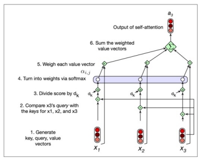
A Sitting Example

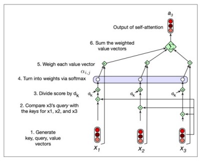

a_3
Output of self-attention
6. Sum the weighted value vectors
5. Weigh each value vector
$\alpha_{i,j}$
4. Turn into weights via softmax
3. Divide score by d_k
2. Compare x3's query with the keys for x1, x2, and x3
1. Generate key, query, value vectors
x_1
x_2
x_3

- Consider the token sequence: "The cat sat <end>"
- 4 tokens
- Embedding dimension: 4
- Step 1: Produce input embeddings:

Token 1 "The":	[1.0, 0.5, 0.2, 0.1]
Token 2 "cat":	[0.5, 1.0, 0.3, 0.2]
Token 3 "sat":	[0.3, 0.2, 1.0, 0.5]
Token 4 "<end>":	[0.1, 0.1, 0.1, 1.0]

9

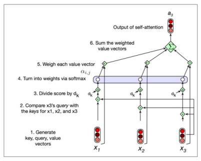
A Sitting Example


a_3
Output of self-attention
6. Sum the weighted value vectors
5. Weigh each value vector
$\alpha_{i,j}$
4. Turn into weights via softmax
3. Divide score by d_k
2. Compare x3's query with the keys for x1, x2, and x3
1. Generate key, query, value vectors
x_1
x_2
x_3

- Step 2: Determine Attention Score of "The":
  - Token 1 attending to Token 1:
    [1.0, 0.5, 0.2, 0.1] · [1.0, 0.5, 0.2, 0.1] = 1.0 + 0.25 + 0.04 + 0.01 = **1.30**
  - Token 1 attending to Token 2:
    [1.0, 0.5, 0.2, 0.1] · [0.5, 1.0, 0.3, 0.2] = 0.5 + 0.5 + 0.06 + 0.02 = 1.08
  - Token 1 attending to Token 3:
    [1.0, 0.5, 0.2, 0.1] · [0.3, 0.2, 1.0, 0.5] = 0.3 + 0.1 + 0.2 + 0.05 = 0.65
  - Token 1 attending to Token 4:
    [1.0, 0.5, 0.2, 0.1] · [0.1, 0.1, 0.1, 1.0] = 0.1 + 0.05 + 0.02 + 0.1 = 0.27

10

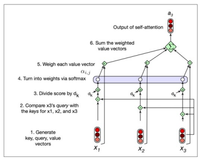
A Sitting Example


a_3
Output of self-attention
6. Sum the weighted value vectors
5. Weigh each value vector
$\alpha_{i,j}$
4. Turn into weights via softmax
3. Divide score by d_k
2. Compare x3's query with the keys for x1, x2, and x3
1. Generate key, query, value vectors
x_1
x_2
x_3

- Step 3: Normalize
  - Divide each score by dk, which is the square root of the dimensionality.
  - Dimensionality = 4.
  - Hence divide by 2.
  - Token 1 attending to Token 1: 1.30 / 2 = 0.65
  - Token 1 attending to Token 2: 1.08 / 2 = 0.54
  - Token 1 attending to Token 3: 0.65 / 2 = 0.325
  - Token 1 attending to Token 4: 0.27 / 2 = 0.135

11

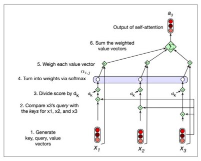
A Sitting Example


a_3
Output of self-attention
6. Sum the weighted value vectors
5. Weigh each value vector
$\alpha_{i,j}$
4. Turn into weights via softmax
3. Divide score by d_k
2. Compare x3's query with the keys for x1, x2, and x3
1. Generate key, query, value vectors
x_1
x_2
x_3

- Step 4: Apply Softmax
  - exp(0.65) = 1.916
  - exp(0.54) = 1.716
  - exp(0.325) = 1.384
  - exp(0.135) = 1.145
  - Sum = 6.161
- Attention weights
  [1.916/6.161, 1.716/6.161, 1.384/6.161, 1.145/6.161] = [0.311, 0.278, 0.225, 0.186]
- These weights tell us:
  Token 1 pays 31.1% attention to itself, 27.8% to "cat", 22.5% to "sat", and 18.6% to "<end>".

12

A Sitting Example


a_3
Output of self-attention
6. Sum the weighted value vectors
5. Weigh each value vector
$\alpha_{i,j}$
4. Turn into weights via softmax
3. Divide score by d_k
2. Compare x3's query with the keys for x1, x2, and x3
1. Generate key, query, value vectors
x_1
x_2
x_3

- Step 5: Output for token 1 ("The")

```
   0.311 × [1.0, 0.5, 0.2, 0.1] ("The")
 + 0.278 × [0.5, 1.0, 0.3, 0.2] ("cat")
 + 0.225 × [0.3, 0.2, 1.0, 0.5] ("sat")
 + 0.186 × [0.1, 0.1, 0.1, 1.0] ("<end>")

 = [0.311, 0.156, 0.062, 0.031]
 + [0.139, 0.278, 0.083, 0.056]
 + [0.068, 0.045, 0.225, 0.113]
 + [0.019, 0.019, 0.019, 0.186]

 = [0.537, 0.498, 0.389, 0.386]
```

- This output vector for "The" is now context-aware.
- It contains information weighted from all tokens in the sequence.

13