Recurrent Networks: Language Models, Gates, and Translation

Lecture Notes Consolidated with Claude AI — CS6140 Machine Learning

---

Instructor's lecture plan for the note of the same title (`RNN_claude_1017.pdf`). Section numbers (§) refer to that note. Two lectures, 100 minutes each, mostly on the board.

Calendar. Lecture A, Thu 11/19 (two days after HW6 opens); Lecture B, Mon 11/23; Thanksgiving takes Thu 11/26; HW6 due Tue 12/1, so everything HW6 needs is taught eight days before the deadline. Lecture A must let students *start* both HW6 Problem 3 (language model) and Problem 4 (translator): after it they can do P3(A) and P4(A–B). Lecture B covers the gates, decoding and BLEU (P3(B–C), P4(C–D), P5).

Lecture A — Thu 11/19: from a hidden state to a translator

Time	On the board	Worked example / demo
0–8	**Put the sentence up first and leave it there:** `tom gave lily his red ball , and she kept it in her bag .` — four pronouns, four *different* referents, 3/6/5/10 tokens back. Then why other ideas don't work: the data are not IID; the chain rule; Plan A (Markov, τ steps) vs Plan B (a hidden state = the context). Draw the unrolled RNN (Figure 2) and leave it up. (§1)	—
8–22	The cell $H_t = \tanh(X_t W_{xh} + H_{t-1} W_{hh} + b_h)$ with shapes (IDs (n, T) in, time-major core); "3 matrices, independent of t"; `nn.RNN`'s transposed weights and two biases. **Then the question they will actually ask** — what *is* in H_t? Walk the cast list of §2.1 down the running sentence: after `tom`, one character; after `lily`, two; `red` adds nothing worth keeping; at `she`, look one up *and keep the others.* (§2.1)	The hand step: net $(0.35, -0.05)$, $H_t = (0.3364, -0.0500)$.
22–34	Tokens → IDs → embeddings → targets shifted by one, on one TinyStories sentence. Why whole words are fine here; one sentence on why real LLMs use subwords (Week 13). (§3)	`tom | gave | lily | his | red | ball | , ...`; 5,274 words cover 99.6%.
34–44	Language model = classifier over the vocabulary at every position; perplexity = exp(mean cross-entropy) = effective number of choices. The ladder. (§4)	uniform 5,274 → unigram 303.5 → RNN ≈24.
44–56	First show the measured plain-RNN park-prompt table: what did it actually retain? Then BPTT = a product of per-step factors (NN_BASICS again, over time). Exploding vs vanishing; clipping, and where it goes (between `backward` and `step`). (§5)	Plain RNN: widths 128/512, 1,000/5,000 updates; $0.9^{50} = 0.005$, $1.1^{50} = 117$; $(3, 4) \to (0.6, 0.8)$.
56–70	Encoder–decoder: two RNNs; what `nn.GRU` returns (outputs (T, n, H), final state (L, n, H), one per layer); the context $\mathbf{c}$ = top layer's final state, fed at every step; "the decoder is the LM conditioned on a source". (§7)	Draw Figure 8 (it is on the running sentence: *tom le dio a lily su pelota*); the shapes strip.
70–92	**Protected block.** Training vs generation, side by side (Figure 10); teacher forcing (one call; the recurrence still steps inside); padding and the masked loss; the whole-target = one-word-at-a-time check (same state, same supplied tokens, `eval()` mode). (§8)	The 2×4 padded batch: 0.75 vs 1.31.
92–100	HW6 P3(A), P4(A–B): what to start on tonight. (§10)	—

Lecture B — Mon 11/23: gates, decoding, BLEU

Time	On the board	Worked example / demo
0–10	Recap: the product of factors; why learning to use words far back can be difficult. (§5)	Figure 6: the straight falling line.
10–40	Gates as learned switches; then the separate GRU park-prompt table with the same recipe. Equal width is not equal parameter count, and one seed is not a general ranking. The GRU in the board's words (reset, update; the candidate is a proposal); the LSTM memory cell, forget/input/output; the additive path through F_t (Figure 7). Parameter cost 3× / 4×. (§6)	$0.99^{50} = 0.61$ vs $0.9^{50} = 0.005$.
40–50	How to read a gradient-vs-distance curve (straight line vs levelling off, on a log scale); what to expect in the perplexity table. Generating text: temperature sampling. (§5, §2.1, §6)	Temperature on logits $(2, 1, 0)$; demo (5 min): stories at $\tau = 0.5$ and 1.0.
50–60	Greedy decoding: the loop, `<eos>`, the finished mask; beam search in one picture (pointer). (§9)	—
60–80	BLEU from n-grams: clipped precision (max over references), brevity penalty, corpus vs sentence level; variants exist. (§9)	*the cat sat on mat*: 0.579; *the the the cat*: 3, not 4.
80–90	BLEU by length; the bottleneck as a hypothesis HW7 tests. (§9)	Demo Segment 10 (greedy translations); a 3-word vs a 10-word sentence: how many 4-grams each has.
90–100	HW6 map and common bugs; "HW7 needs your own full-run results"; the bridge to HW7 (subwords, attention, GPT on the same data). (§10)	—

If Lecture A runs long, shorten 44–56 to the two factor examples and where clipping goes (the full BPTT discussion moves into Lecture B's recap). Do *not* move the protected block: P4(A–B) needs teacher forcing and the masked loss, and students should be able to start P4 right after Lecture A.

The story, in six beats

1. **Order matters, and the past grows.** Every model so far treated rows as independent. Sequences are the opposite. The chain rule says "predict each step from everything before", and "everything before" keeps growing — that is the whole problem. *Do this on the sentence, not in the abstract:* `she` at position 9 needs `lily` at position 3, past a closer noun that suggests *it*; `her` at 13 reaches ten tokens back. A fixed window of k words is beaten by any sentence slightly longer than k.

2. **Carry a summary.** Instead of the whole past, keep a fixed-size vector and update it with each step, with the same weights every time. That is the RNN, and it is just a dense layer that also reads its own last output. *The question they will ask next is what is actually in that vector*, and the cast list is the answer worth giving: a little register of who is in play, updated at every step, looked up when a pronoun arrives. Say plainly that it is a description of the task, not of the trained net — then beat 4 measures the difference.

3. **Language modelling makes it concrete.** Predict the next word of a story; measure the effective number of choices. Going from 303 (no memory) to about 24 (an RNN) is the evidence that memory helps.

4. **Test the memory, then explain the learning difficulty.** First use the measured plain-RNN table in §5: does changing Lily to Tom change the next-word odds? BPTT multiplies per-step factors, so gradients can shrink or explode; clipping addresses explosions. Keep the behavior probe distinct from a gradient measurement. After introducing gates, compare the separate GRU table: the 128-unit runs show little separation, while the 512-unit run scores +3.40. These runs do not establish an architectural limit or prove that more training could never help.

5. **Gates keep a lane open.** The LSTM's cell is updated by a learned forget factor and an addition. Where the factor stays near 1, memory and gradient can survive many steps. The GRU filler probe drops from +3.50 at prediction position 7 to +0.11 at position 47; that is evidence about this model's output, not a direct measurement of individual gates. On the running sentence it prefers `he` to the supplied continuation `she`. Both can be grammatical: this does not prove that the recipient is absent from the hidden state. Attention will provide a direct route back to earlier positions, rather than relying only on a summary.

6. **Two RNNs translate.** One reads, one writes; the writer is the language model with the reader's summary as an extra input. Training feeds the truth; generation feeds its own guesses. BLEU falls on long sentences, and the single summary vector is the suspect that Week 13 investigates.

   The one slide to end on, if there is time. The Spanish for the running sentence contains `su` twice — *tom le dio a lily* ***su*** *pelota roja , y ella la guardó en* ***su*** *bolso.* Spanish does not mark whose: the first must come out *his*, the second *her*, and nothing in the word says which. With one fixed context that had to be decided before any English was produced. Next week the decoder simply looks back — at `tom` for the first and `lily` for the second — and the attention map shows it doing so. That is the whole bridge to Week 13, in one word of Spanish.