作业1 LSTM RNN GRU 对比试验

RNN介绍

输入与输出格式

- **输入**：
  - **`input`**
    - 形状为 `(seq_len, batch_size, input_size)`，如果设置 `batch_first=True` 则为 `(batch_size, seq_len, input_size)`。
    - `seq_len`：序列长度（时间步数）
    - `batch_size`：每批数据的样本数量
    - `input_size`：每个时间步输入的特征维度
  - **初始隐藏状态 `h_0`**
    - 形状为 `(num_layers * num_directions, batch_size, hidden_size)`
    - 可省略，默认初始化为全 0。
- **输出**：
  - **`output`**
    - 形状为 `(seq_len, batch_size, num_directions * hidden_size)`，包含每个时间步的输出。
  - **`h_n`**
    - 形状为 `(num_layers * num_directions, batch_size, hidden_size)`，为每层最后一个时间步的隐藏状态。

初始化参数说明

参数名	类型	默认值	含义
`input_size`	int	无	每个时间步输入的特征维度
`hidden_size`	int	无	隐藏状态的特征维度
`num_layers`	int	1	RNN 的堆叠层数
`nonlinearity`	str	'tanh'	激活函数，可选 `'tanh'` 或 `'relu'`

bias	bool	True	是否使用偏置项
batch_first	bool	False	若为 True，则输入输出维度变为（batch，seq，feature)
dropout	float	0.0	除最后一层外，各层之间的 dropout 概率
bidirectional	bool	False	是否为双向 RNN

LSTM运行结果

```
vocab_size:  20001
ImdbNet(
  (embedding): Embedding(20001, 64)
  (lstm): LSTM(64, 64)
  (linear1): Linear(in_features=64, out_features=64, bias=True)
  (act1): ReLU()
  (linear2): Linear(in_features=64, out_features=2, bias=True)
)
Train Epoch: 1 Loss: 0.581733    Acc: 0.676567
Test set: Average loss: 0.4569, Accuracy: 0.7848
Train Epoch: 2 Loss: 0.391332    Acc: 0.826078
Test set: Average loss: 0.3824, Accuracy: 0.8277
Train Epoch: 3 Loss: 0.302817    Acc: 0.873502
Test set: Average loss: 0.3598, Accuracy: 0.8483
Train Epoch: 4 Loss: 0.234376    Acc: 0.910294
Test set: Average loss: 0.3626, Accuracy: 0.8505
Train Epoch: 5 Loss: 0.188259    Acc: 0.931809
Test set: Average loss: 0.3734, Accuracy: 0.8546
```

RNN运行结果

```
ImdbNet_RNN(
  (embedding): Embedding(20001, 64)
  (rnn): RNN(64, 64)
  (linear1): Linear(in_features=64, out_features=64, bias=True)
  (act1): ReLU()
  (linear2): Linear(in_features=64, out_features=2, bias=True)
)
Train Epoch: 1 Loss: 0.597444    Acc: 0.669928
Test set: Average loss: 0.4790, Accuracy: 0.7696
Train Epoch: 2 Loss: 0.415444    Acc: 0.811152
Test set: Average loss: 0.3947, Accuracy: 0.8210
Train Epoch: 3 Loss: 0.329002    Acc: 0.862071
Test set: Average loss: 0.3684, Accuracy: 0.8402
Train Epoch: 4 Loss: 0.268788    Acc: 0.893021
Test set: Average loss: 0.3575, Accuracy: 0.8463
Train Epoch: 5 Loss: 0.219605    Acc: 0.918331
Test set: Average loss: 0.3807, Accuracy: 0.8503
```

GRU运行结果

```
ImdbNet_GRU(
  (embedding): Embedding(20001, 64)
  (gru): GRU(64, 64)
  (linear1): Linear(in_features=64, out_features=64, bias=True)
  (act1): ReLU()
  (linear2): Linear(in_features=64, out_features=2, bias=True)
)
Train Epoch: 1 Loss: 0.564370    Acc: 0.688249
Test set: Average loss: 0.4314, Accuracy: 0.8056
Train Epoch: 2 Loss: 0.372965    Acc: 0.836711
Test set: Average loss: 0.3677, Accuracy: 0.8428
Train Epoch: 3 Loss: 0.289450    Acc: 0.882538
Test set: Average loss: 0.3416, Accuracy: 0.8554
Train Epoch: 4 Loss: 0.231660    Acc: 0.911342
Test set: Average loss: 0.3494, Accuracy: 0.8572
Train Epoch: 5 Loss: 0.184861    Acc: 0.933207
Test set: Average loss: 0.3591, Accuracy: 0.8576
```

作业2 手写LSTM实验

参考 `pytorch` 的实现：

$$
\begin{aligned}
i_t &= \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{t-1} + b_{hi}) \\
f_t &= \sigma(W_{if}x_t + b_{if} + W_{hf}h_{t-1} + b_{hf}) \\
g_t &= \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{t-1} + b_{hg}) \\
o_t &= \sigma(W_{io}x_t + b_{io} + W_{ho}h_{t-1} + b_{ho}) \\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
h_t &= o_t \odot \tanh(c_t)
\end{aligned}
$$

itftgtotctht = σ(Wiixt + bii + Whiht−1 + bhi) = σ(Wif xt + bif + Whf ht−1 + bhf) = tanh(Wigxt + big

+ Whght−1 + bhg) = σ(Wioxt + bio + Whoht−1 + bho) = ft ⊙ ct−1 + it ⊙ gt = ot ⊙ tanh(ct)

可以证明这种形式和 PPT 中使用 `concat` 的形式是等价的。

运行结果

```
Net(
  (embedding): Embedding(20001, 64)
  (lstm): LSTM(
    (Wii): Linear(in_features=64, out_features=64, bias=True)
    (Whi): Linear(in_features=64, out_features=64, bias=False)
    (Wif): Linear(in_features=64, out_features=64, bias=True)
    (Whf): Linear(in_features=64, out_features=64, bias=False)
    (Wig): Linear(in_features=64, out_features=64, bias=True)
    (Whg): Linear(in_features=64, out_features=64, bias=False)
    (Wio): Linear(in_features=64, out_features=64, bias=True)
    (Who): Linear(in_features=64, out_features=64, bias=False)
    (sigmoid): Sigmoid()
    (tanh): Tanh()
  )
  (fc1): Linear(in_features=64, out_features=64, bias=True)
  (fc2): Linear(in_features=64, out_features=2, bias=True)
)
Train Epoch: 1 Loss: 0.588408    Acc: 0.667083
Test set: Average loss: 0.4970, Accuracy: 0.7625
Train Epoch: 2 Loss: 0.388743    Acc: 0.828325
Test set: Average loss: 0.3876, Accuracy: 0.8240
Train Epoch: 3 Loss: 0.295549    Acc: 0.880341
Test set: Average loss: 0.3535, Accuracy: 0.8410
Train Epoch: 4 Loss: 0.234457    Acc: 0.909545
Test set: Average loss: 0.3444, Accuracy: 0.8528
Train Epoch: 5 Loss: 0.180492    Acc: 0.934056
Test set: Average loss: 0.3655, Accuracy: 0.8570
```

调整网络结构

- **hidden_size：**128
- **embedding_size：**128

```
Net(
  (embedding): Embedding(20001, 128)
  (lstm): LSTM(
    (Wii): Linear(in_features=128, out_features=128, bias=True)
    (Whi): Linear(in_features=128, out_features=128, bias=False)
    (Wif): Linear(in_features=128, out_features=128, bias=True)
    (Whf): Linear(in_features=128, out_features=128, bias=False)
    (Wig): Linear(in_features=128, out_features=128, bias=True)
    (Whg): Linear(in_features=128, out_features=128, bias=False)
    (Wio): Linear(in_features=128, out_features=128, bias=True)
    (Who): Linear(in_features=128, out_features=128, bias=False)
    (sigmoid): Sigmoid()
    (tanh): Tanh()
  )
  (fc1): Linear(in_features=128, out_features=128, bias=True)
  (fc2): Linear(in_features=128, out_features=2, bias=True)
)
Train Epoch: 1 Loss: 0.538069    Acc: 0.718600
Test set: Average loss: 0.4113, Accuracy: 0.8157
Train Epoch: 2 Loss: 0.335955    Acc: 0.854932
Test set: Average loss: 0.3534, Accuracy: 0.8477
Train Epoch: 3 Loss: 0.244695    Acc: 0.902506
Test set: Average loss: 0.3480, Accuracy: 0.8491
Train Epoch: 4 Loss: 0.173830    Acc: 0.935753
Test set: Average loss: 0.3604, Accuracy: 0.8610
Train Epoch: 5 Loss: 0.110160    Acc: 0.961811
Test set: Average loss: 0.4108, Accuracy: 0.8582
```

更改损失函数

损失函数改为 `NLLLoss`，并将 `output` 进行 `log_softmax` 处理。

```
Net(
  (embedding): Embedding(20001, 64)
  (lstm): LSTM(
    (Wii): Linear(in_features=64, out_features=64, bias=True)
    (Whi): Linear(in_features=64, out_features=64, bias=False)
    (Wif): Linear(in_features=64, out_features=64, bias=True)
    (Whf): Linear(in_features=64, out_features=64, bias=False)
    (Wig): Linear(in_features=64, out_features=64, bias=True)
    (Whg): Linear(in_features=64, out_features=64, bias=False)
    (Wio): Linear(in_features=64, out_features=64, bias=True)
    (Who): Linear(in_features=64, out_features=64, bias=False)
    (sigmoid): Sigmoid()
    (tanh): Tanh()
  )
  (fc1): Linear(in_features=64, out_features=64, bias=True)
  (fc2): Linear(in_features=64, out_features=2, bias=True)
)
Train Epoch: 1 Loss: 0.587833    Acc: 0.667732
Test set: Average loss: 0.4572, Accuracy: 0.7838
Train Epoch: 2 Loss: 0.391795    Acc: 0.827177
Test set: Average loss: 0.3932, Accuracy: 0.8234
Train Epoch: 3 Loss: 0.301120    Acc: 0.875849
Test set: Average loss: 0.3539, Accuracy: 0.8471
Train Epoch: 4 Loss: 0.240825    Acc: 0.904802
Test set: Average loss: 0.3546, Accuracy: 0.8534
Train Epoch: 5 Loss: 0.188787    Acc: 0.931510
Test set: Average loss: 0.3544, Accuracy: 0.8527
```

调整训练超参数

- **`batch_size`**：128
- **`learning_rate`**：5e-3

```
Net(
  (embedding): Embedding(20001, 64)
  (lstm): LSTM(
    (Wii): Linear(in_features=64, out_features=64, bias=True)
    (Whi): Linear(in_features=64, out_features=64, bias=False)
    (Wif): Linear(in_features=64, out_features=64, bias=True)
    (Whf): Linear(in_features=64, out_features=64, bias=False)
    (Wig): Linear(in_features=64, out_features=64, bias=True)
    (Whg): Linear(in_features=64, out_features=64, bias=False)
    (Wio): Linear(in_features=64, out_features=64, bias=True)
    (Who): Linear(in_features=64, out_features=64, bias=False)
    (sigmoid): Sigmoid()
    (tanh): Tanh()
  )
  (fc1): Linear(in_features=64, out_features=64, bias=True)
  (fc2): Linear(in_features=64, out_features=2, bias=True)
)
Train Epoch: 1 Loss: 0.537051    Acc: 0.710042
Test set: Average loss: 0.3672, Accuracy: 0.8379
Train Epoch: 2 Loss: 0.274269    Acc: 0.890426
Test set: Average loss: 0.3041, Accuracy: 0.8715
Train Epoch: 3 Loss: 0.165040    Acc: 0.940237
Test set: Average loss: 0.3595, Accuracy: 0.8645
Train Epoch: 4 Loss: 0.086922    Acc: 0.971188
Test set: Average loss: 0.4336, Accuracy: 0.8566
Train Epoch: 5 Loss: 0.038015    Acc: 0.988953
Test set: Average loss: 0.5297, Accuracy: 0.8539
```