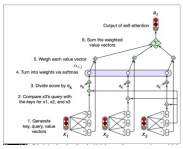
Back to Cats



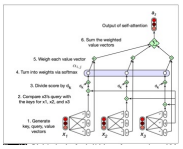
Embedding tokens.

More realistically, tokens from prior processing.

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

19

Back to Cats



- Create Q, K, V with Different Weight Matrices

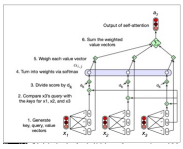
W_Q (transforms X to "what am I looking for?"):
W_Q = [[1.0, 0.0, 0.5, 0.0],
[0.0, 1.0, 0.0, 0.5],
[0.5, 0.0, 1.0, 0.0],
[0.0, 0.5, 0.0, 1.0]]

W_K (transforms X to "what do I offer?"):
W_K = [[1.0, 0.2, 0.0, 0.0],
[0.2, 1.0, 0.0, 0.0],
[0.0, 0.0, 1.0, 0.3],
[0.0, 0.0, 0.3, 1.0]]

W_V (transforms X to "what information will I contribute?"):
W_V = [[0.8, 0.0, 0.0, 0.1],
[0.0, 0.9, 0.1, 0.0],
[0.0, 0.1, 0.8, 0.0],
[0.1, 0.0, 0.0, 0.9]]

20

Back to Cats



- Computing q = X @ W_Q

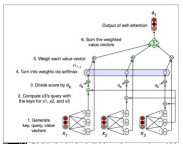
For token 1: "The" =
[1.0, 0.5, 0.2, 0.1] @ [[1.0, 0.0, 0.5, 0.0],
[0.0, 1.0, 0.0, 0.5],
[0.5, 0.0, 1.0, 0.0],
[0.0, 0.5, 0.0, 1.0]]

[1.0×1.0 + 0.5×0.0 + 0.2×0.5 + 0.1×0.0,
1.0×0.0 + 0.5×1.0 + 0.2×0.0 + 0.1×0.5,
1.0×0.5 + 0.5×0.0 + 0.2×1.0 + 0.1×0.0,
1.0×0.0 + 0.5×0.5 + 0.2×0.0 + 0.1×1.0]
= [1.1, 0.55, 0.7, 0.35]

"cat": [0.65, 1.1, 0.55, 0.7]
"sat": [0.8, 0.45, 1.15, 0.6]
<end>: [0.15, 0.6, 0.15, 1.05]

21

Back to Cats



- Computing k = X @ W_K

For token 1: "The" =
[1.0, 0.5, 0.2, 0.1] @ [[1.0, 0.2, 0.0, 0.0],
[0.2, 1.0, 0.0, 0.0],
[0.0, 0.0, 1.0, 0.3],
[0.0, 0.0, 0.3, 1.0]]

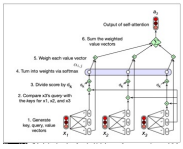
[1.0×1.0 + 0.5×0.2 + 0.2×0.0 + 0.1×0.0,
1.0×0.2 + 0.5×1.0 + 0.2×0.0 + 0.1×0.0,
1.0×0.0 + 0.5×0.0 + 0.2×1.0 + 0.1×0.3,
1.0×0.0 + 0.5×0.0 + 0.2×0.3 + 0.1×1.0]
= [1.1, 0.7, 0.23, 0.16]

"cat": [0.7, 1.1, 0.36, 0.29]
"sat": [0.34, 0.26, 1.15, 0.8]
<end>: [0.12, 0.12, 0.33, 1.03]

K = [[1.1, 0.7, 0.23, 0.16],
[0.7, 1.1, 0.36, 0.29],
[0.34, 0.26, 1.15, 0.8],
[0.12, 0.12, 0.33, 1.03]]

22

Back to Cats

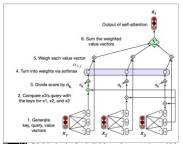


- Computing v = X @ W_V

"The": [0.82, 0.47, 0.17, 0.19]
"cat": [0.7, 1.1, 0.36, 0.29]
"sat": [0.34, 0.26, 1.15, 0.8]
<end>: [0.12, 0.12, 0.33, 1.03]

23

Back to Cats



- Calculating attention score for "The": q @ K

K = [[1.1, 0.7, 0.34, 0.12],
[0.7, 1.1, 0.26, 0.12],
[0.23, 0.36, 1.15, 0.33],
[0.16, 0.29, 0.8, 1.03]]

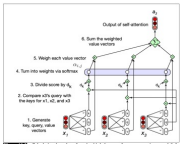
Q for "The" = [1.1, 0.55, 0.7, 0.35]

"The" & "The": 1.1×1.1 + 0.55×0.7 + 0.7×0.23 + 0.35×0.16 =
1.21 + 0.385 + 0.161 + 0.056 = 1.812
"The" & "cat": 1.1×0.7 + 0.55×1.1 + 0.7×0.36 + 0.35×0.29 =
0.77 + 0.605 + 0.252 + 0.102 = 1.729
"The" & "sat": 1.1×0.34 + 0.55×0.26 + 0.7×1.15 + 0.35×0.8 =
0.374 + 0.143 + 0.805 + 0.28 = 1.602
"The" & "<end>": 1.1×0.12 + 0.55×0.12 + 0.7×0.33 + 0.35×1.03 =
0.132 + 0.066 + 0.231 + 0.361 = 0.790

24

Back to Cats

- Scaling, i.e. divide by square root of dimensionality of key.
  "The" score: [1.812/2, 1.729/2, 1.602/2, 0.790/2] =
  [0.906, 0.865, 0.801, 0.395]
- Softmax:
  exp(0.906) = 2.475, exp(0.865) = 2.375, exp(0.801) = 2.228, exp(0.395) = 1.484
  Sum = 8.562
  Attention weights for "The":
  [2.475/8.562, 2.375/8.562, 2.228/8.562, 1.484/8.562] =
  [0.289, 0.277, 0.260, 0.173]
  Interpretation: "The" pays:
  - 28.9% attention to itself
  - 27.7% attention to "cat"
  - 26.0% attention to "sat"
  - 17.3% attention to "<end>"



25

Back to Cats

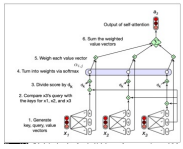
- Output for "The"
  $0.289 \times V^{\text{The}} + 0.277 \times V^{\text{cat}} + 0.260 \times V^{\text{sat}} + 0.173 \times V^{\text{<end>}} =$

  $0.289 \times [0.82, 0.47, 0.17, 0.19] +$
  $0.277 \times [0.42, 0.93, 0.34, 0.23] +$
  $0.260 \times [0.26, 0.19, 0.82, 0.48] +$
  $0.173 \times [0.18, 0.10, 0.09, 0.91]$

  $= [0.237, 0.136, 0.049, 0.055] +$
  $[0.116, 0.258, 0.094, 0.064] +$
  $[0.068, 0.049, 0.213, 0.125] +$
  $[0.031, 0.017, 0.016, 0.157]$

  $= [0.452, 0.460, 0.372, 0.401]$ ⟸

  - This is the context-aware output vector for "The"
  - It has been enriched with weighted information from all other tokens.



26