activity19_partd

April 3, 2025

Question 1-d

[1]:
```
import numpy as np

# Define the query, keys, and values
q = np.array([0, 1, 0])
keys = np.array([[0, 1, 0], [1, 0, 0], [0, 0, 1], [0, 1, 0]])
values = np.array([[1, 2], [0, 3], [4, 5], [2, 2]])

# Step 1: Compute dot product similarities
dot_products = np.dot(keys, q)  # q.T * k_i for each key k_i
similarities = dot_products / np.sqrt(3)  # Normalize by sqrt(r), r=3

# Step 2: Compute attention weights
exp_similarities = np.exp(similarities)
attention_weights = exp_similarities / np.sum(exp_similarities)

# Step 3: Compute the output
output = np.sum(attention_weights[:, np.newaxis] * values, axis=0)

# Results
print("Dot Product Similarities:", similarities)
print("Attention Weights:", attention_weights)
print("Output Vector:", output)
```

```
Dot Product Similarities: [0.57735027 0.         0.         0.57735027]
Attention Weights: [0.32022874 0.17977126 0.17977126 0.32022874]
Output Vector: [1.67977126 2.71908505]
```

[ ]:
```

```