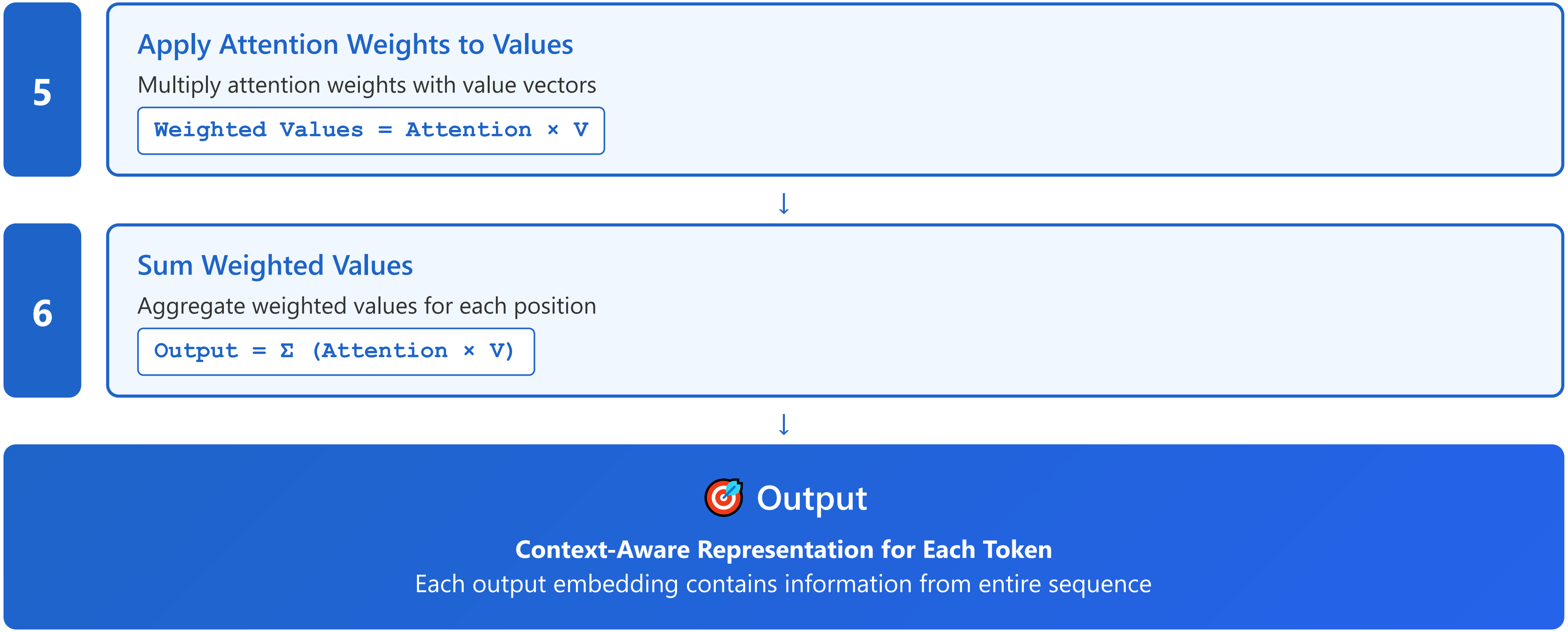
Self-Attention Computation Process (2/2)

From Attention Weights to Context-Aware Representations



5

Apply Attention Weights to Values

Multiply attention weights with value vectors

```
Weighted Values = Attention × V
```

↓

6

Sum Weighted Values

Aggregate weighted values for each position

```
Output = Σ (Attention × V)
```

↓

🎯 Output

Context-Aware Representation for Each Token

Each output embedding contains information from entire sequence

⚡

Parallel
computation for all
positions simultaneously

🔄

Differentiable
end-to-end for
backpropagation

🌐

Global context
in every output
representation

💡 Numerical Example (Continued)

📋 **Recap from Part 1:**

We computed attention weights using softmax on scaled scores.
Now we'll use these weights to create context-aware representations.

Attention Weights Matrix (5×5) - After Softmax:

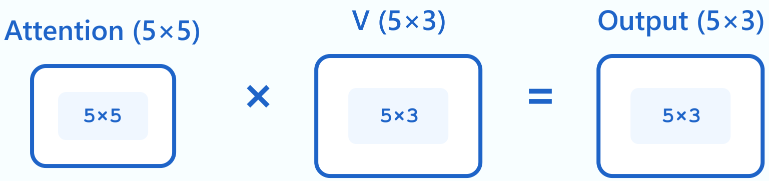
0.182	0.204	0.207	0.192	0.198
0.170	0.279	0.222	0.173	0.186
0.162	0.206	0.272	0.184	0.208
0.188	0.213	0.224	0.219	0.215
0.178	0.212	0.239	0.204	0.244

Each row sums to 1.0 and represents how much each token attends to others

Value Matrix V (5×3) - Same as X in our example:

1.0	0.5	0.2
0.8	1.2	0.3
0.6	0.9	1.1
1.1	0.4	0.7
0.9	0.7	0.8

⑤ Weighted Values = Attention × V (Matrix Multiplication):



Attention (5×5)

5×5

×

V (5×3)

5×3

=

Output (5×3)

5×3

⑥ Final Output Matrix (5×3) - Context-Aware Embeddings:

0.87	0.74	0.59
0.83	0.88	0.58
0.83	0.78	0.68
0.88	0.73	0.63
0.84	0.78	0.70

Each row is a new embedding that incorporates information from all tokens

🎯 Token 1's Output Example

Original embedding:

1.0 0.5 0.2

Context-aware embedding:

0.87 0.74 0.59

✓ The output is a weighted combination of all tokens' values
✓ Token 1 now "knows about" the entire sequence context
✓ Different attention patterns → Different contextual representations

💭 How Token 1's Output was Computed:

$$\text{Output}[1] = 0.182\times[1.0, 0.5, 0.2] + 0.204\times[0.8, 1.2, 0.3] + 0.207\times[0.6, 0.9, 1.1] + 0.192\times[1.1, 0.4, 0.7] + 0.198\times[0.9, 0.7, 0.8]$$

= [0.87, 0.74, 0.59]

This is a weighted average where weights come from attention scores!