AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

Page1.

안녕하세요. 이어서 발표를 진행하게될 김동휘입니다. 먼저 남은 3개의 식을 설명하기에 앞서서 트렌스포머에 사용되는 attention의 원리에 대해 설명하고 multi-head self-attention에 대해 설명하면서 같이 수식에 대해 같이 설명 드리겠습니다.

attention은 우선 쿼리, 키, 벨류로 구성됩니다. 쿼리는와 벨류의 값사이의 내적을 통해 두 값 사이의 유사도를 가중치로 바꾸고 이 가중치 내적값을 Softmax를 통해 정규화하고, 모든 가중치의 합이 1이 되도록 만들어 입력 벡터 간의 중요도를 나타내는 확률 분포로 바꿉니다. 이후 Value벡터의를 가중합을 함으로써 유사도가 높을수록 정보가 많이 포함되는 값이 attention의 결과로 출력되게 됩니다.

Self-attention은 이때 각 Key값 사이의 주변 맥락정보에 대해 학습하기 위해 사용되는데 attention의 경우 단순히 쿼리와 키 사이의 내적으로 유사도를 가중치화하여 벨류 벡터를 계산하는데 이는 키의 순서가 어떻게 되든 유사도는 동일하기 때문에 순서와 상관없이 동일한 출력값을 내보내게 됩니다. 또한 주변에 어떤 키가 있는지에 대한 정보는 단순한 attention 방식으로는 학습하기 어렵습니다. 처음 Embedding시 입력해준 position정보를 사용하여 각 패치 키의 위치정보를 학습하고 Query와 Key를 동일한 값을 사용하는 self-attention 방식을 사용하여 입력된 각 패치가 다른 모든 패치와의 관계를 계산하게합니다. 만약 Z0의 1번패치의 Key값에 대한 value vector를 Self attention하여 구한다면 결과로 나온 패치1번의 Z1 값은 전체 패치의 가중치합하여 전체 이미지의 정보를 유사한 만큼 담고있는 패치가 되게된다. 이를 통해 각 패치 사이의 상관관계를 파악하고 학습할 수 있습니다.

이때 여기 self attention의 식을 보시면 입력값 Z는 NxD의 형태를 보이는데 q, k, v는 Dh의 차원을 가지게 됩니다. 뒤의 멀티헤드 어텐션에서 다루겠지만 Dh는 헤드의 겟수로 나눈 차원이라 만약 D의 차원이 50이었고 헤드의 겟수가 5개라면 Dh의 차원은 10이 될 것입니다. 각각의 Q, K, V는 self-attention이기 때문에 동일한 차원을 가지게되고 따라서 3개의 행렬을 합치면 3Dh의 차원을 가지게 됩니다. Uqkv는 입력 Z를 q, k, v로 나누어주는 가중치 행렬이기 떄문에 Dx3Dh의 형태를 가지게 되고 Uqkv를 z와 행렬곱셈을 하여 Q, K, V 벡터로 나눠주고 아래의 경우 아까 설명한거와 같이 Q와 K를 통해 attention score를 만들어주고 루트 Dh로 나누어주는데 이는 둘사이의 내적값결과가 어느하나가 너무큰경우 다른 정보가 거의 반영되지 않는 문제를 해결하기위해 나누어주어Smooding 기법을 적용한 것으로 볼 수 있고 Softmax를 적용하여 정규화하여 v와 행렬곱셈을 하여 V의 각 벡터에 A의 가중합을 수행하여 어텐션의 결과를 도출하게된다.

이제 다음으로 ViT의 Transfoemer Encoder에서 주요하게 사용되는 Multi head attention에 대해서 설명하자면 Multi head attention은 입력 벡터 Z를 K개의 헤드로 나누어 병렬로 Self-Attention을 수행하고, 그 결과를 결합하여 최종 출력을 생성합니다. 이를 통해 입력 데이터의 다양한 관계를 학습하고 풍부한 표현을 생성할 수 있게 합니다. 그리고 만들어진 Dh차원의 attention 결과를 모두 concatenate하여 N x (k x Dh)의 행렬로 합치고 여기에 k x Dh X D의 가중치 행렬 U\_msa을 통해 선형변환하여 처음 입력했던 차원 D로 변환합니다.

이제 앞에 식으로 돌아와서 식 2번의 경우 이제 위에 설명한 multi head attention을 포함하여 전체 Transformer의 encoder블록을 L번 반복하게 되는데 Zl-1을 입력으로하여 Multi head attention의 출력값 Z’l에 대해서 나타낸 식입니다. 식을 보시면 Zl-1번째 encoder의 출력값을 다시 encoder의 입력값으로 입력하여 layer normalization을 해주고 multi head attention 과정을 거치고 마지막으로 Residual connection을 더해주기 위해 Zl-1을 더해주면 multi head attention 구간의 결과값인 Z’l이 되고 이 결과를 MLP의 입력으로하여 결과적으로 출력되는 encoder의 값을 다시 입력으로 넣는 과정을 L번 반복하게 됩니다.

다음 3번째 공식은 encoder의 Multilayer perceptron부분인데 Multi head attention의 결과로 나온 Z’l을 마찬가지로 Layer normalization하고 이를 두 개의 레이어와 GELU 활성함수로 구성된 다층 퍼셉트론에 입력하여 self-attention결과의 복잡한 패턴을 학습하도록 합니다. 이후 MLP 출력값에 Residual connection으로 이전 출력값을 더해 최종 적으로 Z\_l을 출력하게 됩니다.

이렇게 마지막에 출력된 z\_l에서 class 토큰의 최종 상태인 z0\_L에 Layer Normalization(LN)이 적용되어 최종 출력 표현 y를 생성하는걸 나타낸 식이다. 이렇게 생성된 y는 ViT에서 이미지의 **전체 표현**을 나타내며, 분류 작업을 수행할 때 **분류 헤드**에 입력되어 이미지가 어떤 클래스에 속하는지 이미지 클래스 에측 수행에 사용하게 됩니다. Fine tunnig 시에는 위의 MLP Head가 아닌 Single linear layer를 통과하도록 한다.

다음으로 Inductive bias를 CNN과 비교했을 때 vision transformer가 가지는 특징에 대해서 설명하고 논문이 제안한 CNN과 transformer의 하이브리드 아키텍처를 설명하겠습니다.

**Q & A**

**Pre-Layer Normalization을 하는이유는?**

ViT에서 사용하는 사전 정규화(pre-layer normalization) 방식은 Self-Attention과 Feed-Forward Network를 통과하기 전에 정규화를 수행하여, 그 뒤의 계산이 더 균일하게 이루어질 수 있도록 합니다. 이는 학습 시 각 레이어의 출력을 안정적으로 유지해, 더 깊은 네트워크에서도 안정적인 학습을 가능하게 합니다.

**왜 Layer Norm을 사용하는가?**

Layer Normaliztion은 RNN에서 사용하는 정규화 기법 중 하나이다.

Batch Norm은 각 Feature 들을 Normalization을 한다면, Layer Norm은 배치마다 시행한다.

즉 동일 배치에 있는 시점의 시퀀스들을 모아서 정규화 하는 것이다. 시계열 데이터는 시점마다 다른 통계치가 존재하기 때문에 첫번째 시퀀스들끼리, 두번째 시퀀스들끼리 등으로 하는 것이 효율적이라는 것이다. Transformer는 자연어 처리를 기반으로 만들어진것이고, VIT 또한 이에 맞춰 Patch 시퀀스를 만들었기 때문에 당연한 정규화 과정이라고도 할 수 있을 것 같다.

ReLU대신 GELU를 사용하는 이유?

**ReLU**는 불연속적인 활성화 함수라 역전파 시 미분값이 불연속적인 구간이 있지만, **GELU**는 부드러운 연속성을 가져 학습 중 그래디언트 전파가 더욱 안정적입니다.

**ReLU**는 단순하고 계산이 빠르며, 입력이 양수일 때만 활성화되므로 뉴런의 일부가 죽는 현상("dying ReLU") 문제가 발생할 수 있습니다. 반면, **GELU**는 음수 입력도 일부 활성화되기 때문에 이러한 문제를 완화합니다.