# Attention: Queries, Keys, and Values

Lecture Notes Consolidated with Claude AI — CS6140 Machine Learning

---

*Instructor's lecture plan for the note of the same title (`ATTENTION_claude_1017.pdf`). Section numbers (§) refer to that note.*

Two sessions: **Lecture A** (Thu 12/3, 100 minutes) unlocks HW7 Problems 2(A–C) and 3; **Lecture B** is the first half (about 50 minutes) of Mon 12/7, unlocking 2(D) and Problem 6; its second half starts the TRANSFORMER note. The TOKENS lecture (Mon 11/30) comes before; TRANSFORMER Part B follows in the extra lecture (or on Thu 12/10). Keep the HW6 translator picture (Figure 1(a)) and the soft $k$-NN formula on a side board for both sessions.

## Lecture A — Thu 12/3: attention as a soft lookup (100 minutes)

| Time | On the board | Worked example / demo |
|---|---|---|
| 0–10 | The HW6 bottleneck: one vector for the whole sentence; BLEU by length. "A translator looks back." Draw Figure 1(a), then (b). (§1) | HW6 reference run: 37.0 (5–6 tokens) → 18.6 (9–12). |
| 10–28 | **Flashback** (the 8/6 p5 board): soft $k$-NN as a weighted average of one-hot labels; Nadaraya–Watson; "$\alpha$ high ⇒ pay attention to $x_i$"; the hard vs soft dictionary lookup. Rename: query, keys, values. Parts table. (§2, Table 1) | The spam example $0.951, 0.908, 0.994 \to 0.65$; keys $(1, 2, 4)$, values $(10, 20, 40)$, $z = 2.5 \to 22.12$. Figure 2. |
| 28–34 | Scores as log-kernels ($k = e^a$); state the result of expanding $-\frac{1}{2}\|q - k\|^2$ (only $q^\top k$ matters when keys have equal length) and leave the expansion to the note; dot product vs additive score. (§3) | — |
| 34–54 | $\text{softmax}(QK^\top/\sqrt{d})V$ with the shape table; the 2-token example by hand; why $\sqrt{d}$ (variance $d$; saturation; $\text{diag}(w) - ww^\top$). (§4) | $A = [[0.6225, 0.3775], [0.3775, 0.6225]]$. Demo (3 min): Figure 4. |
| 54–64 | Masks: $-\infty$ before the softmax; why not zero after; the conventions table, "True = may attend" in the HW. (§5) | Valid length 1: $[[1, 0], [1, 0]]$; zeroing after gives rows summing to 0.62 and 0.38. |
| 64–79 | Multi-head: $Q = XW_q$ etc. as "demand / offer / hand over"; split, attend, merge, $W_o$; the shape walk; the transpose; each head scales by $\sqrt{d/h}$; $4(d^2 + d)$. (§6) | The shape table with $B = 3$, $T = 6$, $d = 16$, $h = 4$; "without the transpose head 1 holds position 1's features". |
| 79–97 | Back to the translator: HW6's decoder + one attention step (Figure 8). **Trace one decoder query with shapes**: GRU output $s_t$ $(n, H)$ → scores against the encoder outputs $(n, S, H)$ by broadcasting $(n, 1, 1, H) + (n, 1, S, H)$ → weights $(n, 1, S)$ → context $(n, 1, H)$ → $\tanh(W_c[s_t; c_t])$; Bahdanau's wiring as a remark; results. (§10) | 27.9 → 37.6 BLEU; the reverse task 97% → 10% vs ≈100%. |
| 97–100 | HW7 2(A–C), 3 map. (§12) | |

# Lecture B — Mon 12/7, first half: order, positions, heads (about 50 minutes)

| Time | On the board | Worked example / demo |
|---|---|---|
| 0–5 | Recap: one formula, three roles; masks. | |
| 5–17 | Self-attention (the 8/8 p2 board: "each token attends each step", "run in parallel"); the CNN/RNN/self-attention table; permutation equivariance. (§7) | "Dog bites man" vs "man bites dog" give the same set of vectors. |
| 17–32 | Positional encoding (the 8/8 p8 board): the binary counter, made continuous; the formula; relative position = rotation. (§8) | Bits 0–7; the $d = 4$ table $P_0 \dots P_3$; $R(\delta)$ maps $P_2$'s first pair to $P_3$'s. Figure 7. |
| 32–40 | The causal mask: no peeking; one pass trains every position. (§9) | All-zero scores: outputs $1, 1.5, 2, 2.5$ (running mean). |
| 40–50 | Why several heads: one softmax = one blend; two heads keep two lookups apart; same parameters. HW7 2(D), 6 map. (§11, §12) | Values $(1, 0)$, $(0, 1)$: one head $\to (0.5, 0.5)$; two heads $\to (1, 0, 0, 1)$. |

# The story, in six beats

1. **HW6's translator had to memorize.** One vector per sentence, and long sentences suffer. People don't translate that way: they look back. That is the whole motivation.

2. **Looking back is something they have already done.** Soft $k$-NN weighted every training point by its similarity to the query; kernel regression did it with real targets. Write the weighted average once and rename the pieces: query, keys, values. If this beat lands, the rest of the week is bookkeeping. Spend the time here.

3. **The similarity becomes a dot product, and it is learned.** The Gaussian kernel is a dot product in disguise; projections $W_q, W_k, W_v$ let the network decide what "similar" means. The $\sqrt{d}$ keeps the softmax soft.

4. **Several lookups at once, at no extra cost.** Heads are slices of one big projection; the only hard part is the reshape, so draw the shapes.

5. **Attention alone is a bag of words.** Shuffle the input and the output shuffles along. Positions are added on purpose, as a timestamp; the causal mask keeps a language model honest.

6. **It works.** One attention step on HW6's own decoder: about 10 BLEU, and a task where the fixed vector collapses becomes easy. The alignments are real but soft: say so.

**If you remember one line:** *attention is soft k-NN with a learned similarity, run inside the network.*