

# Ներկառուցող շերտ: Դիրքային կոդավորում

## Հայկ Կարապետյան

### 1 Embedding շերտ

Սկզբից մեր մոտեցային նախադասությունը վերածում ենք token-ների, tokenizer-ի միջոցով: Օրինակ՝ "Կարմիր կովը սև կաշին չի փոխում" նախադասությունը կարող է վերածվել [2810, 4002, 3001, 487, 125, 30] list-ի, որ ամեն անդամ ցույց է տալիս token-ի դիրքը բառարանում: Այս թվերը հետո վերածվում են one hot վեկտորների և վեկտորի չափողականությունը հավասար է բառարանի չափին (vocab\_size): One hot վեկտորները այնքան էլ շատ ինֆորմացիա չեն պարունակում բառի մասին: Եթե վերցնենք և հաշվենք "սև" և "կարմիր" բառերի միջև կոսինուսային նմանություն (cosine similarity), կստանանք 0: Միևնույն ժամանակ երկու բառերը իրար շատ նման են՝ երկուսն էլ գույն են: Այդ պատճառով One hot վեկտորները անցկացնում ենք embedding շերտի միջով: Այդ շերտը իրենից ներկայացնում է W մատրից, որի չափերն են (vocab\_size, embedding\_size): Օրինակ transformer-ի հոդվածում embedding\_size=512: Ամեն token ունի իրեն համապատասխան embedding\_size չափողականությամբ վեկտորը embedding շերտում: W մատրիցը պարամետր է և թարմացվում է ուսուցանման ընթացքում: Ցանցի ուսուցման վերջում, կարող ենք "սև" և "կարմիր" բառերի one hot վեկտորները անցկացնել embedding շերտով և կտեսնենք, որ cosine similarity-ի արժեքը մոտ է մեկին:

### 2 Positional Encoding

"Կարմիր" և "սև" բառերի embedding վեկտորները իրար շատ նման են ստացվում, բայց կարմիր բառը գտնվում է նախադասության սկզբում, իսկ "սև" բառը նախադասության մեջտեղում: Embedding վեկտորը այդ ինֆորմացիան նույնպես պետք է պարունակի: Դրա համար օգտագործում ենք positional encoding-ը: Եթե վերցնենք ուղղակի համարակալենք "սև" բառը 0-ով, իսկ "կարմիր" բառը 2-ով, այդ դեպքում չենք ունենա բավարար արժեքներ embedding վեկտորի բոլոր անդամներին գումարելու համար (embedding վեկտորի չափողականությունը embedding\_size է): Իսկ եթե վերցնենք այդ վեկտորի բոլոր անդամներին գումարենք 2, ապա վեկտորը ամբողջականությամբ կփոխվի: Այդ պատճառով վեկտորին գումարում ենք կոսինուս կամ սինուս արժեքներ: Միայն positional encoding ստանալու բանաձևը.

$$PE(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$
$$PE(pos, 2i + 1) = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

i-ն ցույց է տալիս անդամի տեղը նախադասության մեջ: Օրինակ "Կարմիր կովը սև կաշին չի փոխում" նախադասության մեջ "սև" բառի համար i=2: i-ն փոխվում է 0-ից նախադասության անդամների քանակ միջակայքում: pos-ը ցույց է տալիս, թե embedding վեկտորի, որ ինդեքսի համար ենք ուզում ստանալ positional encoding: pos-ը փոխվում է 0-ից  $d_{emb}$  միջակայքում: Որտեղ  $d_{emb}$ -ը, embedding վեկտորի չափողականությունն է: Positional encoding շերտը նաև կարող է ուսուցանվող լինել:

p0	p1	p2	p3	
0.000	0.841	0.909	0.141	i=0
1.000	0.540	-0.416	-0.990	i=1
0.000	0.638	0.983	0.875	i=2
1.000	0.770	0.186	-0.484	i=3

Նկար 1: Positional Encoding-ը պատկերավոր ( $d_{emb} = 4$ )

Embedding շերտի միջոցով ստացված վեկտորները, գումարում ենք positional encoding վեկտորներին և ստանում ենք ավելի շատ ինֆորմացիա պարունակող embedding վեկտորներ: