1. Residual Connection (잔차 연결)

▌1-1. 왜 **ResNet**을 사용하게 되었는가**?**

문제점	설명
딥한 **CNN** 구조의 성능 한계	GoogleNet, VGG 등 깊은 CNN은 오히려 학습 성능이 나빠지는 현상 발생 (기울기 소실, 과적합 등)
병렬적 학습 불가	기존 CNN은 순차적으로 학습해야 하며, 깊어질수록 학습의 효율성이 떨어짐
Shallow 모델보다 성능 저하	깊은 모델이 일반화 성능이 떨어져, 얕은 네트워크보다 성능이 낮아지는 경우 발생

➡ **ResNet**은 입력을 그대로 전달하는 경로를 추가하여 학습 경로를 단축하고, **Convolution** 연산의 병렬적 처리 성격과 결합하여 학습을 빠르고 안정적으로 만들 수 있게 함.

▌1-2. **Residual** 구조의 핵심 원리

- 입력을 그대로 다음 블록에 더해줌으로써, 정보가 사라지지 않고 흐르게 함
- 학습 목표는 전체 출력이 아니라 잔차만을 학습
- 잔차차가 작고 국소적인 변화만 반영하면 되므로 학습이 간단하고 빠름
- 이 덕분에 병렬화에 유리한 **Convolution** 연산 구조를 효과적으로 사용할 수 있음

Layer Normalization (레이어 정규화)

항목	내용
정의	각 레이어의 출력을 feature 차원별로 정규화하여 학습 안정화
방식	Batch 단위가 아닌 각 **sample** 내의 **feature**에 대해 평균·표준편차를 기준으로 정규화
공식	$\mathrm{LayerNorm}(x) = \gamma \cdot \frac{x - \mu}{\sigma} + \beta$

✅ 수식 구성 요소 설명

기호	의미
x	입력 벡터 (예: 하나의 토큰의 임베딩 벡터)
μ	입력 x의 **평균** (feature 차원 기준)
σ	입력 x의 **표준편차** (feature 차원 기준)
$\frac{x-\mu}{\sigma}$	정규화된 입력 (평균 0, 표준편차 1로 변환)
γ	정규화된 입력에 곱하는 **스케일(확장)** 파라미터 (학습 가능)
β	정규화된 입력에 더하는 **시프트(이동)** 파라미터 (학습 가능)

✅ 요약 정리

- μ, σ:
  👉 입력 벡터의 분포를 정규화하기 위한 **통계량** (계산 결과, 학습되지 않음)
- γ, β:
  👉 정규화 후에도 모델이 **표현력을 잃지 않도록 학습하는 보정 계수**
  (즉, 모델이 "필요하다면 다시 스케일을 조절"할 수 있게 함)

🎯 한 줄 요약

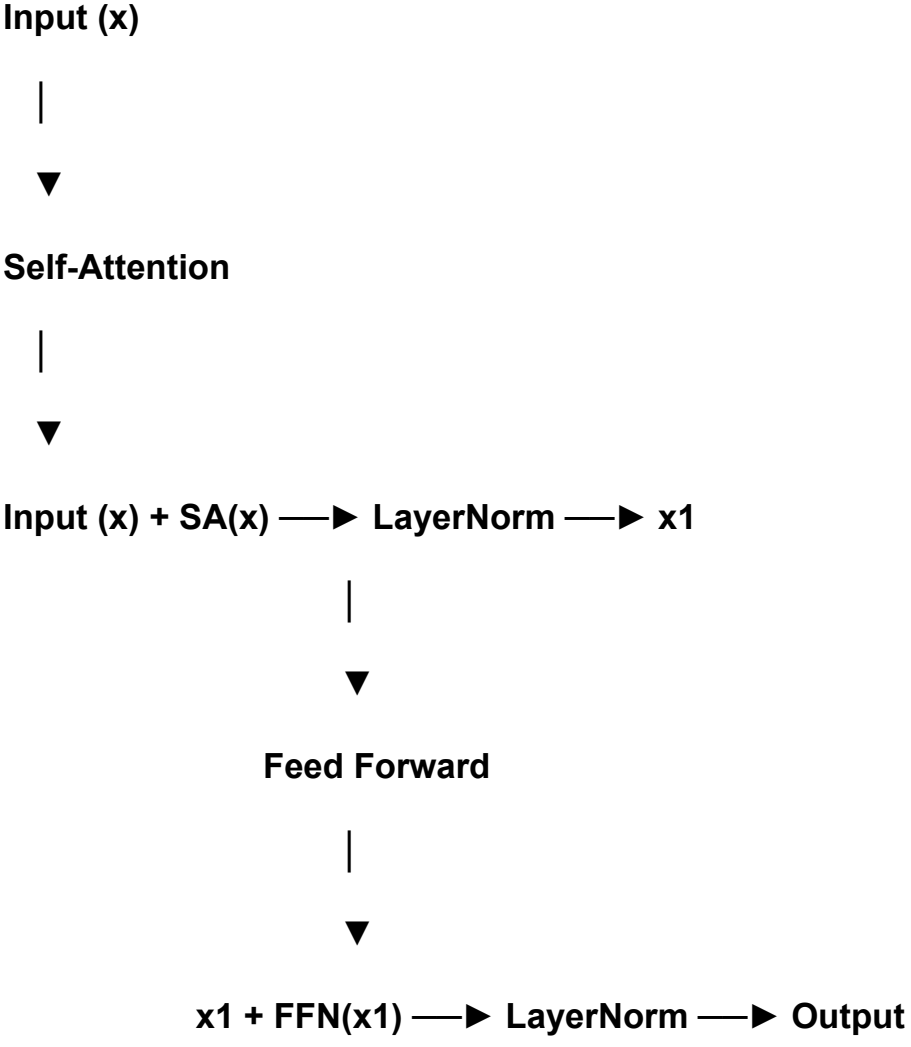
> γ는 정규화된 입력의 **스케일 조정**, β는 **값 이동 조정**을 위해 학습되는 파라미터이며, 정규화로 인해 사라질 수 있는 표현력을 보존합니다.

2. **Transformer**에서의 구조적 적용 방식

위치	구조
Self-Attention 이후	LayerNorm(x+SelfAttention(x))
FFN 이후	LayerNorm(x+FFN(x))

Transformer는 ResNet에서 착안한 **Residual Connection +** 정규화 구조를 모든 서브 레이어에 적용하여 학습 안정성과 정보 흐름을 개선함.

3. **Transformer**에서의 구조적 적용 방식 시각화



Transformer는 ResNet에서 착안한 **Residual Connection +** 정규화 구조를 모든 서브 레이어에 적용하여 학습 안정성과 정보 흐름을 개선함.

4. 해볼만한 질문들

1. 왜 깊은 신경망이 성능이 오히려 떨어지는 경우가 발생할까**?**

→ 깊은 신경망은 기울기 소실·폭주로 학습이 어려워지고, 입력 정보도 점차 사라진다. 파라미터 수가 많아지며 과적합 위험도 커지고, 최적화가 복잡해진다. Residual Connection은 입력을 우회 전달해 이러한 문제를 완화하고 학습을 안정화한다.

2. Residual Connection이 정보의 손실을 줄이는 이유는**?**

→ Residual Connection은 입력을 출력에 직접 더함으로써, 정보가 중간 연산에서 사라지지 않고 깊은 네트워크 전체에 안정적으로 전달되게 합니다.

3. 왜 Transformer는 Residual Connection + 정규화를 모든 서브 레이어에 적용했는가**?**

→ Transformer는 ResNet에서의 경험을 반영하여, 깊은 구조에서도 정보와 기울기가 잘 흐르게 만들기 위해 Residual Connection을 도입했고, 각 **Sublayer** 출력의 수렴 안정성과 스케일 보정을 위해 Layer Normalization을 결합했습니다. 이 구조 덕분에 수백, 수천 층의 모델도 비교적 안정적으로 학습이 가능해진 것입니다.

Reference

1. “Vaswani et al., 2017 (Transformer)”
2. https://velog.io/@xuio/Transformer-%EB%85%BC%EB%AC%B8-%EB%A6%AC%EB%B7%B0-%EC%A0%84-%ED%94%84%EB%A6%AC%EB%B7%B02Positional-Encoding%EA%B3%BC-Residual-Connection