

用神经网络处理NER命名实体识别问题



不会停的蜗牛 (/u/7b67af2e61b3) [+ 关注](#)

2016.09.05 11:02* 字数 1014 阅读 6572 评论 9 喜欢 18

(/u/7b67af2e61b3)

本文结构：

1. 什么是命名实体识别（NER）
2. 怎么识别？

cs224d Day 7: 项目2-用DNN处理NER问题

课程项目描述地址 ([https://link.jianshu.com?](https://link.jianshu.com?t=https://cs224d.stanford.edu/assignment2/index.html)

[t=https://cs224d.stanford.edu/assignment2/index.html](https://cs224d.stanford.edu/assignment2/index.html))

什么是NER？

命名实体识别（NER）是指识别文本中具有特定意义的实体，主要包括人名、地名、机构名、专有名词等。命名实体识别是信息提取、问答系统、句法分析、机器翻译等应用领域的重要基础工具，作为结构化信息提取的重要步骤。摘自BosonNLP

(<https://link.jianshu.com?t=http://docs.bosonnlp.com/ner.html>)

怎么识别？



先把解决问题的逻辑说一下，然后解释主要的代码，有兴趣的话，完整代码请去[这里看](https://link.jianshu.com?t=https://github.com/AliceDudu/Named-Entity-Recognition) (https://link.jianshu.com?t=https://github.com/AliceDudu/Named-Entity-Recognition)。
代码是在 Tensorflow 下建立只有一个隐藏层的 DNN 来处理 NER 问题。

(/apps/download?
utm_source=sbc) ×

1.问题识别：

NER 是个分类问题。

给一个单词，我们需要根据上下文判断，它属于下面四类的哪一个，如果都不属于，则类别为0，即不是实体，所以这是一个需要分成 5 类的问题：

- Person (PER)
- Organization (ORG)
- Location (LOC)
- Miscellaneous (MISC)

我们的训练数据有两列，第一列是单词，第二列是标签。

```
EU ORG
rejects 0
German MISC
Peter PER
BRUSSELS LOC
```

2.模型：

接下来我们用神经网络对其进行训练。

模型如下：

输入层的 $x^{(t)}$ 为以 x_t 为中心的窗口大小为3的上下文语境， x_t 是 one-hot 向量， x_t 与 L 作用后就是相应的词向量，词向量的长度为 $d = 50$ ：

$$\mathbf{x}^{(t)} = [x_{t-1}L, x_tL, x_{t+1}L] \in \mathbb{R}^{3d}$$



我们建立一个只有一个隐藏层的神经网络，隐藏层维度是 100， \hat{y} 就是得到的预测值，维度是 5：

$$\begin{aligned} \mathbf{h} &= \tanh(\mathbf{x}^{(i)}\mathbf{W} + \mathbf{b}_1) \\ \hat{\mathbf{y}} &= \text{softmax}(\mathbf{h}\mathbf{U} + \mathbf{b}_2) \end{aligned}$$

(/apps/download?utm_source=sbc) ×

用交叉熵来计算误差：

$$J(\theta) = \text{CE}(\mathbf{y}, \hat{\mathbf{y}}) = - \sum_{i=1}^5 y_i \log \hat{y}_i$$

J 对各个参数进行求导：

$$\frac{\partial J}{\partial \mathbf{U}} \quad \frac{\partial J}{\partial \mathbf{b}_2} \quad \frac{\partial J}{\partial \mathbf{W}} \quad \frac{\partial J}{\partial \mathbf{b}_1} \quad \frac{\partial J}{\partial \mathbf{L}_i}$$

$$\mathbf{U} \in \mathbb{R}^{100 \times 5} \quad \mathbf{b}_2 \in \mathbb{R}^5 \quad \mathbf{W} \in \mathbb{R}^{150 \times 100} \quad \mathbf{b}_1 \in \mathbb{R}^{100} \quad \mathbf{L}_i \in \mathbb{R}^{50}$$

得到如下求导公式：

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{U}} &= \mathbf{h}^T (\mathbf{y} - \hat{\mathbf{y}}) \\ \frac{\partial J}{\partial \mathbf{b}_2} &= (\mathbf{y} - \hat{\mathbf{y}}) \\ \frac{\partial J}{\partial \mathbf{h}} &= (\mathbf{y} - \hat{\mathbf{y}}) \mathbf{U}^T \\ \frac{\partial J}{\partial \mathbf{W}} &= (\mathbf{x}^{(i)})^T \left(\frac{\partial J}{\partial \mathbf{h}} \odot \tanh'(2(\mathbf{x}^{(i)}\mathbf{W} + \mathbf{b}_1)) \right) \\ \frac{\partial J}{\partial \mathbf{b}_1} &= \left(\frac{\partial J}{\partial \mathbf{h}} \odot \tanh'(2(\mathbf{x}^{(i)}\mathbf{W} + \mathbf{b}_1)) \right) \\ \frac{\partial J}{\partial \mathbf{x}^{(i)}} &= \left(\frac{\partial J}{\partial \mathbf{h}} \odot \tanh'(2(\mathbf{x}^{(i)}\mathbf{W} + \mathbf{b}_1)) \right) \mathbf{W}^T \end{aligned}$$



在 TensorFlow 中求导是自动实现的，这里用Adam优化算法更新梯度，不断地迭代，使得loss越来越小直至收敛。

3.具体实现

在 `def test_NER()` 中，我们进行 `max_epochs` 次迭代，每次，用 training data 训练模型得到一对 `train_loss, train_acc`，再用这个模型去预测 validation data，得到一对 `val_loss, predictions`，我们选择最小的 `val_loss`，并把相应的参数 `weights` 保存起来，最后我们是要用这些参数去预测 test data 的类别标签：

(/apps/download?
utm_source=sbc)



```

def test_NER():

    config = Config()
    with tf.Graph().as_default():
        model = NERModel(config) # 最主要的类

        init = tf.initialize_all_variables()
        saver = tf.train.Saver()

    with tf.Session() as session:
        best_val_loss = float('inf') # 最好的值时, 它的 loss 它的 迭代次数 epoch
        best_val_epoch = 0

        session.run(init)
        for epoch in xrange(config.max_epochs):
            print 'Epoch {}'.format(epoch)
            start = time.time()
            ###
            train_loss, train_acc = model.run_epoch(session, model.X_train,
                                                    model.y_train) # 1.把 train 数据放进
            val_loss, predictions = model.predict(session, model.X_dev, model.y_dev) #
            print 'Training loss: {}'.format(train_loss)
            print 'Training acc: {}'.format(train_acc)
            print 'Validation loss: {}'.format(val_loss)
            if val_loss < best_val_loss: # 用 val 数据的loss去找最小的loss
                best_val_loss = val_loss
                best_val_epoch = epoch
                if not os.path.exists("./weights"):
                    os.makedirs("./weights")

            saver.save(session, './weights/ner.weights') # 把最小的 loss 对应的 weight
            if epoch - best_val_epoch > config.early_stopping:
                break
            ###
            confusion = calculate_confusion(config, predictions, model.y_dev) # 3.把 de
            print_confusion(confusion, model.num_to_tag)
            print 'Total time: {}'.format(time.time() - start)

        saver.restore(session, './weights/ner.weights') # 再次加载保存过的 weights, 用
        print 'Test'
        print '===='
        print 'Writing predictions to q2_test.predicted'
        _, predictions = model.predict(session, model.X_test, model.y_test)
        save_predictions(predictions, "q2_test.predicted") # 把预测结果保存起来

if __name__ == "__main__":
    test_NER()

```

(/apps/download?
utm_source=sbc)



4.模型是怎么训练的呢？

- 首先导入数据 training , validation , test :

```
# Load the training set
docs = du.load_dataset('data/ner/train')

# Load the dev set (for tuning hyperparameters)
docs = du.load_dataset('data/ner/dev')

# Load the test set (dummy labels only)
docs = du.load_dataset('data/ner/test.masked')
```

- 把单词转化成 one-hot 向量后，再转化成词向量：

```
def add_embedding(self):
    # The embedding lookup is currently only implemented for the CPU
    with tf.device('/cpu:0'):

        embedding = tf.get_variable('Embedding', [len(self.wv), self.config.embed_size])
        window = tf.nn.embedding_lookup(embedding, self.input_placeholder)
        window = tf.reshape(
            window, [-1, self.config.window_size * self.config.embed_size])

        return window
```

- 建立神经层，包括用 xavier 去初始化第一层，L2 正则化和用 dropout 来减小过拟合的处理：

(/apps/download?
utm_source=sbc)



```
def add_model(self, window):

    with tf.variable_scope('Layer1', initializer=xavier_weight_init()) as scope:
        W = tf.get_variable(
            'W', [self.config.window_size * self.config.embed_size,
                  self.config.hidden_size])
        b1 = tf.get_variable('b1', [self.config.hidden_size])
        h = tf.nn.tanh(tf.matmul(window, W) + b1)
        if self.config.l2:
            tf.add_to_collection('total_loss', 0.5 * self.config.l2 * tf.nn.l2_loss(W))

    with tf.variable_scope('Layer2', initializer=xavier_weight_init()) as scope:
        U = tf.get_variable('U', [self.config.hidden_size, self.config.label_size])
        b2 = tf.get_variable('b2', [self.config.label_size])
        y = tf.matmul(h, U) + b2
        if self.config.l2:
            tf.add_to_collection('total_loss', 0.5 * self.config.l2 * tf.nn.l2_loss(U))
        output = tf.nn.dropout(y, self.dropout_placeholder)

    return output
```

(/apps/download?
utm_source=sbc)



关于 L2正则化 和 dropout 是什么, 如何减小过拟合问题的, 可以看这篇博客, 总结的简单明了。([https://link.jianshu.com?](https://link.jianshu.com?t=http://blog.csdn.net/u012162613/article/details/44261657)

[t=http://blog.csdn.net/u012162613/article/details/44261657](https://link.jianshu.com?t=http://blog.csdn.net/u012162613/article/details/44261657))

- 用 cross entropy 来计算 loss :

```
def add_loss_op(self, y):

    cross_entropy = tf.reduce_mean(
        tf.nn.softmax_cross_entropy_with_logits(y, self.labels_placeholder))
    tf.add_to_collection('total_loss', cross_entropy)          # Stores value in the
                                                                # collections are not
                                                                # added to the total loss

    loss = tf.add_n(tf.get_collection('total_loss'))           # Adds all input tensors
                                                                # to the total loss

    return loss
```

- 接着用 Adam Optimizer 把loss最小化 :



```
def add_training_op(self, loss):  
  
    optimizer = tf.train.AdamOptimizer(self.config.lr)  
    global_step = tf.Variable(0, name='global_step', trainable=False)  
    train_op = optimizer.minimize(loss, global_step=global_step) # 2.关键步骤：用 A  
  
    return train_op
```

(/apps/download?
utm_source=sbc)



每一次训练后，得到了最小化 loss 相应的 weights。

这样，NER 这个分类问题就搞定了，当然为了提高精度等其他问题，还是需要查阅文献来学习的。下一次先实现个 RNN。

[cs224d]

Day 1. 深度学习与自然语言处理 主要概念一览

(<https://www.jianshu.com/p/6993edef96e4>)

Day 2. TensorFlow 入门 (<https://www.jianshu.com/p/6766fbcd43b9>)

Day 3. word2vec 模型思想和代码实现 (<https://www.jianshu.com/p/86134284fa14>)

Day 4. 怎样做情感分析 (<https://www.jianshu.com/p/1909031bb1f2>)

Day 5. CS224d - Day 5: RNN快速入门 (<https://www.jianshu.com/p/bf9ddfb21b07>)

Day 6. 一文学会用 Tensorflow 搭建神经网络

(<https://www.jianshu.com/p/e112012a4b2d>)

Day 7. 用深度神经网络处理NER命名实体识别问题

(<https://www.jianshu.com/p/581832f2c458>)

Day 8. 用 RNN 训练语言模型生成文本 (<https://www.jianshu.com/p/b4c5ff7c450f>)

Day 9. RNN与机器翻译 (<https://www.jianshu.com/p/23b46605857e>)

Day 10. 用 Recursive Neural Networks 得到分析树

(<https://www.jianshu.com/p/403665b55cd4>)

Day 11. RNN的高级应用 (<https://www.jianshu.com/p/0e840f92b532>)



我是 不会停的蜗牛 Alice
85后全职主妇
喜欢人工智能，行动派
创造力，思考力，学习力提升修炼进行中
欢迎您的喜欢，关注和评论！

(/apps/download?
utm_source=sbc) ×

📖 技术博文 (/nb/5173140)

举报文章 © 著作权归作者所有



不会停的蜗牛 (/u/7b67af2e61b3)

+ 关注

写了 224835 字，被 3243 人关注，获得了 1969 个喜欢
(/u/7b67af2e61b3)

我是 Alice 喜欢人工智能，行动派 创造力，思考力，学习力提升修炼进行中 欢迎志同道合的小伙伴们和我一...

喜欢就点赞，有用就随意打赏吧 😊 Run with AI ！

赞赏支持

♥ 喜欢 (/sign_in?utm_source=desktop&utm_medium=not-signed-in-like-button) | 18



更多分享

(http://cwb.assets.jianshu.io/notes/images/5562522/weibo/image_2



下载简书 App ▶
随时随地发现和创作内容



(/apps/download?utm_source=nbc)





登录 (/sign-in?source=desktop&utm_medium=not-signed-in-comment-form)

(/apps/download?
utm_source=sbc)



9条评论

只看作者

按喜欢排序 按时间正序 按时间倒序



zy6job (/u/ea5be52cf272)

2楼 · 2016.09.28 10:01

(/u/ea5be52cf272)

你好，我是一个对神经网络的初学者，这里有一个疑问：神经网络在训练的时候对数据量有要求吗？比如我的训练数据不是很多，可以得到一个好的模型吗？

👍 赞 💬 回复

不会停的蜗牛 (/u/7b67af2e61b3)：@zy6job (/users/ea5be52cf272) 数据多的效果比少量的好，在工业应用中也是，feature重于算法，数据重于feature

2016.09.28 10:17 💬 回复

SoloworkHB (/u/d0178caf54b3)：@不会停的蜗牛 (/users/7b67af2e61b3) feature重于算法？好的算法往往能提炼出好的feature。数据重于feature？数据重于算法比较准确吧

2017.12.20 16:27 💬 回复

✍️ 添加新评论



dreamingc (/u/92fe68e581a8)

3楼 · 2017.04.22 20:21

(/u/92fe68e581a8)

你好，请问数据在哪里能找到？

👍 赞 💬 回复

不会停的蜗牛 (/u/7b67af2e61b3)：@dreamingc (/users/92fe68e581a8) 文章中github里有，还有一个wordvector文件在这取：链接: <https://pan.baidu.com/s/1cAdVIG> (https://pan.baidu.com/s/1cAdVIG) 密码: rpfh

2017.04.22 20:43 💬 回复



dreamingc (/u/92fe68e581a8) : @不会停的蜗牛 (/users/7b67af2e61b3) 请问, 这个词向量文件是自己生成的吗? 怎样生成的? 这个词向量文件和train有什么关系? 我现在想做中文的ner, 用这个程序可以吗? 十分感谢!

2017.05.26 21:45 回复

不会停的蜗牛 (/u/7b67af2e61b3) : @dreamingc (/users/92fe68e581a8) 这是一篇训练词向量的文章, 可以看看原理简介, <http://www.jianshu.com/p/16bf97e3f43a> (<http://www.jianshu.com/p/16bf97e3f43a>); 你可以用中文语料库, 用word2vec去训练词向量; 这个程序应该需要修改一些地方, 对中文的处理, 你可以先搜一下中文的程序, 没有再自己改

2017.05.26 22:22 回复

添加新评论 | 还有1条评论, 展开查看



再没 (/u/cb71595ab85f)

4楼 · 2017.10.19 16:23

(/u/cb71595ab85f)

你好, 为什么我跑你的程序, 出来的预测结果都是3?

赞 回复

被以下专题收入, 发现更多相似内容



数据科学家 (/c/0adc32d3cf07?utm_source=desktop&utm_medium=notes-included-collection)



坚持写作100天 (/c/6f43264f8299?utm_source=desktop&utm_medium=notes-included-collection)



程序员 (/c/NEt52a?utm_source=desktop&utm_medium=notes-included-collection)



NLP (/c/173cbecfcfc2?utm_source=desktop&utm_medium=notes-included-collection)

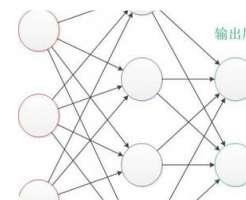
(/apps/download?utm_source=sbc)





机器学习迷 (/c/5c6f39c9167c?utm_source=desktop&utm_medium=notes-included-collection)

(/p/77e69ff5cfae?



(/apps/download?
utm_source=sbc)



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

重磅！神经网络浅讲：从神经元到深度学习 (/p/77e69ff5cfae?utm_campai...

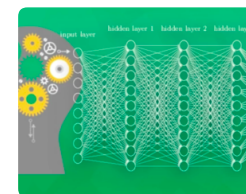
神经网络是一门重要的机器学习技术。它是目前最为火热的研究方向—深度学习的基础。学习神经网络不仅可以让你掌握一门强大的机器学习方法，同时也可以更好地帮助你理解深度学习技术。本文以一种简单的，循..



BURIBURI_ZAEMON (/u/00d1ed2b53ae?

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/e112012a4b2d?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

一文学会用 Tensorflow 搭建神经网络 (/p/e112012a4b2d?utm_campaign=...

cs224d-Day 6: 快速入门 Tensorflow 本文是学习这个视频课程系列的笔记，课程链接是 youtube 上的，讲的很好，浅显易懂，入门首选，而且在github有代码，想看视频的也可以去他的优酷里的频道找。Tensorflo...



不会停的蜗牛 (/u/7b67af2e61b3?

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/dbc48e9c2f56?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)



神经网络 (/p/dbc48e9c2f56?utm_campaign=maleskine&utm_content=n...

转载的他人的文章，适合神经网络刚开始学习的人：http://www.cnblogs.com/subconscious/p/5058741.html

神经网络浅讲：从神经元到深度学习 神经网络是一门重要的机器学习技术。它是目前最为火热的研究方向-...



hailiu13 (/u/3172ba2c7721?

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/apps/download?

utm_source=sbc)



(/p/4d37813c0952?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

从神经元到深度学习 (/p/4d37813c0952?utm_campaign=maleskine&utm_...

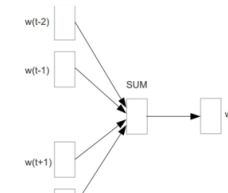
转载自：神经网络浅讲：从神经元到深度学习 神经网络是一门重要的机器学习技术。它是目前最为火热的研究方向--深度学习的基础。学习神经网络不仅可以让你掌握一门强大的机器学习方法，同时也可以更好地帮...



Daniel大人 (/u/2ff5bd422ca1?

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/6993edef96e4?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

深度学习与自然语言处理 主要概念一览 (/p/6993edef96e4?utm_campaign=...

CS224d - Day 1: 要开始系统地学习 NLP 课程 cs224d，今天先来一个课程概览。课程一共有16节，先对每一节中提到的模型，算法，工具有个总体的认识，知道都有什么，以及它们可以做些什么事情。简介：1....



不会停的蜗牛 (/u/7b67af2e61b3?

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)


(/p/e37dd97f64a9?

utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)



专为时间紧凑的人打造---桂林阳朔两天两夜游 (/p/e37d...

前言 去桂林的想法去年就有了，然而去年的工作是大小周，也就是一周六天一周五天，节假日呢又不想出去玩，因为到处人山人海，于是想法就被埋没了。...

 冬天只爱早晨 (/u/223a1314e818?

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)




(/apps/download?
utm_source=sbc)



凌晨四点零六分 (/p/c8dbb81e6311?utm_campaign=maleskine&utm_con...


飞驰的火车怎么刹住 爆炸的夏天如何留住 我后悔卖掉时光机器 往昔的日子一点一滴刻在电影胶片里 用力也想不起那个梦 我炸掉那座无名山 赶走所有野生动物 ...

 张常委 (/u/b3da84d4d728?

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

ObjectAnimator基本使用 (/p/4d55ee31f40c?utm_campaign=maleskine&...

Android自定义控件三部曲文章索引 ObjectAnimator基本使用

 凉城花祭八回梦 (/u/21894248e8ed?

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)


(/p/5b0b8ccb0d0f?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

致阿楠 (/p/5b0b8ccb0d0f?utm_campaign=maleskine&utm_content=not...

你要走，我不挽留。我知道我留不住。时光荏苒了岁月，似水年华住我握不住。亲爱的阿楠，初遇你，你若清风徐来，一转间，浮动了我的光年。温婉如你，北方之女。亲爱的阿楠，相处已久，太多无法叙尽，记...


 小二婆 (/u/a9005b95c257?

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

小白学炒股（一） (/p/592a21b59251?utm_campaign=maleskine&utm_c...



为了实现财务自由，我下定决心从今天开始学着买股票 首先明白股票是什么，它是股份公司发给股东证明其所入股份的一种有价证券，它可以作为买卖对象和抵押品，是资金市场主要的长期信用工具之一 好吧，今...

 绿冰鸢 (/u/7418ff3c2ccb?)

utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation) (/apps/download?utm_source=sbc)

