01204211 Discrete Mathematics
Lecture 8b: Vectors and Matrices

Jittat Fakcharoenphol

October 16, 2025

What is a vector?

You can think of a **vector** as an "ordered" list of elements (which are typically numbers). For example:

- $[1, 2, 5, 20]$
- $[0, 0, 1, 1, 0, 0, 0, 1]$

You can also view a vector as a **function**, e.g., you can view $\boldsymbol{u} = [1, 2, 5, 20]$ as a function $\boldsymbol{u}$ that maps

$$0 \mapsto 1, \quad 1 \mapsto 2, \quad 2 \mapsto 5, \quad 3 \mapsto 20.$$

Each element in the vector is typically a real number ($\mathbb{R}$), but can be an element from other sets with appropriate property (more on this later).

Remark: Mathematically, a vector is an element of a vector space. We will understand this more later.

What can be represented as a vector?

Applications in machine learning

Viewing vectors: vectors in $\mathbb{R}^2$

Viewing vectors: vectors in $\mathbb{R}^3$

n-vectors over $\mathbb{R}$

- We mostly deal with vectors with finite number of elements.
- This is a **4-vector**: $[10, 20, 500, 4]$.
- We sometimes also write it as a column vector:

$$\begin{bmatrix} 10 \\ 20 \\ 500 \\ 4 \end{bmatrix}$$

- When every element of a vector is from some set, we say that it is a vector **over** that set. For example, $[10, 20, 500, 4]$ is a 4-vector over $\mathbb{R}$.

Vector operations

- As discussed in the previous slides, when working with a system of linear equations, we mostly deals with **linear combinations** of vectors.
- We will look at the operations we do to vectors to obtain their linear combinations.
- The operations are:
  - Vector additions
  - Scalar multiplications
- These operations motivate the definition of vector spaces.

Vector additions

Given two n-vectors

$$\boldsymbol{u} = [u_1, u_2, \ldots, u_n]$$

and

$$\boldsymbol{v} = [v_1, v_2, \ldots, v_n],$$

we have that

$$\boldsymbol{u} + \boldsymbol{v} = [u_1 + v_1, u_2 + v_2, \ldots, u_n + v_n].$$

Vector additions, in picture

Zero vectors

A zero n-vector $\mathbf{0} = [0, 0, \ldots, 0]$ is an additive identity, i.e., for any vector $\boldsymbol{u}$,

$$\mathbf{0} + \boldsymbol{u} = \boldsymbol{u} + \mathbf{0} = \boldsymbol{u}.$$

Scalar multiplications

For a vector over $\mathbb{R}$, we refer to an element α in $\mathbb{R}$ as a scalar. For an n-vector

$$\boldsymbol{u} = [u_1, u_2, \ldots, u_n],$$

we have that

$$\alpha \cdot \boldsymbol{u} = [\alpha \cdot u_1, \alpha \cdot u_2, \ldots, \alpha \cdot u_n],$$

Scalar multiplications, in pictures

Linear combinations

For any scalar

$$\alpha_1, \alpha_2, \ldots, \alpha_m$$

and vectors

$$\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_m,$$

we say that

$$\alpha_1 \boldsymbol{u}_1 + \alpha_2 \boldsymbol{u}_2 + \cdots + \alpha_m \boldsymbol{u}_m$$

is a **linear combination** of $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m$.

Examples:

A linear system with 3 variables

Give the following linear system.

$$\begin{array}{ccccccc} 2x_1 & + & 4x_2 & + & 3x_3 & = & 7 \\ x_1 & + & & & 5x_3 & = & 12 \\ 4x_1 & + & 2x_2 & + & 3x_3 & = & 10 \end{array}$$

If we rewrite the system as

$$\begin{bmatrix} 2 \\ 1 \\ 4 \end{bmatrix} \cdot x_1 + \begin{bmatrix} 4 \\ 0 \\ 2 \end{bmatrix} \cdot x_2 + \begin{bmatrix} 3 \\ 5 \\ 3 \end{bmatrix} \cdot x_3 + = \begin{bmatrix} 7 \\ 12 \\ 10 \end{bmatrix}.$$

This becomes the problem of expressing a vector as linear combination of other vectors. I.e., given vectors

$$\boldsymbol{u}_1 = [2, 1, 4], \quad \boldsymbol{u}_2 = [4, 0, 2], \quad \boldsymbol{u}_3 = [3, 5, 3]$$

we would like to find coefficients x_1, x_2, x_3 such that

$$x_1 \cdot \boldsymbol{u}_1 + x_2 \cdot \boldsymbol{u}_2 + x_3 \cdot \boldsymbol{u}_3 = [7, 12, 10].$$

Span

A set of all linear combination of vectors $\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_m$ is called the **span** of that set of vectors.
It is denote by $\mathrm{Span}\{\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_m\}$.

Examples:

Convex combination

For any scalar

$$\alpha_1, \alpha_2, \ldots, \alpha_m,$$

such that $\alpha_1 + \alpha_2 + \ldots + \alpha_m = 1$ and $\alpha_i \geq 0$ for all i, and vectors

$$\boldsymbol{u}_1, \boldsymbol{u}_2, \ldots, \boldsymbol{u}_m,$$

we say that

$$\alpha_1 \boldsymbol{u}_1 + \alpha_2 \boldsymbol{u}_2 + \cdots + \alpha_m \boldsymbol{u}_m$$

is a **convex combination** of $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m$.

Examples:

What is a matrix?

Matrices arise in many places. We will see that there are essentially two ways to look at matrices.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{bmatrix} = \left[\begin{array}{c|c|c} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{array}\right] = \left[\begin{array}{ccc} 1 & 2 & 3 \\ \hline 4 & 5 & 6 \\ \hline 7 & 8 & 9 \\ \hline 10 & 11 & 12 \end{array}\right]$$

A matrix from a system of linear equations

Consider the following system of linear equations:

$$\begin{array}{ccccccc} x_1 & + & x_2 & + & x_3 & = & 5 \\ 2x_1 & + & x_2 & + & 2x_3 & = & 10 \\ 3x_1 & + & x_2 & + & 2x_3 & = & 4 \end{array}$$

Again we can view it as a vector equation:

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} x_1 + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} x_2 + \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix} x_3 = \begin{bmatrix} 5 \\ 10 \\ 4 \end{bmatrix}$$

A matrix from a system of linear equations

From the following system of linear equations

$$\begin{array}{ccccccc} x_1 & + & x_2 & + & x_3 & = & 5 \\ 2x_1 & + & x_2 & + & 2x_3 & = & 10 \\ 3x_1 & + & x_2 & + & 2x_3 & = & 4 \end{array}$$

We can also view variables x_1, x_2, x_3 as a vector, i.e., let $\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$.

The coefficients form a nice rectangular "matrix" A:

$$A = \begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & 2 \\ 3 & 1 & 2 \end{bmatrix},$$

and rewrite the system as

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & 2 \\ 3 & 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 10 \\ 4 \end{bmatrix}$$

Size

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & 2 & 5 \\ 3 & 1 & 2 & 4 \end{bmatrix}$$

The **size** of a matrix is determined by the number of rows and columns. A matrix with m rows and n columns is referred to as an m-by-n matrix or an $m \times n$ matrix. We refers to m and n as its **dimensions**.

Matrix-Vector Multiplication

How would we understand the multiplication

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & 2 \\ 3 & 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

By rows. Consider the first row of A:

$$\begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 1 \cdot x_1 + 1 \cdot x_2 + 1 \cdot x_3.$$

Let's look at another two rows:

$$\begin{bmatrix} 2 & 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 2 \cdot x_1 + 1 \cdot x_2 + 2 \cdot x_3, \qquad \begin{bmatrix} 3 & 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 3 \cdot x_1 + 1 \cdot x_2 + 2 \cdot x_3,$$

Matrix-Vector Multiplication **by Rows**

We look at matrix-vector multiplication with “row perspective”. This is a common way to view matrix-vector multiplication.

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & 2 \\ 3 & 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \cdot x_1 + 1 \cdot x_2 + 1 \cdot x_3 \\ 2 \cdot x_1 + 1 \cdot x_2 + 2 \cdot x_3 \\ 3 \cdot x_1 + 1 \cdot x_2 + 2 \cdot x_3 \end{bmatrix}$$

Recall:

$$\begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 1 \cdot x_1 + 1 \cdot x_2 + 1 \cdot x_3.$$

$$\begin{bmatrix} 2 & 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 2 \cdot x_1 + 1 \cdot x_2 + 2 \cdot x_3,$$

$$\begin{bmatrix} 3 & 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = 3 \cdot x_1 + 1 \cdot x_2 + 2 \cdot x_3,$$

Review: Dot product

Definition

For n-vectors $\boldsymbol{u} = [u_1, u_2, \ldots, u_n]$ and $\boldsymbol{v} = [v_1, v_2, \ldots, v_n]$, the **dot product** of $\boldsymbol{u}$ and $\boldsymbol{v}$, denoted by $\boldsymbol{u} \cdot \boldsymbol{v}$, is

$$u_1 \cdot v_1 + u_2 \cdot v_2 + \cdots + u_n \cdot v_n$$

Matrix-Vector Multiplication **by Rows**

We look at matrix-vector multiplication with "row perspective", which can be written nicely with **dot product**.
I.e., from:

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & 2 \\ 3 & 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \cdot x_1 + 1 \cdot x_2 + 1 \cdot x_3 \\ 2 \cdot x_1 + 1 \cdot x_2 + 2 \cdot x_3 \\ 3 \cdot x_1 + 1 \cdot x_2 + 2 \cdot x_3 \end{bmatrix}$$

we have

$$\left[\begin{array}{c} \boldsymbol{r}_1 \\ \hline \boldsymbol{r}_2 \\ \hline \boldsymbol{r}_3 \end{array}\right] \boldsymbol{x} = \left[\begin{array}{c} \boldsymbol{r}_1 \cdot \boldsymbol{x} \\ \hline \boldsymbol{r}_2 \cdot \boldsymbol{x} \\ \hline \boldsymbol{r}_3 \cdot \boldsymbol{x} \end{array}\right],$$

where

$$\boldsymbol{r}_1 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}, \quad \boldsymbol{r}_2 = \begin{bmatrix} 2 & 1 & 2 \end{bmatrix}, \quad \boldsymbol{r}_{\boldsymbol{3}} = \begin{bmatrix} 3 & 1 & 2 \end{bmatrix}.$$

Dot-product perspective

The matrix-vector product is a vector of **dot products** between each rows and the vector.

Matrix-Vector Multiplication **by Columns**

However, another nice way to look at matrix-vector multiplication is **by columns**.
Notice that:

$$\begin{bmatrix} 1 & 1 & 1 \\ 2 & 1 & 2 \\ 3 & 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \cdot x_1 + 1 \cdot x_2 + 1 \cdot x_3 \\ 2 \cdot x_1 + 1 \cdot x_2 + 2 \cdot x_3 \\ 3 \cdot x_1 + 1 \cdot x_2 + 2 \cdot x_3 \end{bmatrix}$$

can be written as

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} x_1 + \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} x_2 + \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix} x_3 = \begin{bmatrix} 5 \\ 10 \\ 4 \end{bmatrix}$$

Linear combination perspective

The matrix-vector product is a **linear combination** of column vectors.

Two perspectives: Matrix-Vector multiplication

Dot products between rows and the vector

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} a_{11} \cdot x_1 + a_{12} \cdot x_2 + a_{13} \cdot x_3 \\ a_{21} \cdot x_1 + a_{22} \cdot x_2 + a_{23} \cdot x_3 \\ a_{31} \cdot x_1 + a_{32} \cdot x_2 + a_{33} \cdot x_3 \\ a_{41} \cdot x_1 + a_{42} \cdot x_2 + a_{43} \cdot x_3 \end{bmatrix}$$

Linear combination of column vectors

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ a_{41} \end{bmatrix} \cdot x_1 + \begin{bmatrix} a_{12} \\ a_{22} \\ a_{32} \\ a_{42} \end{bmatrix} \cdot x_2 + \begin{bmatrix} a_{13} \\ a_{23} \\ a_{33} \\ a_{43} \end{bmatrix} \cdot x_3$$

Dimensions

If the matrix has n columns, the vector should be an n-vector.

Document search

- You have 1,000,000 documents in a library. Given another document, you would like to find similar documents from the library. How can you do that?
- You need some way to measure document **similarity**.
- Suppose that you nave N documents in the library: $d_1, d_2, \ldots, d_N$. Given a query document q, you want to find document d_i that maximize

$$sim(d_i, q),$$

where $sim(d, d')$ is the similarity score between documents d and d'.

Document vector models

What is a document? It's just a list of words. If you throw all the ordering away, a document is simply a set of words.

Let's start with an example. Suppose that we only care about 5 words: `dog`, `cat`, `food`, `restaurant`, and `coffee`.

Consider the following 4 (very short) documents:

- d_1: People love pets. Most famous pets are cats and dogs.
  $d_1 = \{\texttt{dog}, \texttt{cat}\}$
- d_2: Bar Mai has many restaurants with cheap foods.
  $d_2 = \{\texttt{restaurant}, \texttt{food}\}$
- d_3: Cat cafe used to be popular in Thailand. People buy coffee and play with cats there.
  $d_3 = \{\texttt{coffee}, \texttt{cat}\}$
- d_4: Dogs are human's best friends. They were around in civilization for a long long time.
  $d_4 = \{\texttt{dog}\}$

How can we translate these sets into vectors?

Document vector models

We assign a fixed co-ordinate for each word, and if a set contain a particular word, we put 1 in that co-ordinate.
Here are our 5 words: `dog`, `cat`, `food`, `restaurant`, and `coffee`.
Each document becomes:

- d_1: People love pets. Most famous pets are cats and dogs.
  $d_1 = \{\texttt{dog}, \texttt{cat}\}$, $\boldsymbol{d}_1 = [1, 1, 0, 0, 0]$
- d_2: Bar Mai has many restaurants with cheap foods.
  $d_2 = \{\texttt{restaurant}, \texttt{food}\}$, $\boldsymbol{d}_2 = [0, 0, 1, 1, 0]$
- d_3: Cat cafe used to be popular in Thailand. People buy coffee and play with cats there.
  $d_3 = \{\texttt{coffee}, \texttt{cat}\}$, $\boldsymbol{d}_3 = [0, 1, 0, 0, 1]$
- d_4: Dogs are human's best friends. They were around in civilization for a long long time.
  $d_4 = \{\texttt{dog}\}$, $\boldsymbol{d}_4 = [1, 0, 0, 0, 0]$

Document vector models

Words: `dog`, `cat`, `food`, `restaurant`, and `coffee`.

Suppose that we have query document:

q: I love cats and coffee. What restaurant should I visit?

as a set: $q = \{\texttt{cat}, \texttt{coffee}, \texttt{restaurant}\}$

as a vector: $\boldsymbol{q} = [0, 1, 0, 1, 1]$

Our documents are:

- d_1: People love pets. Most famous pets are cats and dogs.
  $d_1 = \{\texttt{dog}, \texttt{cat}\}$ $\boldsymbol{d}_1 = [1, 1, 0, 0, 0]$
- d_2: Bar Mai has many restaurants with cheap foods.
  $d_2 = \{\texttt{restaurant}, \texttt{food}\}$ $\boldsymbol{d}_2 = [0, 0, 1, 1, 0]$
- d_3: Cat cafe used to be popular in Thailand. People buy coffee and play with cats there.
  $d_3 = \{\texttt{coffee}, \texttt{cat}\}$ $\boldsymbol{d}_3 = [0, 1, 0, 0, 1]$
- d_4: Dogs are human's best friends. They were around in civilization for a long long time.
  $d_4 = \{\texttt{dog}\}$ $\boldsymbol{d}_4 = [1, 0, 0, 0, 0]$

How can we define "similarity" measure?

Dot products as a similarity measure

From the previous example, we see that the dot products between $\boldsymbol{d}_i$'s and $\boldsymbol{q}$ count the number of common words.

This simple idea can be extended in many ways.

- We can increase our "dictionary"'s size to include more words.
- We can group similar words into the same "co-ordinates".
- In fact, the dot product measures the "angle" between vectors. For vectors over $\mathbb{R}$, we have that

$$\boldsymbol{u} \cdot \boldsymbol{v} = |\boldsymbol{u}||\boldsymbol{v}| \cos \theta,$$

  where θ is the angle between vectors $\boldsymbol{u}$ and $\boldsymbol{v}$.

Computing all similarity scores

If we have documents $\boldsymbol{d}_1, \boldsymbol{d}_1, \ldots, \boldsymbol{d}_N$, as vectors, and a query $\boldsymbol{q}$, how can we compute all similarity scores?

By performing matrix-vector multiplication:

$$\begin{bmatrix} \boldsymbol{d}_1 \\ \hline \boldsymbol{d}_2 \\ \hline \vdots \\ \hline \boldsymbol{d}_N \end{bmatrix} \begin{bmatrix} \boldsymbol{q} \end{bmatrix} = \begin{bmatrix} sim(\boldsymbol{d}_1, \boldsymbol{q}) \\ sim(\boldsymbol{d}_2, \boldsymbol{q}) \\ \vdots \\ sim(\boldsymbol{d}_N, \boldsymbol{q}) \end{bmatrix}$$

Vector-matrix multiplication

Let's consider another direction.
What is

$$\begin{bmatrix} x_1 & x_2 & x_3 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} ?$$

As a linear combination

As dot products

Matrix-matrix multiplication

Consider

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}.$$

Matrix-matrix multiplication (based on matrix-vector multiplication)

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{bmatrix} \left[\begin{array}{c|c|c|c} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{array}\right].$$

Matrix-matrix multiplication (based on vector-matrix multiplication)

$$\left[\begin{array}{ccc} x_{11} & x_{12} & x_{13} \\ \hline x_{21} & x_{22} & x_{23} \end{array}\right] \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}.$$

Matrix transpose

If A is an $m \times n$ matrix

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix},$$

the **transpose** of A, denoted by A^T is an $n \times m$ matrix

$$\begin{bmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ a_{13} & a_{23} & \cdots & a_{m3} \\ \vdots & \vdots & & \vdots \\ a_{1n} & a_{m2} & \cdots & a_{mn} \end{bmatrix}.$$

Remark: We usually view a vector as a column vector. Therefore, a dot product between m-vectors can be viewed also as a matrix multiplication:

$$\boldsymbol{u} \cdot \boldsymbol{v} = \boldsymbol{u}^T \boldsymbol{v}$$

Matrix multiplication and transpose

What is $(AB)^T$?

Key-Value database

Suppose you have a database of key-value pairs:

$$\{(somchai, 10), (somying, 14), (somnuk, 23), (somjai, 50), (somsom, -40)\}$$

Given a query q, you want to find a value v such that (q, v) is in the database. E.g.,
Let's see how we could do that (very **inefficiently**) with matrix multiplication.

Vector encodings of keys and queries

- You want to have **distinct** keys: $\boldsymbol{k}_1, \boldsymbol{k}_2, \ldots, \boldsymbol{k}_n$

- You want a query $\boldsymbol{q}$ to **match** with an appropriate key. (Maybe the key which is exactly the same.)

Example

- Key encoding:

$$somchai = [0,1,0,0,0,0],\ \ somying = [0,0,0,0,1,0],\ \ somnuk = [1,0,0,0,0,0],$$

$$somjai = [0,0,1,0,0,0], \qquad somsom = [0,0,0,0,0,1]$$

- A value table (or vector): $\boldsymbol{v} = \begin{bmatrix} 10 \\ 14 \\ 23 \\ 50 \\ -40 \end{bmatrix}$
- A query $\boldsymbol{q}$ is a 5-vector. A query matches key $\boldsymbol{k}_i$ if

$$\boldsymbol{k}_i^T \boldsymbol{q} = 1$$

Example (cont)

- Key matrix $K = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
- A value table (or vector): $\boldsymbol{v} = \begin{bmatrix} 10 \\ 14 \\ 23 \\ 50 \\ -40 \end{bmatrix}$
- Let's try querying with various $\boldsymbol{q}$
- The final formula is

$$(K\boldsymbol{q})^T\boldsymbol{v} = (\boldsymbol{q}^T K^T)\boldsymbol{v}$$

Key-Value database (with vector values)

Suppose you have a database of key-value pairs, where a value is a 2-vector:

$$\{(somchai, [10, 20]), (somying, [14, -2]), (somnuk, [23, 3]), (somjai, [50, -10])\}$$

Given a query q, can you find a 2-vector $\boldsymbol{v}$ such that $(q, \boldsymbol{v})$ is in the database?

Understanding self-attention formula

Self-attention mechanisms are key steps in transformers, work horses for all chatbots you have been using recently. The formula looks like (from wikipedia)

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$