GAI Project 2.a

顏呈育 F64096106

1. Load the data and split them into train and validation (10 pts). You can use **pandas**, **Dataset** and **Dataloader**.

```
df = pd.read_csv('/content/gdrive/MyDrive/colab_input/data.csv')
from sklearn.model_selection import train_test_split
(train_x_split, valid_x_split,train_y_split, valid_y_split) = train_test_split(
    df['src'],
    df['tgt'],
    train_size=0.8,
    random_state=64096106
)
df.head(5)
```

2. Generate your own version of data (10 pts), At least three two-digit addition and subtraction problems, and if you'd like, you can also try multiplication and division.

	ques	answ
0	27-3-24=	0
1	14+6+1=	21
2	6+12-7=	11
3	15+17-22=	10
4	21*6-6=	120

我嘗試生成包含加、減、乘的表達式，並且與給定的data一樣，限制生成的三個數字都在30以下，一開始忘記考慮先乘除後加減，後來考慮之後accuracy才提升上來

3. Tokenize the text (10 pts). You can design your own tokenizer or use any API (if available).

[71]
```
char_to_id = {}
id_to_char = {}

char_to_id['<pad>'] = 0
char_to_id['<eos>'] = 1
id_to_char[0] = '<pad>'
id_to_char[1] = '<eos>'

for char in set(df['src'].str.cat()):
    ch_id = len(char_to_id)
    char_to_id[char] = ch_id
    id_to_char[ch_id] = char

vocab_size = len(char_to_id)
print('字典大小: {}'.format(vocab_size))
char_to_id
```

```
字典大小: 16
{'<pad>': 0,
 '<eos>': 1,
 '2': 2,
 '0': 3,
 '8': 4,
 '-': 5,
 '6': 6,
 '7': 7,
 '=': 8,
 '5': 9,
 '4': 10,
 '1': 11,
 '3': 12,
 '9': 13,
 '*': 14,
 '+': 15}
```

2. Generation Models (30 pts)

    1. Model design (10 pts). You can use RNN, GRU or LSTM... whatever. (**at least one sequential model**)

```python
class CharRNN(torch.nn.Module):
    def __init__(self, vocab_size, embed_dim, hidden_dim):
        super(CharRNN, self).__init__()

        # Embedding層
        self.embedding = torch.nn.Embedding(num_embeddings=vocab_size,
                                            embedding_dim=embed_dim,
                                            padding_idx=char_to_id['<pad>'])
        # RNN層
        self.rnn_layer1 = torch.nn.LSTM(input_size=embed_dim,
                                        hidden_size=hidden_dim,
                                        batch_first=True)
        #self.rnn_layer2 = torch.nn.LSTM(input_size=hidden_dim,
        #                                hidden_size=hidden_dim,
        #                                batch_first=True)
        # output層
        self.linear = torch.nn.Sequential(torch.nn.Linear(in_features=hidden_dim,
                                                          out_features=hidden_dim),
                                          torch.nn.ReLU(),
                                          torch.nn.Linear(in_features=hidden_dim,
                                                          out_features=vocab_size))

```

2. Train(finetune) the model (10 pts).
3. Evaluate your model when you are training. (10 pts)

```
QA =  4+4*2= 12 , Prediction =  4+4*2=12
QA =  18-8*22= -158 , Prediction =  18-8*22=-158
QA =  2-15+25= 12 , Prediction =  2-15+25=12
QA =  18-29+17= 6 , Prediction =  18-29+17=6
QA =  12+22*2= 56 , Prediction =  12+22*2=56
QA =  22-3-17= 2 , Prediction =  22-3-17=2
QA =  2-13-19= -30 , Prediction =  2-13-19=-30
QA =  2*9+10= 28 , Prediction =  2*9+10=28
QA =  20+10-2= 28 , Prediction =  20+10-2=28
QA =  0+14-16= -2 , Prediction =  0+14-16=-2
QA =  1-8+4= -3 , Prediction =  1-8+4=-3
QA =  5-5+14= 14 , Prediction =  5-5+14=14
QA =  8-20*17= -332 , Prediction =  8-20*17=-332
QA =  25-11+1= 15 , Prediction =  25-11+1=15
QA =  12+6-20= -2 , Prediction =  12+6-20=-2
QA =  22-29*5= -123 , Prediction =  22-29*5=-123
QA =  0-10-13= -23 , Prediction =  0-10-13=-23
accuray=0.7
epoch 195: 100%|          | 80/80 [00:01<00:00, 48.06it/s, loss=0.236]
 49%|     | 49/100 [00:00<00:00, 241.76it/s]QA =  21+21+12= 54 , Prediction =  21+21+12=54
QA =  2-15-10= -23 , Prediction =  2-15-10=-23
QA =  15+15-6= 24 , Prediction =  15+15-6=24
QA =  15+15-25= 5 , Prediction =  15+15-25=5
QA =  7*16-16= 96 , Prediction =  7*16-16=96
QA =  25-25*24= -575 , Prediction =  25-25*24=-575
QA =  2*6+1= 13 , Prediction =  2*6+1=13
QA =  25-28-28= -31 , Prediction =  25-28-28=-31
QA =  3+15+2= 20 , Prediction =  3+15+2=20
QA =  23+26+6= 55 , Prediction =  23+26+6=55
QA =  8-22-16= -30 , Prediction =  8-22-16=-30
QA =  29-9+24= 44 , Prediction =  29-9+24=44
QA =  23+9-20= 12 , Prediction =  23+9-20=12
```

```
accuray=0.54
epoch 992: 100%|          | 200/200 [00:02<00:00, 86.48it/s, loss=0.00138]
100%|          | 100/100 [00:00<00:00, 297.52it/s]
accuray=0.64
epoch 993: 100%|          | 200/200 [00:03<00:00, 65.58it/s, loss=0.0011]
100%|          | 100/100 [00:00<00:00, 252.34it/s]
accuray=0.61
epoch 994: 100%|          | 200/200 [00:02<00:00, 83.15it/s, loss=0.00119]
100%|          | 100/100 [00:00<00:00, 365.05it/s]
accuray=0.62
epoch 995: 100%|          | 200/200 [00:02<00:00, 87.46it/s, loss=0.000892]
100%|          | 100/100 [00:00<00:00, 365.96it/s]
accuray=0.59
epoch 996: 100%|          | 200/200 [00:02<00:00, 86.69it/s, loss=0.00134]
100%|          | 100/100 [00:00<00:00, 380.67it/s]
accuray=0.68
epoch 997: 100%|          | 200/200 [00:02<00:00, 86.80it/s, loss=0.00101]
100%|          | 100/100 [00:00<00:00, 306.55it/s]
accuray=0.62
epoch 998: 100%|          | 200/200 [00:03<00:00, 64.62it/s, loss=0.00117]
100%|          | 100/100 [00:00<00:00, 281.92it/s]
accuray=0.71
epoch 999: 100%|          | 200/200 [00:02<00:00, 83.40it/s, loss=0.703]
100%|          | 100/100 [00:00<00:00, 371.70it/s]
accuray=0.48
epoch 1000: 100%|          | 200/200 [00:02<00:00, 86.26it/s, loss=0.0465]
100%|          | 100/100 [00:00<00:00, 370.27it/s]accuray=0.57
```

3. Analysis (40 pts)

    1. Model analysis (10 pts)
       Include your model design and the process of loss reduction and the results of validation and testing (if available).
    2. Dataset analysis(15 pts)
       What are the characteristics of the datasets you have generated, and what is your method or understanding of it? What adjustments did you make to the dataset? How did they impact your results? (At least 2 variations, 5pts per variation)
    3. Discussion (15 pts)
       During the training process, what impact does different learning rates have? What impact does different batch sizes have? What are the characteristics of the model you are using, and why do you think it is suitable(not suitable) for this task?

1. 模型分析

- Model design：
    - 模型包括一層LSTM layer和一層FNN layer（兩層LSTM layers的效果較不如一層LSTM layer）。而Embedding layer用於將char序列轉換為嵌入向量。兩個LSTM layers分別處理嵌入向量序列。最透過FNN layer將LSTM layers的輸出映射到詞彙空間，以獲得對下一個char的prediction。
    - 執行時torch的tensor搬到Colab提供之CUDA上執行，提高速度
- Loss reduction：
    - 使用了cross entropy來衡量模型預測與真實值之間的差距。使用了Adam來更新模型參數，並使用clip_grad來防止梯度爆炸。
- Validation：
    - 在每個epoch結束後進行validate，輸入隨機生成的ques，並將模型的預測與answ進行比較。如果模型的預測完全匹配真實答案，則記為一次成功。

3. Analysis (40 pts)

    1. Model analysis (10 pts)
       Include your model design and the process of loss reduction and the results of validation and testing (if available).
    2. Dataset analysis(15 pts)
       What are the characteristics of the datasets you have generated, and what is your method or understanding of it? What adjustments did you make to the dataset? How did they impact your results? (At least 2 variations, 5pts per variation)
    3. Discussion (15 pts)
       During the training process, what impact does different learning rates have? What impact does different batch sizes have? What are the characteristics of the model you are using, and why do you think it is suitable(not suitable) for this task?

2. 資料集分析

- 資料集特點:
  - 資料集包含ques和對應的answ。每個ques由三個隨機生成的二位數字和兩個隨機operator構成。
- 資料集調整:
  - 使用digit_gen()函數生成ques和answ。數字的範圍原本使用範圍0~99，後來發現會導致預測結果不好，於是更改為與dataGen裡相同的0~29。
- 調整對結果的影響:
  - 使用自定函數生成的ques和answ，使模型學習到更多不同形式的算式。
  - 一開始忘記考慮先乘除後加減，後來考慮之後accuracy才提升上來
  - 將數字的範圍設置為0~30，增加數據的多樣性，有助於提高模型的泛化能力。

3. Analysis (40 pts)

    1. Model analysis (10 pts)
       Include your model design and the process of loss reduction and the results of validation and testing (if available).
    2. Dataset analysis(15 pts)
       What are the characteristics of the datasets you have generated, and what is your method or understanding of it? What adjustments did you make to the dataset? How did they impact your results? (At least 2 variations, 5pts per variation)
    3. Discussion (15 pts)
       During the training process, what impact does different learning rates have? What impact does different batch sizes have? What are the characteristics of the model you are using, and why do you think it is suitable(not suitable) for this task?

3. 討論

- 不同learning rates的影響：
  - 在採用較高的lr（如lr = 0.001）時可以提高loss的收斂速度，但是到一個固定值的時候會無法再繼續減少
  - 採用較低的lr（如lr = 0.0001）會導致訓練過程比較慢，但更容易達到較好的收斂效果
- 不同Batch sizes的影響：
  - 較大的Batch sizes可以提高訓練速度，但loss有時比較無法收斂
  - 較小的Batch sizes可能會增加訓練時間，但比Batch sizes大時收斂的效果更好
- 模型特點、適用性：
  - model使用LSTM結構，適用於處理數學問題生成這樣的GAI Task。
  - 通過Embedding layer將字符序列轉換為密集向量表示，可以使model「理解」字符之間的關係。