## Intutive way to understand no loop KNN →

I spent a lot of time on this and after reading through so many solution's and working it on paper I finally got it working.

So the problem in hand is

| | test | train | goal |
|---|---|---|---|
| shape | 500 X 3072 | 5000 X 3072 | 500 X 5000 |

→ (A)

Let's start with scalars.

we will mostly see euclidean-square here.

for 2 scalars a and b the distance square would be

$$(a-b)^2$$

by our high school formula it is

$$(a-b)^2 = a^2 + b^2 - 2ab$$

we will apply this same thing

but before that let me just relax the dimensions a bit

let's make eq$^n$ (A) as

| | test | train | goal |
|---|---|---|---|
| shape | 2 X 2 | 3 X 2 | 2 X 3 |

↳ D = # of dimensions

let's put in values

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

$$\begin{bmatrix} 5 & 6 \\ 7 & 8 \\ 9 & 10 \end{bmatrix}$$ → train

all distinct values so that we won't confuse

Let me write the distance intuitively — no magic here

$$\begin{bmatrix} (1-5)^2+(3-6)^2 & (1-7)^2+(3-8)^2 & (1-9)^2+(3-10)^2 \\ (2-5)^2+(4-6)^2 & (2-7)^2+(4-8)^2 & (2-9)^2+(4-10)^2 \end{bmatrix}$$

now let me apply $(a-b)^2 = a^2+b^2-2ab$ (our friend)

$$\begin{bmatrix} 1^2+5^2-2\times1\times5 + 3^2+6^2-2\times3\times6 & 1^2+7^2-2\times7\times1 + 3^2+8^2-2\times3\times8 & 1^2+9^2-2\times9\times1+3^2+10^2-2\times3\times10 \\ 2^2+5^2-2\times2\times5 + 4^2+6^2-2\times4\times6 & 2^2+7^2-2\times7\times2 + 4^2+8^2-2\times4\times8 & 2^2+9^2-2\times2\times9+4^2+10^2-2\times4\times10 \end{bmatrix}$$

now ~~that is~~ those are lot of numbers.
what we will do now is seperate them.
and see the sum would be same for each cell.

$$\begin{bmatrix} 1^2+3^2 & 1^2+3^2 & 1^2+3^2 \\ 2^2+4^2 & 2^2+4^2 & 2^2+4^2 \end{bmatrix} + \begin{bmatrix} 5^2+6^2 & 7^2+8^2 & 9^2+10^2 \\ 5^2+6^2 & 7^2+8^2 & 9^2+10^2 \end{bmatrix}$$

$$-2 * np.dot\left(\begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}, \begin{bmatrix} 5 & 7 & 9 \\ 6 & 8 & 10 \end{bmatrix}\right)$$

trust me I did verify it

now let's take advantage of numpy broadcast you can see it in Jake video I have included the link

$$\begin{bmatrix} 1^2+3^2 \\ 2^2+4^2 \end{bmatrix} + \begin{bmatrix} 5^2+6^2 & 7^2+8^2 & 9^2+10^2 \end{bmatrix}$$

$$-2 * np.dot(test, train.T)$$

now if we the first sum.

$$\begin{bmatrix} 1^2 + 3^2 \\ 2^2 + 4^2 \end{bmatrix}$$

what has happened we squared the test and added in the rows.

Same with train

$$\begin{bmatrix} 5^2+6^2 & 7^2+8^2 & 9^2+10^2 \end{bmatrix}$$

but here we have first transposed the train and then added in columns.

now to the code. it is simple.

1st ↑

`np.square(test).sum(axis=1, keepdims=True)` — it will still let it be a matrix else it becomes a vector

2nd ↑

`+ np.square(train.T).sum(axis=0, keepdims=True)`

`- 2 * np.dot(test, train.T)`

↓ 3rd.

— hence coded / proved

☺