深度学习
Lab6-recurrent neural network

兰韵诗

本次Lab有作业，请在4月7号结束之前提交！！！

Lab5参考答案

```python
class Conv2D(nn.Module):
    def __init__(self, out_channels, in_channels, kernel_size):
        super(Conv2D, self).__init__()
        self.weight = np.random.randn(out_channels, in_channels, kernel_size[0], kernel_size[1])
        self.bias = np.random.randn(out_channels)

    def forward_np(self, X):
        '''
        X --> (B, I, H, W) where B = batch size, I = in_channel, H = height of feature map, W = width of feature map
        你需要利用以上初始化的参数weight和bias实现一个卷积层的前向传播
        Y should have size (B, O, H-h+1, W-w+1)
        '''

        Y = corr2d(X, self.weight) + np.reshape(self.bias, (1, -1, 1, 1))
        return Y
```

```python
def corr2d(X, K):
    '''
    X --> (B, I, H, W) where B = batch size, I = in_channel, H = height of feature map, W = width of feature map
    K --> (O, I, h, w) where O = out_channel, I = in_channel, h = height of kernel, w = width of kernel
    你需要实现一个Stride为1, Padding为0的窄卷积操作
    Y should have size (B, O, H-h+1, W-w+1)
    '''

    B, I, H, W = X.shape
    o, _, h, w = K.shape
    Y = np.zeros((B, o, H - h + 1, W - w + 1))
    for i in range(Y.shape[-2]):
        for j in range(Y.shape[-1]):
            Y[:, :, i, j] = np.sum(np.sum(np.sum(np.repeat(np.expand_dims(X[:, :, i: i + h, j: j + w], axis = 2), repeats = o, axis = 2) \
                * np.transpose(np.reshape(K, (1, o, I, h, w)), (0, 2, 1, 3, 4)), -1), -1), 1)
    return Y
```

Lab5参考答案

```python
class MaxPool2D(nn.Module):
    def __init__(self, pool_size):
        super(MaxPool2D, self).__init__()
        self.pool_size = pool_size


    def forward_np(self, X):
        '''
        X --> (B, I, H, W) where B = batch size, I = in_channel, H = height of feature map, W = width of feature map
        K --> (h, w) where h = height of kernel, w = width of kernel
        你需要利用以上pool_size实现一个汇聚层的前向传播，汇聚层的子区域间无覆盖
        Y should have size (B, I, H/h, W/w)
        '''
        B, I, H, W = X.shape
        p_h, p_w = self.pool_size
        Y = np.zeros((B, I, int(H/p_h), int(W/p_w)))
        for i in range(Y.shape[-2]):
            for j in range(Y.shape[-1]):
                Y[:, :, i, j] = np.amax(np.amax(X[:, :, i*p_h: (i+1)*p_h, j*p_w: (j+1)*p_w], -1), -1)
        return Y
```

Lab5参考答案

```python
class linear(nn.Module):
    def __init__(self, num_outputs, num_inputs):
        super(linear, self).__init__()
        self.weight = np.random.randn(num_inputs, num_outputs)
        self.bias = np.random.randn(num_outputs)

    def forward_np(self, X):
        '''
        X --> (B, I, H, W) where B = batch size, I = in_channel, H = height of feature map, W = width of feature map
        你需要利用以上初始化的参数weight和bias实现一个卷积层的前向传播
        Y should have size (B, O, H-h+1, W-w+1)
        '''

        Y = np.matmul(X, np.transpose(self.weight)) + np.expand_dims(self.bias, axis=0)
        return Y
```

Lab6

1.熟悉文本生成任务的流程

2.补全rnn_hard_version.py 文件中的基于GRU的歌词预测模型

Recurrent Neural Network

- 根据提示，补全**基于GRU的歌词预测模型**代码
  - 利用设定好的输入完成**GRU的前向传播**和**歌词预测模型主体**
  - **正确定义和初始化GRU中的参数**
  - 不能调用其他工具包，不能调用pytorch内置的GRU模块，只能在 “to do” 下面书写代码
  - 提交之后，测试集上的准确率应该在一个正确的范围内可多次提交。即使对自己的代码没有自信也一定要提交，我们会酌情给过程分
- **TO DO**： 完成《Recurrent Neural Network》项目。补全rnn_hard_version.py文件使exercise_rnn.py文件中的train_with_RNN_hard()可以顺利执行。

Evaluation脚本

```python
def compute_acc(pred_file):
    with open('data/jaychou_test_y.txt') as f:
        gold = f.readlines()
    gold = [sent.strip() for sent in gold]

    with open(pred_file) as f:
        pred = f.readlines()
    pred = [sent.strip() for sent in pred]
    correct_case = [i for i, _ in enumerate(gold) if gold[i] == pred[i]]

    acc = len(correct_case)*1./len(gold)
    print('the predicted accuracy is %s' %acc)

if __name__ == '__main__':
    pred_file = 'data/predict.txt'
    compute_acc(pred_file)
```

Note：为了测试方便，这里我们使用准确率作为我们的生成评估标准。实际生成任务一般采用BLEU，Rouge等