Write, Attend and Spell: End-to-end Free-style Handwriting Recognition Using Smartwatches

基础知识

GRU

GRU（Gate Recurrent Unit）是循环神经网络（Recurrent Neural Network, RNN）的一种。和LSTM（Long-Short Term Memory）一样，也是为了解决长期记忆和反向传播中的梯度等问题而提出来的。

与LSTM相比的优势：我们在我们的实验中选择GRU是因为它的实验效果与LSTM相似，但是更易于计算。

GRU的输入输出结构：：

在当前时刻输入x^t和上一时刻隐状态h^{t-1}，结合新的输入和上一时刻输出，GRU得到当前隐节点的输出y^t和要传给下一时刻的隐状态h^t。

内部结构：两个门r和z，r控制重复，z控制更新，r和z使用sigmoid函数。

$h^{t-1'} = h^{t-1} \cdot r$

将$h^{t-1'}$和x^t拼接通过tanh得到h'，h'主要包含了当前的输入数据

更新表达式：$h^t = (1-z) \cdot h^{t-1} + z \cdot h'$。

我们使用了同一个门控 z 就同时可以进行遗忘和选择记忆（LSTM则要使用多个门控）。

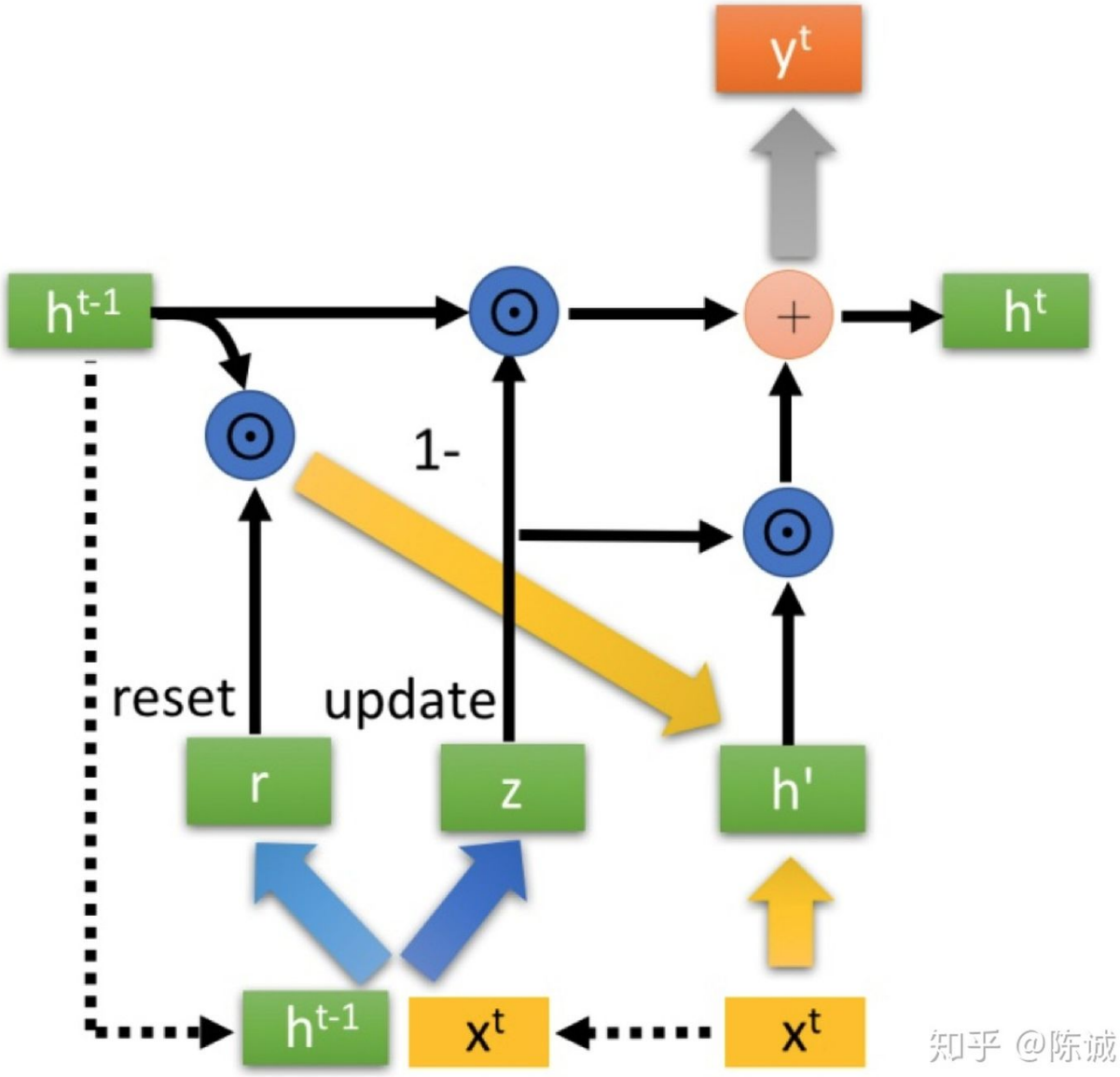
$(1-z)\cdot h^{t-1}$表示对隐藏状态的遗忘

$z\cdot h'$表示对当前输入信息的选择性记忆

可以看到，这里的选择z和遗忘 $1-z$ 是联动的。也就是说，对于传递进来的维度信息，我们会进行选择性遗忘，则遗忘了多少权重 $(1-z)$，我们就会使用包含当前输入的 h' 中所对应的权重进行弥补 z 。以保持一种”恒定“状态。

与LSTM的对比：我们这里的1-z相当于LSTM的遗忘门，z相当于LSTM的输入选择门。这里的输入h'相当于LSTM中的输入z，但是LSTM中z直接由x^t和h^{t-1}拼接而来，这里的z拼接时先对h^{t-1}做了一次重复选择。

GRU cell:



LSTM cell:

注意力机制

当我们人在看一样东西的时候，我们当前时刻关注的一定是我们当前正在看的这样东西的某一地方，换句话说，当我们目光移到别处时，注意力随着目光的移动也在转移，这意味着，当人们注意到某个目标或某个场景时，该目标内部以及该场景内每一处空间位置上的注意力分布是不一样的。

注意力机制包含 一下两个部分：

1. 注意力机制需要决定整段输入的哪个部分需要更加关注；
2. 从关键的部分进行特征提取，得到重要的信息。

从Attention的作用角度出发，我们就可以从两个角度来分类Attention种类：**Spatial Attention 空间注意力(图片)**和**Temporal Attention 时间注意力(序列)**。更具实际的应用，也可以将Attention分为**Soft Attention**和**Hard Attention**。**Soft Attention是所有的数据都会注意，都会计算出相应的注意力权值，不会设置筛选条件。Hard Attention会在生成注意力权重后筛选掉一部分不符合条件的注意力，让它的注意力权值为0，即可以理解为不再注意这些不符合条件的部分。**

Encoder-Decoder框架可以这么直观地去理解：可以把它看作适合处理由一个句子（或篇章）生成另外一个句子（或篇章）的通用处理模型。对于句子对<X,Y>，我们的目标是给定输入句子X，期待通过Encoder-Decoder框架来生成目标句子Y。X和Y可以是同一种语言，也可以是两种不同的语言，而X和Y分别由各自的单词序列构成。

理解AM模型的关键就是这里，即由固定的中间语义表示C换成了根据当前输出单词来调整成加入注意力模型的变化的Ci。

attention机制的本质思想：

对于一个q，计算q与编码层输出所有向量的相似度，然后对其进行加权求和得到attention value c_t。

在解码器中，$y_t = Decoder(h_{t-1}, c_t, y_{t-1})$。将attention value、解码器上一时刻的隐藏状态、解码器上一时刻的输出送入到解码器中求得下一时刻的输出，然后通过FC层得到输出。

步骤：

- 首先计算query和key的相似度或者相关性，计算方式有：点积、cosine相似度、MLP网络等。
- 第一步产生的分值根据具体产生的方法不同其数值取值范围也不一样，第二步引入类似SoftMax的计算方式对第一步的得分进行数值转换，**一方面可以进行归一化**，将原始计算分值整理成所有元素权重之和为1的概率分布；另一方面也可以通过SoftMax的内在机制更加突出重要元素的权重。
- 将第二步的计算结果做为Value对应的权重系数，然后进行加权求和即可得到Attention数值。

详见知乎

个人认为：比较重要的是理清楚哪个是query，哪个是key和value。

用在当前论文里，LSTM可以学习到字母之间的关系，加入attention可以使得字母的上下文关系更关注与其相关的重点字母，增加精度。

CTCLoss

基于RNN文字识别算法主要有两个框架：

1. CNN+RNN+CTC(CRNN+CTC)
2. CNN+Seq2Seq+Attention

整个CRNN网络可以分为三个部分：

假设输入图像大小为 $(32, 100, 3)$，注意提及图像都是 $(\mathrm{Height}, \mathrm{Width}, \mathrm{Channel})$ 形式。

- Convlutional Layers

这里的卷积层就是一个普通的CNN网络，用于提取输入图像的Convolutional feature maps，即将大小为 $(32, 100, 3)$ 的图像转换为 $(1, 25, 512)$ 大小的卷积特征矩阵。

- Recurrent Layers

这里的循环网络层是一个深层双向LSTM网络，在卷积特征的基础上继续提取**文字序列特征**。

由于CNN输出的Feature map是$(1, 25, 512)$大小，所以对于RNN最大时间长度 $T = 25$ （即有25个时间输入，每个输入 x_t 列向量有 $D = 512$ ）。

- Transcription Layers

将RNN输出做softmax后，为字符输出。

对于Recurrent Layers，如果使用常见的Softmax cross-entropy loss

CTC是一种Loss计算方法，用CTC代替Softmax Loss，训练样本无需对齐。CTC特点：

- 引入blank字符，解决有些位置没有字符的问题
- 通过递推，快速计算梯度

CTC的做法：

对于给定LSTM的输入x的情况下，输出为l的概率为：$p(l|x) = \sum_{\pi \in B_l^{-1}} p(\pi|x)$。其中$\pi \in B_l^{-1}$表示经过B变换（去掉-以及-间重复的字母之后的结果）之后是l的路径π。

那么，对于任意一条路径π有：$p(\pi|x) = \prod_{t=1}^{T} y_{\pi_t}^t$，表示在路径中t时刻的输出为$\pi_t$。

eg：对于T=12的路径π_1，$\pi_1 = (--stta-t---e)$，
$p(\pi_1|x) = y_{\_}^1 y_{\_}^2 y_s^3 y_t^4 y_t^5 y_a^6 y_{\_}^7 y_t^8 y_{\_}^9 y_{\_}^{10} y_{\_}^{11} y_e^{12}$。在实际计算中，$\pi \in B_l^{-1}$路径会有很多条，无法一一计算，因此需要一种快速计算方法。

CTC借用了HMM的"向前—向后"(forward-backward)算法来计算 $p(l|x)$，通过调整参数w使得输入样本为$\pi \in B_l^{-1}$时$p(l|x)$达到最大。

具体算法见知乎

CRNN+CTC总结

- 首先CNN提取图像卷积特征
- 然后LSTM进一步提取图像卷积特征中的序列特征

- 最后引入CTC解决训练时字符无法对齐的问题

处理流程

信号预处理

单词分割

在这篇论文中，本质上依旧无法解决连续输入的问题，只是将按照字母分割识别改为了按照单词分割来识别，在每一个单词中间会有短暂的间隔，通过计算每一时刻的能量来推断此处是否应该间隔，通过在一个短暂片段内的求和求得能量A，然后与一个阈值比较来判断是否处于单词间隔期 。

数据增强

由于每个人写字的速度、字体大小、手表朝向都有所不同，因此需要进行矫正。

对于写字速度的不同，使用time wrap技术将每一个字的时常都压缩拓展到固定的时间长度。

对于不同的字体大小，在组标轴幅度上进行压缩。

对于手表朝向不同的问题，通过提取重力传感器的数据，使用一个旋转矩阵来矫正不同位置（具体实现可以忽略，我们暂时不会遇到这个问题）。

网络结构

多模态CNN

由于使用手表可以获取到来自陀螺仪、加速度计、重力传感器3个数据，这3个数据对于手写姿势的描述是单独的，因此使用多模态CNN，使用3个CNN去提取特证，最后采用3×3的二维卷积来提取跨模态特征，通过一个FC层后送入GRU。

多任务编码-解码网络

这里与基础知识中讲诉的比较重复，使用CNN去提取图像中的特证，送入GRU进行编码，将编码器的输出通过一个softmax层来得到每一个字母的输出概率得到预测输出的路径，然后通过CTC计算Loss，同时使用attention机制的seq2seq model来显式的使用历史来推断当前标签，然而注意对齐不具有从左到右的单调性约束，因为它允许解码器参与每一步的输出顺序。将ctc和attention融合，
$L_{word} = (1-\lambda)L_{attentiion} + L_{ctc}$。

实验验证

基本介绍

评价指标：CER。

并且在训练阶段，对每个样本进行数据增强，使得样本数量翻20倍。

独立用户测试

使用5折交叉验证CER为0.6%，数据集中单词的平均长度为4.32。

非独立用户测试

平均CER为8.1%，范围为$4.7\% - 11.8\%$。

生词测试

每次选择20个单词作为测试集，CER为9.7%。

....(其他略)