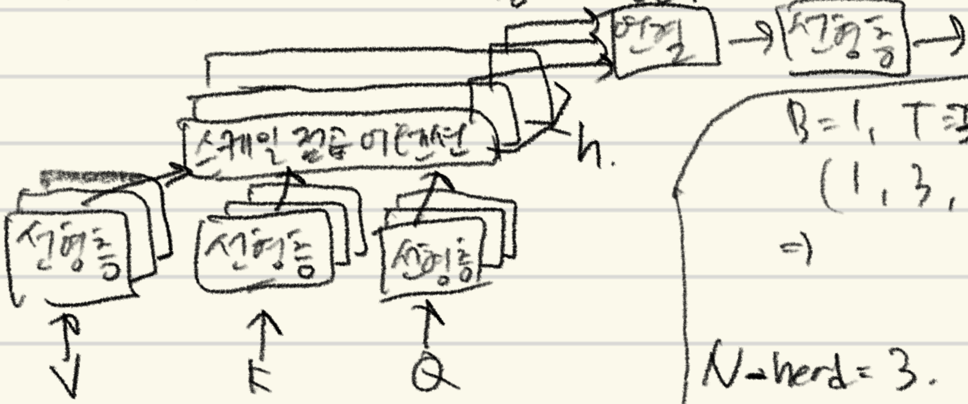
< 멀티헤드 어텐션 >

• 여러 어텐션 연산을 동시 적용 → 성능↑



• 개의 Attention Head 와 유사

① 쿼리, 키, 값을 n_head개로 쪼개고

② 각각의 어텐션 계산.

③ 입력과 같은 형태로 변환.

④ 선형층 통과 후 결과 반환.

코드 상 텐서변환 이해

B=1, T=3, C=3.

(1, 3, 3) → "나는" "배고파서" "햄버거를"

⇒ (0.2, 0.3, 0.4) (0.1, 0.7, 0.8) (0.3, 0.5, 0.6)

N_head = 3.

.View(B, T, N, C//N)

: X를 N의 헤드로 즉 각 부분할당 → 각 헤드를 독립적으로 다른 표현을 학습

각 토큰의 임베딩 벡터를 여러 개로 나눈 부분집합 중 하나를 의미

⇒ (0.2), (0.3), (0.4)

N이 3인 경우, 각 헤드.

⇒ (1, 3, 3, 1)

```
⇒ [
    [ # Token별
        [0, 2],
        [0, 3],
        [0, 4],
    ],
    [ ], [ ]
]
```

transpose(1, 2)

⇒ (1, 3, 3, 1)

```
⇒ [
    [ # 같은 위치의 헤드별
        [0, 2],
        [0, 1],
        [0, 3],
    ],
    [ ], [ ]
]
```