# Reading microgpt: Supplement

What do the three name-tag tables (wte, wpe, lm_head) represent?

— The meaning of the numbers on the training monitor —

Planned and edited by Yoshio Okamoto / Written by Claude Opus 5.5 (Anthropic)

Ver. 0.1 October 24, 2026

Note: The following content was generated by AI. The editor has checked it, but please be aware that it may still contain errors or mistaken assumptions.

© 2026 Yoshio Okamoto. Free to use for educational purposes (please cite the source). The microgpt program is the work of Andrej Karpathy.

## Introduction

By adding a few lines to the microgpt program, we can keep showing the values of the parameters (knobs) in a separate terminal window while the model is training. This supplement explains the three tables shown on that monitor:

- `wte` (27 rows × 16 columns): the **letters' name tags**
- `wpe` (16 rows × 16 columns): the **name tags of the positions (seats)**
- `lm_head` (27 rows × 16 columns): the **score sheets at the exit**

Using analogies, we explain what the lists of numbers represent, and what it means when they change during training. This is a re-reading of Sections 3 and 4 of the main booklet, *Reading microgpt*, linked to what appears on the monitor screen.

The screen values in the text are excerpts from the monitor screen during an actual training run (one example of how training goes midway).

## 1 Where the three tables are used

`gpt()` receives one letter per call and returns scores for the "next letter," one for each of the 27 kinds. The three tables are used at its entrance and exit (Figure 1).

- `wte`: a **name tag** for each letter. **Only that letter's one row** is taken from the table.
- `wpe`: a **seat tag** for each position (which letter of the name). **Only that position's one row** is taken.
- `lm_head`: a **score sheet** for each letter. **All 27 rows** are used to score the candidates for the next letter.

In all three, one row is a list of 16 numbers (a 16-dimensional vector). The number 16 is the "number of lanes" chosen by the author, and no column has a meaning decided by a human.

## 2 wte: the letters' name tags

### 2.1 Analogy: profile cards

Imagine that each of the 27 members of a class (a–z and BOS) has a **profile card with 16 items**. One row of `wte` is one member's card. But the item names are not written anywhere. Nobody decided what "item 1" is; the meanings grow naturally during training.

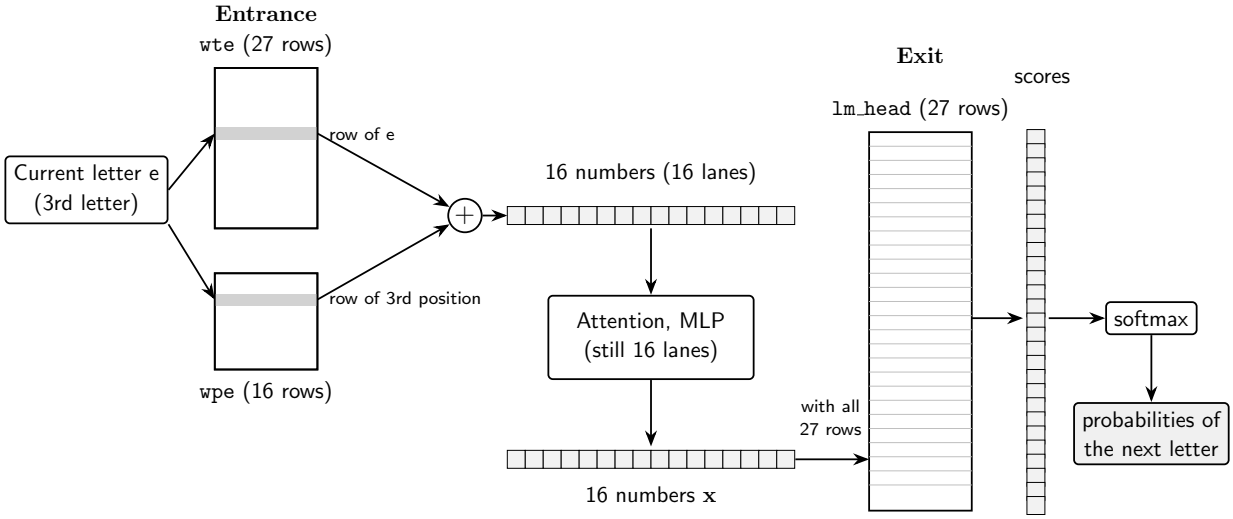



Figure 1 How the three tables are used. At the entrance, just one row is taken from each of `wte` and `wpe`, and the two are added. At the exit, the 16 numbers $\mathbf{x}$ that arrive are compared with all 27 rows of `lm_head` to make the scores of the 27 letters.

Before training, the cards are filled with **random numbers**, as if written by rolling dice (random numbers with mean 0 and spread 0.08). At this stage there is no reason for the cards of a and e to be similar, nor for them to be different.

## 2.2 What it means when the numbers change

Each step uses just one name. For example, in the step that learns `yuheng`, only the 7 rows of y, u, h, e, n, g and BOS are rewritten, based on "what came after this letter." It is like **rewriting only the cards of the students called on in that day's lesson**[1].

Of the 1000 names used in 1000 training steps, 729 contain an a, but only 3 contain a q. On the monitor, the row of a should move often, and the row of q should hardly ever move.

As the rewriting is repeated hundreds of times, **the cards of letters that are used in similar ways within names come to have similar lists of numbers**. In Appendix B of the main booklet, we drew this as arrows for a small example reduced to 3 numbers per letter (Figure 2). The arrows of i and v, which pointed in different directions before training, point in almost the same direction after training.

## 2.3 On the monitor screen

From the screen at about step 490 of the actual training (16 numbers per letter), we extracted the rows of the vowels a, e, o, u and of the consonants l and r (Figure 3). Looking at the gray columns (5, 6, 10 and 14), a, e, o and u share a similar pattern of signs: "columns 5 and 6 negative, column 10 positive, column 14 negative." Meanwhile, l and r both have large positive values in columns 3 and 6, resembling each other in a pattern different from the vowels.

[1]Strictly speaking, the refinement in how the knobs are turned (Adam) remembers the "momentum" of recent turning, so rows of letters that did not appear in that step also move slightly while the momentum lasts.

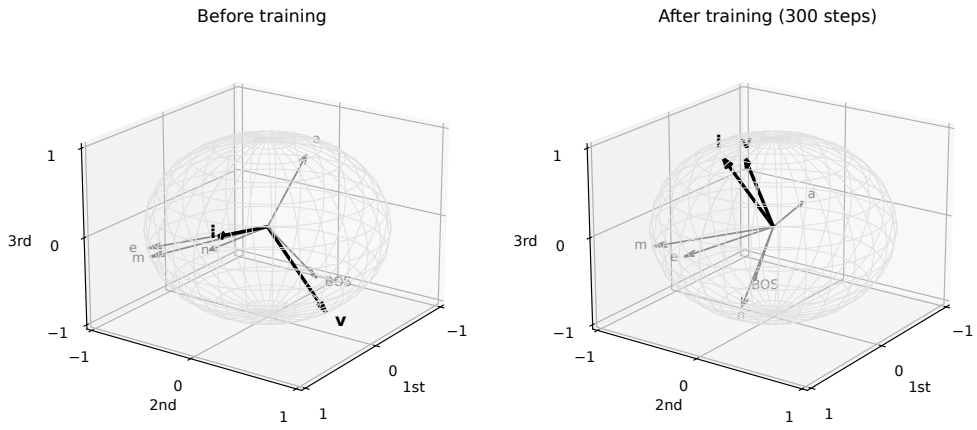



Figure 2 Name tags (example reduced to 3 dimensions) drawn as arrows (left: before training; right: after 300 steps). Reprinted from the main booklet. Black: i and v. In the small example (the four names emma, ava, mia, anna), i appears in mia and v in ava, and in both cases the next letter is always a, so they ended up with name tags pointing the same way (cosine similarity in the 3D example: 0.04 before training, 0.98 after).

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| a | +0.14 | +0.13 | +0.19 | −0.07 | +0.01 | −0.27 | −0.32 | +0.00 | −0.05 | +0.09 | +0.32 | +0.11 | −0.08 | −0.20 | −0.19 | −0.39 |
| e | −0.05 | −0.26 | +0.26 | +0.03 | +0.06 | −0.31 | −0.26 | −0.22 | +0.06 | +0.05 | +0.32 | −0.03 | +0.09 | −0.05 | −0.37 | +0.01 |
| o | −0.09 | −0.26 | +0.35 | −0.29 | +0.08 | −0.16 | −0.22 | +0.25 | −0.01 | +0.10 | +0.37 | −0.09 | +0.05 | −0.09 | −0.37 | −0.34 |
| u | +0.20 | −0.04 | −0.12 | −0.19 | +0.09 | −0.38 | −0.31 | −0.31 | +0.02 | +0.30 | +0.21 | −0.14 | +0.01 | −0.01 | −0.57 | −0.15 |
| l | +0.13 | +0.18 | +0.01 | +0.41 | −0.22 | +0.05 | +0.41 | −0.11 | −0.19 | −0.06 | −0.05 | +0.02 | +0.23 | +0.16 | +0.06 | +0.53 |
| r | +0.14 | +0.14 | +0.05 | +0.33 | −0.27 | +0.04 | +0.28 | +0.03 | −0.06 | +0.02 | −0.03 | +0.07 | +0.09 | −0.07 | +0.04 | +0.15 |

Figure 3 Excerpt from the `wte` monitor screen (about step 490). Rows are letters; columns are the numbers 0–15 of the 16 numbers.

This is an impression from looking at one screen midway through training, and it has not been checked carefully. Still, there are signs that the name tags of vowels, and of consonants, are becoming similar to each other. In names, vowels and consonants often alternate, so "which letter tends to come next" is similar among vowels; this is probably the reason.

# 3 wpe: the seat tags

## 3.1 Analogy: seat-number tags

Imagine that the letters of a name sit in **seats** lined up in order: 1st letter, 2nd letter, and so on. One row of `wpe` is a **seat tag with 16 items** attached to each seat. The letter's name tag (a row of `wte`) plus the tag of the seat where that letter sits (a row of `wpe`) is the starting point that flows through the model.

The seat tags are needed because Attention, the calculation that later compares letters with each other, cannot tell the order of the letters by itself. Even for the same e, the letter likely to come next differs between an e in the 1st position and an e in the 5th position. Adding the seat tag lets the model tell "which position this e is in."

### 3.2 What it means when the numbers change: seats nobody sits in

The seat tags also start out as random numbers. Only **the tags of seats that were actually sat in** by that step's name are rewritten.

This is where the length of names matters. Most names are 5–7 letters long, and only 38 of the 32,033 names have 13 letters or more. For the 1000 names used in 1000 training steps, here is how many times each seat was sat in[2]:

| Position (row) | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Times used | 1000 | 1000 | 1000 | 999 | 986 | 903 | 679 | 361 | 152 | 61 | 21 | 5 | 2 | 1 | **0** | **0** |

**Nobody ever sits** in the seats at positions 14 and 15 during the 1000 steps. So the last two rows of `wpe` **keep their random starting values and do not change at all**, from the beginning of training to the end[3]. Positions 11–13 are also hardly used, so they stay almost as they started.

On the monitor, the upper rows (positions 0–6) should move often, the lower rows less and less, and the last two rows should appear frozen. Figure 4 is an excerpt from the screen at about step 190. The row of position 0 already has a large value such as −0.50, while the rows of positions 12–15 remain small, about the same size as the initial random numbers.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | −0.09 | −0.12 | −0.50 | +0.05 | +0.03 | +0.00 | −0.12 | +0.01 | +0.08 | +0.12 | +0.01 | +0.08 | −0.08 | −0.05 | −0.15 | +0.08 |
| 1 | −0.05 | −0.02 | −0.17 | −0.01 | −0.26 | −0.04 | +0.19 | −0.06 | −0.19 | +0.06 | −0.00 | −0.20 | +0.17 | +0.04 | +0.07 | +0.04 |
| 2 | +0.01 | −0.04 | −0.19 | −0.17 | −0.08 | −0.23 | +0.08 | −0.03 | −0.10 | +0.06 | −0.01 | −0.25 | +0.28 | +0.10 | −0.19 | +0.01 |
| ⋮ | | | | | | | | | | | | | | | | |
| 12 | +0.00 | +0.02 | −0.02 | +0.09 | −0.09 | +0.07 | −0.10 | −0.12 | −0.02 | +0.06 | +0.01 | −0.09 | +0.06 | −0.04 | +0.01 | −0.01 |
| 13 | −0.01 | +0.05 | −0.01 | −0.17 | +0.03 | +0.06 | +0.02 | +0.00 | +0.11 | +0.06 | −0.05 | +0.12 | −0.10 | +0.16 | −0.03 | +0.02 |
| 14 | −0.07 | −0.02 | −0.04 | +0.20 | +0.07 | −0.03 | +0.05 | −0.09 | −0.03 | +0.03 | +0.04 | +0.08 | +0.01 | +0.14 | +0.01 | +0.04 |
| 15 | +0.03 | −0.12 | −0.01 | +0.05 | +0.01 | +0.03 | −0.12 | −0.14 | +0.05 | +0.06 | +0.02 | +0.00 | +0.05 | −0.10 | −0.14 | +0.11 |

Figure 4 Excerpt from the `wpe` monitor screen (about step 190). Rows are positions (which letter of the name). The two gray rows (positions 14 and 15) keep their values from before training until the end.

## 4 lm_head: the score sheets at the exit

### 4.1 Analogy: 27 judges

Imagine that at the exit there are **27 judges**, one for each of the 27 letters. One row of `lm_head` is one judge's **score sheet**, with a "weight" written for each of its 16 items.

When the 16 numbers **x** that have passed through the model arrive, every judge multiplies their own score sheet by **x** item by item, adds the results up, and gives a score. This is the "matrix × vector" calculation from the main booklet: the inner product of one row of the matrix with the vector gives one score. **If x has a pattern of signs similar to the judge's score sheet, the score is high; if the pattern is opposite, the score is low.** For example, if an **x** similar to the score sheet of the judge for e arrives, the score for "next is e" becomes high.

The 27 judges' scores are turned into probabilities that "add up to 1" by softmax (Figure 5).

[2]The seat at position 0 is taken by BOS, which marks the start of a name. A name of length $L$ uses the $L+1$ seats at positions 0 to $L$ (at most 16).

[3]The gradient is always 0, so Adam's "momentum" also stays 0, and the knobs do not turn at all.

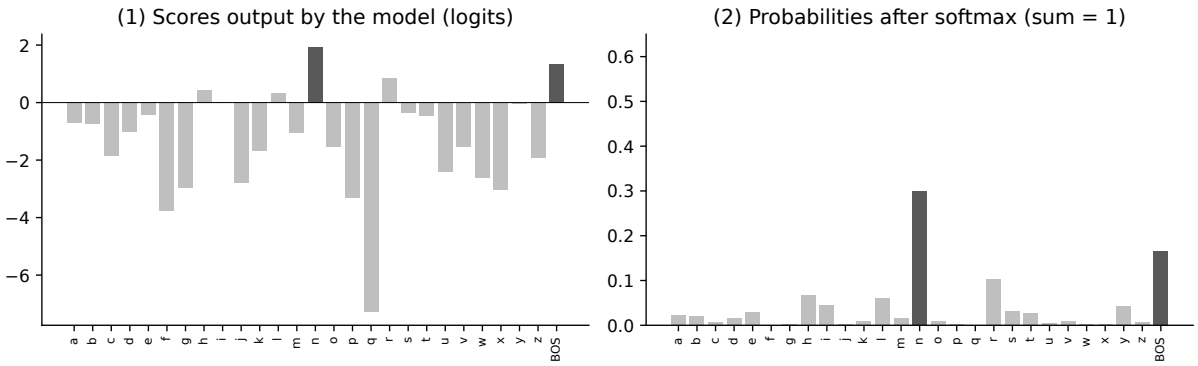



Figure 5 The scores of the 27 judges (1) and the probabilities after softmax (2), for the letter after the final a when emma is fed into the trained model. Reprinted from the main booklet.

`wte` and `wpe` were tables from which "just one row is taken," but a big difference with `lm_head` is that **all 27 rows are used every time**. The judge for BOS decides "the name ends here." When BOS gets a high score, the name ends there.

### 4.2 What it means when the numbers change: every judge revises their sheet

Suppose the correct answer at some position was e. After scoring, all 27 judges revise their score sheets.

- **The judge for the correct letter e** revises the sheet in the direction that brings it **closer to x**, so that next time a similar **x** arrives, the score for e is even higher.
- **The other 26 judges** revise their sheets in the direction that moves them **away from x**. A judge who gave a high probability to a wrong letter is corrected more strongly.

So in `lm_head`, **every row moves at every step**, including the rows of letters that do not appear in that name. This is in contrast with `wte`, where only the rows of letters in the name moved. Placing the two tables side by side on the monitor shows this difference clearly.

### 4.3 On the monitor screen

From the screen at about step 190, we extracted the rows of the common letters a, e, n and of the rare letters j, q, x (Figure 6). The rows of j, q and x contain large values of ±0.5–0.7, and in the gray columns (columns 5, 8 and 13 positive, column 6 negative) the three have very similar patterns.

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| a | −0.12 | +0.03 | −0.19 | −0.01 | +0.03 | +0.02 | +0.20 | +0.01 | −0.18 | +0.22 | −0.08 | −0.15 | +0.10 | −0.03 | −0.12 | +0.21 |
| e | +0.01 | +0.08 | −0.09 | +0.05 | −0.08 | −0.00 | +0.10 | −0.13 | −0.03 | −0.02 | −0.13 | −0.03 | +0.07 | +0.13 | −0.04 | +0.22 |
| n | +0.05 | −0.04 | +0.29 | −0.23 | +0.25 | +0.01 | −0.07 | +0.08 | −0.07 | +0.16 | +0.28 | −0.00 | −0.02 | −0.08 | −0.19 | −0.18 |
| j | +0.13 | −0.32 | −0.19 | −0.02 | −0.09 | +0.40 | −0.36 | −0.02 | +0.33 | −0.05 | −0.13 | −0.07 | −0.27 | +0.26 | +0.50 | −0.04 |
| q | +0.31 | −0.29 | +0.60 | −0.01 | −0.21 | +0.60 | −0.34 | +0.15 | +0.55 | −0.60 | −0.49 | +0.09 | −0.03 | +0.41 | +0.69 | −0.41 |
| x | +0.11 | −0.08 | +0.30 | −0.18 | +0.00 | +0.72 | −0.13 | +0.09 | +0.65 | −0.19 | −0.04 | +0.11 | −0.04 | +0.51 | +0.29 | −0.18 |

Figure 6 Excerpt from the `lm_head` monitor screen (about step 190). Rows are letters (judges).

This, too, is an impression from looking at a mid-training screen, but it can be explained as follows. j, q and x are rarely the correct answer. So almost every time they are on the "move away" side, and their score sheets **keep being pushed in much the same direction**. Being pushed again and again away from the shape of the **x** that usually arrives, the three judges' sheets

probably became similar and their values grew large. This way, when an ordinary **x** arrives, j, q and x get low scores, and the model correctly predicts "rare letters" as unlikely.

## 5 Why the changes gradually become smaller

If you watch the monitor to the end, the changes in the numbers become smaller in the later part of training, and at the end the numbers look almost frozen. There are two reasons.

The first is that **the amount by which the knobs are turned is itself made smaller as training proceeds**. In the program, the turning amount (learning rate) starts at 0.01 and is reduced linearly as the steps go on, to almost 0 at step 1000 (line 175). The idea is to search for the rough position with large, coarse turns, then settle down with fine turns at the end.

The second is that as training proceeds and the predictions get better, **the need for correction itself becomes smaller**. The loss (how wrong the predictions are) in Figure 7 drops sharply in the first few hundred steps and then falls only slowly.

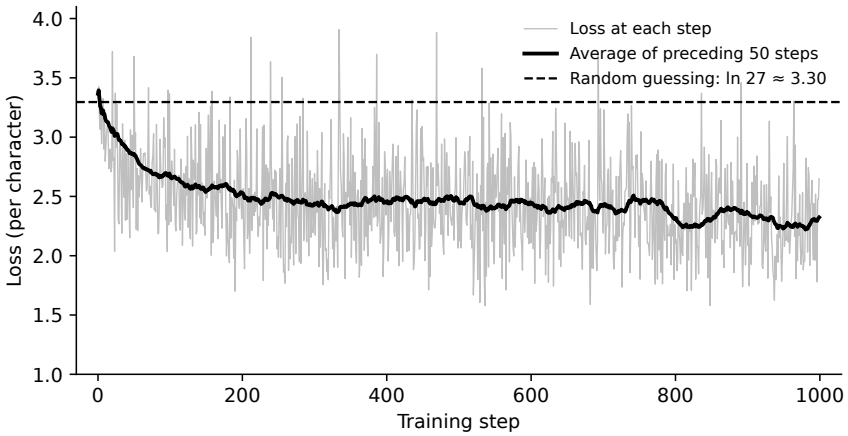



Figure 7 The change of the loss (reprinted from the main booklet). Gray: the loss at each step; black line: the average over the preceding 50 steps.

> What the monitor shows are 1,120 of the 4,192 knobs: those of `wte`, `wpe` and `lm_head` (Figure 8). Each number moves only a little and seems to jitter at random, but as a whole the knobs are being turned "in the direction that makes the loss smaller."

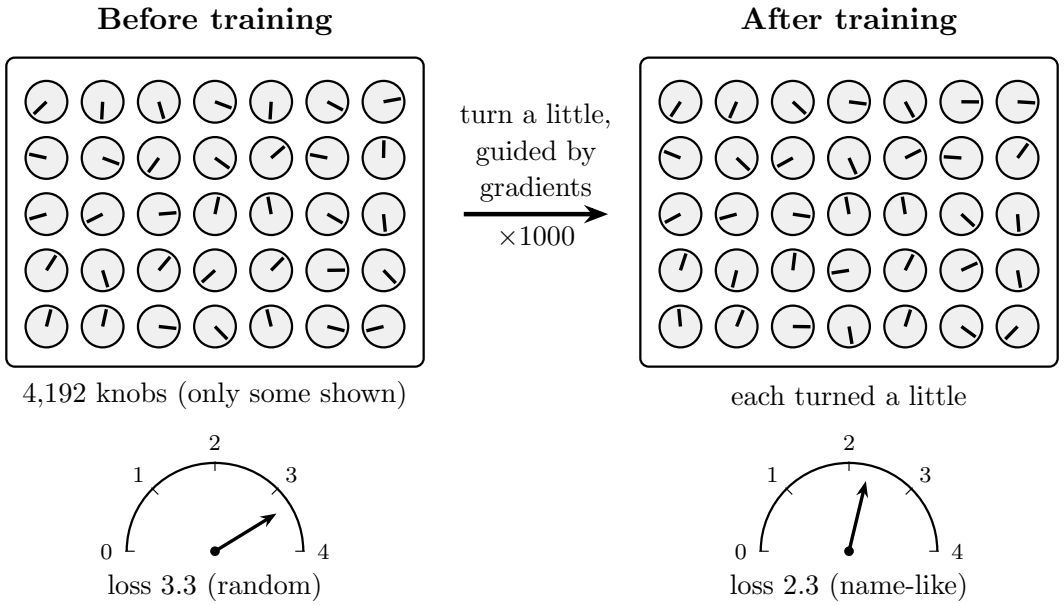



Figure 8 An illustration likening the parameters to "knobs" (reprinted from the main booklet). One number in the monitor's tables corresponds to one knob.

# 6 Summary

| Table | Analogy | Rows that move in one step | What you see on the monitor |
|---|---|---|---|
| `wte` (27 rows) | letters' profile cards | rows of the letters in that name (and BOS) | The more common the letter, the more its row moves. Vowels, and consonants, gradually come to have similar patterns. |
| `wpe` (16 rows) | seat tags | rows of the positions used by that name | Upper rows move more. Rows 14 and 15 never change from start to finish. |
| `lm_head` (27 rows) | judges' score sheets | all rows | All rows move every time. Rows of rare letters grow large and similar. |

None of the 16 numbers has a meaning decided by a human. Through training, **things used in similar ways come to have similar lists of numbers, and things used in different ways come to have different lists**. The changing numbers on the monitor show these "patterns" taking shape little by little.

# A The changes of the three tables in formulas

**Entrance (`wte`, `wpe`)** When letter $c$ is at position $t$, the starting vector is

$$\mathbf{x}_0 = \mathbf{w}_c^{\text{te}} + \mathbf{w}_t^{\text{pe}}$$

($\mathbf{w}_c^{\text{te}}$ is row $c$ of `wte`). The gradient of the loss $\mathcal{L}$ flows only to the rows used, $\mathbf{w}_c^{\text{te}}$ and $\mathbf{w}_t^{\text{pe}}$; the gradients of the other rows are 0. If the same letter or position appears several times in a name, their gradients are added together.

**Exit (`lm_head`)** Let $\mathbf{x}$ be the final vector and $\mathbf{w}_i$ row $i$ of `lm_head`. Then the score of letter $i$ is the inner product $z_i = \mathbf{w}_i \cdot \mathbf{x}$, and its probability is $p_i = e^{z_i} / \sum_j e^{z_j}$. With correct letter $y$ and loss $\mathcal{L} = -\log p_y$,

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}_i} = (p_i - \delta_{iy})\,\mathbf{x}, \qquad \delta_{iy} = \begin{cases} 1 & (i = y) \\ 0 & (i \neq y) \end{cases}$$

Since the knobs are turned opposite to the gradient,

- the correct row ($i = y$) moves in the direction of $(1 - p_y)\mathbf{x}$, that is, **toward $\mathbf{x}$**;
- the other rows ($i \neq y$) move in the direction of $-p_i\mathbf{x}$, that is, **away from $\mathbf{x}$**.

Since all $p_i$ are positive, none of the 27 rows has a zero gradient, and every row moves at every step. The actual size of each move is adjusted by Adam, so it is not simply proportional to the size of the gradient, but the direction is as given by this formula.

**Learning-rate decay** The learning rate at step $s$ ($0 \leq s < 1000$) is $\eta_s = 0.01 \times (1 - s/1000)$, which is 0.00001 at the last step. For a row whose gradient is 0, Adam's first and second moments also stay 0 throughout, so its update is exactly 0 (rows 14 and 15 of `wpe`).