word2vec 说明

1、genism 库

gensim 库中提供了 word2vec 的 cbow 模型和 skipgram 模型的实现，可直接调用

```
sentences = word2vec.Text8Corpus(input_corpus)  # 加载语料
model = word2vec.Word2Vec(sentences, size=100, window=8, min_count=3, iter=8)
model.save(model_path)
model.wv.save_word2vec_format(model_path, binary=False)
```

完整版参考代码

2、tensorflow 实现 skipgram 模型

skipgram 模型使用中心词预测上下文，网上介绍很多了也可直接去看论文

本模型实验采用的数据时 wiki 百科中文数据，有原版和分词后版本，数据量大下载请移步

实现详细直接看代码，代码中关键处都有注释，这里提一下 word2vec 中常用的 nce loss 损失函数，nce loss 函数个参数定义如下

```
def nce_loss(
    weights,   #模型网络权重W，weights.shape=(N,K)，N是数据类数即单词数，K是每个单词Embedding Size
    biases,    #模型网络偏置b，biases.shape=(N)
    inputs,    #输入数据即单词的初始化，这里课采用one-hot初始化，也可以采用随机初始化
    labels,    #skipgram模型输入数据是中心词，则labels便是上下文词
    num_sampled,      #负采样的样本的个数
    num_classes,      #num_classes=N，在word2vec中既是词表的大小vocabulary_size
    num_true=1,       #实际正样本个数
    sampled_values=None,            #word2vec中负采样，sampled_values=None是定义了负采样的方式
    remove_accidental_hits=False,   #如果在负采样中采样到了正样本，要不要去掉
    partition_strategy="mod",       #对weights进行embedding_lookup时并行查表时的策略
    name="nce_loss"
)
```

解释一下参数 sampled_values，从 tensorflow 的 nce_loss 源代码中可以看到当 sampled_values=None 时采样方式，word2vec 中负采样过程其实就是优选采样词频高的词作负样本

```
if sampled_values is None:
    sampled_values = candidate_sampling_ops.log_uniform_candidate_sampler(
        true_classes=labels,
        num_true=num_true,
        num_sampled=num_sampled,
        unique=True,
        range_max=num_classes
    )
```

在上图中展现了 nce_loss 在实际使用过程中参数列表以及各个参数的含义，下面我们看一下 tensorflow 源码中对于 nce_loss 函数的实现逻辑：

```
def nce_loss(weights, biases, labels, inputs, num_sampled, num_classes, num_true=1,
             sampled_values=None,  remove_accidental_hits=False,
             partition_strategy="mod", name="nce_loss"):

    logits, labels = _compute_sampled_logits(
        weights=weights, biases=biases, labels=labels, inputs=inputs,
        num_sampled=num_sampled, num_classes=num_classes, num_true=num_true,
        sampled_values=sampled_values, subtract_log_q=True,
        remove_accidental_hits=remove_accidental_hits,
        partition_strategy=partition_strategy, name=name)
```

_compute_sampled_logits函数计算出正样本和负采样得到的负样本的output和label

```
    sampled_losses = sigmoid_cross_entropy_with_logits(
        labels=labels, logits=logits, name="sampled_losses")
```

通过正样本和负样本的logits和labels将word2vec本来一个多分类问题转变成一个二分类问题，使用sigmoid cross entropy计算loss，进行反向传播更新网络参数，使用的是交叉熵损失函数

```
    # sampled_losses is batch_size x {true_loss, sampled_losses...}
    # We sum out true and sampled losses.
    return _sum_rows(sampled_losses)
```

Tensorflow 实现 skipgram 模型完整细节参考代码，训练测试效果可参见下图：

```
Nearest to 中国: 中国，多汗，劳力士，长照，男中音，董事局，逐年，地理知识，
Nearest to 学院: 学院，充电，设备齐全，压差，zunp，文言，望去，marina，
Nearest to 中心: 最佳，由人，信山，土卫二，ruficeps，涩味，深衣，暴卒，
Nearest to 北京: 北京，成天，比古，菊池，响应，存者，推普，穆索尔，
Nearest to 大学: 明星，岐州，希运，丝毫，张凯，再版，六成，颇重，
Nearest to 爱: 文学，保护区，声名大噪，方敏，家，粒，感受器，分得，
Nearest to 不错: 地铁，萧纪，温差，国瑞，盖革，可道，东荟城，日向秋，
Nearest to 中文: 广告，二部曲，訚，占，韩昭侯，梁钊峰，未了，金属制品，
Nearest to 幸福: 荣誉，开季，第二任，董洁，陡河，莫康时，干涩，公共汽车，
Average loss at step 100 : 8.354138759613036
Average loss at step 200 : 7.764778663635254
Average loss at step 300 : 7.424072135925293
Average loss at step 400 : 7.0825538787841795
Average loss at step 500 : 7.352655693054199
Average loss at step 600 : 6.941441452026367
Average loss at step 700 : 6.831985248565674
Average loss at step 800 : 6.939995124816894
Average loss at step 900 : 6.595024318695068
Average loss at step 1000 : 6.538598918914795
```

训练前效果