Supplementary Algorithm S2 **Working of Back Propagation through time (BPPT)**

|  |
| --- |
| **Algorithm 2: Working of Back Propagation through time (BPPT)** |
| **Input:** A sequence of data points *x*, denoted as $x=\left[x\_{t-1, }x\_{t}, x\_{t+1}…\right]$ where $x\_{t}$represents the input at a specific time step "*t*" within the sequence.1. The hidden state at time step "*t*", denoted as $f\_{t}$, is calculated by combining the current input $x\_{t}$with the previous hidden state $f\_{t-1}$. This computation is expressed as

$$f\_{t}=σ\left(αx\_{t}+θf\_{t-1}\right)$$where $σ$, α and θ have the usual meanings.1. The loss function, such as Mean Squared Error (MSE) or cross-entropy loss, is applied to calculate the loss at each time step. We can represent the loss at a specific time step *‘t’* as $l\_{t}$.

The gradients of the loss with respect to the parameters of the network is calculated from the final time step. Subsequently, these gradients are propagated in a backward direction across previous time steps. The gradient at time step *‘t’*, which signifies the impact of the loss on the hidden state at *‘t’* can be represented as:$$\frac{∂l\_{t}}{∂f\_{t}}$$1. The gradients are accumulated at each step as they propagate through time ensuring that dependencies between different time steps are captured effectively. Accumulated gradients are represented as

$$\frac{∂l}{∂f\_{t}}$$1. The parameters of the network are updated by utilizing optimization algorithms such as Adam, SGD and RMSprop. The parameter update for each weight can be computed as:

$$∆w=-η\*\sum\_{}^{}\frac{∂l}{∂w}$$where $∆w$ is the update in weight and $η$is the learning rate.  |