RT Modelling with RNNs (and more)

Tim Lin
yl0124@princeton.edu

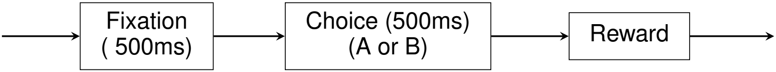
September 12, 2025

PRINCETON Neuroscience INSTITUTE

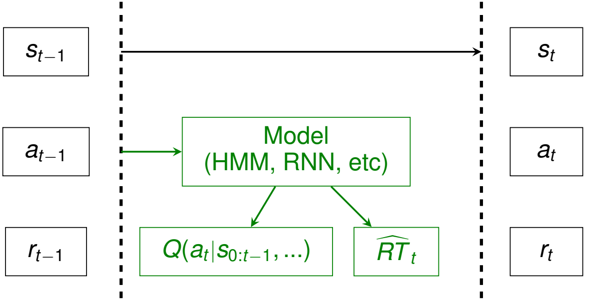
NYU

Task to model

Behavioral Task



Modelling Abstraction



Outline

Task to model

Modelling Approaches

- RNN Models
- Finite-state Automata

Modelling Approaches

Making Choice and RT predictions

- RTified RNN
- RT RNN (Generalized RTified RNN)

Output: $\texttt{choice\_pred} \in [0, 1]^{T\times 2}$, $\texttt{rt\_pred} \in \mathbb{R}_+^T$

Modelling the entire decision-making trajectory

- Finite-state Automata
- Time-discretized RNN

Output: $\texttt{action\_pred} \in [0, 1]^{T\times 3}$ (hold, left, right)

RTified RNN (factorized)

Model:

$$\begin{aligned}
\mathbf{h}_0^{\text{slow}} &\sim \mathcal{N}(0, I) \\
\mathbf{h}_{1:\text{trials}}^{\text{slow}} &= \text{RNN}^{\text{slow}}(\mathbf{h}_{0:\text{trials}-1}, \mathbf{u}_{1:\text{trials}}) \\
\hat{\mathbf{x}}_i^{\text{choice}} &= \text{MLP}^{\text{choice}}(\mathbf{h}_i^{slow}) \\
\mathbf{h}_{0,i}^{\text{fast}} &= \text{MLP}^{\text{init}}(\mathbf{h}_i^{\text{slow}}) \\
\mathbf{h}_{1:S,i}^{\text{fast}} &= \text{RNN}^{\text{fast}}(\mathbf{h}_{0:S-1,i}^{\text{fast}}, \mathbf{u}_{1:S,i}) \\
\hat{\mathbf{x}}_i^{\text{RT}} &= \tau(\Phi(\mathbf{h}_{1:S,i}^{\text{fat}}), \theta)
\end{aligned}$$

Loss:

$$\mathcal{L} = \left\langle \sum_{i=1}^{\text{trials}} \text{CE}(\mathbf{x}_i^{\text{choice}}, \hat{\mathbf{x}}_i^{choice}) + \lambda_{\text{RT}} \text{MSE}(\mathbf{x}_i^{\text{RT}}, \hat{\mathbf{x}}_i^{RT}) \right\rangle_{\text{batch}}$$

RTified RNN (shared)

T : trials $\times$ S.

Model:

$$
\begin{aligned}
\mathbf{h}_0 &\sim \mathcal{N}(0, I) \\
\mathbf{h}_{1:T} &= \mathrm{RNN}(\mathbf{h}_{0:T-1}, \mathbf{u}_{1:T}) \\
\hat{\mathbf{x}}_i^{\mathrm{choice}} &= \mathrm{MLP}^{\mathrm{choice}}(\mathbf{h}_{Si+1:Si+S}) \\
\hat{\mathbf{x}}_i^{\mathrm{RT}} &= \tau(\Phi(\mathbf{h}_{Si+1:Si+S}), \theta)
\end{aligned}
$$

Loss:

$$
\mathcal{L} = \left\langle \sum_{i=1}^{\mathrm{trials}} \mathrm{CE}(\mathbf{x}_i^{\mathrm{choice}}, \hat{\mathbf{x}}_i^{choice}) + \lambda_{\mathrm{RT}} \mathrm{MSE}(\mathbf{x}_i^{\mathrm{RT}}, \hat{\mathbf{x}}_i^{RT}) \right\rangle_{\mathrm{batch}}
$$

RT RNN (shared)

Model:

$$\begin{aligned} \mathbf{h}_0 &\sim \mathcal{N}(0, I) \\ \mathbf{h}_{1:T} &= \mathrm{RNN}(\mathbf{h}_{0:T-1}, \mathbf{u}_{1:T}) \\ \hat{\mathbf{x}}_i^{\text{choice}} &= \mathrm{MLP}^{\text{choice}}(\mathbf{h}_{Si:Si+S}) \\ \hat{\mathbf{x}}_i^{\text{RT}} &= \mathrm{MLP}^{\text{RT}}(\mathbf{h}_{Si:Si+S}) \end{aligned}$$

Loss:

$$\mathcal{L} = \left\langle \sum_{i=1}^{\text{trials}} \mathrm{CE}(\mathbf{x}_i^{\text{choice}}, \hat{\mathbf{x}}_i^{choice}) + \lambda_{\text{RT}} \mathrm{MSE}(\mathbf{x}_i^{\text{RT}}, \hat{\mathbf{x}}_i^{RT}) \right\rangle_{\text{batch}}$$

Time-discretized RNN

Instead of turning "discretized-trajectories" into RT and choice predictions, use RT to reconstruct the entire trial trajectory.

Ask the model to reconstruct the entire trial trajectory $\mathbf{x}_{1:T}$ given control inputs (that the subject received) $\mathbf{u}_{1:T}$, where T is the total number of discretized time steps across all trials.

- Separation of different sub-trial events; dissect phase-planes and fixed-point structures.
- Direct interpretability of the model's internal dynamics. Amenable to joint behavioral-neural data modelling.
- Extra model complexity needed (empirically show this and comment on parameter retrieval)[a][b]

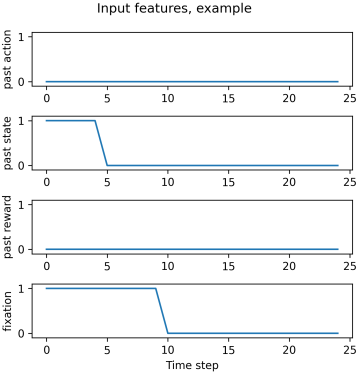
---

[a] we don't necessarily care so much about the evidence accumulation in classical cognitive model settings?

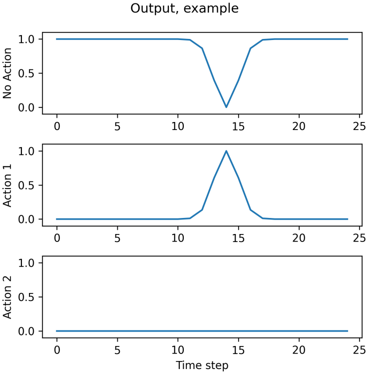
[b] also RNNs, where inputs come in additively, are not necessarily the most appropriate models here.

Time-discretization Example

Example **u** within a trial:



Example **x** within a trial (gaussian-smoothed):



Time-discretized RNN

Model:

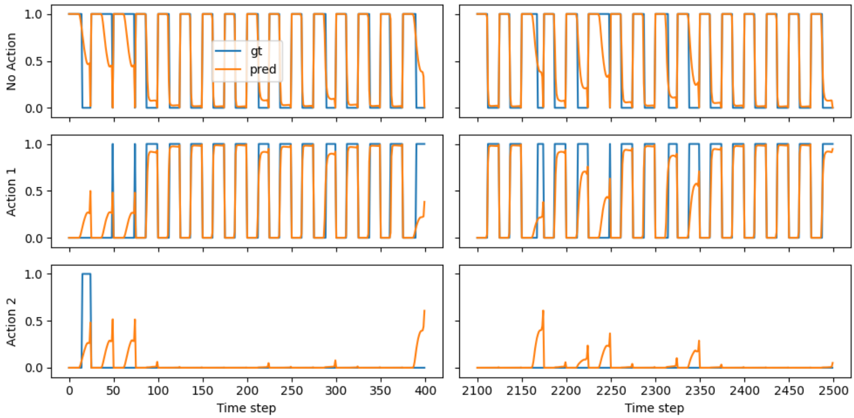
$$\mathbf{h}_0 \sim \mathcal{N}(0, I)$$
$$\mathbf{h}_{1:T} = \mathrm{RNN}(\mathbf{h}_{0:T-1}, \mathbf{u}_{1:T})$$
$$\mathbf{x}_{1:T} = \mathrm{MLP}(\mathbf{h}_{1:T})$$

Loss:

$$\mathcal{L} = \left\langle \sum_{t=1}^{T} \mathrm{CE}(\mathbf{x}_t, \mathbf{x}_t) \right\rangle_{\text{batch}}$$

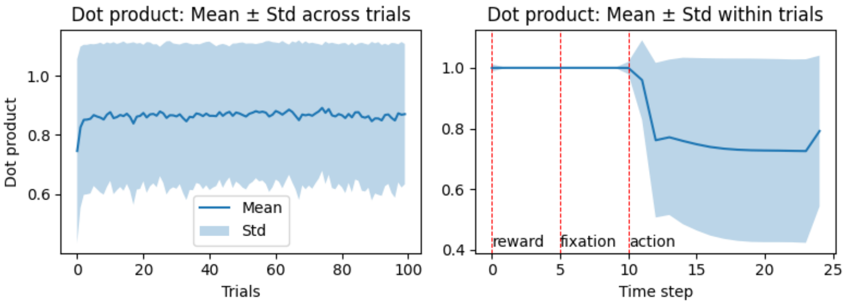
Time-discretization RNN Trained Result

Example $\mathbf{x}$ vs. $\hat{\mathbf{x}}$ (taken from one trial sequence)

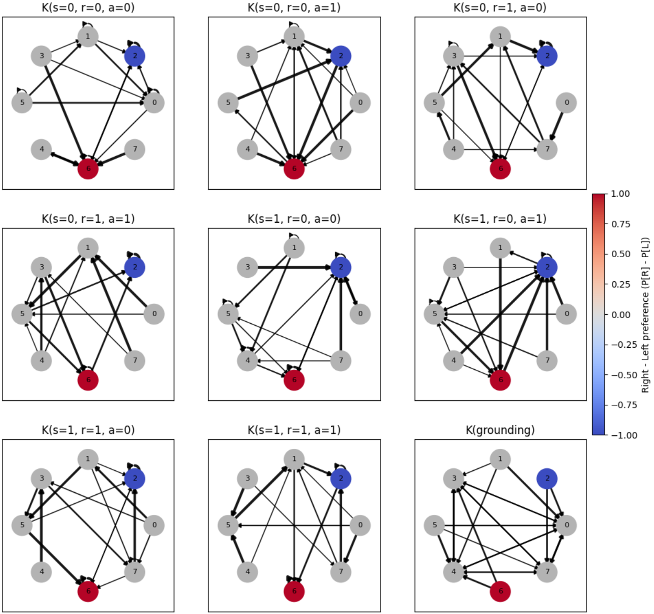


Time-discretization RNN Trained Result 2

Example $\langle \mathbf{x}, \hat{\mathbf{x}} \rangle$, averaged across vs. within trials.



Finite-state Automata



SFSA Model Appendix (credits to Prakhar)

The sequential decision setting, e.g. n-armed Bernoulli bandit. The (time-dependent) decision function / policy based on stochastic finite state automaton defined as follows:
Let there be given some environment with:

- S a finite set of environment states;
- A a finite set of actions that an agent can take, with $\bar{A} = A \cup \{*\}$, where '$*$' represents the empty action;
- T a probabilistic transition $S \times \bar{A} \to S$, such that for all $a \in \bar{A}$, $T_a \in^{S \times S}$ is a probability matrix.

*Bandit Example. The environment can be an iterated n-armed Bernoulli bandit. Here, the environment states would be the outcomes of pulling an arm, e.g. $S = 0, 1$, and the actions would be the n arms. The bandit environment provides for any arm $a \in \bar{A}$ chosen, a probability p_a, such that the follow-up state is 1 with probability p_a and to 0 with probability $1 - p_a$. If the empty action $* \in \bar{A}$ is chosen, then the state changes to 0 with probability 1, i.e., $p_* = 0$. In the bandit case, the current state has not effect on the transition probabilities.

Let $(\Omega, Q, \{P_o\}_{o\in\Omega})$ be a probabilistic automaton, where

- Ω is a partition of $\bar{A} \times S$ denoting the set of inputs for the automaton; e.g., in an n-armed bandit setting this might be a set of tuples representing an action taken and the *resulting* environment state; in a more general setting, one might require that Ω is a partition of $S \times \bar{A} \times S$; for notational simplicity, we assume Ω to be the finest partition and simplify to $\Omega = \bar{A} \times S$;
- Q is a finite set of states;
- for $\omega \in \Omega$, P_ω is a right-stochastic matrix such that for any $q, q' \in S$, $[P_\omega]_{qq'}$ is the probability for a transition from state q to q' under input ω, i.e. for $q \xrightarrow{\omega} q'$.

Let further

- $\pi : Q \to D(\bar{A})$ be the probabilistic action readout, where $D(\bar{A})$ is the set of probability distributions over $\bar{A}$, yielding for any given automaton state either an actual action or the empty action;
- $R : \Omega \to$ be the agent's/automaton's reward function.

*Conditional Transitions. The resulting stochastic process then looks as follows

$\ldots \to a_{t-1} \to s_t \to q_t \to a_t \to s_{t+1} \to q_{t+1} \to a_{t+1} \to s_{t+2} \ldots$

and the conditional transitions look as follows:

$$
\begin{aligned}
\Pr(s_{t+1} = s' | s_t, q_t) &= \sum_{a \in \bar{A}} \Pr(a_t = a | q_t) \Pr(s_{t+1} = s' | s_t, a) \\
&= \sum_{a \in \bar{A}} \pi_{q_t}(a) T((s_t, a) \to s') = \sum_{a \in \bar{A}} \pi_{q_t}(a) [T_a]_{s_t s'} \\
\Pr(q_{t+1} = q' | s_t, q_t) &= \sum_{a \in \bar{A}} \Pr(a_t = a | q_t) \sum_{s' \in S} \Pr(s_{t+1} = s' | (s_t, a)) [P_{a,s'}]_{q_t q'} \\
&= \sum_{s' \in S} \sum_{a \in \bar{A}} \pi_{q_t}(a) T((s_t, a) \to s') [P_{a,s'}]_{q_t q'} \\
&= \sum_{s' \in S} \sum_{a \in \bar{A}} \pi_{q_t}(a) [T_a]_{s_t s'} [P_{a,s'}]_{q_t q'}
\end{aligned}
$$

Conditional Transitions (contd.)

$$\begin{aligned}
\Pr(s_{t+1} = s', q_{t+1} =& q' | s_t = s, q_t = q) \\
&= \sum_{a \in \bar{A}} \pi_q(a) T((s,a) \to s') [P_{a,s'}]_{qq'} \\
&= \sum_{a \in \bar{A}} \pi_q(a) [T_a]_{ss'} [P_{a,s'}]_{qq'} =: [M]_{(s,q),(s',q')} \\
\Pr(a_{t+1} = a' | s_t, q_t) &= \sum_{q' \in Q} \Pr(q_t \to q' | s_t) \Pr(a_{t+1} = a' | q') \\
&= \sum_{q' \in Q} \Pr(q_t \to q' | s_t) \pi_{q'}(a')
\end{aligned}$$

Stationarity and Evolution of Probabilities

Overall, due to the stationary of the transitions, we have for all $h, t \in_0$:

$$\Pr(s_{h+t} = s', q_{h+t} = q' | s_h = s, q_h = q) = [M^t]_{(s,q),(s',q')}$$
$$= \Pr(s_t = s', q_t = q' | s_0 = s, q_0 = q) \qquad (*)$$

*Evolution of Probabilities. Given a probability distribution ρ_t over states Q, and a probability distribution σ_t over environment states S at some time t, we have:

$$\sigma_{t+1}(s') = \Pr(s_{t+1} = s' | \sigma_t, \rho_t) = \sum_{s \in S} \sum_{q \in Q} \sigma_t(s) \rho_t(q) \Pr(s \to s' | q)$$
$$\rho_{t+1}(q') = \Pr(q_{t+1} = q' | \sigma_t, \rho_t) = \sum_{s \in S} \sum_{q \in Q} \sigma_t(s) \rho_t(q) \Pr(q \to q' | s)$$
$$\text{for } s' \in S, q' \in Q.$$

*Sequence of Rewards. From this, we have the sequence of rewards $\{R_t\}_{t \in}$ with $R_t = R(a_{t-1}, s_t)$, with suitable choice for $t = 0$, e.g., $R_t = 0$.