Pattern Chain Analysis of a Saturated RNN via the Isomorphic Universal Model

Claude MJC

2026-02-06

Abstract

We analyze a saturated RNN (0.079 bpc on 1024 bytes) by reading its weights as Universal Model patterns under the doubled-E isomorphism. Tracing *pattern chains* through hidden activations reveals a flat depth distribution (depths 0–8 equally represented), the signature of overtraining. A direct n-gram UM with no hidden layer matches the RNN at order 11 (0.081 bpc) and surpasses it at higher orders (0.010 bpc at order 50). Together these results show that (1) the RNN fills all available depths uniformly with dataset patterns, (2) only patterns up to word length generalize, and (3) the UM standard learning function recovers the same statistics in one pass that backprop requires 4,000 epochs to learn.

1 Background

The doubled-E isomorphism maps an Elman RNN exactly onto a Universal Model: each hidden neuron h_j becomes a binary event space $\{h_j, \bar{h}_j\}$, weight magnitude becomes pattern strength ($2|w|$), and tanh falls out of softmax over each binary ES. This has been validated at 0.00% bpc difference on full enwik9 (see `rnn-um-mapping.pdf`, archive 20260131).

A pattern with strength k means $\approx 2^k$ co-occurrences of the linked events. On our 1024-byte dataset, the maximum possible strength is $\log_2(1024) = 10$.

Under doubled-E, only positive activations (h_j^+) propagate as events. Negative activations ($\bar{h}_j$) absorb negative weights. We trace only positive events through positive patterns.

2 Pattern Chains Through the Hidden Layer

A *pattern chain* connects an input event to an output event through hidden activations:

$$\text{input}(x_t) \xrightarrow{W_{xh}} h_j^+ \xrightarrow{W_{hh}} h_k^+ \xrightarrow{W_{hy}} \text{output}(y)$$

Chain strength $= \min_i(2|w_i|)$, following the UM product pattern rule. A chain of depth d encodes a $(d+2)$-gram.

We ran the saturated model forward on all 1024 bytes and traced chains backwards from each correct prediction (`sat_chains.c`).

Ten hub neurons (h15, h8, h112, h90, h52, h50, h68, h17, h61, h16) account for the majority of chain routing. Seven positions have surprisal > 3 bits with no strong chains, all involving rare transitions (`/"`, `/X`, `"-`, etc.).

Depth	Chains	Fraction
0	2,777	13.9%
1	2,777	13.9%
2	2,633	13.2%
3	2,318	11.6%
4	2,071	10.4%
5	1,954	9.8%
6	1,834	9.2%
7	1,867	9.4%
8	1,704	8.5%
Total	19,935	(avg 19.5/step)

Table 1: Chain depth distribution. Nearly flat across all available depths, indicating the RNN uses its full recurrence capacity uniformly—the signature of overtraining.

3 Direct n-gram UM (No Hidden Layer)

The hidden layer is an *embedding* of n-gram patterns, not the source of them. We can find the same patterns directly by counting n-grams in the dataset—the UM standard learning function operating on the byte-level event space (`sat_ngram_um.c`).

Order	BPC	Bits saved	Note
1	4.739	3,336	unigram
2	2.046	2,793	bigram
3	0.560	1,497	trigram
4	0.308	261	
5	0.263	53	
6	0.207	53	
7	0.148	49	
8	0.138	19	
9	0.115	5	
10	0.110	28	
11	0.081	4	$\approx$ RNN (0.079)
20	0.066		
30	0.052		
40	0.024		
50	0.010		

Table 2: n-gram UM performance by order. The UM matches the RNN at order 11 and surpasses it at higher orders. "Bits saved" counts total bits saved across all 1023 predictions when this depth first improves a prediction over shorter contexts.

4 The Overtraining Explanation

Tables 1 and 2 together explain overtraining:

1. The RNN has an effective depth of ∼11 (limited by gradient vanishing and BPTT truncation). It fills all available depths uniformly with patterns from the training data (Table 1).
2. Depths 0–2 (unigram through trigram) account for 93% of the total bits saved (7,626 of 8,174 bits). These patterns generalize to any English text.
3. Depths 3–10 add lexical patterns (word-level). These mostly generalize because the lexicon repeats across Wikipedia.
4. Depths 11+ encode dataset-specific long-range patterns that do not generalize. The n-gram UM continues improving past order 11 (down to 0.010 bpc at order 50), but the RNN cannot access these depths.

The RNN's 0.079 bpc residual is not a failure to learn—it has learned everything accessible within its depth limit. The remaining 0.079 bpc comes from patterns at depths 12–50 that require longer context than the RNN can carry through its hidden recurrence.

5 Implications

1. **One-pass learning.** The order-11 n-gram UM achieves 0.081 bpc in a single pass over the data, matching 4,000 epochs of backprop. This validates the CMP paper's claim that event counting recovers the same statistics as gradient descent.
2. **The hidden layer is a bottleneck.** With order 50, the UM reaches 0.010 bpc—the architecture (128 hidden neurons, tanh, BPTT) limits the RNN, not the learning algorithm.
3. **Patterns exist independently of the embedding.** The UM finds patterns first; the RNN's hidden layer is one possible embedding. The projection onto 128 neurons can be solved as a separate linear algebra problem.
4. **Overtraining is depth-uniform pattern filling.** The flat chain distribution (Table 1) is the mechanism. A model that could selectively learn only depths 0–10 would generalize; the RNN has no such selectivity under gradient descent.

6 Open Questions

1. **Reverse isomorphism.** Compute n-gram strengths from event counts, embed into 128 hidden neurons via SVD, write as RNN weights. Does this reproduce 0.079 bpc without any backprop?
2. **Generalization test.** Train on bytes 0–1023, evaluate on bytes 1024–2047. The n-gram UM with order cutoff at 10 should generalize better than the full-depth RNN, confirming the overtraining diagnosis.
3. **Scaling.** At what dataset size does the RNN's depth limit become the binding constraint rather than overtraining?
4. **Strength calibration.** Chain strengths (avg 0.92) are much lower than $\log_2$ of actual n-gram counts due to the bottleneck effect (min over links). How much information is lost in the hidden embedding vs. the direct n-gram representation?

7 Reproducibility

Repository. `https://github.com/inimino/hutter` (commit: `23a9427`).

Model. Elman RNN, 256 → 128 → 256, tanh activation. Trained with SGD, cosine LR decay (0.01 → 0), gradient clipping at 5, BPTT length 50, 4,000 epochs. Checkpoint: `sat_model.bin` (329 KB), included in this archive and downloadable from `https://cmpr.ai/hutter/archive/20260206/sat_model.bin`.

Data. First 1024 bytes of enwik9 (`http://mattmahoney.net/dc/enwik9.zip`).

To reproduce:

```
git clone https://github.com/inimino/hutter.git
cd hutter
git checkout 23a9427
bash reproduce.sh
```

Or manually:

```
gcc -O3 -o sat_chains sat_chains.c -lm
gcc -O3 -o sat_ngram_um sat_ngram_um.c -lm
head -c 1024 enwik9 > enwik_1024.txt
wget https://cmpr.ai/hutter/archive/20260206/sat_model.bin
./sat_chains enwik_1024.txt sat_model.bin
./sat_ngram_um enwik_1024.txt 50
```

Source files. `sat_train.c` (training), `sat_chains.c` (chain analysis), `sat_ngram_um.c` (n-gram UM), `reproduce.sh` (all-in-one).