Gated Recurrent Convolution Neural Network for OCR

2018 年 6 月 30 日 星期日　　杜臣

摘要：

1、在原来的 Recurrent Convolution Layer (RCL)基础上,添加了 gate 设计了 Gated RCNN(GRCNN)进行场景字符串识别。通过添加门（gate）来控制循环卷积层的感受野，平衡其中的前向信息和循环信息。

2、用于字符串识别的整体网络结构为 GRCNN-BLSTM-CTC，实验证明 GRCNN 能较好的进行特征表示。

背景：

Gated Recurrent Convolution Neural Network (GRCNN)

1、循环连接在大脑皮层中广泛存在，而且循环连接在视觉识别中对上下文语义的融合起着重要的作用。前向模型只能获取拥有更大感受野的高层神经元隐含的上下文信息（语义信息，contex），却没有更好的融合用来识别小目标的低层神经元。Recurrent Convolution Layer (RCL)在卷积层中添加了循环卷积来扩大单层卷积层的感受野以及融合高底层信息。

RCNN 结构如下：详见论文 Recurrent Convolutional Neural Network for Object Recognition http://openaccess.thecvf.com/content_cvpr_2015/papers/Liang_Recurrent_Convolutional_Neural_2015_CVPR_paper.pdf

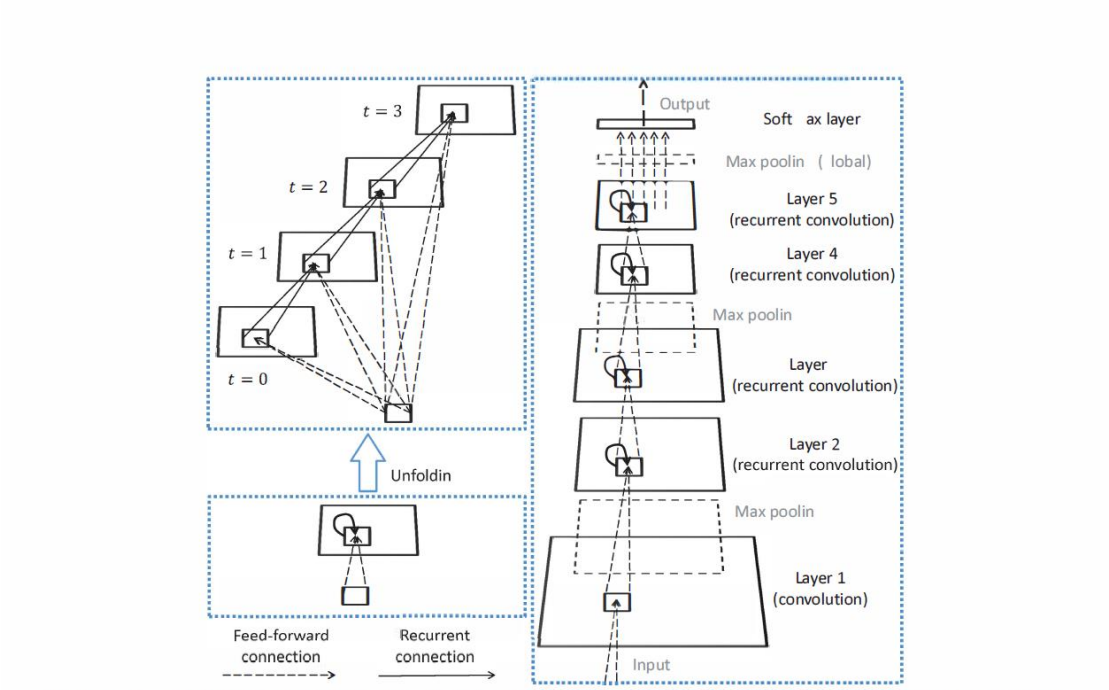

Figure 3. The overall architecture of RCNN. Left: An RCL is unfolded for $T = 3$ time steps, leading to a feed-forward subnetwork with the largest depth of 4 and the smallest depth of 1. At $t = 0$ only feed-forward computation takes place. Right: The RCNN used in this paper contains one convolutional layer, four RCLs, three max pooling layers and one softmax layer.

2、在 RCNN 中，如果增加迭代次数（T），每个卷积层的有效感受野将急剧增加，（与所基于的生物学事实不符），所以需要引入一个控制机制来限制有效感受野的增长。

3、为提升图像特征提取的性能，需要控制网络中神经元对周围语义信息的融合，如下图中

字符串识别中，对于‘h’的识别，RCNN 最大的感受野应该控制到覆盖整个‘h’字符，但是当卷积核的感受野大到覆盖到字符‘p’时，此时卷积核并没有带来更好的的 context（上下文信息），所以在 RCNN 中应该弱化一些不必要的 contex 以及提供一种更灵活的方式来融合前向信息和循环信息，于是提出了在 RCNN 中加入 gate 实现以上的目的。

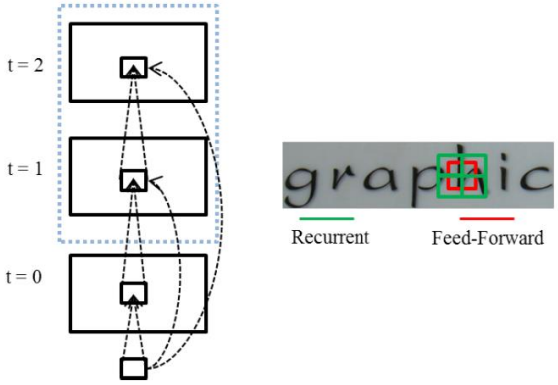

Figure 1: Illustration of using RCL with $T = 2$ for OCR.

论文主要内容：

1、 Recurrent Convolution Layer and RCNN

The RCL [22] is a module with recurrent connections in the convolutional layer of CNN. Consider a generic RNN model with feed-forward input $u(t)$. The internal state $x(t)$ can be defined as:

$$x(t) = \mathcal{F}(u(t), x(t-1), \theta) \tag{1}$$

where the function $\mathcal{F}$ describes the nonlinearity of RNN (e.g. ReLU unit) and θ is the parameter. The state of RCL evolves over discrete time steps:

$$x(t) = \mathcal{F}((w^f * u(t) + w^r * x(t-1)) \tag{2}$$

where "*" denotes convolution, $u(t)$ and $x(t-1)$ denote the feed-forward input and recurrent input respectively, w^f and w^r denote the feed-forward weights and recurrent weights respectively.

2、 Gated Recurrent Convolution Layer and GRCNN

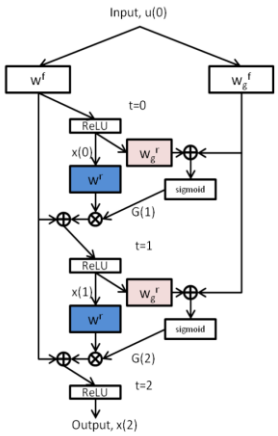

Figure 2: Illustration of GRCL with $T = 2$. The convolutional kernels in the same color use the same weights.

门控信号的计算：

$$G(t) = \begin{cases} 0 & t = 0 \\ sigmoid(BN(w_g^f * u(t)) + BN(w_g^r * x(t-1))) & t > 0 \end{cases} \tag{3}$$

Inspired by the Gated Recurrent Unit (GRU) [4], we let the controlling gate receive the signals from the feed-forward input as well as the states at the last time step. We use two 1×1 kernels, w_g^f and w_g^r, to convolve with the feed-forward input and recurrent input separately. w_g^f denotes the feed-forward weights for the gate and w_g^r denotes the recurrent weights for the gate. The recurrent weights are shared over all time steps (Figure 2). Batch normalization (BN) [13] is used to improve the performance and accelerate convergence. The GRCL can be described by:

每一时刻输出状态的计算：

$$x(t) = \begin{cases} ReLU(BN(w^f * u(t)) & t = 0 \\ ReLU(BN(w^f * u(t)) + BN(BN(w^r * x(t-1)) \odot G(t))) & t > 0 \end{cases} \tag{4}$$

In the equations, "$\odot$" denotes element-wise multiplication. Batch normalization (BN) is applied after each convolution and element-wise multiplication operation. The parameters and statistics in BN are not shared over different time steps. It is assumed that the input to GRCL is the same over time t, which is denoted by $u(0)$. This assumption means that the feed-forward part contributes equally at each time step. It is important to clarify that the time step in GRCL is not identical to the time associated with the sequential data. The time steps denote the iterations in processing the input.

3、Overall Architecture GRCNN-BLSTM Model

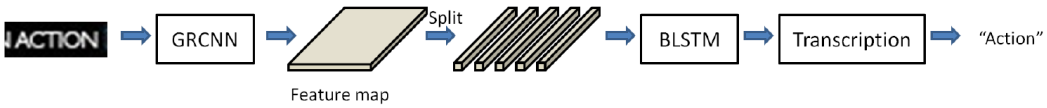

Figure 3: Overall pipeline of the architecture.

使用 GRCNN 做特征提取，然后接入 BLSTM 和 CTC 进行识别

使用的特征提取 GRCNN 结构如下：

Table 1: The GRCNN configuration

Conv	MaxPool	GRCL	MaxPool	GRCL	MaxPool	GRCL	MaxPool	Conv
3×3	2×2	3×3	2×2	3×3	2×2	3×3	2×2	2×2
num: 64		num: 64		num: 128		num: 256		num: 512
sh:1 sw:1	sh:2 sw:2	sh:1 sw:1	sh:2 sw:2	sh:1 sw:1	sh:2 sw:1	sh:1 sw:1	sh:2 sw:1	sh:1 sw:1
ph:1 pw:1	ph:0 pw:0	ph:1 pw:1	ph:0 pw:0	ph:1 pw:1	ph:0 pw:1	ph:1 pw:1	ph:0 pw:1	ph:0 pw:0

num: denotes the number of feature maps

sh: denotes the stride of the kernel along the height;

sw: denotes the stride along the width;

"ph" and "pw" denote the padding value of height and width respectively;

实验：

数据集：ICDAR2003、IIIT5K、Street View Text (SVT)、Synth90k

训练细节：The input is a gray-scale image which is resized to 100×32. Before input to the network, the pixel values are rescaled to the range (-1, 1). The final output of the feature extractor is a feature sequence of 26 frames. The recurrent layer is a bidirectional LSTM with 512 units without dropout. The ADADELTA method is used for training with the parameter ρ=0.9. The batch size is set to 192 and training is stopped after 300k iterations.

实验 1、迭代次数（T）对性能的影响，加入 gate 后对性能的影响

实验 2、不同的 lstm 结构对性能的影响

Table 2: Model analysis over the IIIT5K and SVT (%). Mean and standard deviation of the results are reported.

(a) GRCNN analysis

Model	IIIT5K	SVT
Plain CNN	77.21±0.54	77.69±0.59
RCNN(1 iter)	77.64±0.58	78.23±0.56
RCNN(2 iters)	78.17±0.56	79.11±0.63
RCNN(3 iters)	78.94±0.61	79.76±0.59
GRCNN(1 iter)	77.92±0.57	78.67±0.53
GRCNN(2 iters)	79.42±0.63	79.89±0.64
GRCNN(3 iters)	80.21±0.57	80.98±0.60

(b) LSTM's variants analysis

LSTM variants	IIIT5K	SVT
LSTM$_{\{\gamma_1=0,\gamma_2=0,\gamma_3=0\}}$	77.92±0.57	78.67±0.53
LSTM-F$_{\{\gamma_1=0,\gamma_2=1,\gamma_3=0\}}$	77.26±0.61	78.23±0.53
LSTM-I$_{\{\gamma_1=1,\gamma_2=0,\gamma_3=0\}}$	76.84±0.58	76.89±0.63
LSTM-O$_{\{\gamma_1=0,\gamma_2=0,\gamma_3=1\}}$	76.91±0.64	78.65±0.56
LSTM-A$_{\{\gamma_1=1,\gamma_2=1,\gamma_3=1\}}$	76.52±0.66	77.88±0.59

实验 3、：整体的 GRCNN-BLSTM Model 性能与已有的方法的比较

Table 3: The text recognition accuracies in natural images. "50","1k" and "Full" denote the lexicon size used for lexicon-based recognition task. The dataset without lexicon size means the unconstrained text recognition

Method	SVT-50	SVT	IIIT5K-50	IIIT5K-1k	IIIT5K	IC03-50	IC03-Full	IC03
ABBYY [36]	35.0%	-	24.3%	-	-	56.0%	55.0%	-
wang et al. [36]	57.0%	-	-	-	-	76.0%	62.0%	-
Mishra et al. [25]	73.2%	-	-	-	-	81.8%	67.8%	-
Novikova et al. [27]	72.9%	-	64.1%	57.5%	-	82.8%	-	-
wang et al. [38]	70.0%	-	-	-	-	90.0%	84.0%	-
Bissacco et al. [3]	90.4%	78.0%	-	-	-	-	-	-
Goel et al. [6]	77.3%	-	-	-	-	89.7%	-	-
Alsharif [2]	74.3%	-	-	-	-	93.1%	88.6%	-
Almazan et al. [1]	89.2%	-	91.2%	82.1%	-	-	-	-
Lee et al. [20]	80.0%	-	-	-	-	88.0%	76.0%	-
Yao et al. [40]	75.9%	-	80.2%	69.3%	-	88.5%	80.3%	-
Rodriguez et al. [28]	70.0%	-	76.1%	57.4%	-	-	-	-
Jaderberg et al. [16]	86.1%	-	-	-	-	96.2%	91.5%	-
Su and Lu et al. [33]	83.0%	-	-	-	-	92.0%	82.0%	-
Gordo [7]	90.7%	-	93.3%	86.6%	-	-	-	-
Jaderberg et al. [14]	93.2%	71.1%	95.5%	89.6%	-	97.8%	97.0%	89.6%
Baoguang et al. [30]	**96.4%**	80.8%	97.6%	94.4%	78.2%	98.7%	97.6%	89.4%
Chen-Yu et al. [21]	96.3%	80.7%	96.8%	94.4%	78.4%	97.9%	97.0%	88.7%
ResNet-BLSTM	96.0%	80.2%	97.5%	94.9%	79.2%	98.1%	97.3%	89.9%
Ours	96.3%	**81.5%**	**98.0%**	**95.6%**	**80.8%**	**98.8%**	**97.8%**	**91.2%**