Position-Wise Feed Forward Network의 개념 정리

1. Position-wise Feed Forward Network는 각 토큰에 독립적으로 작용하는 함수열로, attention layer 이후 정보를 비선형적으로 확장하고 정제한다. 이 함수열은 레이어를 거치며 각 위치의 표현을 점별 수렴 형태로 깊이 있게 학습한다. Attention은 관계성만 학습하는 반면 Feed-Forward-Network(FFN)은 토큰 자체의 표현력을 강화해 최종적으로 softmax에 의해 수렴할 수 있는 표현 기반을 만든다. (반면, 균등수렴이 불필요한 이유는 토큰의 갯수가 유한하므로 각각의 토큰마다 어떤 값으로 수렴하면 된다.)

핵심 용어 정리

2. **Position-wise**는 각 토큰 위치마다 동일한 함수를 독립적으로 적용하는 방식이며, 이는 각 위치별 표현이 점점 이상적인 형태로 수렴하는 함수열의 점별수렴처럼 작동한다.
   Feed Forward는 attention 이후 각 레이어가 하나의 함수처럼 작용하며, 이들이 순차적으로 합성된 함수열 구조를 통해 점진적으로 복잡한 표현을 만들어간다.(함수열들이 합성해나가며 특정 함수로 수렴한다.)
   Non-linearity는 모델이 단순한 선형 계산을 넘어서 다양한 관계를 학습할 수 있게 해주며, 동시에 역전파 시 gradient vanishing 문제도 줄여준다. (PyTorch 사용 시, 오차역전파는 자동 미분을 통해 계산된다.)

동작 요약 스케치

3. Input → Attention → FFN(ReLU 포함) → Output(Also input) → 다음 Attention → FFN → ... → Final Output

“왜 Attention만 있으면 안 되고, FFN이 추가되어야할까?”

4. Attention은 문장 내 단어들 간의 상호작용, 즉 어떤 단어가 다른 단어를 얼마나 주목해야 하는지를 계산하는 역할을 수행합니다. 이때 단어 간의 관계는 주로 내적(dot product) 또는 코사인 유사도(cosine similarity)를 통해 측정되며, 이는 의미적으로 비슷한 단어일수록 더 강한 연관성을 가지도록 설계되어 있습니다. 하지만 이러한 연산은 단어 간의 관계에 집중할 뿐, 각 단어 자체의 의미 표현을 충분히 확장하거나 정제하지는 못합니다. 따라서 FFN이 도입되어, 각 단어 위치마다 동일한 함수를 반복 적용하는 함수열처럼 작동함으로써 개별 표현을 점진적으로 강화하고, 모델의 표현력을 향상시키는데 기여합니다.