WEEK 4
MACHINE TRANSLATION

• Transforming word vectors

- Rotation matrix $R=\begin{bmatrix}2 & 0\\ 0 & -2\end{bmatrix}$
- word vector $x=[[1,1]]$
- np.dot $(x, R) = [[2,-2]]$
- Loss $= \|XR-Y\|_F$

  $g=\frac{d}{dR}\text{Loss}$ (gradient)

  $R=R-\alpha g$ (update)
- Frobenius norm

  $A=\begin{bmatrix}2 & 2\\ 2 & 2\end{bmatrix}$

  $\|A\|_F=\sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{i,j}|^2}=4$

  $\|A\|_F^2=16$
- Gradient

  $\text{Loss}=\|XR-Y\|_F^2$

  $g=\frac{d}{dR}\text{Loss}=\frac{2}{m}(X^T(XR-Y))$

• K-nearest neighbors

"hello" ⟶ R matrix ⟶ [transformed] ⟶ find similar words

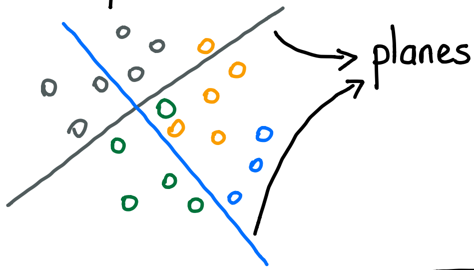
• Hash tables and hash functions.

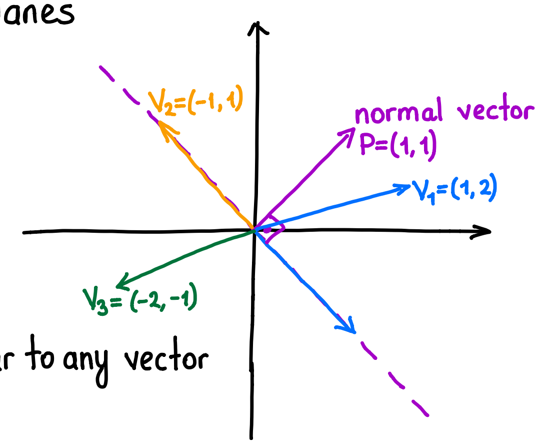
- a function that assigns a hash value is called a hash function

  hash function (vector) ⟶ hash value
- hash table design should ensure to store the similar words in the same bucket

•Locality sensitive hashing

-reduces the computational cost of finding k-nearest neighbors.





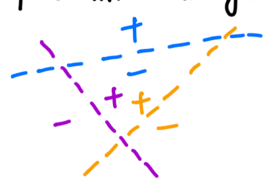
- the normal vector is perpendicular to any vector that lie on the plane.

$\underbrace{PV_1^T=3,}_{\text{the vectors are on the same side}}\ PV_2^T=0\ \underbrace{PV_3^T=-3,}_{\text{opposite side}}$

•Multiple planes.

-Multiple planes → dot products → hash values.

-to divide the vector space into managable regions, use more than one plane.



$P_1 v^T=3$, $sign_1=+1$, $h_1=1$

$P_2 v^T=5$, $sign_2=+1$, $h_2=1$

$P_3 v^T=-2$, $sign_3=-1$, $h_3=0$

$hash=2^0\times h_1+2^1\times h_2+2^2\times h_3$

$hash=3$

-Multiple planes, single hash value!

•Approximate nearest neighbors

-create multiple sets of random planes

-by using multiple sets of random planes for locality-sensitive hashing, we have more robust way of searching the vector space

-do not search on entire space ⟶ faster