# A Bergson-Inspired Adaptive Time Constant for the Multiple Timescales Recurrent Neural Network Model

Thomas F. Burns (P)[1], Fabien C. Y. Benureau[1], (and) Jun Tani[1]

[1] Cognitive Neurorobotics Research Unit, Okinawa Institute of Science and Technology Graduate University
E-mail: t.f.burns@gmail.com

**Abstract—**We introduce an adaptive time constant for the multiple timescales recurrent neural network (MTRNN) inspired by the work of philosopher Henri Bergson (1859-1941). Observed variations in the recent activity of MTRNN neurons is used to adapt the time constant. We analyse how the time constant adapts in response to neuronal activity for a simple one-dimensional function-fitting task.

***Keywords*—Multiple Timescales Recurrent Neural Network, Time Constant, Henri Bergson, Adaptation, Duration**

### 1.1. Background on MTRNN

Introduced by Yamashita & Tani [1], the MTRNN consists of neurons whose firing rates are partially determined by their past firing rate and their current inputs, the balance between which is determined by a time constant, $\tau$. We describe the state of a MTRNN neuron $i$ at time $t+1$ as

$$u_{i,t+1} = \left(1 - \frac{1}{\tau}\right) u_{i,t} + \frac{1}{\tau}\left[\sum_{j\in N} w_{ij} x_{j,t}\right] \tag{1}$$

where $w_{ij}$ is the weight from neuron $i$ to presynaptic neuron $j$ and $x_{j,t}$ is the activity of neuron $j$ at time $t$.

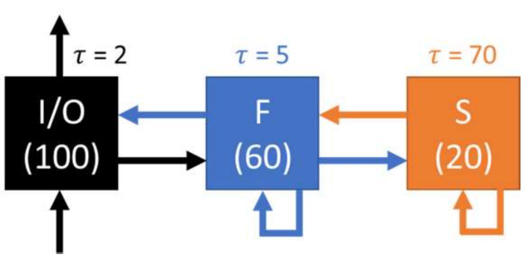


*Figure 1. Example MTRNN model with three layers, illustrating their selective all-to-all connection and flow of information.*

A MTRNN is organized in hierarchical layers. The bottom one receives the input, and the others communicate with the layers above and below. Figure 1 illustrates an example MTRNN model which has an input-output layer (I/O), a fast (F) layer, and a slow (S) layer, with time constants of 2, 5, and 70, and 100, 60, and 20 neurons, respectively. This is the MTRNN configuration introduced in [1].

### 1.2. Inspiration from Henri Bergson

Henri Bergson (1859-1941) was a philosopher famous for his conceptions of 'qualitative multiplicity' and 'duration'. He defines 'qualitative multiplicity' as the simultaneous and heterogeneous (yet continuous) conscious states which "are organized into a whole, permeate one another, [and] gradually gain a richer content" [2]. Bergson believed this organized whole of conscious states was what we subjectively experience as "the immediate data of consciousness" in some "duration" of experience.

Bergson further described his ideas using a model of memory and experience, illustrated in Figure 2. He describes bodily sensation (S) as our mode of direct interaction with our environment (P). 'Memories' stretch from the present (S) to the base of the inverted cone (AB), which are our 'pure memories'. Bergson states that consciousness is an experience of "unceasingly going backwards and forwards between the plane of action and that of pure memory." [3]

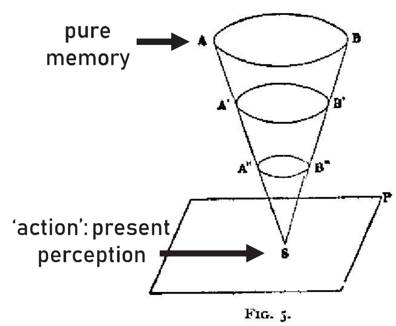


*Figure 2. Bergson's memory cone. From [3].*

We can interpret Bergson's ideas as positing that representation and memory is fundamentally a dynamic interaction between multiple simultaneous processes. Drawing inspiration from this view, we provide MTRNN layers (processes) with a means of temporal 'movement' by adapting its time constant according to the recent experiences of that layer.

### 2. Method of adapting $\tau$

Instead of setting the time constant, $\tau$, in Equation 1 to a fixed value for each layer, we let

$$\tau = (1 - v_{l,t})\, \mathrm{T}_l \quad (2)$$

where $v_{l,t}$ is a measure of variation (with respect to past activity) of the immediate neuronal activity in the layer $l$ at time $t$ and $\mathrm{T}_l$ is the upper bound of the time constant of that layer $l$. $v$ is a value between 0 and 1, where values approaching 1 represent a high degree of deviation from the mean in the current neuronal activity. Thus,

$$v_{i,t} = \frac{\min\left(\frac{|x_{l,t} - \mu_{l,H_l}|}{\sigma_{l,H_l} + \varepsilon}, \mathrm{Z}\right)}{\mathrm{Z}} \quad (3)$$

where $x_{l,t}$ is the sum of activations of neurons in layer $l$ at time $t$, and $\mu$ and $\sigma$ are the mean and standard deviation, respectively, of the sum of activations of neurons in layer $l$ for the $H_l$ last timesteps. $\varepsilon$ is added for numerical stability, and we minimize what is effectively a custom absolute z-score to a range of no more than Z.

## 3. Adaptation of $\tau$ in a function-fitting task

We modified a classical MTRNN with structure identical to Figure 1 and used $\mathrm{Z} = 3$, $\mathrm{T}_l = [2, 5, 70]$, and $\varepsilon = 10^{-5}$. We tested a range of one-dimensional functions which included sinusoidal, exponential, and sigmoidal components varying over time. We tested how $H$ adapted $\tau$ across layers 0 (I/O), 1 (F), and 2 (S) by setting $H_l = [\,2\mathrm{T}_l, \mathrm{T}_l, \frac{\mathrm{T}_l}{2}]$.

Figure 3 shows a representative example of $\tau$ adapting over timesteps during closed-loop generation. We can see sinusoidal activity present in this example, especially in the S layer. This shows that certain individual signal components can directly influence how $\tau$ adapts. However, in this example the sinusoidal activity has a period of approximately half of the original signal's. The range of $\tau$ for each layer shows that $v$ is rarely < 0.5, however when it was it generated the interesting phenomenon of layers 'changing order', i.e. a higher layer adopting a $\tau$ which is less than its lower layer(s). Typically, such 'order changes' did not last longer than a few timesteps.

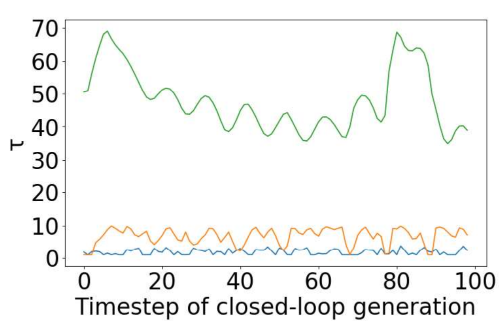


*Figure 3. Example of $\tau$ adapting over timesteps across three layers during closed-loop generation. (Green line=S layer, orange line=F layer, blue line=I/O layer.)*

Table 1 shows the mean (± S.D.) τs for closed-loop generation tested after each epoch in the training, comparing between the first and last 500 epochs. This allows us to observe how the network learnt weights which adapted the $\tau$ by influencing neuronal activity variation in certain layers. Three separate networks were tested by setting $H_l = [\,2\mathrm{T}_l, \mathrm{T}_l, \frac{\mathrm{T}_l}{2}]$.

$\tau$ becomes systematically larger as $H_l$ becomes large with respect to $\mathrm{T}_l$. We also see that the degree of $\tau$ variability is generally larger in the higher layers, where $\mathrm{T}_l$ is also higher and $\tau$ therefore has a greater range. After training, almost all higher layers had increased mean $\tau$s and, except for $H_l = \mathrm{T}_l/2$, layer 0 mean $\tau$s decreased.

*Table 1. $\tau$s across the first and last 500 epochs per layer (mean ± S.D.)*

| $H_l = 2\mathrm{T}_l$ | | | |
|---|---|---|---|
| Layer | 0 | 1 | 2 |
| First $\tau$s | 2.43 ± 0.99 | 7.01 ± 2.37 | 51.74 ± 15.07 |
| Last $\tau$s | 2.08 ± 0.97 | 7.26 ± 1.92 | 53.26 ± 14.47 |
| $H_l = \mathrm{T}_l$ | | | |
| Layer | 0 | 1 | 2 |
| First $\tau$s | 2.05 ± 1.01 | 6.48 ± 2.83 | 51.64 ± 15.81 |
| Last $\tau$s | 1.89 ± 0.85 | 6.72 ± 2.77 | 47.96 ± 16.55 |
| $H_l = \mathrm{T}_l/2$ | | | |
| Layer | 0 | 1 | 2 |
| First $\tau$s | 1.65 ± 0.92 | 5.22 ± 3.09 | 45.62 ± 18.54 |
| Last $\tau$s | 1.71 ± 0.96 | 5.38 ± 3.10 | 38.20 ± 19.97 |

All pairs of first and last $\tau$s were significantly different ($p<0.001$) from one another as tested by a Welch's t-test.

## 4. Discussion

Adapting $\tau$ based on recent layer activity in classical MTRNN models generally results in higher layers having higher $\tau$s and/or higher layers increasing their $\tau$s with training on a simple task, and vice-versa for lower layers. This reiterates the importance of hierarchical structuring in the MTRNN and shows that such a dynamical structure infers longer computational considerations at higher levels of processing.

Bergson's conceptions of qualitative multiplicity and duration are also well supported by these results, since our method of adapting $\tau$ (as tested in this task) generated phenomena similar to that described in Bergson's model of memory. One such example was that layers could 'change order' – higher layers' $\tau$s could dip below that of lower layers – suggesting that in some contexts higher-order processing can be sped up and act more quickly than lower-order processing when that higher-order processor experiences significant contextual changes.

## References


[1] Yamashita Y, Tani J (2008) Emergence of Functional Hierarchy in a Multiple Timescale Neural Network Model: A Humanoid Robot Experiment. PLOS Computational Biology 4(11):e1000220

[2] Bergson H (1889) Time and Free Will. p. 122

[3] Bergson H (1911) Matter and Memory. p. 210