Step-by-Step Tasks

目录

Think of these as **guided nudges**, not exams. Each task has a single intellectual target.

Step 00 - Bigram Model

Concept: next-token prediction without context

Task

1. Change the training text to:

   `"abababababab"`

2. Retrain the model.
3. Generate text starting from `"a"`.

Questions to answer

- Why does the model always alternate?
- What happens if you start from `"b"`?
- Can this model learn `"aba"` patterns? Why or why not?

Intended realization

The model has *no memory*. It literally only knows one-step statistics.

Step 01 - Tokenizer

Concept: text → discrete symbols

Task

1. Add one sentence with **non-ASCII characters** (e.g. accents or emojis).
2. Retrain the tokenizer.
3. Compare:
   - number of tokens
   - encoded token lengths

Questions to answer

- Which characters become single tokens?
- Which get split?
- Why is `[UNK]` needed at all?

Intended realization

Tokenization defines the “atoms” of the model’s world.

Step 02 - Dataset Construction

Concept: self-supervision via shifting

Task

1. Print one `(x, y)` pair from the batch.
2. Write them as integers *and* as arrows:

```
x: 10 → 11 → 12
y: 11 → 12 → 13
```

Questions to answer

- Where do the labels come from?
- How many training examples can one string produce?

Intended realization

Labels are free. Data is infinite.

Step 03 - Embeddings

Concept: tokens are vectors, not numbers

Task

1. Print the embedding vectors for:

   `x[0, 0]` **`and`** `x[0, 1]`

2. Verify they are different even if token IDs are close.

Questions to answer

- Does token ID magnitude matter?
- What happens if positional embeddings are removed?

Intended realization

Token IDs are identifiers, not measurements.

Step 04 - MLP Language Model

Concept: depth without interaction

Task

1. Increase the MLP depth from 2 to 4 layers.
2. Observe loss behavior.
3. Compare with a shallower model.

Questions to answer

- Does deeper always help?
- Why can this model *not* copy text from earlier positions?

Intended realization

Depth $\neq$ context.

Step 05 - Self-Attention (Single Head)

Concept: tokens can communicate

Task

1. Print the attention matrix:

   ```
   att[0]
   ```

2. Check that rows sum to 1.

Questions to answer

- What does one row of attention represent?
- What would happen if softmax were removed?

Intended realization

Attention is a learned weighted average.

Step 06 - Multi-Head Attention

Concept: parallel subspaces

Task

1. Change the number of heads from 4 to 1.
2. Observe output shape and behavior.

Questions to answer

- Why must `embed_dim` be divisible by `num_heads`?
- What does a "head" buy us conceptually?

Intended realization

Heads are viewpoints, not extra layers.

Step 07 - Transformer Block

Concept: residual pathways and normalization

Task

1. Remove the residual connection once.
2. Train or forward-pass the model.

Questions to answer

- What breaks?
- Why is LayerNorm placed *before* attention?

Intended realization

Residuals are not optional glue.

Step 08 - Full Transformer

Concept: depth via stacking

Task

1. Increase the number of layers from 2 to 4.
2. Compare runtime and loss behavior.

Questions to answer

- What changes with depth?
- What stays the same?

Intended realization

Transformers scale by repetition.

Step 09 - Training

Concept: optimization as part of modeling

Task

1. Change the learning rate by $\times 10$ and $\div 10$.
2. Observe stability.

Questions to answer

- What does "divergence" look like?
- Why does clipping exist?

Intended realization

Training is physics, not math.

Step 10 - Generation

Concept: model as an autoregressive process

Task

1. Replace `argmax` with sampling.
2. Generate several sequences.

Questions to answer

- Why does sampling introduce diversity?
- Why does temperature matter?

Intended realization

> Generation is a choice, not a fact.

Optional Meta-Task (Highly Recommended)

After Step 10, let us:

> Draw the entire training + generation pipeline on paper.

No code. Just arrows.

Anyone who can do that has genuinely learned transformers.