Saturation Experiment: RNN vs UM on 1024 Bytes

Claude

2026-02-06

Abstract

We train an Elman RNN (128 hidden units, 82K parameters) to convergence on the first 1024 bytes of enwik9 and compare with the Universal Model learning function (joint event counting). The RNN reaches 0.079 bpc after 4000 epochs with cosine learning rate decay, near-memorizing the sequence. A UM bigram model achieves 2.05 bpc in a single pass; a trigram model achieves 0.55 bpc. We analyze the relationship between RNN weights and UM pattern strengths derived from joint event counts, finding moderate correlations ($r = 0.21$–0.46) under the doubled-E discretization.

1 Setup

Data. First 1024 bytes of enwik9 (Wikipedia XML). Maximum pattern strength from this data: $\log_2(1024) = 10$.

RNN. Standard Elman architecture: 256 input (one-hot byte) $\rightarrow$ 128 hidden (tanh) $\rightarrow$ 256 output (softmax). Total parameters: 82,304. Trained with SGD, truncated BPTT (length 50), gradient clipping at 5.

Training. Initial learning rate 0.01 with cosine annealing over 4000 epochs. Each epoch: one full pass over 1024 bytes ($\sim$20 BPTT steps). Best model saved by training loss.

UM. The standard learning function counts joint events. For bigrams: a 256×256 contingency table from consecutive byte pairs. For trigrams: conditioned on the previous two bytes.

2 Training Results

The RNN reaches the UM bigram baseline (2.05 bpc) at approximately epoch 350, after which recurrent connections (W_{hh}) provide sequence context beyond bigrams. By epoch 2000 the model is near-memorized; cosine decay stabilizes the final convergence.

Without learning rate decay, the model oscillates around 0.05 bpc and diverges (observed in preliminary runs with constant lr = 0.01).

3 UM Baselines

The trigram model captures most of the structure. The gap from 0.55 to 0.079 bpc represents longer-range patterns that the RNN encodes via its recurrent state.

Epoch	Train BPC	Eval BPC	LR
1	7.872	7.724	0.0100
10	5.004	5.011	0.0100
100	4.377	4.392	0.0100
200	3.204	3.229	0.0099
300	2.522	2.592	0.0099
400	1.994	2.137	0.0098
500	1.594	1.764	0.0096
1000	0.696	0.684	0.0085
2000	0.256	0.196	0.0050
3000	0.119	0.110	0.0015
4000	0.096	0.081	0.0001

Table 1: RNN training on 1024 bytes. Best eval: 0.079 bpc at convergence.

Model	BPC
Uniform (8 bits)	8.000
Unigram (byte frequencies)	$\sim$4.5
UM bigram (1 pass)	2.047
UM trigram (1 pass, with backoff)	0.555
RNN (4000 epochs)	0.079

Table 2: Comparison of models on 1024 bytes. The UM models use the standard learning function (joint event counting). All evaluate on the training data.

4 Weight Analysis

4.1 Weight Statistics

Layer	Min	Max	RMS
W_{xh} (input$\rightarrow$hidden)	-1.988	1.787	0.080
W_{hh} (hidden$\rightarrow$hidden)	-1.266	1.393	0.133
W_{hy} (hidden$\rightarrow$output)	-4.082	3.315	0.140

Table 3: Weight ranges and RMS of the saturated model.

4.2 UM Pattern Strengths

Under the isomorphism (strength $= 2|w|$), the saturated model contains:

- W_{xh}: 205 patterns with strength > 1 (max: 3.98)
- W_{hh}: 207 patterns with strength > 1 (max: 2.79)
- W_{hy}: 523 patterns with strength > 1 (max: 8.16)

The maximum observed strength is 8.16 out of a theoretical maximum of 10 (since $\log_2(1024) = 10$). The output layer has the strongest patterns, consistent with the model learning sharp predictions for specific contexts.

4.3 Hidden State Events

All 128 hidden neurons are "mixed" (active positive 10–90% of the time). None are locked to one sign. This contrasts with the full enwik9 model where some neurons specialize (e.g., word boundary detector h2).

5 RNN Weights vs UM Event Counts

The core question: do the RNN weights encode the same statistics as UM joint event counting?

We run the converged RNN forward on all 1024 bytes, recording hidden states at every timestep. Using the doubled-E discretization ($h_j > 0 \Rightarrow$ event h_j, $h_j < 0 \Rightarrow$ event $\bar{h}_j$), we count joint events across all three layers and compute correlations with the corresponding RNN weights.

Layer	Comparison	Correlation
W_{xh}	weight vs log-odds($h_i > 0 \mid$ input$= j$)	0.206
W_{hy}	weight vs log-odds($h_j > 0 \mid$ output$= o$)	0.421
W_{hh}	weight vs log-odds-ratio($h_i > 0 \mid h_j$ sign)	0.456

Table 4: Correlation between RNN weights and UM event count statistics under doubled-E discretization. All correlations are positive, indicating the same direction of encoding.

The correlations are moderate, not high. This is expected: the doubled-E discretization discards activation magnitude, and the RNN's tanh nonlinearity means the relationship between weights and event statistics is nonlinear.

The ordering $W_{hh} > W_{hy} > W_{xh}$ suggests the recurrent layer most closely matches the event-counting statistics, while the input layer's relationship is obscured by the interaction between input encoding and recurrent state.

6 Discussion

6.1 What the RNN Learned

On 1024 bytes, the RNN near-memorizes the sequence (0.079 bpc $\approx$ 5.4% of full entropy). It does this by:

1. Learning byte transition statistics (matching UM bigram by epoch $\sim$350)
2. Learning trigram-level context via recurrence (dropping below 0.55 bpc by epoch $\sim$1200)
3. Learning longer patterns specific to the 1024-byte sequence (0.55 $\rightarrow$ 0.079 bpc)

6.2 Why Correlations Are Moderate

The doubled-E discretization (sign only) is the simplest mapping from continuous hidden states to discrete UM events. Better mappings would:

- Use multiple thresholds per neuron (quantized activation levels)
- Account for the nonlinear tanh relationship
- Consider the bias terms (which shift the threshold)

The moderate correlations confirm the weights encode event co-occurrence statistics, but the exact quantitative relationship requires the full isomorphism machinery (as established in the doubled-E exact match on full enwik9).

6.3 Comparison with Full enwik9 Model

On 1024 bytes, the model memorizes; on enwik9, it generalizes. The saturation experiment shows the endpoint of training: all learnable statistics from the data are captured in the weights. The UM learning function reaches this point in one pass (for each pattern order), while the RNN requires $\sim$3000 epochs of gradient descent.

6.4 Next Steps

1. **Higher-order UM comparison**: Count n-gram events for $n = 4, 5, \ldots$ until the UM matches the RNN's 0.079 bpc. This determines the effective memory depth of the trained RNN.
2. **Better discretization**: Use the full doubled-E isomorphism instead of sign-only to achieve higher correlations.
3. **Direct weight derivation**: Compute UM pattern strengths from event counts, convert to RNN weights via the inverse isomorphism, and compare directly with the trained weights.

7 Reproduction

Source: `sat_train.c`, `sat_analyze.c` in the `hutter/` repository.

```
gcc -O3 -o sat_train sat_train.c -lm
gcc -O3 -o sat_analyze sat_analyze.c -lm
head -c 1024 enwik9 > enwik_1024.txt
./sat_train enwik_1024.txt models/sat 4000 500
./sat_analyze enwik_1024.txt models/sat_best.bin
```