

# Report of assignment-4

## Requirements

1.

### CNN

#### Data:

```
categories = ['alt.atheism', 'talk.religion.misc', 'comp.graphics', 'sci.space', 'rec.motorcycles']
```

#### Structure:

```
self.embed = encoder.Embedding(init_embed)
self.conv_pool = encoder.ConvMaxpool(in_channels=self.embed.embedding_dim, out_channels=kernel_nums,
                                     kernel_sizes=kernel_sizes, padding=padding)
self.dropout = nn.Dropout(dropout)
self.fc = nn.Linear(sum(kernel_nums), num_classes)
```

#### Hyperparameter:

```
model = cnn((len(vocab), 50), num_classes=5, padding=2, dropout=0.1)
```

#### Result:

```
[tester]
AccuracyMetric: acc=0.956772
```

#### Data:

```
categories = ['alt.atheism', 'comp.graphics', 'comp.os.ms-windows.misc', 'comp.sys.ibm.pc.hardware', 'comp.sys.mac.hardware',
             'comp.windows.x', 'misc.forsale', 'rec.autos', 'rec.motorcycles', 'rec.sport.baseball',
             'rec.sport.hockey', 'sci.crypt', 'sci.electronics', 'sci.med', 'sci.space',
             'soc.religion.christian', 'talk.politics.guns',
             'talk.politics.mideast', 'talk.politics.misc', 'talk.religion.misc']
```

#### Structure:

```
self.embed = encoder.Embedding(init_embed)
self.conv_pool = encoder.ConvMaxpool(in_channels=self.embed.embedding_dim, out_channels=kernel_nums,
                                     kernel_sizes=kernel_sizes, padding=padding)
self.dropout = nn.Dropout(dropout)
self.fc = nn.Linear(sum(kernel_nums), num_classes)
```

#### Hyperparameter:

```
model = cnn((len(vocab), 200), num_classes=20, padding=2, dropout=0.1)
```

#### Result:

```
[tester]
AccuracyMetric: acc=0.937244
```

### RNN

#### Data:

```
categories = ['alt.atheism', 'talk.religion.misc', 'comp.graphics', 'sci.space', 'rec.motorcycles']
```

#### Structure:

```
self.word_embeddings = nn.Embedding(vocab_size, embedding_length)
self.lstm = nn.LSTM(embedding_length, hidden_size)
self.label = nn.Linear(hidden_size, output_size)
```

#### Hyperparameter:

```
model = lstm(vocab_size=len(vocab), embedding_length=50, hidden_size=32, output_size=5)
```

#### Result:

```
[tester]
AccuracyMetric: acc=0.899135
```

#### Data:

```
categories = ['alt.atheism', 'comp.graphics', 'comp.os.ms-windows.misc', 'comp.sys.ibm.pc.hardware', 'comp.sys.mac.hardware',  
             'comp.windows.x', 'misc.forsale', 'rec.autos', 'rec.motorcycles', 'rec.sport.baseball',  
             'rec.sport.hockey', 'sci.crypt', 'sci.electronics', 'sci.med', 'sci.space',  
             'soc.religion.christian', 'talk.politics.guns',  
             'talk.politics.mideast', 'talk.politics.misc', 'talk.religion.misc']
```

Structure:

```
self.word_embeddings = nn.Embedding(vocab_size, embedding_length)  
self.lstm = nn.LSTM(embedding_length, hidden_size)  
self.label = nn.Linear(hidden_size, output_size)
```

Hyperparameter:

```
model = lstm(vocab_size=len(vocab), embedding_length=200, hidden_size=128, output_size=20)
```

**Result:**

```
[tester]  
AccuracyMetric: acc=0.864256
```

## 2.

我觉得总体上给人感觉非常好。我主要有两个小的建议。

### 一、命名规则显得非常生硬

一般来说，我们实现模型的时候，都是遵循以下的逻辑：先制作数据集，然后编写模型，然后实现模型的训练，最后实现模型的测试。但是 FastNLP 里面的命名规则让人感觉是模型决定了数据集的 field 的命名，决定了训练和测试过程中，loss 和 metrics 的参数值，显得有些和逻辑相背。

对此，我有两个想法。

第一，放弃 Const，在 tutorial 中告诉使用人员，模型的返回字典的 key 要和数据集的输入 field 名保持一致，训练和测试中 loss 和 metrics 中 pred 参数的值要和数据集的输入 field 名保持一致，target 参数的值要和数据集的目标 field 名保持一致，这样就是数据集决定模型，训练，测试，和逻辑相符。

第二，保留 Const，并且在 tutorial 中强调 Const 是 FastNLP 提供的一个命名接口，用户可以用 Const 进行数据集、模型、训练和测试中 loss 和 metrics 中相应名称的命名。

### 二、tutorial,README 中有 bug

#### 1、

Vocabulary 的使用中，Vocabularyindex\_dataset()中间没有加点。

#### 2、

编写自己的模型中，LSTMText 中，output, (hidden, cell) = self.lstm(embedded)

输入参数应该缺少了 h\_0 和 t\_0。

#### 3、

| 类型      | 功能                | 例子                               |
|---------|-------------------|----------------------------------|
| encoder | 将输入编码为具有具有表示能力的向量 | embedding, RNN, CNN, transformer |

README 的这里多打了一个具有。