect of these experiments involves the incorporation of a novel mechanism within the EB-JANET architecture, which is designed to enhance its learning capabilities and efficiency in processing sequential data.

The effectiveness of the RNN architectures in this research was evaluated through complex tasks. To ensure proper experimental conduct, a variety of datasets were utilized. This approach enabled a comprehensive evaluation of the performance of the models across various domains and challenges, contributing to an improved understanding of their capabilities and limitations. The results were derived from three independent experiments, with the outcomes being averaged. Experiments were sometimes repeated in response to oscillations. If oscillations were frequent,

an experiment was deemed unsuccessful. In certain experiments, changing the initial value of the event buffer from 0 to $\mathcal{U}[-1, 1]$ significantly enhanced the neural network's resilience to failures. Default initialization was zero unless otherwise specified. Furthermore, a consistent set of hyperparameters was maintained across all experiments to ensure fairness and facilitate meaningful comparisons. The final model was selected based on the best validation loss, with all reports generated at the epoch level.

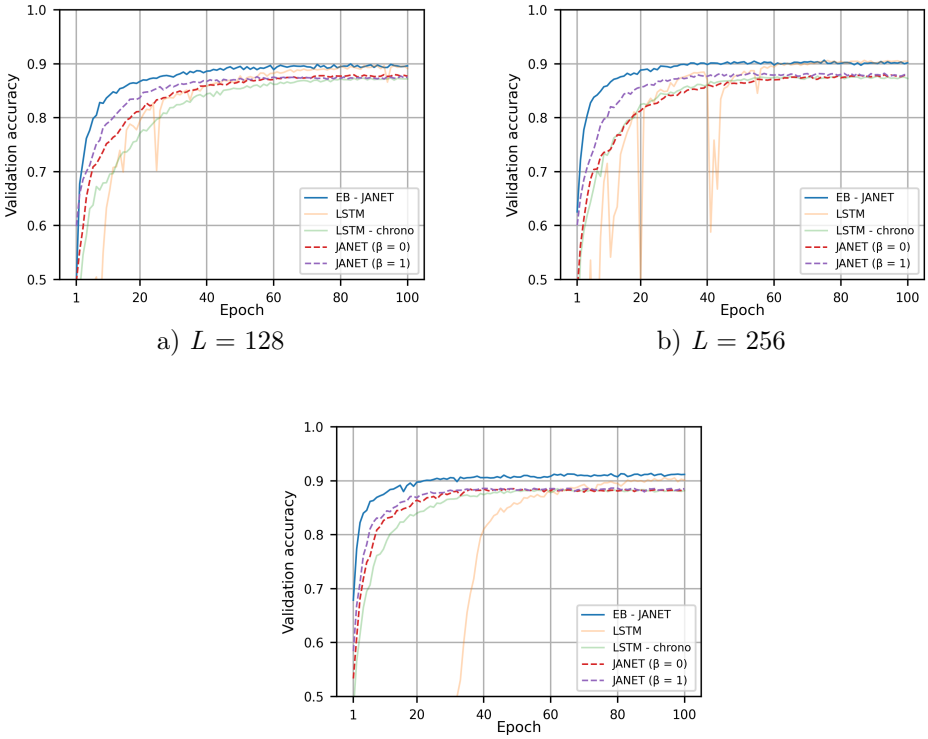
6.1 Experiments – real and publicly available datasets

Real and publicly available datasets serve as valuable benchmarks for testing models in real-world scenarios and optimizing them for various challenges. All models were trained by using the Adam optimizer with a learning rate of 0.001 and a mini-batch size of 200. The weight decay factor was set to 1e-5 and the dropout value of 0.1 was used on the output of the recurrent layers. The number of passes through the entire training dataset (epochs) was set to 100 or sometimes 200 (highlighted in the corresponding graphs), and the gradient norm was clipped at the value of 5. The number of hidden layer units (L) varied for each task (highlighted in the corresponding graphs).

6.1.1 Fashion MNIST classification

Fashion MNIST (fMNIST) [41] is a dataset of clothing images from Zalando, designed to replace the original MNIST dataset [34]. It is used for benchmarking machine learning algorithms, providing a more challenging classification task than the original MNIST. The dataset consists of 28×28 pixels grayscale images of clothes that are annotated with a label indicating the correct garment (dress, shirt, sneaker, etc.). Despite the dataset's nature, similar to the sequential MNIST task [38], fMNIST images are transformed into a sequential format of 784 individual pixels (raster of 28×28 pixels), presented one at a time to the neural network. Furthermore, the dataset is divided into a training set of 60,000 images and a test set of 10,000 images. For the validation set, 10% of the training data was used. The primary metric employed for evaluating the performance of the models was accuracy.

In this experiment, one-dimensional arrays of transformed fMNIST images frequently contain zero sequences. This implies the new model could efficiently analyze theoretical events, leveraging its enhanced memory mechanism when handling longterm dependencies in sequential data. As shown in Fig. 3, EB-JANET achieves higher accuracy on the validation set, indicating that it potentially generalizes more effectively to unseen data compared to other models. The high performance on the test set, as shown in Tab. I, confirms the model's ability to generalize well to unseen data. While the LSTM model sometimes shows competitive performance, its training variability can cause inconsistent outcomes. Another advantage of the EB-JANET model is its faster convergence to the optimal minimum loss compared to other models, evident from Fig. 3, underscoring its training efficiency.



a) $L = 128$ b) $L = 256$

c) $L = 512$

Fig. 3 *The mean accuracy values, calculated by assessing the models on the validation set during three separate training runs, utilizing varying numbers of hidden layer units (L).*

Model	$L = 128$	$L = 256$	$L = 512$
EB-JANET	**0.9013**	**0.9068**	**0.9147**
JANET ($\boldsymbol{\beta} = 0$)	0.8801	0.8799	0.8856
JANET ($\boldsymbol{\beta} = 1$)	0.8773	0.8833	0.8865
LSTM-chrono	0.8737	0.8778	0.8853
LSTM	0.8967	0.9065	0.9060

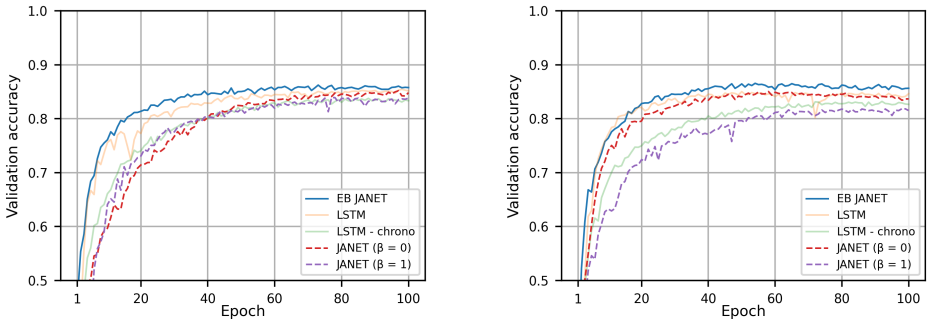
Tab. I *The mean accuracy [%] values, calculated by assessing the models on the test set after each of the three separate training runs, are listed in the table. The best accuracies obtained from the experiments are presented in bold.*

6.1.2 Permuted Fashion MNIST classification

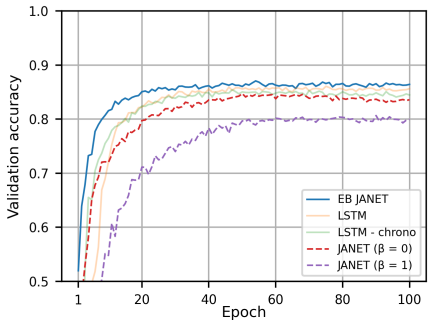
The permuted Fashion MNIST (pfMNIST) dataset is a transformation of the fMNIST dataset, as introduced in this paper. It represents a more complex variant

in which the pixels in every image from the fMNIST dataset have been permuted in the same order. The labels are kept. Furthermore, each image within the pfMNIST dataset is also converted into a sequence of 784 individual pixels (raster of 28×28 pixels). The dataset is divided into a training set of 60,000 images and a test set of 10,000 images. For the validation set, 10% of the training data was used. The primary metric employed for evaluating the performance of the models was accuracy.

The EB-JANET model outperforms the other models with a faster ascent to optimum accuracy on the validation set, as detailed in Fig. 4, and demonstrates enhanced generalization abilities on the test set, as shown in Tab. II. This experiment introduces a more challenging classification task by permuting the pixels in all images using the same order, unlike the previous one. This alteration introduces longer-range patterns and can lead to a dispersion of characteristic shapes within the input data. Notably, in this more intricate task, the LSTM model exhibited fewer oscillations compared to the previous experiment.



a) $L = 128$ b) $L = 256$



c) $L = 512$

Fig. 4 *The mean accuracy values, calculated by assessing the models on the validation set during three separate training runs, utilizing varying numbers of hidden layer units (L).*

Model	$L = 128$	$L = 256$	$L = 512$
EB-JANET	**0.8528**	**0.8539**	**0.8552**
JANET ($\boldsymbol{\beta} = 0$)	0.8447	0.8456	0.8461
JANET ($\boldsymbol{\beta} = 1$)	0.8287	0.8094	0.8003
LSTM-chrono	0.8317	0.8329	0.8464
LSTM	0.8445	0.8458	0.8511

Tab. II *The mean accuracy [%] values, calculated by assessing the models on the test set after each of the three separate training runs, are listed in the table. The best accuracies obtained from the experiments are presented in bold.*

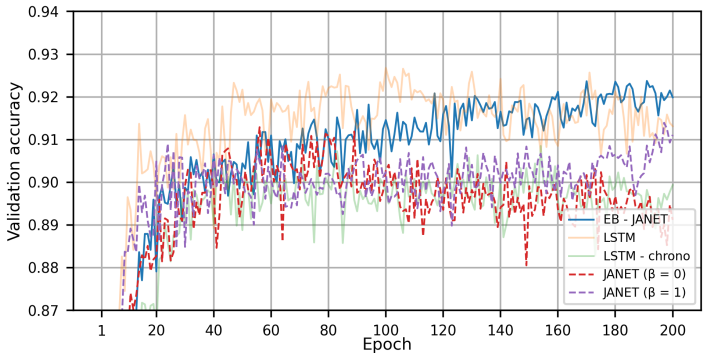
6.1.3 Human activity recognition

Human activity recognition (HAR) using smartphones [42] is a challenging task that involves classifying a person's activity based on sensor data influenced by their movements. The study involved participants between the ages of 19 to 48, who were instructed to perform six distinct activities while wearing a smartphone mounted on their waist. These activities included walking, walking upstairs, walking downstairs, sitting, standing, and laying down. The experiments were video recorded to manually label the data with the goal of classifying the activities into one of the six predefined categories. The dataset is partitioned, with 70% set aside for training, and the remaining portion reserved for testing. Additionally, as part of the training process, 10% of the training data was utilized for a validation set. The primary metric employed for evaluating the performance of these models was accuracy.

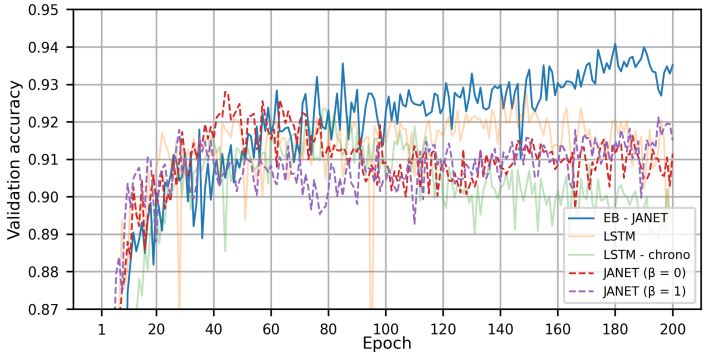
In this experiment, sequences in the dataset are comprised entirely of non-zero values, indicating the presence of a single event. The question concerns whether the memory components of the new architecture can handle long-term dependencies in sequential data more effectively than the other models, even in scenarios involving a single event. Consequently, EB-JANET exhibited enhanced anticipatory power compared to JANET for both $\boldsymbol{\beta} = 0$ and $\boldsymbol{\beta} = 1$, and also when compared to the LSTM-chrono model, as can be observed cumulatively in Fig. 5 and Tab. III. In experiments with hidden layers containing 128 units, the LSTM showed slightly better performance. However, as the number of units increased, EB-JANET displayed greater prediction accuracy on the test set, as shown in Tab. III.

Model	$L = 128$	$L = 256$	$L = 512$
EB-JANET	0.9237	**0.9408**	**0.9517**
JANET ($\boldsymbol{\beta} = 0$)	0.9129	0.9280	0.9282
JANET ($\boldsymbol{\beta} = 1$)	0.9149	0.9212	0.9223
LSTM-chrono	0.9085	0.9196	0.9217
LSTM	**0.9268**	0.9279	0.9252

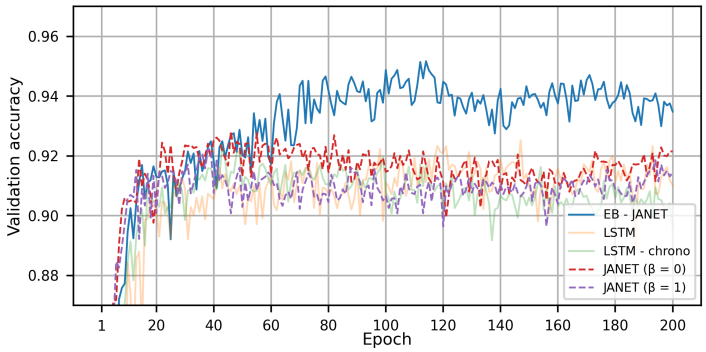
Tab. III *The mean accuracy [%] values, calculated by assessing the models on the test set after each of the three separate training runs, are listed in the table. The best accuracies obtained from the experiments are presented in bold.*



a) $L = 128$



b) $L = 256$



c) $L = 512$

Fig. 5 *The mean accuracy values, calculated by assessing the models on the validation set during three separate training runs, utilizing varying numbers of hidden layer units (L).*

6.1.4 Time series analysis

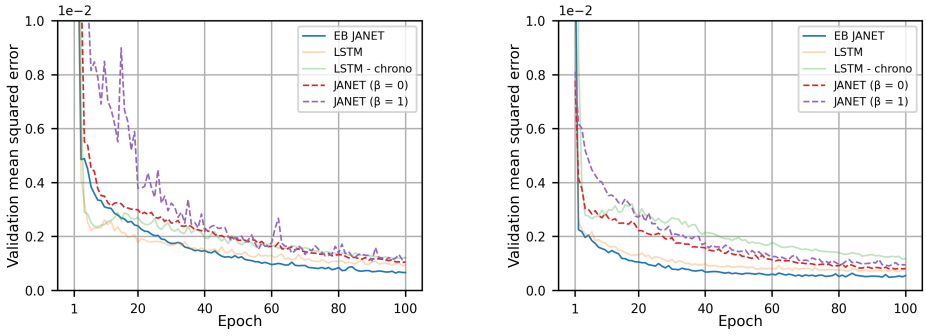
Time series analysis involves examining and extracting insights from a series of data points collected over a set period. In this experiment, two datasets were chosen to give a rough idea of how the models perform in predicting trends and patterns in similar data. Below are the abbreviations and their explanations:

- **NDX**: The NASDAQ-100 index consists of the top 100 non-financial companies globally, listed on the broader NASDAQ stock market, selected based on their market capitalization. This index encompasses a diverse range of companies spanning major industry sectors, including computer software, retail, and biotechnology.
- **HG = F**: Copper futures are standardized contracts traded on the New York Mercantile Exchange (NYMEX). As the third most widely used metal globally, following iron and aluminum, copper holds pivotal roles in industries like construction and industrial machinery manufacturing.

The datasets cover four years of information up to the paper's writing, updated daily, and used for predicting closing prices based on historical trends. The sequence lengths were standardized at 30, with a focus on predicting the next element in the sequence. The performance of the models was evaluated using mean squared error (MSE) as a primary metric. The models underwent testing with hidden layer units (L) set at 16 and 32, employing a mini-batch size of 32. This analysis provided valuable insights into the strengths, limitations, and forecasting applicability of the models.

Across both validation sets of this experiment, EB-JANET achieves a lower MSE and converges towards the optimal minimum faster than the other models, as shown in Figs. 6 and 7, suggesting better generalization to unseen data. Furthermore, the analysis of evaluation results on both test datasets in this experiment,

NDX (NASDAQ-100 index)



a) $L = 16$ b) $L = 32$

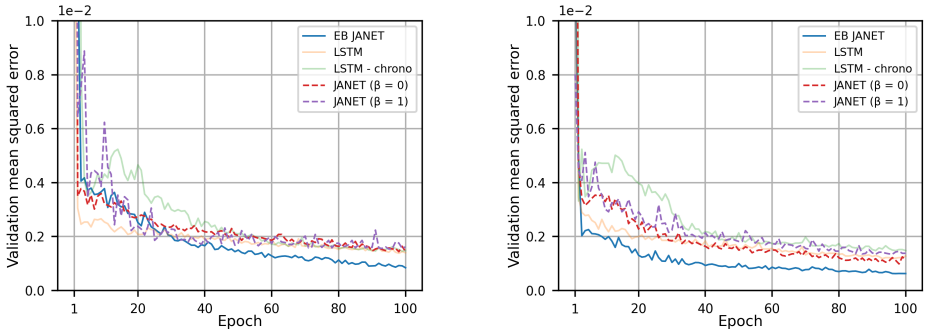
Fig. 6 *The mean MSE results obtained by evaluating the models on the validation set across three independent training runs, utilizing varying numbers of hidden layer units (L).*

as shown in Tabs. IV and V, highlights significant advantages favoring the EB-JANET model. Similar to the prior experiment, there are no assumptions about the number of events. This confirms that integrating the new WM mechanism improves anticipatory abilities, irrespective of the input data sequence construction, showcasing its versatility and robustness across varied situations.

Model	$L = 16$	$L = 32$
EB-JANET	**0.0004**	**0.0003**
JANET ($\boldsymbol{\beta} = 0$)	0.0009	0.0008
JANET ($\boldsymbol{\beta} = 1$)	0.0011	0.0009
LSTM-chrono	0.0011	0.0011
LSTM	0.0008	0.0006

Tab. IV *The mean MSE values, calculated by assessing the models on the test set after each of the three separate training runs, are listed in the table. The best MSE results results from the experiments are presented in bold.*

HG = F (copper futures)



a) $L = 16$ b) $L = 32$

Fig. 7 *The mean MSE results obtained by evaluating the models on the validation set across three independent training runs, utilizing varying numbers of hidden layer units (L).*

Model	$L = 16$	$L = 32$
EB-JANET	**0.00028**	**0.00019**
JANET ($\boldsymbol{\beta} = 0$)	0.00084	0.00065
JANET ($\boldsymbol{\beta} = 1$)	0.00093	0.00088
LSTM-chrono	0.00087	0.00081
LSTM	0.00089	0.00069

Tab. V *The mean MSE values, calculated by assessing the models on the test set after each of the three separate training runs, are listed in the table. The best MSE results results from the experiments are presented in bold.*

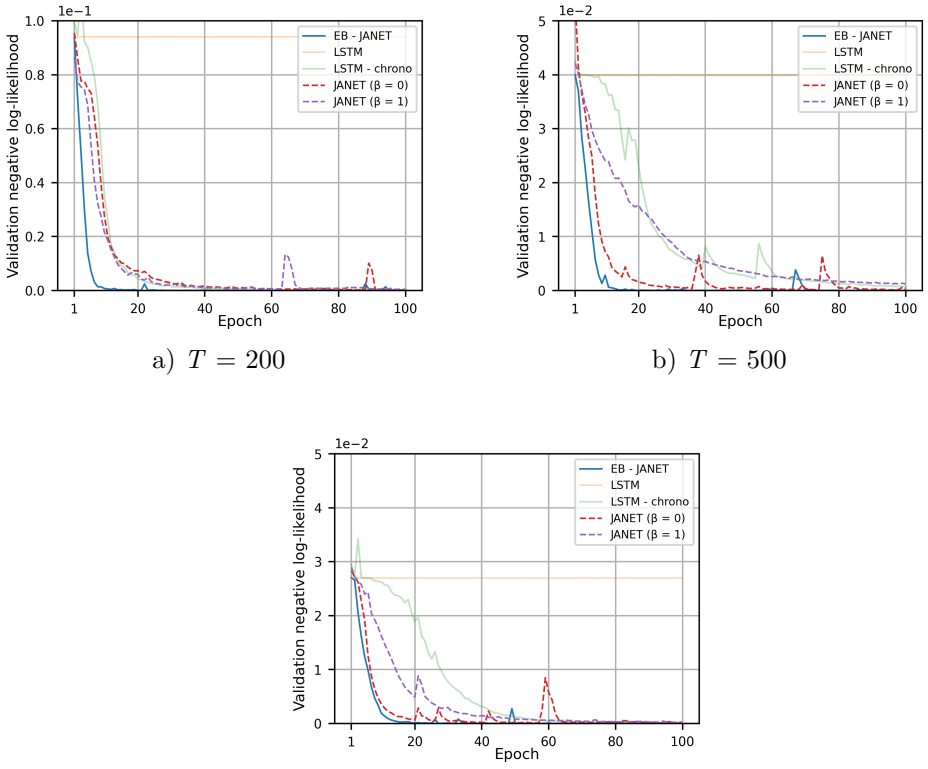
6.2 Experiments – synthetic datasets

Synthetic datasets enable the testing, evaluation, and optimization of models in controlled conditions with various variables, providing insights into how models perform in real-world situations and how to adapt them for different challenges and scenarios. In the subsequent experiments, the models underwent testing on synthetic tasks designed to expose them to various perspectives, with the goal of evaluating their performance in a wide range of scenarios and challenges. Diverse sequence lengths were employed to ensure a comprehensive evaluation of the models' adaptability and effectiveness. The Adam optimizer with a learning rate of 0.001 and a mini-batch size of 50 was utilized for model training, with the number of units in the hidden layer set to 128. The training set comprised 100,000 instances, while the validation set consisted of 10,000 instances, and the test set included 40,000 instances. The number of passes of the entire training dataset (epochs) was set to 100, with the gradient norm being clipped at a value of 5. Additionally, a dropout value of 0.1 was applied to the output of the recurrent layers.

6.2.1 Copying problem

The copying memory task, originally introduced in [3], represents a synthetic task that highlights how RNN-based models manage the LTM and the ability to recollect information seen in the distant past. This characteristic establishes the copy task as a robust benchmark for evaluating RNNs in proficiently addressing long-term dependencies within sequences [43]. The same setup as in [10] was followed, as briefly outlined here. $A = \{a_i\}_{i=1}^{K}$ is a set of K symbols and $S, T \in \mathbb{N}$ are arbitrarily chosen. The input embraces a $T + 2S$ length sequence of categories, where the first S entries need to be remember, all the while being sampled uniformly and independently and with replacement from $\{a_i\}_{i=1}^{K}$. The following $T - 1$ inputs are set to a_{K+1} and they represent a dummy or blank category. Furthermore, the following input a_{K+2} indicates that the network should predict the initial S entries of the input, and it can be considered a delimiter. The remaining S inputs are set to a_{K+1}. The expected output sequence consists of the $T + S$ repeated entries of a_{K+1}, followed by the first S categories of the input sequence kept exactly in the same order. The main objective is to minimize the average cross-entropy of the predictions at each timestep of the sequence, which boils down to memorizing a categorical sequence of the given length S for T timesteps. The most that a memoryless model can do in the mentioned task is make a random prediction in relation to the possible characters and imply the exact cross-entropy loss [6, 11], depending on a defined dataset considering the aforementioned (referred to as the baseline performance). The EB-JANET's event buffer was initialized as $\mathcal{U}[-1, 1]$. The primary metric employed for evaluating the performance of these models was the negative log-likelihood.

Upon examining Fig. 8, illustrating evaluations of the models on the validation dataset, it can be noted that EB-JANET performs better compared to the others, demonstrating faster convergence and reaching a more favorable optimum. This implies that the implemented enhancements improve the capacity to effectively address problems characterized by extensive sequences of steps. As for the evaluation



a) $T = 200$ b) $T = 500$

c) $T = 750$

Fig. 8 *The mean negative log-likelihood results obtained by evaluating the models on the validation set across three independent training runs, utilizing varying event lengths.*

on the test dataset presented in Tab. VI, EB-JANET achieves results differing by at least an order of magnitude compared to the other models. The conventionally initialized LSTM is inadequate in solving the copy memory problem [6, 10, 43].

Model	$T = 200$	$T = 500$	$T = 750$
EB-JANET	$\mathbf{10^{-6}}$	$\mathbf{10^{-6}}$	$\mathbf{10^{-6}}$
JANET ($\boldsymbol{\beta} = 0$)	10^{-4}	10^{-4}	10^{-4}
JANET ($\boldsymbol{\beta} = 1$)	10^{-4}	10^{-3}	10^{-4}
LSTM-chrono	10^{-5}	10^{-4}	10^{-5}
LSTM	10^{-2}	10^{-2}	10^{-2}

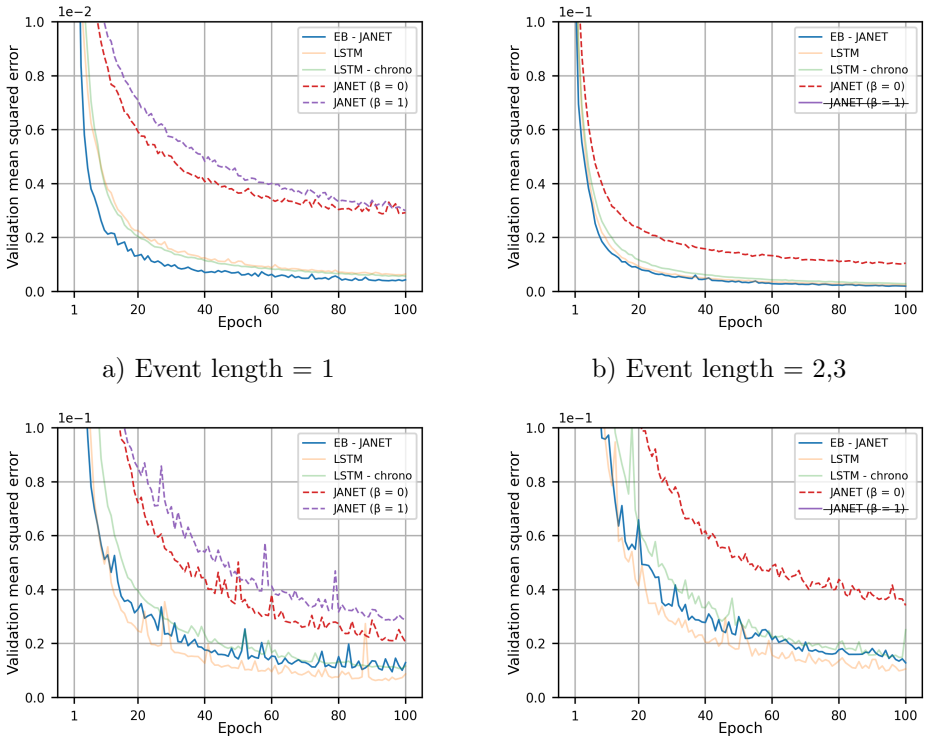
Tab. VI *The mean negative log-likelihood values, calculated by assessing the models on the test set after each of the three separate training runs, are listed in the table. The best MSE results results from the experiments are presented in bold. Due to the differences in the results, the tabular presentation has been simplified.*

6.2.2 Partial sums problem

The partial sums problem, introduced in this paper, involves remembering non-zero inputs in a sequence and calculating partial sums up to each non-zero element. Four experiments were conducted for each defined sequence length, using different event lengths. Sequences in the dataset, containing grouped non-zero values $x_i \in \mathbb{R}[-1, 1]$, consistently featured 10 events. Event lengths were limited to two distinct values (highlighted in the corresponding graphs). The objective was to assess the models' robustness and performance when working with zero-dense sequences and their resilience to failures, while considering variations in result quality based on event width and sequence length. The primary metric employed for evaluating the performance of the models was MSE. It should be noted that the models were unable to complete the last specified experiment in its entirety.

Through the conducted experiments, the EB-JANET and LSTM-chrono models were able to complete the largest number of experiments, where the EB-JANET model demonstrated the best performance in most cases, as can be visually observed in Fig. 9. The JANET model rarely finished experiments, more often with $\boldsymbol{\beta} = 0$, but generally struggled to capture the complexity of the data and frequently failed to finish experiments. When the sequence length was set to $T = 750$ or higher, and the event length was set to 6 or more, all models experienced catastrophic failures.

Sequence length $T = 200$



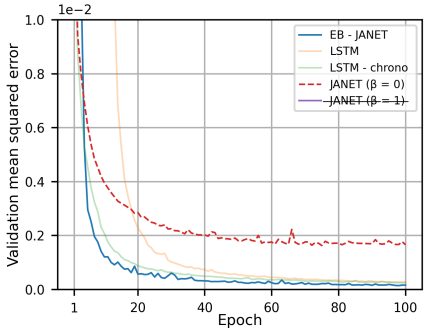
a) Event length = 1

b) Event length = 2,3

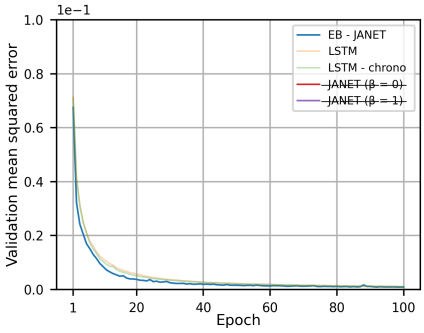
c) Event length = 6,7

d) Event length = 9,10

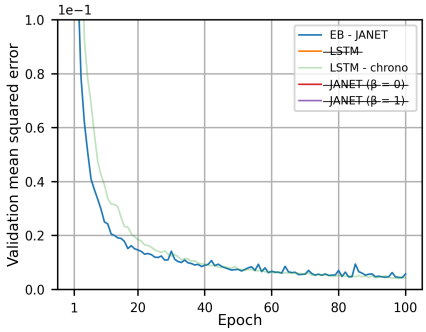
Sequence length $T = 500$



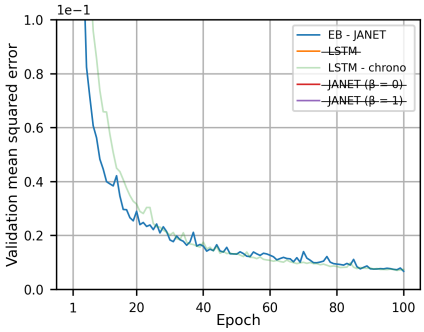
a) Event length = 1



b) Event length = 2,3

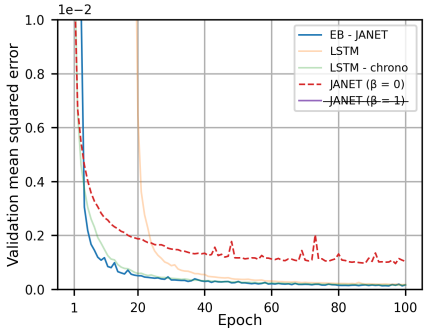


c) Event length = 6,7

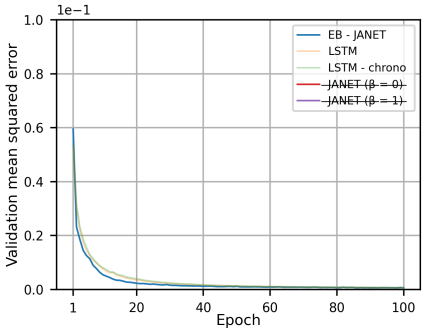


d) Event length = 9,10

Sequence length $T = 750$



a) Event length = 1



b) Event length = 2,3

Fig. 9 *The mean MSE results obtained by evaluating the models on the validation set across three independent training runs, utilizing varying event lengths.*

For the test dataset, the average and tabular presentation of results are provided for the models that successfully performed the task, as presented in Tab. VII. Displaying individual results has been omitted due to abundance.

Model	$T = 200$	$T = 500$	$T = 750$
EB-JANET	0.0061	**0.0029**	**0.0003**
JANET ($\boldsymbol{\beta} = 0$)	0.0169	–	–
JANET ($\boldsymbol{\beta} = 1$)	–	–	–
LSTM-chrono	0.0072	0.0031	0.0004
LSTM	**0.0047**	–	0.0004

Tab. VII *The mean MSE values, calculated by assessing the models on the test set after each of the three separate training runs, are listed in the table. Different event lengths were experimented with for each sequence length (T), resulting in the averaging of results. The best MSE results from the experiments are presented in bold. The absence of data in the table is related to models that did not successfully complete the experiment.*

7. Conclusion

This research delved into human memory theory with the aim of creating an artificial neural network that mirrors these principles. It combined established memory models and recent findings to bridge real-world and artificial methods. This paper introduced the event buffering JANET (EB-JANET) architecture, an advanced JANET model with a newly developed event buffer for improved handling of dynamic events in working memory.

A surveillance of the EB-JANET model was conducted through empirical testing on a variety of datasets. By comparing EB-JANET with JANET for both $\boldsymbol{\beta} = 0$ and $\boldsymbol{\beta} = 1$ values, it has been demonstrated that EB-JANET outperforms JANET in the majority of experiments. These experiments indicate the versatility of the EB-JANET model across various problem domains, in contrast to JANET, which is primarily suited for a limited range of scenarios.

The secondary findings of this research indicate that EB-JANET consistently outperforms both the LSTM and LSTM-chrono in capturing long-term dependencies and effectively utilizing sequential information across nearly all conducted experiments. While the LSTM remains widely used, the EB-JANET model, with fewer learnable parameters, has the potential to be a better alternative to the LSTM and LSTM-chrono models.

The study has presented EB-JANET, purposefully designed to be a relatively psychologically plausible model, outperforming its predecessor, the JANET model. Artificial intelligence has the potential to contribute to our understanding of the brain and cognition, assuming the reversibility of this process. Utilizing this architecture as a memory model concept within the field of psychology could make it easier to harmonize theory, practice, and artificial architectures. This synergistic approach with these three elements may better tackle questions about the human brain and memory, offering insights into unsolved mysteries of cognition.

In terms of the development environment used in this study, Python was employed as the primary programming language. The research extensively utilized the Keras functional API and TensorFlow within the Google Colab environment for advanced data processing and model development.

References

[1] SCHMIDHUBER J. Deep learning in neural networks: An overview. *Neural Networks*. 2015, 61, pp. 85–117, doi: 10.1016/j.neunet.2014.09.003.

[2] SHERSTINSKY A. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. *Physica D: Nonlinear Phenomena*. 2020, 404(8), pp. 1–40, doi: 10.1016/j.physd.2019.132306.

[3] HOCHREITER S., SCHMIDHUBER J. Long short-term memory. *Neural Computation*. 1997, 9(8), pp. 1735–1780, doi: 10.1162/neco.1997.9.8.1735.

[4] LIPTON Z.C., BERKOWITZ J., ELKAN C. A Critical Review of Recurrent Neural Networks for Sequence Learning, 2015. Available from: https://arxiv.org/abs/1506.00019.

[5] SRIVASTAVA R.K., GREFF K., KOUTNIK J., STEUNEBRINK B.R., SCHMIDHUBER J. LSTM: A search space odyssey. *IEEE transactions on neural networks and learning systems*. 2015, 28(10), pp. 2222–2232, doi: 10.1109/tnnls.2016.2582924.

[6] ARJOVSKY M., SHAH A., BENGIO Y. Unitary Evolution Recurrent Neural Networks. In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, 2016, pp. 1120–1128.

[7] JOZEFOWICZ R., ZAREMBA W., SUTSKEVER I. An empirical exploration of recurrent network architectures. In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, 2015, pp. 2342–2350.

[8] GERS F., SCHMIDHUBER J., CUMMINS F. Learning to forget: Continual prediction with LSTM. *Neural Computation*. 2000, 12, pp. 2451–2471, doi: 10.1049/cp:19991218.

[9] TALLEC C., OLLIVIER Y. Can recurrent neural networks warp time? In: *International Conference on Learning Representations*, 2018, pp. 1–13.

[10] VAN DER WESTHUIZEN J., LASENBY J. The unreasonable effectiveness of the forget gate, 2018. Available from: https://arxiv.org/abs/1804.04849.

[11] CHANDAR S., SANKAR C., VORONTSOV E., KAHOU S.E., BENGIO Y. Towards Non-saturating Recurrent Units for Modelling Long-term Dependencies. In: *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, 2019, pp. 3280–3287, doi: 10.1609/aaai.v33i01.33013280.

[12] CHIEN H.Y.S., TUREK J.S., BECKAGE N., VO V.A., HONEY C., WILLKE T.L. Slower is Better: Revisiting the Forgetting Mechanism in LSTM for Slower Information Decay, 2021. Available from: https://arxiv.org/abs/2105.05944.

[13] ATKINSON R.C., SHIFFRIN R.M. Human memory: A proposed system and its control processes. In: *Psychology of Learning and Motivation*, 1968, pp. 89–195, doi: 10.1016/S0079-7421(08)60422-3.

[14] CRAIK F.I.M., LOCKHART R.S. Levels of processing: A framework for memory research. *Journal of Verbal Learning and Verbal Behavior*. 1972, 11(6), pp. 671–684, doi: 10.1016/S0022-5371(72)80001-X.

[15] TULVING E. Organization of memory: Quo vadis? In: *The Cognitive Neurosciences*, 1995, pp. 839–847.

[16] DEWALL C.N., MYERS, D.G. Psychology (11th ed.). *Worth Publishers*. 2015.

[17] BADDELEY A.D., HITCH G.J. Working memory. In: *Psychology of Learning and Motivation*, 1974, pp. 47–89, doi: 10.1016/S0079-7421(08)60452-1.

[18] BADDELEY A. Working Memory: Theories, Models, and Controversies. *Annual review of psychology*. 2010, 63, pp. 1–29, doi: 10.1146/annurev-psych-120710-100422.

[19] WU J., LI J., PAN Z., LU J., ZHOU H., HU Y., WANG T., GAO Z. The Event Buffer: A New Storage Buffer of Working Memory. *Journal of Vision*. 2022, 22(14), pp. 3885, doi: 10.1167/jov.22.14.3885.

[20] YU S., REMPEL S., GHOLAMIPOURBAROGH N., BESTE C. A ventral stream-prefrontal cortex processing cascade enables working memory gating dynamics. *Communications Biology*. 2022, 5, pp. 1–11, doi: 10.1038/s42003-022-04048-7.

[21] UNGER K., ACKERMAN L., CHATHAM C.H., AMSO D., BADRE D. Working memory gating mechanisms explain developmental change in rule-guided behavior. *Cognition*. 2016, 155, pp. 8–22, doi: 10.1016/j.cognition.2016.05.020.

[22] BRADY T.F., KONKLE T., ALVAREZ G.A., OLIVA A. Visual long-term memory has a massive storage capacity for object details. *Proceedings of the National Academy of Sciences*. 2021, 105(38), pp. 14325–14329, doi: 10.1073/pnas.0803390105.

[23] BRADY T.F., KONKLE T., ALVAREZ G.A., OLIVA A. A review of visual memory capacity: beyond individual items and toward structured representations. *Journal of vision*. 2011, 11(5), pp. 4, doi: 10.1167/11.5.4.

[24] COWAN N. The magical number 4 in short-term memory: a reconsideration of mental storage capacity. *Behavioral and Brain Sciences*. 2001, 24, pp. 87–114, doi: 10.1017/s0140525x01003922.

[25] ERIKSSON J., VOGEL E.K., LANSNER A., BERGSTRÖM F., NYBERG L. Neurocognitive Architecture of Working Memory. *Neuron*. 2015, 88, pp. 33–46, doi: 10.1016/j.neuron.2015.09.020.

[26] RANGANATH C., BLUMENFELD R.S. Doubts about double dissociations between short- and long-term memory. *Trends in cognitive sciences*. 2005, 9(8), pp. 374–380, doi: 10.1016/j.tics.2005.06.009.

[27] DUDAI Y. The Neurobiology of Consolidations, Or, How Stable is the Engram? *Annual review of psychology*. 2004, 55, pp. 51–86, doi: 10.1146/annurev.psych.55.090902.142050.

[28] BIDERMAN N., LURIA R., TEODORESCU A.R., HAJAJ R., GOSHEN-GOTTSTEIN Y. Working Memory Has Better Fidelity Than Long-Term Memory: The Fidelity Constraint Is Not a General Property of Memory After All. *Psychological Science*. 2019, 30(2), pp. 223–237, doi: 10.1177/0956797618813538.

[29] BENGIO Y., SIMARD P., FRASCONI P., SCHMIDHUBER J. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*. 1994, 5(2), pp. 157–166, doi: 10.1109/72.279181.

[30] PASCANU R., MIKOLOV T., BENGIO Y. On the difficulty of training Recurrent Neural Networks. In: *Proceedings of the 30th International Conference on International Conference on Machine Learning*, 2013, pp. 1310–1318.

[31] HE K., ZHANG X., REN S., SUN J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In: *IEEE International Conference on Computer Vision*, 2015, pp. 1026–1034, doi: 10.1109/ICCV.2015.123.

[32] GLOROT X., BENGIO Y. Understanding the difficulty of training deep feedforward neural networks. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[33] MAHT S., VO V.A., TUREK J.S., HUTH A.G. Multi-timescale representation learning in LSTM Language Models, 2020. Available from: https://arxiv.org/abs/2009.12727.

[34] LECUN Y., CORTES C., BURGES C.J. The MNIST Database of Handwritten Digits. 1998. Available from: http://yann.lecun.com/exdb/mnist/.

[35] ROSALES K.P., SNIJDER J.P., CONWAY A.R., GONTHIER C. Working memory capacity and dual mechanisms of cognitive control: An experimental-correlational approach. *Quarterly Journal of Experimental Psychology*. 2022, 75(10), pp. 1793–1809, doi: 10.1177/17470218211066410.

[36] VERSCHOOREN S., KESSLER Y., EGNER T. Evidence for a single mechanism gating perceptual and long-term memory information into working memory. *Cognition*. 2021, 212, pp. 1–34, doi: 10.1016/j.cognition.2021.104668.

[37] ARTUSO C., PALLADINO P. Long-term memory effects on working memory updating development. *PLOS ONE*. 2019, 14(5), pp. 1–16, doi: 10.1371/journal.pone.0217697.

[38] LE QUOC V., NAVDEEP J., HINTON G.E. A Simple Way to Initialize Recurrent Networks of Rectified Linear Units, 2015. Available from: https://arxiv.org/abs/1504.00941.

[39] AN S., LEE M., PARK S., YANG H., SO J. An Ensemble of Simple Convolutional Neural Network Models for MNIST Digit Recognition, 2020. Available from: https://arxiv.org/abs/2008.10400.

[40] GOODFELLOW I.J., MIRZA M., XIAO D., COURVILLE A., BENGIO Y. An Empirical Investigation of Catastrophic Forgeting in Gradient-Based Neural Networks. In: *2nd International Conference on Learning Representations*, 2014, pp. 1–9.

[41] XIAO H., RASUL K., VOLLGRAF R. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms, 2017. Available from: https://arxiv.org/abs/1708.07747.

[42] ANGUITA D., GHIO A., ONETO L., PARRA X., REYES-ORTIZ L.J. A Public Domain Dataset for Real-Life Human Activity Recognition Using Smartphone Sensors. *Sensors*. 2020, 20(8), pp. 2200, doi: 10.3390/s20082200.

[43] JING L., GULCEHRE C., PEURIFOY J., SHEN Y. Gated Orthogonal Recurrent Units: On Learning to Forget. *Neural Computation*. 2019, 31(4), pp. 765–783, doi: 10.1162/neco_a_01174.