信息网络专题研究课大作业（应用层）

一. 作业任务

1. 内容概述

在本作业中，你将实现循环网络，并将其应用于在微软的 COCO 数据库上进行图像标注。还将探索在 ImageNet 上可视化预训练模型特征的方法，并将该模型应用于风格迁移。

2. 设计目的

1) 理解循环神经网络（RNN）结构，知道它们是如何随时间共享权重来对序列进行操作的。
2) 理解并实现 Vanilla RNN 和长期短期记忆（LSTM）RNN。
3) 理解如何在测试时从 RNN 语言模型中进行采样。
4) 理解如何将卷积神经网络和循环神经网络结合在一起来实现图像标注。
5) 理解一个训练过的卷积神经网络是如何用来从输入图像中计算梯度的。
6) 实现图像梯度的不同应用，比如显著图，愚弄图像，类别可视化。
7) 理解并实现风格迁移。

二. 基础知识

1. Q1：使用普通 RNN 进行图像标注

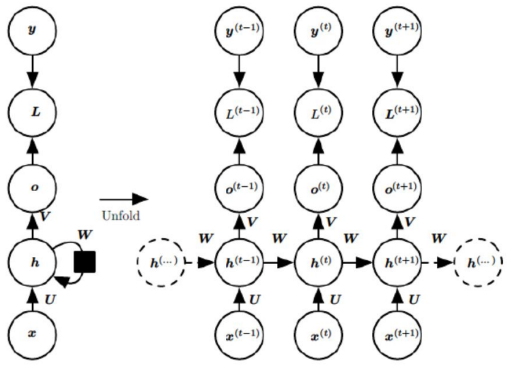
1) 问题描述

训练一个循环神经网络（RNN）来生成一个图片的文字注释（captions）。

2) 数据集

本节将用到的数据集是微软的 COCO 数据集，该数据集是测试为图片加文字注释算法的标准数据集。在该数据集中有大概 80000 张训练图片和 40000 张测试图片，每张图片在 Amazon Mechanical Turk 上征集了五个志愿者来手动写文字说明。数据集已进行了预处理并且通过 VGG-16 网络提取了特征，为了减少时间和内存需求，特征的维数从 4096 降到 512。

3) 循环神经网络（RNN）



x(t)指序列索引号t时训练样本的输入

h(t)指序列索引号t时模型的隐藏状态

o(t)指序列索引号t时模型的输出

L(t)指序列索引号t时模型的损失函数

y(t)指序列索引号t时训练样本序列的真实输出

U,W,V三个矩阵是模型的共享线性关系参数

RNN 是一种输出和模型间有反馈的神经网络，在层之间的神经元之间建立权连接。它广

泛的用于自然语言处理中的语音识别，手写书别以及机器翻译等领域。

RNN 前向传播算法：对于任意一个序列索引号 t，隐藏状态和预测输出分别为：

$$h(t) = \sigma(z(t)) = \sigma(Ux(t) + Wh(t-1) + b)$$

$$\hat{y}(t) = \sigma(o(t)) = \sigma(Vh(t) + c)$$

其中 σ 为 RNN 的激活函数，一般为 tanh，b 为线性关系的偏差值。

RNN 反向传播算法：通过梯度下降法一轮轮的迭代，得到合适的模型参数 U，W，V，由于参数在序列的各个位置是共享的，反向传播时更新相同的参数，激活函数嵌套在里面。

$$L = \sum_{t=1}^{n} L^{(t)} \qquad \frac{\partial L}{\partial \mathrm{V}} = \sum_{t=1}^{n} \frac{\partial L^{(t)}}{\partial o^{(t)}} \cdot \frac{\partial o^{(t)}}{\partial V}$$

$$\frac{\partial L^{(t)}}{\partial \mathrm{W}} = \sum_{k=0}^{t} \frac{\partial L^{(t)}}{\partial o^{(t)}} \cdot \frac{\partial o^{(t)}}{\partial h^{(t)}} \left(\prod_{j=k+1}^{t} \frac{\partial h^{(j)}}{\partial h^{(j-1)}} \right) \frac{\partial h^{(k)}}{\partial W}$$

$$\frac{\partial L^{(t)}}{\partial \mathrm{U}} = \sum_{k=0}^{t} \frac{\partial L^{(t)}}{\partial o^{(t)}} \cdot \frac{\partial o^{(t)}}{\partial h^{(t)}} \left(\prod_{j=k+1}^{t} \frac{\partial h^{(j)}}{\partial h^{(j-1)}} \right) \frac{\partial h^{(k)}}{\partial U}$$

2. Q2：使用 LSTM 进行图像标注

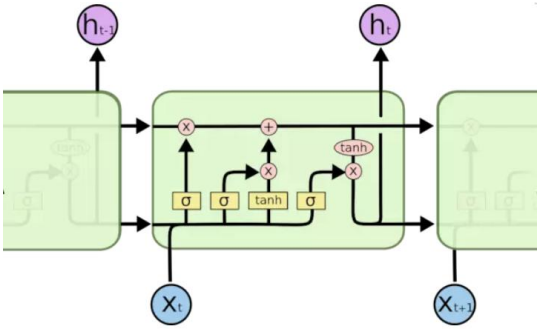
1) 问题描述

训练一个长短时记忆单元（LSTM）来生成一个图片的文字注释（captions）。

2) 数据集

同上

3) 长短时记忆单元（LSTM）



黄色矩形是学习得到的神经网络层

粉色圆形表示一些运算操作，诸如加法乘法

黑色单箭头表示向量的传输

两个箭头合成一个表示向量的连接

一个箭头分开表示向量的复制

LSTM 是 RNN 的一种变体，RNN 由于梯度消失的原因只能有短期记忆，LSTM 网络通过精妙的门控制将加法运算带入网络中，一定程度上解决了梯度消失的问题。不同于单一神经网络层，整体上除了 h 在随时间流动，细胞状态 c 也在随时间流动，细胞状态 c 就代表着长期记忆。而门是一种让信息选择式通过的方法，包含一个 sigmoid 神经网络层和一个乘法操作。LSTM 拥有三个门，来保护和控制细胞状态，虽然功能不同，但操作相同，都是使用 sigmoid 作为选择工具，tanh 作为变换工具，即 sigmoid 选择更新内容，tanh 创建更新候选。

遗忘门： $i_t = \sigma(W_i \cdot [h_{t-1},x_t] + b_i)$ $\widetilde{C}_t = \tanh(W_C \cdot [h_{t-1},x_t] + b_C)$

输入门： $f_t = \sigma(W_f \cdot [h_{t-1},x_t] + b_f)$ $C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t$

输出层： $o_t = \sigma(W_o \cdot [h_{t-1},x_t] + b_o)$ $h_t = o_t * \tanh(C_t)$

3. Q3：网络可视化：显著图，愚弄分类器和类可视化

1) 问题描述

探索图像梯度对于生成新图片的用法。

2) 图像梯度

当训练模型的时候，通常会定义一个损失函数来表示对当前模型性能的不满意程度（和标准答案的差异程度），接着用反向传播来计算损失函数对于模型参数的梯度，并且在模型参数上用梯度下降来最小化损失（loss）。

本节首先将使用一个在 ImageNet 数据集上做了预训练的卷积神经网络模型 SqueezeNet，接着用这个模型来定义一个损失函数，接着用反向传播来计算损失函数对于图片像素的梯度，保持模型不变，对图片进行梯度下降，从而生成一张新图片来降低损失（loss）。

3) 网络可视化

本次作业主要探索三种图片生成的技巧：

① 显著图（Saliency Maps）：可以很快说明图片中的哪些部分影响了模型对于最后分类 label 的判断。

② 愚弄分类器（Fooling Images）：扰乱一个输入的图片，使它对于人类来说看起来是一样的，但是会被训练的模型分错类。

③ 类可视化（Class Visualization）：合成一张图片来最大化一个特定类的打分，可以给我们一些直观感受，即模型在判断图片是当前类时最关注的是图片的哪些部分。

4. Q4：风格迁移

1) 问题描述

风格迁移主要目的是将一幅图的艺术风格转移到另一幅图中去。

2) 损失函数

为了实现上述目的，首先需要使用相应的损失函数以描述在神经网络的特征空间中各图片的内容以及风格，之后对图片中的像素使用梯度下降来使图片向目标风格转变，本节使用的网络是 SqueezeNet，成本函数的计算包括内容成本和风格成本。

内容成本：浅层的一个卷积网络往往检测到较低层次的特征，如边缘和简单的纹理，更深层往往检测更高层次的特征，如更复杂的纹理以及对象分类等。为了生成的图像 G 具有与输入图像 C 相似的内容，通常选择网络中间的一层，以得到最好的的视觉结果。F 表示当前图像的特征图，P 表示源图的特征图，内容成本函数定义为：

$$J_{content}(C,G)=\sum_{i,j}(F_{ij}^{l}-P_{ij}^{l})^{2}$$

风格成本：首先通过将降维的过滤器矩阵与其转置相乘来计算风格矩阵，维度为（nC,nC），其中 nC 是过滤器的数量，Gij 测量了过滤器 i 的激活与过滤器 j 的激活具有多大的相似度。通过捕捉不同类型的特征（Gii）的多少，以及特征的数量，风格矩阵 G 就能测量整个图片的风格。在生成了风格矩阵（Gram matrix / Style matrix）之后，我们的目标是最小化风格图像的 S 与生成的图像 G 之间的距离。G 是由当前图片的特征图算得的 Gram 矩阵，A 是源图片的特征图算得的 Gram 矩阵，单个隐藏层的风格成本定义如下

$$J_{style}(S,G)=\sum_{ij}(G_{ij}^{l}-A_{ij}^{l})^{2}=\sum_{ij}(\sum_{k}F_{ij}^{l}F_{jk}^{l}-A_{ij}^{l})^{2}$$

最后创建一个最小化风格和内容成本的成本函数：

$$J(G)=\alpha J_{content}(C,G)+\beta J_{style}(S,G)$$

三．思路方法

1. Q1：使用普通 RNN 进行图像标注

1) 代码要点（rnn_layers.py + rnn.py）

① RNN 的单步前向传播 rnn_step_forward()：根据公式计算新的网络隐藏状态

② RNN 的单步反向传播 rnn_step_backward()：超正切函数求导 $tanh^{'}x=1-tanh^{2}x$

③ RNN 的前向传播 rnn_forward()：读取一批的标注数据 x，样本数为 N，每条标注的长度为 T，使用这批标注对应图片特征作为网络的初始隐藏状态 h0，通过前向传播过程，获得各个样本在每一步推进中产生的隐藏状态 h，并存储反向传播所需变量

④ RNN 的反向传播 rnn_backward()：利用存储的变量实现反向传播，梯度值累积

⑤ 字词向量化表达 word_embedding_forward()：将图像标注中的词索引 x 转化为向量

⑥ 字词向量化表达 word_embedding_backward()：后向传播时更新字词所对应的向量

⑦ 损失函数 loss()：输入图像特征和正确图片注释，用网络计算损失函数和参数梯度，训练模型，输出对图片注释序列的预测

⑧ 测试过程 sample()：上述步骤中真实的图片注释换成隐藏层输出得分最高的词汇，即给定图片向量，自己生成一个对应的 caption

2) 梯度检验

一方面，利用数字计算的方法，估算 f'(x)，另一方面，利用设计的梯度算法计算梯度，看两者是否近似，如果相差很小，说明梯度算法正确。

函数	实际误差（max）	理论误差
rnn_step_forward()	6e-9	< 1e-8
rnn_step_backward()	2e-10	< 1e-8
rnn_forward()	8e-8	< 1e-7

rnn_backward()	1e-7	< 5e-7
word_embedding_forward()	1e-8	< 1e-8
word_embedding_backward()	3e-12	<1e-11
loss()	3e-12	<1e-10

2. Q2：使用 LSTM 进行图像标注

1) 代码要点（rnn_layers.py + rnn.py）

① LSTM 的单步前向传播 lstm_step_forward()：先计算一个激活向量，然后将该向量分成四个子向量，分别用以计算输入门，遗忘门，输出门和下一个细胞状态。

② LSTM 的单步反向传播 lstm_step_backward()：在多一条关于C_t的传播路径下应用求导的链式法则

③ LSTM 的前向传播 lstm_forward()：读取一小批的标注数据 x，进行前向传播

④ LSTM 的反向传播 lstm_backward()：利用存储的变量实现反向传播，梯度值累积

2) 梯度检验

过程同上

函数	实际误差（max）	理论误差
lstm_step_forward()	6e-9	< 1e-8
lstm_step_backward()	3e-8	< 1e-6
lstm_forward()	9e-8	< 1e-7
lstm_backward()	9e-7	< 1e-7
loss()	2e-12	<1e-10

3. Q3：网络可视化：显著图，愚弄分类器和类可视化

1) 代码要点（NetworkVisualization-PyTorch.ipynb）

① 显著图 compute_saliency_maps()：计算网络终端的该图像的正确类别的得分对于图像像素的梯度，反映了某个像素点值的改变对于分类的打分影响的程度大小

② 愚弄分类器 make_fooling_image()：计算某图片在一个错误分类上的得分关于图片各像素的梯度，然后使用梯度上升法，修改图片，使得网络发生误判。

③ 类可视化 create_class_visualization()：优化目标函数定为某分类的得分再加上正则项；将图片初始为随机噪声，使用梯度上升法，生成最符合该类别的图片。

4. Q4：风格迁移

1) 代码要点（StyleTransfer-PyTorch.ipynb）

① 内容偏差 content_loss()：是生成图片和源内容图片在特征空间中的差异描述

② 风格偏差 style_loss()：是生成图片和源风格图片的 Gram 矩阵之间的差异描述

③ 图像变分惩罚项 tv_loss()：进行整体方差正则化，描述图像的平滑度

四、结果展示

1. Q1：使用普通 RNN 进行图像标注

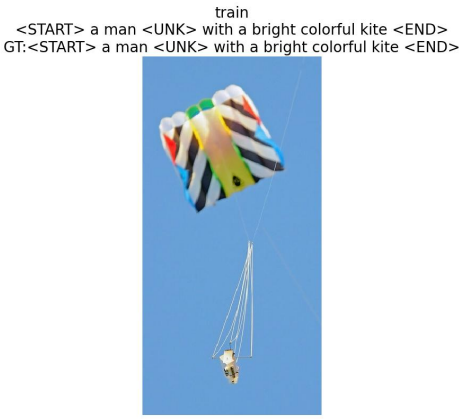
1) 小型数据集应用 RNN 模型训练结果

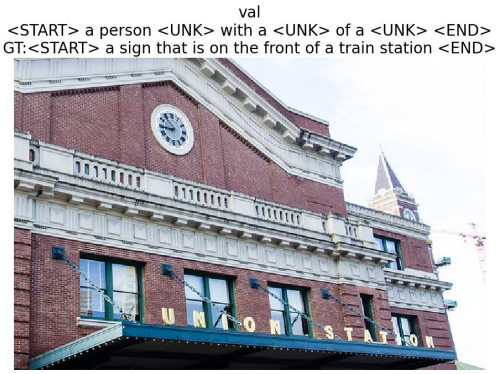
2) RNN 模型应用于图像标注

2. Q2：使用 LSTM 进行图像标注

1) 小型数据集应用 RNN 模型训练结果

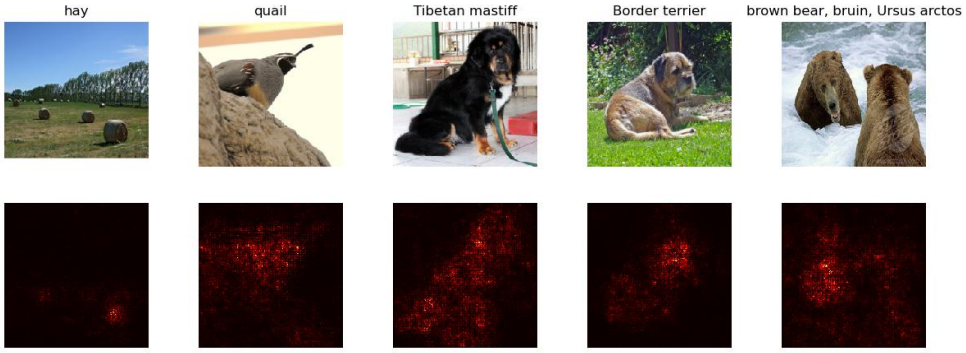
2)　RNN 模型应用于图像标注



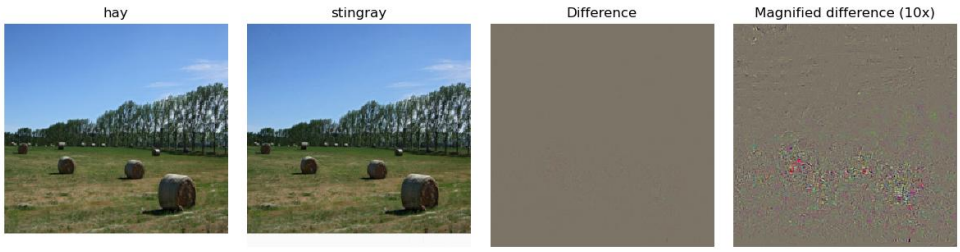


3. Q3：网络可视化：显著图，愚弄分类器和类可视化

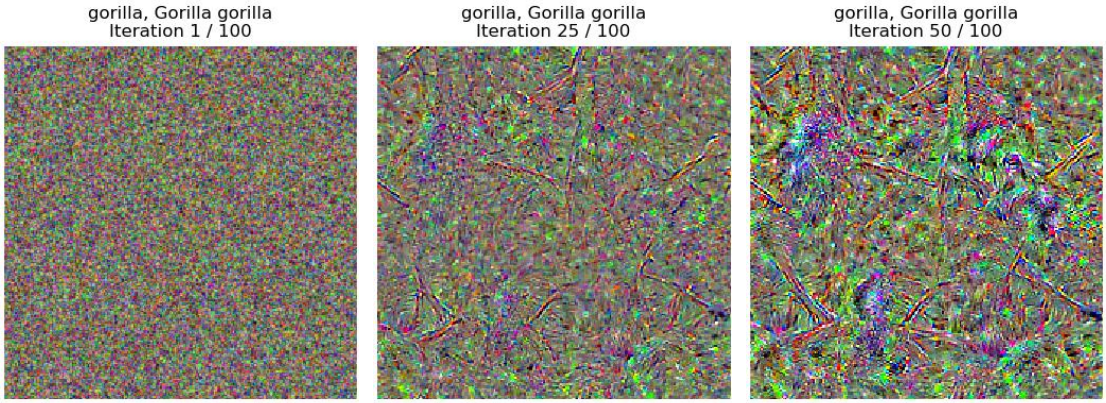
1)　显著图，影响模型对于 label 判断的像素点

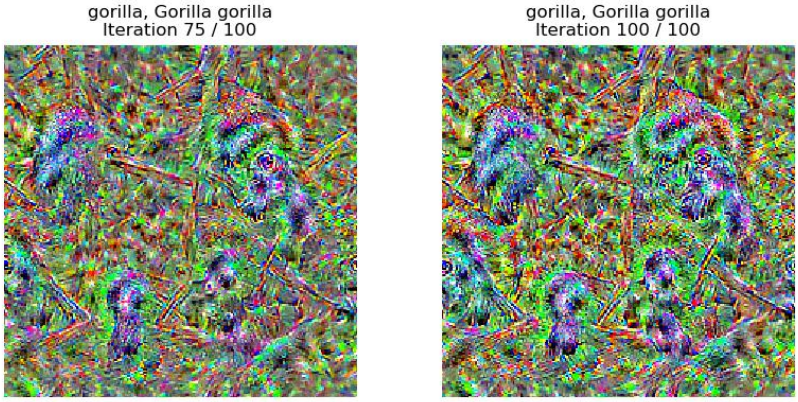


2)　愚弄分类器，网络误判的图片与原图的差异



3)　类可视化，对猩猩类别的反演结果





4. Q4：风格迁移

1) 实现风格迁移

五、思考总结

1. 遇到的问题

1) 在导入数据集的时候发现 windows 下不能运行.sh 文件，依照网上教程安装了 linux 子系统以及 ubantu 软件

2) 在调试程序的时候发现安装包版本不兼容的问题，按照相应操作对某些包进行了降级

3) 对误差不满足要求的子函数在测试时进行了调整

4) 无法直接运行.ipynb 文件，复制相应内容到新建对.py 文件

2. 收获与感想

在本次大作业中，我完整地运行了整个作业任务。从 python 程序与包的安装，到数据集的下载，到基本理论知识的学习，到英文文本提示的阅读，到程序的补充以及调试，最后是结果的可视化展现以及报告的书写。通过此次大作业的完成，我收获很多，除了神经网络方面理论知识的初步掌握，更重要的是提高了独立分析问题，解决问题的能力。