Memory Traces, Integration, and Explanatory Sufficiency

Hutter RNN Project

2026-01-31

1 Setup: Adding Time to the Universal Model

The CMP framework defines $u = (E, T, P, f, \omega)$ for static models. To handle sequences, we introduce **time**.

1.1 Definitions

Let $x_1, x_2, \ldots, x_t$ be a sequence of observations (bytes).

Memory Trace The complete history up to time t:

$$M_t = (x_1, x_2, \ldots, x_t)$$

Sufficient Statistic A compression $h_t = \phi(M_t)$ such that:

$$P(x_{t+1}|M_t) = P(x_{t+1}|h_t)$$

If h_t is sufficient, it captures all predictive information in M_t.

Integration The update rule that computes h_t from h_{t-1} and x_t:

$$h_t = g(h_{t-1}, x_t)$$

This is the RNN recurrence.

Explanatory Sufficiency A statistic h_t is **explanatorily sufficient** if:

$$I(x_{t+1}; M_t|h_t) = 0$$

That is, h_t screens off the future from the past.

2 Memory Traces in Log-Space (N)

Working in $\mathbb{N}$ (log-space), the memory trace becomes a thought vector:

$$\mathbf{t}^{(M_t)} \in \mathbb{N}^{|E|}$$

Each component $t_e^{(M_t)}$ is the log-support for event e given history M_t.

2.1 Integration as Pattern Application

Given patterns P and update function f_P, integration is:

$$\mathbf{t}^{(t)} = f_P(\mathbf{t}^{(t-1)}, x_t)$$

In CMP terms:

$$t_j^{(t)} = \max_i \min(t_i^{(t-1)}, p_{ij}) + \delta_{j,x_t} \cdot \text{obs\_strength}$$

The δ term adds support for the observed event.

3 Factored Memory Traces

For an RNN with hidden state $\mathbf{h} \in \mathbb{R}^H$, the memory trace factors:

$$M_t \approx h_t = (h_t^{(1)}, h_t^{(2)}, \ldots, h_t^{(H)})$$

Each hidden unit $h_t^{(i)}$ is a **partial trace** capturing some aspect of history.

3.1 ES as Coarse Traces

Our ES features are coarse memory traces:

$$ES_t = \pi(x_t) \in \{0, 1, 2, 3, 4\}$$

The ES-level memory is:

$$M_t^{ES} = (ES_1, ES_2, \ldots, ES_t)$$

This is a lossy compression of M_t.

4 Explanatory Sufficiency

4.1 Definition

A representation h_t has **explanatory sufficiency** for predicting x_{t+1} if:

$$H(x_{t+1}|h_t) = H(x_{t+1}|M_t)$$

Equivalently, the mutual information between future and past, given h_t, is zero:

$$I(x_{t+1}; M_t|h_t) = 0$$

4.2 Levels of Sufficiency

1. **Full sufficiency:** $h_t = M_t$ (store everything)
2. **Markov sufficiency:** $h_t = x_t$ (only last observation)
3. **k-Markov sufficiency:** $h_t = (x_{t-k+1}, \ldots, x_t)$
4. **ES-Markov sufficiency:** $h_t = ES_t$
5. **RNN sufficiency:** $h_t = \mathbf{h}_t$ (learned compression)

4.3 Measuring Sufficiency Gap

The **sufficiency gap** of representation h_t is:

$$\Delta_h = H(x_{t+1}|h_t) - H(x_{t+1}|M_t)$$

For ES-Markov:

$$\Delta_{ES} = H(x_{t+1}|ES_t) - H(x_{t+1}|x_t) = ?$$

5 Empirical Questions

1. What is the sufficiency gap for 1-Markov (single byte)?
2. What is the sufficiency gap for ES-Markov?
3. How much does the RNN hidden state reduce the gap?
4. At what depth k does k-Markov become sufficient?

6 Connection to Compression

Explanatory sufficiency relates directly to compression:

$$\text{Optimal code length} = H(x_{t+1}|M_t)$$

If we use h_t instead of M_t:

$$\text{Achievable code length} = H(x_{t+1}|h_t) \geq H(x_{t+1}|M_t)$$

The gap Δ_h is the **compression penalty** for using the coarse representation.