Pytorch 的入门与实战
群问题和资料收集
(不断更新)

资料推荐：

1、[NLP] 秒懂词向量 Word2vec 的本质

https://zhuanlan.zhihu.com/p/26306795

2、关于今日头条的写稿机器人

https://www.jiqizhixin.com/articles/2017-12-26-3
据说这篇 Paper 是做写稿机器人的那个组前一年发的，比较相关。
https://aclweb.org/anthology/P16-1129

3、对 dialogue system 感兴趣的同学可以查一查前两年的 alexa prize，然后看看他们系统的 paper，比如这篇 paper 来自 2017 的 alexa prize winner.

http://alexaprize.s3.amazonaws.com/2017/technical-article/soundingboard.pdf
这里是 2018 年的 alexa prize reports，每个 team 都有一个 report 讲他们的 system
https://developer.amazon.com/alexaprize/challenges/past-challenges/2018

4、语言模型的总结

https://blog.csdn.net/tiffanyrabbit/article/details/72650606
https://blog.csdn.net/TiffanyRabbit/article/details/72654180

5、特征工程书籍

http://fe4ml.apachecn.org/#/

6、pytorch 超全资源表

https://github.com/bharathgs/Awesome-pytorch-list

7、代码风格指南

https://github.com/IgorSusmelj/pytorch-styleguide

8、图解 Seq2Seq Attention 模型

https://zhuanlan.zhihu.com/p/40920384
https://blog.csdn.net/guohao_zhang/article/details/79540014

9、学习笔记：图像风格迁移

https://blog.csdn.net/czp_374/article/details/81185603

10、GAN 的原理入门

https://www.cnblogs.com/bonelee/p/9166084.html

11、如何理解 LSTM

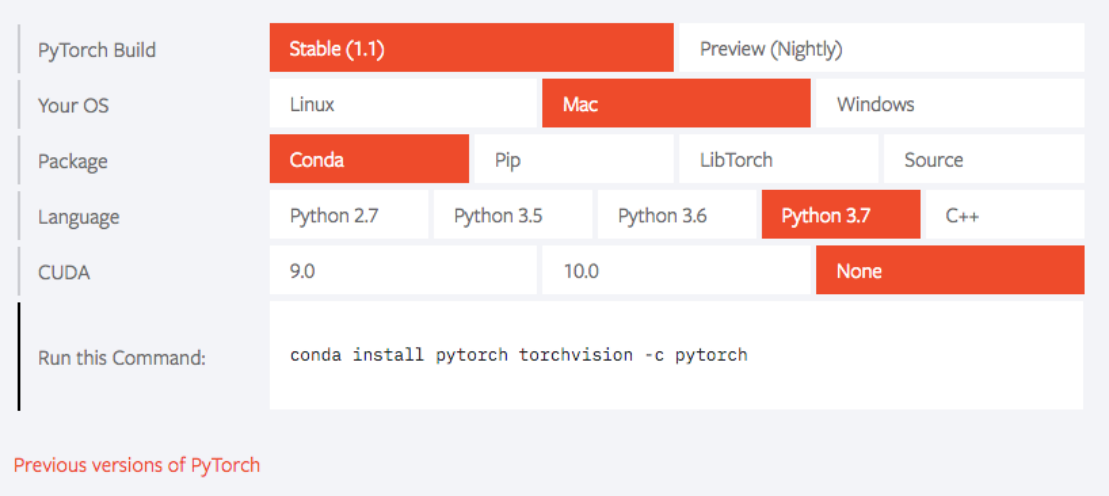
https://www.julyedu.com/question/big/kp_id/26/ques_id/1851

问题收集：

1、 如何安装 pytorch?

推荐官网安装，https://pytorch.org/

往下拉，可以看到安装向导，选中想要安装的格式，复制下面的安装命令到电脑终端（或 cmd）命令行即可。



注：

（1） 不管是 Mac 还是 Windows 暂不支持 GPU，需要有 NVIDA 的显卡才能用 GPU。

（2） 如果是用 conda 安装的话，可以去看一下 jupyter 和 pytorch 是不是装在了同一个环境下。比如在命令行看看 which jupyter 在哪个位置（linux 或 mac），如果显示在 anaconda 目录下，pyroch 也是用 conda 命令装的，那就没问题。

（3） 如果安装的是 anaconda，就用 conda 来安 pytorch。如果独立安装的 jupyter，就用安装 jupyter 的那个 pip 来安装。

2、 为什么 transpose 以后 view 就会报错了呢?

需要.contiguous()然后再 view。

一般来说是因为 transpose 改变了一些 memory 的结构，然后有些本来相邻的 memory 现在不相邻了，然后 pytorch 就需要你 contiguous()把这些

tensor 都搬到一个相邻的 memory 上。其实是一个他们不太好的设计，但是一直都需要这样做。

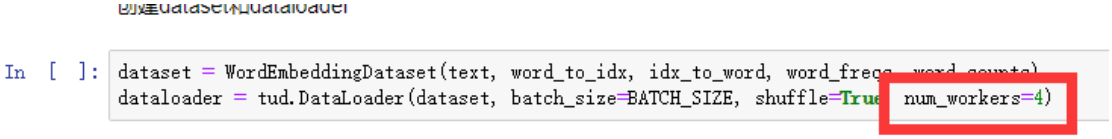
3、 如果分类时，除了文本，还有一个特征是一个数字，怎么将这个特征一起加进去?

一般可以拼接到文本处理后的向量，例如 wordavg 向量

4、 关于 chatbot?

实际上 chatbot 目前的大部分做法和翻译的 seq2seq + attn 模型非常类似，所以学完翻译之后只要有大的对话数据集可以用同样的方式训练。老实说现在的大部分关于 chatbot 的研究都比较水，其实没啥实质性进展。主要的难点都不在于模型的训练，而在于该怎么训练，用什么数据训练。大公司的做法有很多 pipeline 和手工的 engineering 在里面，例如在对话系统中加入 QA 系统, dialogue state tracking 等等。所以 chatbot 是一个很 open 的很大的 topic，涉及到很多的系统，所有的公司都在探索中。我会倾向于在最后几节课给大家讲一下大家做 chatbot 常见的几个 component，一个大的成熟的系统大家其实很难找得到，因为公司都在自己研发没有什么特别实用的开源的系统。网上一般能找到的开源的代码其实还是挺蠢的。

5、 出现 BrokenPipeError: [Errno 32] Broken pipe 错误解决办法?



以上参数 num_workers = 4 改成 1 就可以了，应该是多线程的问题。

6、 改成下面这样就报错了，是为什么呢?

```
    y_pred = h_relu @ w2                    # N * OUT

    #Loss
    loss = (y_pred - y).pow(2).sum()
    print('times is {}, loss is {}'.format(i, loss.item()))
    loss.backward()

    #Backward pass

    with torch.no_grad():
      w1 -= eta * w1.grad
      w2 -= eta * w2.grad
      w1.grad.zero_()
      w2.grad.zero_()
```

```
times is 0, loss is 27252062.0
times is 1, loss is 22909572.0
times is 2, loss is 21310926.0
times is 3, loss is 19621816.0
times is 4, loss is 16762486.0
times is 5, loss is 12897883.0
times is 6, loss is 9025428.0
times is 7, loss is 5888264.0
```

```
    with torch.no_grad():
      w1 = w1 - eta * w1.grad
      w2 = w2 - eta * w2.grad
      w1.grad.zero_()
      w2.grad.zero_()
```

```
times is 0, loss is 27188638.0
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-46-e44e6e45b8ea> in <module>()
     26     w1 = w1 - eta * w1.grad
     27     w2 = w2 - eta * w2.grad
---> 28     w1.grad.zero_()
     29     w2.grad.zero_()

AttributeError: 'NoneType' object has no attribute 'zero_'
```

因为 w1=w1- 是新建了地址保存 w1

因为 w1- = 是原地更新 w1

上面两种仅仅是在 pytorch 环境下，python 命令下两种方式的运行地址都变了。

而 with no_grad 说明生成的新对象 w1 没有梯度，没有梯度的 w1 是没有 zero_属性的。

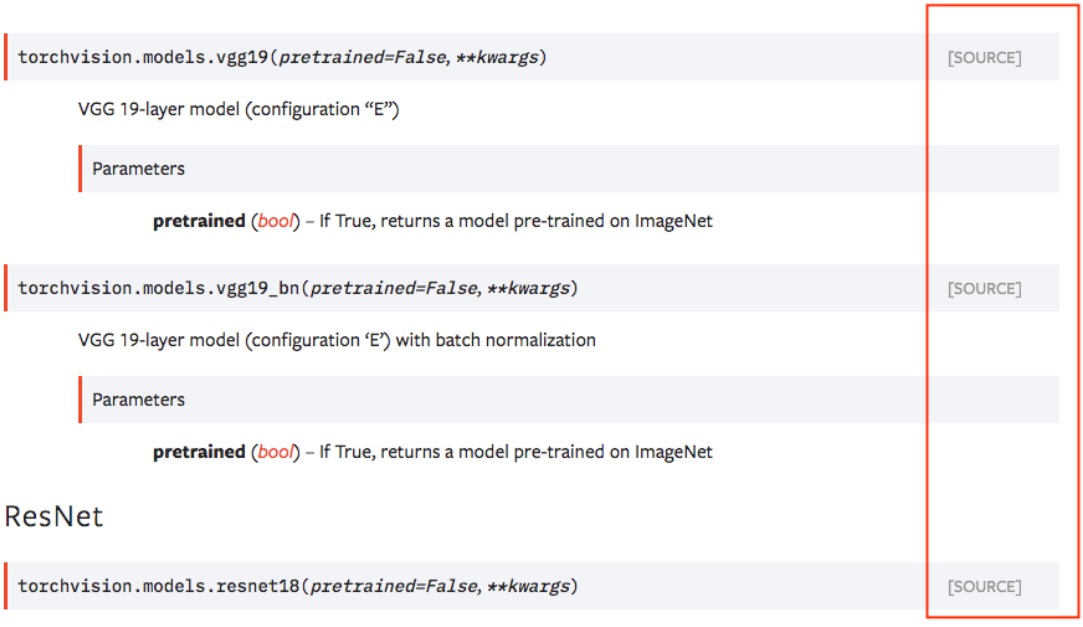
7、 用什么工具看 pytorch 的网络结构和监控 pytorch 网络的训练过程（是通过打印 loss 这种方式来看的吗?

一般直接打印。也可以用 tensorboardX

8、 对于一般的图像数据，做数据增强的目的是干什么？还有如果在训练集做数据增强，测试集是否也要做呢?

训练做数据增强是为了能够引入更多的训练数据，虽然很多时候引入的可能是 noise，但是还是希望可以让模型见到更多的训练数据，更多的情况。
测试集的目的是为了测试你的模型在真实数据上的效果，所以不要做任何人为的数据增强比较好。
在训练集样本不足的情况下，做一些人工的数据增强，来让模型训练的更好

9、 有没有 pytorch 实现常见的 alexnet、vgg 的代码呢 学习下?



`torchvision.models.vgg19(pretrained=False, **kwargs)` [SOURCE]

VGG 19-layer model (configuration “E”)

Parameters

pretrained (*bool*) – If True, returns a model pre-trained on ImageNet

`torchvision.models.vgg19_bn(pretrained=False, **kwargs)` [SOURCE]

VGG 19-layer model (configuration ‘E’) with batch normalization

Parameters

pretrained (*bool*) – If True, returns a model pre-trained on ImageNet

ResNet

`torchvision.models.resnet18(pretrained=False, **kwargs)` [SOURCE]

Constructs a ResNet-18 model

官网上点击 source 就可以看到源码。
也可以到 github 上搜也可以。
https://github.com/pytorch/vision/tree/master/torchvision/models

10、 第一课更新权重，为什么需要 with no_grad?

W1、W2 是没有梯度的，参考第第七题。
不用 with no_grad，那么 W1、W2 这个更新也在 autograd engine 里面，那么整个计算图就乱了。