IIT4316 Deep Learning

Homework #2: 두 자리수 덧셈기 (2025 년 11 월 23 일)

마감: 2025 년 12 월 6 일 23 시 59 분까지

(늦은 제출은 12 월 8 일 오전 8 시 59 분까지만 허용)

CNN 과 Transformer 구조를 이용하여 두 자리수 두 개를 더하는 덧셈기를 구현한다. 예를 들어 72+46=118 의 경우, 입력을 [7, 2, 4, 6]로 주면 출력은 [1, 1, 8]로 얻고자 한다. 또 다른 예로, 입력이 [1, 1, 2, 2]이면 출력은 [0, 3, 3]이 되도록 한다.

입력받는 각 숫자는 0 부터 9 까지이므로 vocabulary size 는 10 이다.(VOCAB_SIZE=10) 각 숫자는 10 차원 one-hot vector 로 변환하여 입력으로 사용한다. 즉, '0'은 [1,0,0,0,0,0,0,0,0,0]로, '6'은 [0,0,0,0,0,0,1,0,0,0]로 변환한다.

CNN 은 다음과 같이 구성된다. MyCNN 클래스를 완성하여 사용한다.

- **Embedding layer**: 주어진 입력을 one-hot 으로 변환하고, 여기에 가중치를 곱해서 embedding 으로 변환한다. embedding 의 차원을 EMBED_DIM 이라 하면, 이 layer 의 출력 크기는 4*EMBED_DIM 이다. MyOneHot 함수와 MyEmbedding 클래스를 완성하여 사용한다.
- **N 번 반복**:
    - **Conv layer**: embedding layer 의 출력을 마치 이미지처럼 3 차원(2 x 2 x EMBED_DIM)으로 reshape 하여 입력으로 사용한다. 필터 크기는 3x3, padding=1, stride=1 을 사용한다. 출력 채널은 NUM_CHANNELS 로 설정한다. MyConv2D 클래스를 완성하여 사용한다.
    - **ReLU**: 구현되어 있는 my_relu 함수를 사용한다.
    - 추가로 skip connection 을 활용할 수도 있다.
- **Linear layer**: conv layer 의 3 차원 출력을 벡터로 만들어 입력으로 받는다. 출력 갯수는 3*VOCAB_SIZE 이다. MyLinear 클래스를 완성하여 사용한다.

편의상 모든 conv layer 의 출력은 모두 같은 수의 channel 을 갖는 것으로 한다.
3*VOCAB_SIZE 개의 출력에서, 각 VOCAB_SIZE 개의 출력 중 최대값의 위치를 찾아 예측 결과를 얻는다.

Transformer 는 decoder 가 없는 encoder-only 모델이다. MyTransformer 클래스를 완성하여 사용한다.

- **Embedding layer**: CNN 의 embedding layer 와 같다.
- **Encoder layer (N 번 반복):**
    - Self-attention: linear layer 를 이용하여 입력을 각각 query, key, value 로 변환한다. 셋 다 입력 임베딩과 같은 차원(EMBED_DIM)이 되도록 한다. scaled dot-product attention 을 사용하되, head 는 하나만 사용한다. 최종적으로 출력에 입력을 더해 skip connection 을 구현한다. MySelfAttention 클래스를 완성하여 사용한다.
    - MLP: 2-layer MLP 를 사용한다. hidden layer 에는 ReLU activation function 을 사용하며, output layer 에는 nonlinear activation 을 사용하지 않는다. embedding 의 차원을 유지하기 위해 MLP 의 출력은 입력과 같은 차원(=EMBED_DIM)을 갖는다. 최종적으로 출력에 입력을 더해 skip connection 을 구현한다. MyLinear 클래스를 완성하여 사용한다.
- **Linear layer**: CNN 의 linear layer 와 같다.

loss 는 cross-entropy 를 사용하며, 학습은 Adam 을 사용한다. learning rate 와 최대 epoch 은 적절히 정하라. 학습 후 100 개의 무작위 테스트를 하여 성공률을 측정한다. (학습과 테스트를 위한 코드는 이미 구현되어 있다.)

주어진 .ipynb 에서 TODO 로 표시한 부분을 작성하여 코드를 완성하라. 해당 연산을 직접 구현하여야 한다. (예를 들어, convolution 과정, self-attention 과정을 이미 구현된 함수를 사용하지 말고 직접 구현한다.) 단, 행렬의 transpose, 행렬의 차원의 순서나 모양을 바꾸는 것, 영행렬 생성 등 기본적인 함수는 사용 가능하다.

필요하다면 TODO 로 표기된 부분 이외에 추가로 작성하여도 무방하다.

(1) 다음의 hyperparameter 에 따른 성능을 비교 관찰한다.(가능한 모든 조합을 다 해 볼 필요는 없다.)

- embedding dimension
- (CNN) Conv(+ReLU) layer 의 수, channel size
- (Transformer) encoder layer 의 수, MLP hidden neuron 갯수

(2) CNN 과 Transformer 를 여러 관점에서 비교하라. 어느 구조가 두 자리 수 덧셈 문제에 더 적합하다고 보는가?

제출항목

- 리포트 (pdf 파일)
- 코드 (ipynb 파일)

리포트 작성 요령

- 우리말로 작성하기
- 표지 만들지 않기 (첫 장 상단에 학번과 이름 표기)
- 완성된 문장으로 작성하기
- 그림과 표가 포함되는 경우, 이를 본문에서 설명하기
- 토의한 사람이 있다면 명시하기