BÁO CÁO TUẦN 11

Người báo cáo: Trần Đức Thọ.

Tuần này em đã chuyển source code của mình lên colab để chạy cho nhẹ

Bộ data Anh-Việt : <https://github.com/stefan-it/nmt-en-vi>

Sau khi tải về và giải nén em thu được 3 tập tiếng anh, 3 tập tiếng việt : train val và test

|  |  |
| --- | --- |
| DATA | SỐ CÂU |
| TRAIN | 133317 |
| VAL | 1553 |
| TEST | 1268 |

Em để tập tiếng anh là tập nguồn và tiếng việt là tập đích:

Text

Description automatically generated

Đầu tiên em cài đặt open-nmt trên colab

!pip install --upgrade pip

!pip install OpenNMT-py

Import các thư viện cần thiết:

import yaml

import torch

import torch.nn as nn

from argparse import Namespace

from collections import defaultdict, Counter

import onmt

from onmt.inputters.inputter import \_load\_vocab, \_build\_fields\_vocab, get\_fields, IterOnDevice

from onmt.inputters.corpus import ParallelCorpus

from onmt.inputters.dynamic\_iterator import DynamicDatasetIter

from onmt.translate import GNMTGlobalScorer, Translator, TranslationBuilder

from onmt.utils.misc import set\_random\_seed

em tạo 1 file yaml để config tập train và tập val:

yaml\_config = """

## Where the samples will be written

save\_data: /content/drive/MyDrive/máy dịch/toy-ende/run/example

## Where the vocab(s) will be written

src\_vocab: /content/drive/MyDrive/máy dịch/toy-ende/run/example.vocab.src

tgt\_vocab: /content/drive/MyDrive/máy dịch/toy-ende/run/example.vocab.tgt

# Corpus opts:

data:

    corpus:

        path\_src: /content/drive/MyDrive/máy dịch/toy-ende/src-train.txt

        path\_tgt: /content/drive/MyDrive/máy dịch/toy-ende/tgt-train.txt

        transforms: []

        weight: 1

    valid:

        path\_src: /content/drive/MyDrive/máy dịch/toy-ende/src-val.txt

        path\_tgt: /content/drive/MyDrive/máy dịch/toy-ende/tgt-val.txt

        transforms: []

"""

config = yaml.safe\_load(yaml\_config)

with open("/content/drive/MyDrive/máy dịch/test.yaml", "w") as f:

    f.write(yaml\_config)

tiếp theo em lấy data vào và cho chạy, trong ví dụ thì người ta chỉ dùng 1000 step train và 500 step val, em thử nâng lên thêm 10 lần : 10000 step train và 5000 step val.(mo hình sử dụng là LSTM)

emb\_size = 100

rnn\_size = 500

# Specify the core model.

encoder\_embeddings = onmt.modules.Embeddings(emb\_size, len(src\_vocab),

                                             word\_padding\_idx=src\_padding)

encoder = onmt.encoders.RNNEncoder(hidden\_size=rnn\_size, num\_layers=1,

                                   rnn\_type="LSTM", bidirectional=True,

                                   embeddings=encoder\_embeddings)

decoder\_embeddings = onmt.modules.Embeddings(emb\_size, len(tgt\_vocab),

                                             word\_padding\_idx=tgt\_padding)

decoder = onmt.decoders.decoder.InputFeedRNNDecoder(

    hidden\_size=rnn\_size, num\_layers=1, bidirectional\_encoder=True,

    rnn\_type="LSTM", embeddings=decoder\_embeddings)

device = "cuda" if torch.cuda.is\_available() else "cpu"

model = onmt.models.model.NMTModel(encoder, decoder)

model.to(device)

# Specify the tgt word generator and loss computation module

model.generator = nn.Sequential(

    nn.Linear(rnn\_size, len(tgt\_vocab)),

    nn.LogSoftmax(dim=-1)).to(device)

loss = onmt.utils.loss.NMTLossCompute(

    criterion=nn.NLLLoss(ignore\_index=tgt\_padding, reduction="sum"),

    generator=model.generator)

lr = 1

torch\_optimizer = torch.optim.SGD(model.parameters(), lr=lr)

optim = onmt.utils.optimizers.Optimizer(

    torch\_optimizer, learning\_rate=lr, max\_grad\_norm=2)

# build the validation iterator

valid\_iter = DynamicDatasetIter(

    corpora={"valid": valid},

    corpora\_info={"valid": {"weight": 1}},

    transforms={},

    fields=vocab\_fields,

    is\_train=False,

    batch\_type="sents",

    batch\_size=8,

    batch\_size\_multiple=1,

    data\_type="text")

valid\_iter = IterOnDevice(valid\_iter, 0)

report\_manager = onmt.utils.ReportMgr(

    report\_every=50, start\_time=None, tensorboard\_writer=None)

trainer = onmt.Trainer(model=model,

                       train\_loss=loss,

                       valid\_loss=loss,

                       optim=optim,

                       report\_manager=report\_manager,

                       dropout=[0.1])

trainer.train(train\_iter=train\_iter,

              train\_steps=10000,

              valid\_iter=valid\_iter,

              valid\_steps=5000)

Đây là kết quả em thu được:

Text

Description automatically generated

Sau đó em dùng tập test để dịch thửu nghĩa từng từ:

for batch in data\_iter:

    trans\_batch = translator.translate\_batch(

        batch=batch, src\_vocabs=[src\_vocab],

        attn\_debug=False)

    translations = builder.from\_batch(trans\_batch)

    for trans in translations:

        print(trans.log(0))

    break

kết quả thu được nghĩa dịch từng từ nhưng không sát nghĩa

Text

Description automatically generated

Text

Description automatically generated with medium confidence