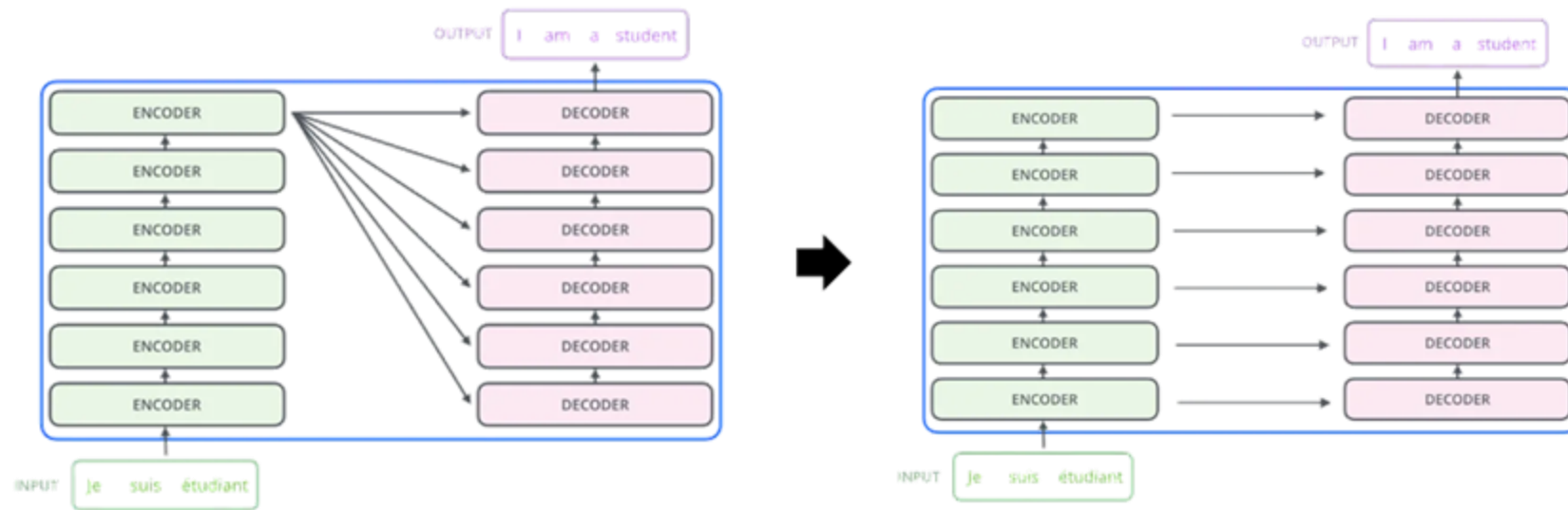


## 8주차: P-Transformer 모델 구현

- 목적: Transformer 모델을 원하는 방식으로 변형해서 쓸 수 있도록 구현 연습
- 내용:
  - Original Transformer를 P-Transformer로 변형



(왼쪽: Original Transformer; 오른쪽: Parallel Transformer (P-Transformer))

- 5주차와 동일한 데이터셋에 대해서도 BLEU-4 Score 0.15 이상 목표

# P-Transformer는 왜 할까?

- 초반 encoder : 저수준 특징 (단어 형태, 문법 패턴)
- 후반 encoder : 문맥 의미, 복잡한 의미 관계
- 유연하게 각 encoder의 정보를 전달해주면 어떨까?

# 추가로 구현할 것

- 각 encoder layer의 output을 저장해주는 리스트
- Decoder에서 `enc_outputs[i]`를 memory로 전달함

# 바뀐 부분

## Encoder 클래스 내부

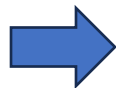
```
def forward(self, x, mask):  
    for layer in self.layers:  
        x = layer(x, mask)  
    return self.norm(x)
```



```
def forward(self, x, mask):  
    ## 수정사항  
    outputs = []  
    for layer in self.layers:  
        x = layer(x, mask)  
        outputs.append(x)  
    outputs[-1] = self.norm(outputs[-1])  
    return outputs
```

## Decoder 클래스 내부

```
def forward(self, x, memory, source_mask, target_mask):  
    for layer in self.layers:  
        x = layer(x, memory, source_mask, target_mask)  
    return self.norm(x)
```



```
def forward(self, x, enc_outputs, source_mask, target_mask):  
    for i, layer in enumerate(self.layers):  
        x = layer(x, enc_outputs[i], source_mask, target_mask)  
    return self.norm(x)
```

# 이런

```
Epoch 1: Train Loss: 3.9808, Valid Loss: 2.8744  
Epoch 2: Train Loss: 2.7259, Valid Loss: 2.2408  
Epoch 3: Train Loss: 2.2438, Valid Loss: 1.9433  
Epoch 4: Train Loss: 1.9555, Valid Loss: 1.7825  
Epoch 5: Train Loss: 1.7626, Valid Loss: 1.7129  
Epoch 6: Train Loss: 1.6181, Valid Loss: 1.6293  
Epoch 7: Train Loss: 1.4995, Valid Loss: 1.6058  
Epoch 8: Train Loss: 1.4004, Valid Loss: 1.5907  
Epoch 9: Train Loss: 1.3181, Valid Loss: 1.6232  
Epoch 10: Train Loss: 1.2524, Valid Loss: 1.6122
```

성능 (beam search of 3)

```
Using device: mps  
Test BLEU score: 0.3615
```

# 성능 (Original Transformer와 비교)

```
Using device: mps  
Test BLEU score: 0.3758
```

# Greedy decoding이라면..?

```
Using device: mps  
Test BLEU score: 0.3631
```

Original transformer

```
Using device: mps  
Test BLEU score: 0.3496
```

Parallel transformer



# 이유 분석 (Original Transformer가 다음)

- 각 층의 출력을 개별적으로 사용하다보니, 노이즈가 발생
- decoder가 정제된/정제되지 않은 다양한 표현을 받다보니 최적화가 복잡해졌을수도 있다.
- 데이터셋 multi30k가 너무 쉬워서 Parallel Transformer에서의 복잡성이 필요없을 것 같다.