



## 一种 Petri 网优化的验证码识别方法

马金林, 陈德光, 马自萍, 魏 麟

(北方民族大学 计算机科学与工程学院, 银川 750021)

**摘 要:** 针对 AlexNet 网络对验证码 (CAPTCHA) 多目标分类问题效果不理想、模型参数量与浮点数计算量过大的问题, 提出一种基于 Petri 网优化的 CAPTCHA 识别方法。利用 Petri 网理论对 AlexNet 和 DenseNet-BC 建模, 并通过所建模型优化网络结构和参数。同时, 根据模型参数量与浮点数计算量的关系, 提出超活性概念, 对 Petri-ANPP-net、Petri-ANPS-net、Petri-DNBC-net 模型进行灵敏度分析。实验结果表明, 经过 Petri 网优化后, Petri-ANPP-net 模型的最高准确度为 60.40%, 且超活性较小, 模型灵敏度较差, Petri-ANPS-net 模型的最高准确度为 97.50%, 但超活性较小, 模型灵敏度较差, Petri-DNBC-net 模型的最高准确度达到 99.24%, 且超活性较大, 模型灵敏度较高。说明 Petri 网能在一定程度上优化网络模型结构和参数, 且超活性对于评价模型的灵敏度具有一定的优越性。

**关键词:** Petri 网; 神经网络; 图像识别; 超活性; 模型优化

开放科学 (资源服务) 标志码 (OSID):



中文引用格式: 马金林, 陈德光, 马自萍, 等. 一种 Petri 网优化的验证码识别方法[J]. 计算机工程, 2020, 46(7): 277-285.

英文引用格式: MA Jinlin, CHEN Deguang, MA Ziping, et al. A CAPTCHA identification method optimized by Petri net[J]. Computer Engineering, 2020, 46(7): 277-285.

## A CAPTCHA Identification Method Optimized by Petri Net

MA Jinlin, CHEN Deguang, MA Ziping, WEI Lin

(School of Computer Science and Engineering, North Minzu University, Yinchuan 750021, China)

**[Abstract]** AlexNet does not perform well in the multi-target classification of verification codes due to the large number of parameters and heavy floating point computation. To address the problem, this paper proposes a CAPTCHA recognition method optimized by Petri net. The method uses the Petri net theory to model AlexNet and DenseNet-BC, and optimizes the network structure and parameters with the built models. At the same time, according to the relationship between the number of model parameters and the amount of floating point computation, the concept of hyperactivity is proposed. Then sensitivity analysis is carried out on Petri-ANPP-net, Petri-ANPS-net, and Petri-DNBC-net models. Experimental results show that after the Petri net-based optimization, the highest accuracy of the Petri-ANPP-net model is 60.40%, and its super activity as well as the model sensitivity is poor. The highest accuracy of the Petri-ANPS-net model is 97.50%, but its superactivity and the model sensitivity is poor. The highest accuracy of the Petri-DNBC-net model is 99.24%, and its superactivity as well as the model sensitivity is high. The results show that Petri net can optimize the network model structure and parameters to a certain extent, and the hyperactivity has certain advantages in evaluating the sensitivity of the model.

**[Key words]** Petri net; neural network; image recognition; superactivity; model optimization

DOI: 10.19678/j.issn.1000-3428.0055178

### 0 概述

随着互联网技术的飞速发展, 验证码 (CAPTCHA)

识别已成为判别人机身份的有效手段, 担负着保护网络账号安全的第一道屏障, 用于防止网站被自动程序进行大批量的恶意攻击<sup>[1]</sup>。基于此, 验证码识别技术

**基金项目:** 国家自然科学基金 (61462002, 61762003, 61862001); 北方民族大学研究项目 (2018XYZJK02); 北方民族大学教育教学重大研究项目 (2018ZHJY01); 宁夏自然科学基金项目 (2020AAC03215); “图像与智能信息处理”民委创新团队项目 (PY1805)。

**作者简介:** 马金林 (1976—), 男, 副教授、博士, 主研方向为机器学习、智能信息处理; 陈德光, 硕士研究生; 马自萍, 副教授、博士; 魏 麟, 硕士研究生。

收稿日期: 2019-06-11

修回日期: 2019-07-22

E-mail: chendeg4086@163.com

的研究有益于验证各种验证码的安全性,帮助人们设计更可靠、安全的验证码。

传统的验证码处理流程可分为验证码预处理、二值化、去除离散噪声、字符分割、归一化、特征提取、训练和字符识别等步骤<sup>[2]</sup>。这些步骤相对独立,整个流程较为复杂,其对粘连较低、噪声较小的验证码图片效果明显,而对于字符相互堆叠、扭曲度较高的验证码效果不理想。近年来,随着深度学习的兴起,有学者提出采用深度神经网络的方法实现噪音较大的验证码识别,并取得了较好的成果。文献[3]提出的基于两极 DCNN 架构的模型能将形变度较大的验证码识别准确度提升至 95% 以上,但是其模型的参数量与浮点数计算量偏高。文献[4]提出的自适应中值滤波算法经过多次迭代后平均准确度达到 98% 以上,但是其步骤较多相对繁琐。文献[5]提出的方法对极其复杂验证码的识别率达到 82% 左右,但是验证码复杂性还缺少评判标准。

神经网络的特征提取与训练过程类似于“黑箱”操作,若要得到识别率高的结果,需要针对网络结构以及模型参数进行大量的繁琐调整,而在国内外研究过程中,通常情况下直接给出模型的合理结

构与最优参数值,并未对结构和参数值的探索过程进行合理的解释,致使网络的泛化能力较差且不利于后续改进。同时,在以往的验证码识别系统中,模型优劣的评判标准更注重准确度,并未讨论该模型所需参数量与浮点数计算总量,存在一定的局限性。

针对上述问题,本文提出一种基于深度卷积神经网络的验证码识别方法。在研究神经网络模型结构的基础上,运用 Petri 网建模,并利用 Petri 网的相关理论分析该模型内部组成及相互关联的关系。在此基础上,提出模型的优化策略并通过相关实验验证优化策略的正确性。同时,针对模型参数量与浮点数计算量之间的关系,提出超活性概念,以对不同模型进行灵敏度分析。

## 1 卷积神经网络和 Petri 网

### 1.1 卷积神经网络

卷积神经网络(Convolutional Neural Network, CNN)是一种建立在传统神经网络基础上的深度神经网络<sup>[2,6]</sup>。在结构上其通常由卷积层、池化层、激励函数(通常使用线性整流单元)及全连接层组成<sup>[7]</sup>。图 1 为 CNN 结构。

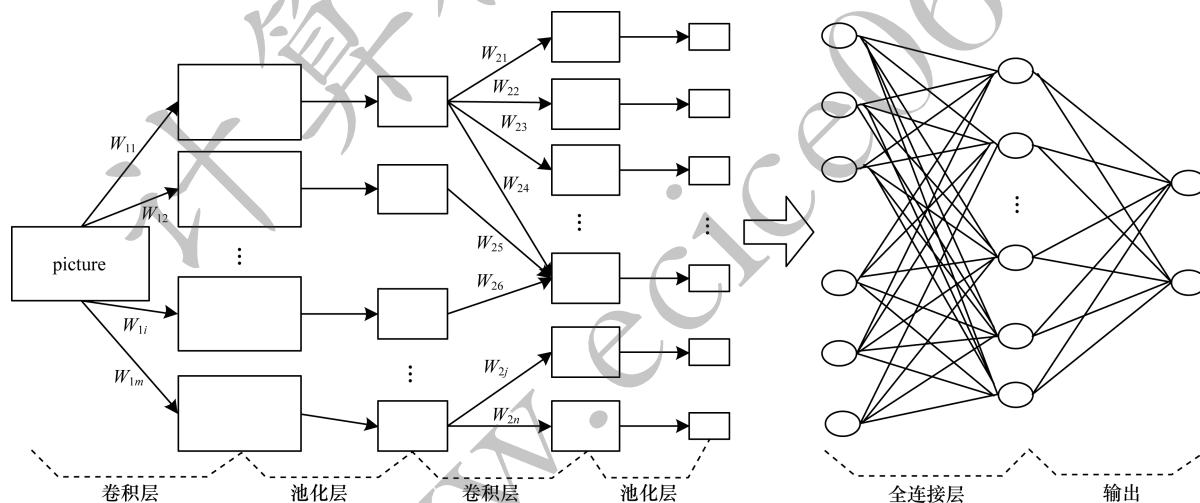


图 1 CNN 结构

Fig. 1 Structure of CNN

### 1.2 Petri 网

Petri 网是离散并行系统的数学表示,其用于描述异步并发的系统模型<sup>[8-9]</sup>,它既具有严格的数学表述方法,同时也具有直观的图形表示方法<sup>[10]</sup>。为了支撑 CNN 模型,需要利用先进的信息化技术手段对其建模。由于 CNN 是发生在训练周期内的活动,因此可参照 Petri 网的概念,即以基本网(EN-net)系统为基础,从而逐步扩展为符合卷积神经模型的网络结构,进而优化网络结构与参数以提高验证码识别准确度。借助 Petri 网重要的理论和关键技术,并根

据 EN-net 的定义,可对 CNN 模型给出以下定义:

**定义 1** 卷积神经基本网(CNE-net)系统是由四元组  $N = (P, T, F, M_0)$  组成。其中:  $P = \{P_0, P_1, \dots, P_m\}$  为有限库所的集合,每个库所包含一批验证码资源;  $T = \{T_0, T_1, \dots, T_n\}$  为有限变迁的集合,每个变迁的发生代表当前库所的资源流向下一个库所,且当前库所中不会剩余资源;  $F$  为流关系,即库所与变迁之间关系的集合,为库所与变迁的笛卡尔积,代表当前阶段训练完成后的信息输出到后续库所中,以方便后续工作开展<sup>[11]</sup>;  $M_0$  为初始情态,即在初始情况下各个库所的资源分布情

况。在此基础上,CNE-net 网络还应满足以下条件:

- 1)  $P \cup T \neq \emptyset$ , 即库所与变迁至少存在一类,它们不能同时为空。
- 2)  $P \cap T = \emptyset$ , 即库所与变迁是 2 类元素,不能交叉。
- 3)  $F \subseteq (P \times T) \cup (T \times P)$ , 即流关系  $F$  既可以由库所流向变迁也可以由变迁流向库所。
- 4)  $M_0 \subseteq 2^P$ , 即  $M_0$  为初始情态,表示初始情况下各库所的资源分布情况。
- 5)  $\text{dom}(F) \cup \text{cod}(F) = P \cup T$ , 即流关系  $F$  的定义域  $\text{dom}(F)$  与值域  $\text{cod}(F)$  的并集为库所  $P$  与变

迁  $T$  的并集,用符号表示为  $\text{dom}(F) = \{x \mid \exists y: (x, y) \in F\}$ ,  $\text{cod}(F) = \{y \mid \exists x: (x, y) \in F\}$ 。

## 2 Petri 网模型优化

### 2.1 AlexNet 网络结构

AlexNet 网络设计结构如图 2 所示,其由 5 层卷积、3 层池化、3 层全连接构成。AlexNet 网络中每层的参数如表 1 中的 AlexNet 所示。基于 AlexNet 网络并根据定义 1 的 CNE-net,给出其对应的 Petri 网模型,如图 3 所示,其模型元素与对应的流程如表 2 所示。

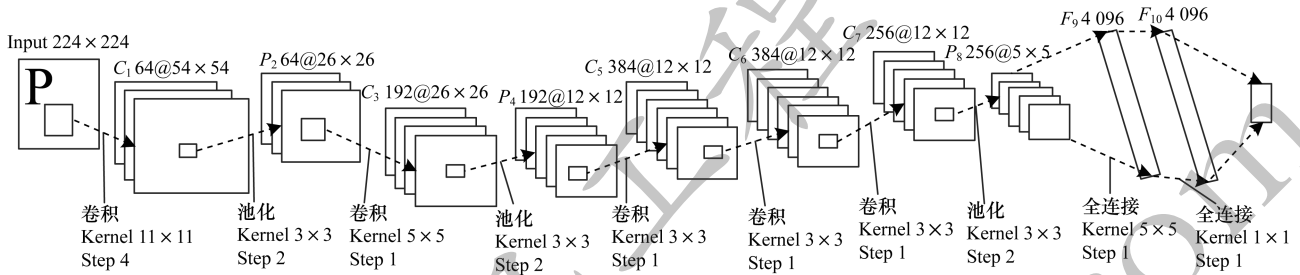


图 2 AlexNet 网络结构

Fig.2 AlexNet network structure

表 1 AlexNet 各层参数设置

Table 1 Parameter settings of each layer of AlexNet

命名	类别	AlexNet	Petri-ANPS-net
$C_1$	卷积层 1	$11 \times 11/4$	$11 \times 11/4$
$P_2$	池化层 1	$3 \times 3/2$	$3 \times 3/2$
$C_3$	卷积层 2	$5 \times 5/1$	$5 \times 5/1$
$P_4$	池化层 2	$3 \times 3/2$	$3 \times 3/2$
$C_5$	卷积层 3	$3 \times 3/1$	$3 \times 3/1$
$C_6$	卷积层 4	$3 \times 3/1$	$3 \times 3/1$
$C_7$	卷积层 5	$3 \times 3/1$	$3 \times 3/1$
$P_8$	池化层 3	$3 \times 3/2$	$3 \times 3/2$
$F_9$	全连接 1	$5 \times 5/1$	$5 \times 5/1$
$F_{10}$	全连接 2	$1 \times 1/1$	$1 \times 1/1$
$F_{11}$	全连接 3	$1 \times 1/1$	$1 \times 1/1$
Output	输出	144	$36 \times 4$

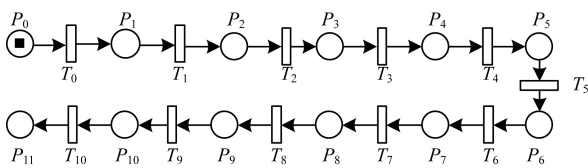


图 3 基于 AlexNet 的 Petri 网模型

Fig.3 Petri net model based on AlexNet

表 2 AlexNet 的 Petri 网模型流程

Table 2 AlexNet's Petri net model procedure

模型元素	对应的流程	模型元素	对应的流程
$P_0$	输入一批验证码	$T_0$	$11 \times 11$ 的 Kernel、步长为 4 做卷积
$P_1$	64 个 $54 \times 54$ 的特征	$T_1$	$3 \times 3$ 的 Kernel、步长为 2 做池化
$P_2$	64 个 $26 \times 26$ 的特征	$T_2$	$5 \times 5$ 的 Kernel、步长为 1 做卷积
$P_3$	192 个 $26 \times 26$ 的特征	$T_3$	$3 \times 3$ 的 Kernel、步长为 2 做池化
$P_4$	192 个 $12 \times 12$ 的特征	$T_4$	$3 \times 3$ 的 Kernel、步长为 1 做卷积
$P_5$	384 个 $12 \times 12$ 的特征	$T_5$	$3 \times 3$ 的 Kernel、步长为 1 做卷积
$P_6$	384 个 $12 \times 12$ 的特征	$T_6$	$3 \times 3$ 的 Kernel、步长为 1 做卷积
$P_7$	256 个 $12 \times 12$ 的特征	$T_7$	$3 \times 3$ 的 Kernel、步长为 2 做池化
$P_8$	256 个 $5 \times 5$ 的特征	$T_8$	$5 \times 5$ 的 Kernel、步长为 1 全连接
$P_9$	4 096 个独立特征	$T_9$	$1 \times 1$ 的 Kernel、步长为 1 全连接
$P_{10}$	4 096 个独立特征	$T_{10}$	分类函数 sigmoid 分 144 类
$P_{11}$	输出分类值		

由上述 Petri 网结构可知,其类似于一个出现网,而出现网仅是将发生过程按时间顺序进行排列,因此其不足以优化 AlexNet 网络模型。基于 AlexNet 网络的特殊性,并结合 CNE-net,下文给出 AlexNet 网络的 Petri 网优化方法。

## 2.2 基于 Petri 网参数优化的 AlexNet

### 2.2.1 预备知识

预备知识主要有以下 4 个方面:

1) 发生权。 $t \in T, M$  为  $N$  的标识,  $t$  在  $M$  下有发生权,记作  $M[t >]$ <sup>[12]</sup>,且满足以下条件:

(1)  $\forall s \in \cdot t: M(s) \geq W(s, t)$ , 即  $s$  在  $t$  变迁前驱发生的条件为  $s$  库所中的资源要大于等于所需要消耗的资源。

(2)  $\forall s \in t: M(s) + W(s, t) \leq K(s)$ , 即库所  $s$  的容量  $K$  要大于等于原库所中的资源容量加上该变迁所产生的资源容量。

2) 后继。若  $M[t >]$ , 即  $t$  在  $M$  条件下具有发生权,  $M'$  记为  $M$  的后继, 后继关系记为  $M[t > M']$ <sup>[13]</sup>, 且满足以下条件:

(1) 如果  $s \in \cdot t - t \cdot$ , 则  $M'(s) = M(s) - W(s, t)$ , 即  $s$  库所要消耗资源  $W(s, t)$ 。

(2) 如果  $s \in t \cdot - \cdot t$ , 则  $M'(s) = M(s) + W(s, t)$ , 即  $s$  库所要生成资源  $W(s, t)$ 。

(3) 如果  $s \in \cdot t \cap t \cdot$ , 则  $M'(s) = M(s) + W(t, s) - W(s, t)$ , 即  $s$  库所既要生成资源  $W(t, s)$ , 又要消耗资源  $W(s, t)$ 。

(4) 如果  $s \notin \cdot t \cup t \cdot$ , 则  $M'(s) = M(s)$ , 即  $s$  库所既不生成资源, 也不消耗资源。

3) 前集与后集。对于  $x \in P \cup T$ , 若有:

(1)  $x^- = \{y | y \in P \cup T \wedge (y, x) \in F\}$ , 则称  $x^-$  为  $x$  的前集<sup>[14]</sup>。

(2)  $x^+ = \{y | y \in P \cup T \wedge (x, y) \in F\}$ , 则称  $x^+$  为  $x$  的后集<sup>[14]</sup>。

4) 公平性。对于  $t_1, t_2 \in T$ , 若存在正整数  $k$ , 使得  $\forall M \in R(M_0)$  和  $\forall \sigma \in T^*: M[\sigma >] \neq \emptyset \leftrightarrow \#(t_j/\sigma) \leq k, i, j \in \{1, 2\}$  且  $i \neq j$ , 则称  $t_1$  和  $t_2$  处于公平关系<sup>[15]</sup>。

### 2.2.2 基于 Petri 网参数优化的 AlexNet 网络

**定义 2** Petri 网参数优化的 AlexNet (Petri-ANPP-net) 由六元组  $\Pi = (P, T; F, K, W, M_0)$  组成, 其中  $N = (P, T; F, M_0)$  为定义 1 中卷积神经的基本网系统, 其在满足基本网系统的前提下还满足以下条件:

1)  $\Pi$  中有一个开始位置  $i \in P$  和一个结束位置  $o \in P$ 。

2)  $K$  为库所  $P$  元素的容量函数, 其代表验证码图片批次的大小, 即有  $K: P \rightarrow \{1, 2, 3, \dots\} \cup \{\infty\}$ 。

3)  $W$  为流关系  $F$  上的权函数, 其代表验证码图片的产生或消耗的量多少, 本文采用学习率来表示流关系, 即有  $W: F \rightarrow \{1, 2, 3, \dots\}$ 。

4)  $M_0$  为初始资源分布情况, 且还应该满足  $\forall s \in$

$S: M_0(s) \leq K(s)$ 。

5) 具有合理性, 即:  $\Pi$  具有安全性, 在任何时刻,  $\Pi$  中的所有库所中至少包含一类验证码资源; 具有规范性, 任何库所  $P$  均有容纳资源 (token) 多少的表示, 而非用一个 token 表示一类资源, 这样的 Petri 网描述能力更强。同理, 每个变迁  $t$  均具有发生权  $M[t >]$  和后继  $M[t > M']$ ;  $\Pi$  不包含无法执行的死任务, 即从开始位置  $\{i\}$  起, 任何一个变迁均可以被激活用来执行任务。

### 2.3 基于 Petri 网结构优化的 AlexNet

Petri 网是一个异步并发系统, 而定义 2 中 Petri-ANPP-net 给出的 Petri 网模型是一个顺序执行系统且系统参数较多, 将顺序执行系统扩展为异步并发系统且减少参数的量。由并发公理可知, 系统性能将会得到进一步改善且网络更容易解读。

**定义 3** Petri 网结构优化的 AlexNet (Petri-ANPS-net) 是四元组  $\Sigma = (S, T, W', M_0)$ , 其中  $S, T, M_0$  与 Petri-ANPP-net 中相同, 其应满足以下条件:

1) 如果对于  $\forall (x, y) \in S \times T \cup T \times S$ , 即当  $(x, y) \in F$  时, 有  $W'(x, y) = W(x, y)$ , 则称  $W': S \times T \cup T \times S \rightarrow \{0, 1, 2, \dots\}$  为  $\Sigma$  的广义权函数<sup>[16]</sup>。

2) 如果  $M[t >]$  发生, 则  $\forall s \in S: M(s) \geq W'(s, t)$  且  $M(s) + W'(s, t) \leq K(s)$ <sup>[16-17]</sup>, 即变迁  $t$  在  $M$  标识下具有发生权的条件为库所  $s$  中的 token 要大于等于所消耗的 token 数量, 且库所  $s$  中所容纳的 token 数量要大于等于其原本库所中的数量加上新产生的 token 数量。

3) 如果  $M[t > M']$  发生, 则  $M[t >]$  且  $\forall s \in S$ , 有  $M'(s) = M(s) + W'(t, s) - W'(s, t)$ <sup>[18-19]</sup>, 即  $M$  若有后继  $M'$ , 则应满足  $t$  在  $M$  下有发生权, 且  $M'$  的 token 数量为原库所中的 token 数量加上产生的 token 数量再减去消耗的 token 数量。

4) 由于 Petri-ANPS-net 是异步并发系统, 且每个验证码字母地位相当, 则当其并发时, 各个子系统间的权重相同。

为使更改后的模型与原模型变动最小, 在图 3 模型中  $P_9$  库所后设置 4 组并发操作 (这是因为验证码的字符长度为 4)。并发操作完成后, 汇聚于漏库所  $O$ 。这样改进后, 将原来的顺序执行系统变为异步并发系统, 具体优势有以下 3 点:

(1) 模型简洁, 由定义 2 的 Petri-ANPP-net 六元组改造为定义 3 的 Petri-ANPS-net 的四元组, 降低了模型的复杂度。

(2) 模型参数更新合理化,  $P_{10} \sim P_{13}$  共享  $P_9$  中的 token, 但是各个分支的关系相互独立, 则各个分支的权值与偏置值的更新与其余分支无关。

(3) 加速模型收敛, 由于异步并发系统中 4 条分支同时训练, 相对于顺序执行收敛速度更快。

图 4 所示为基于 ANPS-net 的模型 (黑色变迁为隐变迁, 表示前面的若干库所与变迁, 由于图 4 中前面的库所变迁与图 3 中的  $P_0 \sim P_9$  相同, 因此采用隐变迁代替)。相关参数见表 1 (右) 表示的 Petri-ANPS-net 网络结构。

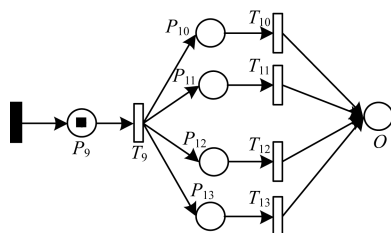


图 4 Petri-ANPS-net 模型  
Fig. 4 Petri-ANPS-net model

## 2.4 基于 Petri 网优化的 DenseNet-BC

### 2.4.1 DenseNet 网络

密集型卷积神经网络 DenseNet 由文献[20]提出,其从神经网络的特征入手,简化模型参数,同时达到了更好的效果。它的主要思想是跨层连接,网络的每一层输入都是前面所有层输出的并集,而该层学习到的特征也会被直接传给后面所有层作为输入。在上述过程中,信息流进行整合,避免了信息在层与层之间传递丢失的问题,极大地减轻了梯度弥散。

DenseNet 主要由密集块和过渡层组成。在每个密集块内,层与层之间的非线性转换函数  $H_i(\cdot)$  包含 3 个连续操作,即批量标准化、线性整流函数

ReLU 以及  $3 \times 3$  卷积 ( $3 \times 3 \text{Conv}$ )<sup>[20]</sup>。定义超参数  $k$  为增长率,如果  $H_i(\cdot)$  每层产生  $k$  个特征映射,则第  $l$  层将有  $k_0 + k(l-1)$  个特征映射输入 ( $k_0$  为首个输入层的特征数)。

### 2.4.2 DenseNet-BC 网络

DenseNet 网络的密集连接方式会导致后续密集层有大量输入特征,从而导致参数量急剧增长,因此,在每个密集层的  $3 \times 3 \text{Conv}$  前引入  $1 \times 1 \text{Conv}$  作为瓶颈层,来减少特征映射输入的数量<sup>[21]</sup>,同时,在每个过渡层中引入一个  $1 \times 1 \text{Conv}$  使其输出特征减少,以增强 GPU 内存的利用率。

由于实现朴素的 DenseNet-BC 可能需要大量的 GPU 内存,基于此,引入共享内存分配机制,所有层都使用共享内存分配来存储中间结果。后续图层会覆盖先前图层的中间结果,但可以在后向传递期间以最低成本重新填充它们的值,这样做可以将特征映射内存消耗从二次减少到线性。图 5 所示为 DenseNet-BC 网络结构模型,由 3 个密集块与 2 个过渡层组成,图 6 第 3 列为 DenseNet-BCm 网络参数,图 6 第 4 列、第 5 列给出 Petri-DNBC-net 的网络参数。

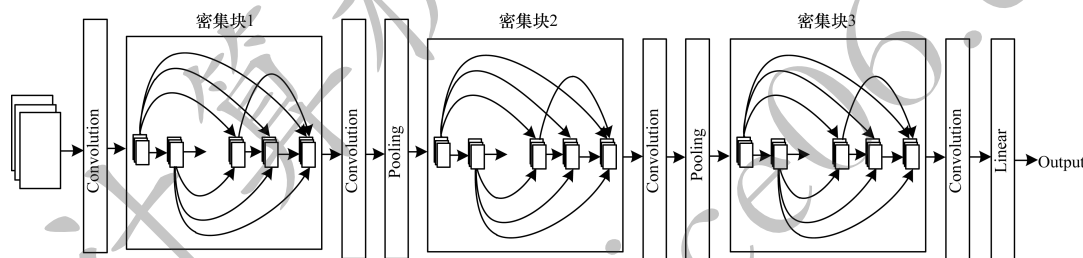


图 5 DenseNet-BC 网络结构  
Fig. 5 DenseNet-BC network structure

Layers	Output Size	DenseNet-BCm	Petri-DNBC-net_40	Petri-DNBC-net_100
卷积层	$56 \times 56$	$7 \times 7$ Conv, stride 2	$4 \times 4$ Conv, stride 4	
池化层	$28 \times 28$	$3 \times 3$ max pooling, stride 2		
密集块1	$28 \times 28$	$\begin{bmatrix} 11 \times \text{Conv} \\ 33 \times \text{Conv} \end{bmatrix} \times m$	$\begin{bmatrix} 11 \times \text{Conv} \\ 33 \times \text{Conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 11 \times \text{Conv} \\ 33 \times \text{Conv} \end{bmatrix} \times 16$
过渡层1	$28 \times 28$	$1 \times 1$ Conv		
	$14 \times 14$	$2 \times 2$ average pooling, stride 2		
密集块2	$14 \times 14$	$\begin{bmatrix} 11 \times \text{Conv} \\ 33 \times \text{Conv} \end{bmatrix} \times m$	$\begin{bmatrix} 11 \times \text{Conv} \\ 33 \times \text{Conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 11 \times \text{Conv} \\ 33 \times \text{Conv} \end{bmatrix} \times 16$
过渡层2	$14 \times 14$	$1 \times 1$ Conv		
	$7 \times 7$	$2 \times 2$ average pooling, stride 2		
密集块3	$7 \times 7$	$\begin{bmatrix} 11 \times \text{Conv} \\ 33 \times \text{Conv} \end{bmatrix} \times m$	$\begin{bmatrix} 11 \times \text{Conv} \\ 33 \times \text{Conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 11 \times \text{Conv} \\ 33 \times \text{Conv} \end{bmatrix} \times 16$
分类层	$1 \times 1$	$7 \times 7$ global average Pool	Out0: $7 \times 7$ average pooling	
			Out1: $7 \times 7$ average pooling	
			Out2: $7 \times 7$ average pooling	
			Out3: $7 \times 7$ average pooling	
		144D fully-connected, softmax	Out0: 36D fully-connected, softmax	
			Out1: 36D fully-connected, softmax	
			Out2: 36D fully-connected, softmax	
			Out3: 36D fully-connected, softmax	

图 6 DenseNet-BC 网络参数  
Fig. 6 DenseNet-BC network parameters

从图 6 可以看出,该模型由 3 个密集块与 2 个过渡层组成,并将原来的单分类层改为并行的四分分类层。

#### 2.4.3 基于 Petri 网优化的 DenseNet-BC

**定义 4** Petri 网优化的 DenseNet-BC (Petri-DNBC-net) 在定义 3 的条件下还应满足以下条件:

- 1)  $\forall x \in S \cup T, x \cap x \cdot \neq \emptyset$  即对元素  $x$  的前集与后集相交不为空(网络不为纯网)。
- 2)  $t_1, t_2 \in T$  时,  $t_1$  和  $t_2$  不一定处于公平状态。

### 2.5 活性及超活性对模型的性能评价

#### 2.5.1 活性

根据定义 1 中的 CNE-net, 如果对任意  $C \in R(M_0)$ , 都存在  $M' \in R(M)$ , 使得  $M' \models t$ , 则称变迁  $t$  是活的。如果  $t \in T$  都是活的, 则称  $N$  为活的 Petri 网<sup>[14]</sup>。

#### 2.5.2 超活性

活性用来评判网络是否存在死锁或者陷阱等不足, 但是其不能用来评判网络的反应灵敏度。基于此, 提出了超活性的概念。

**定义 5** 超活性( $\mathfrak{R}$ ) 是反映一个网络系统灵敏度的指标, 计算公式为:  $\mathfrak{R} = \frac{FPN}{PQ}$ , 其中, FPN 为模型浮点数计算量, PQ 为模型参数量, 且满足以下条件:

- 1) 在同构系统中, 超活性( $\mathfrak{R}$ ) 相同时, 则灵敏度相同。
- 2) 在异构系统中, 比较超活性  $\mathfrak{R}_1$  和  $\mathfrak{R}_2$  的值,  $\mathfrak{R}$  较大者则灵敏度相对较高。

#### 2.5.3 3 种网络结构的超活性比较

Petri-ANPP-net 和 Petri-ANPS-net 网络结构大体相同(其区分在于最后一个全连接层是否进行并发处理), 因此, 这 2 个模型的超活性  $\mathfrak{R}$  近似相同, 即在同等条件下比较其准确度即可。Petri-DNBC-net 与 Petri-ANPS-net (或 Petri-ANPP-net) 结构不同, 则采用公式  $\mathfrak{R} = \frac{FPN}{PQ}$  进行超活性分析以确定超活性性能。

## 3 实验结果与分析

本文实验基于 Win10 平台, CPU 为 Inter(R) Xeon(R) E5-1620 v3 3.5 GHz, 内存 8 GB, 显卡为 NVIDIA RTX 2080。

### 3.1 数据集

在深度学习的训练过程中需要大量的数据来训练构建的网络模型。由于目前没有公开的验证码数据集, 本文采用验证码生成器生成 4 字验证码数据集, 在生成验证码的过程中对字符进行不同程度的旋转、扭转、平移、加入噪点及字符重叠等处理<sup>[22]</sup>。验证码图片大小为  $60 \times 160$ , 由阿拉伯数字 0~9 以及小写字母 a~z 共 36 类随机组成。通过

对生成的验证码图片进行灰度化处理, 以提高训练和测试的速度, 然后将得到的灰度图片制作成 Tensorflow 易读取的 TFrecords 格式的数据集。图 7 为部分验证码图片。

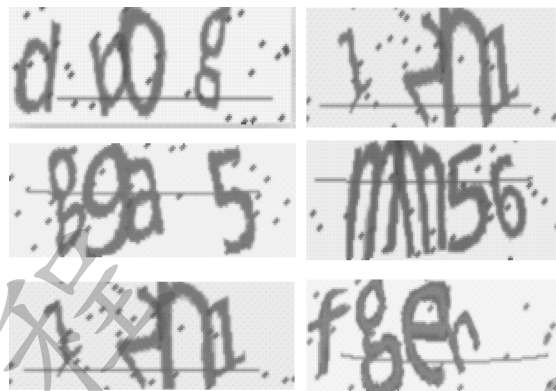


图 7 部分验证码图片

Fig.7 Part of the CAPTCHA picture

### 3.2 数据集选择

选择基于框架 Tensorflow 的实现, 在综合比较后, 损失函数采用 sigmoid 交叉熵损失函数, 优化器采用 AdamOptimizer 来使损失函数 sigmoid 交叉熵最小化, 其学习率初始固定为 0.000 3, 初始训练批次大小为 20。在训练过程中还添加了 dropout 层, 能够在一定程度上防止过拟合, 其参数值为 0.5。图 8 显示了不同训练集下的准确度。

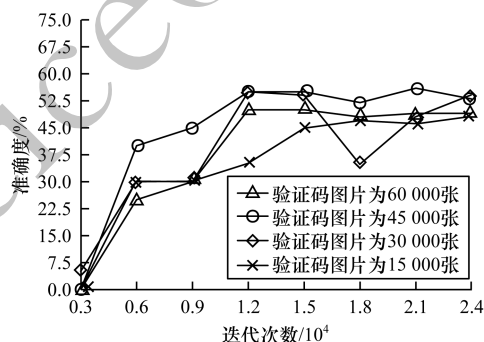


图 8 不同训练集的准确度

Fig.8 Accuracy of different training sets

从图 8 可以看出, 当选用 45 000 张验证码图片时(其中训练集为 40 000 张, 测试集为 5 000 张), 其准确度最高为 55% 左右(测试方式为测试 10 个批次, 每个批次为 40 张, 将每个批次的准确度相加再除以 10 即为最终的准确度), 较其余 3 组数据集性能更为优越。因此, 在下文的实验中, TFrecords 数据集均采用 45 000 张图片制作。

### 3.3 Petri-ANPP-net 实验

由 Petri-ANPP-net 可知, Petri 网在结构确定的情况下, 其性能由库所  $P$  中资源和流关系  $F$  上的权函数共同决定。预备实验中的学习率(权函数)与批次大小(资源)是随机设置的, 存在较大的局限性, 而同时调整 2 个参数会导致模型变得异常复杂。因

此,假设当 token 发生变化时,权函数  $W$  为理想条件,即  $M[t > M']$  一定发生。同理,当调整权函数  $W$  时,其资源一定满足所有条件即不发生冲突、冲撞等异常情况。

**实验 1** 结合预备实验,将初始学习率设为 0.000 3,即流关系  $F$  的权值设置为 0.000 3(这里将 0.000 3 看成无穷大,资源均能快速通过),设置 4 组不同批次的对照实验。第 1 组为 10 张/批次,第 2 组为 20 张/批次,第 3 组为 25 张/批次,第 4 组为 30 张/批次。在初始库所  $P_0$  中放置不同的 token 数量,以测试该系统的最佳 token 性能。图 9 显示了不同大小的批次量在不同迭代次数下的准确度。

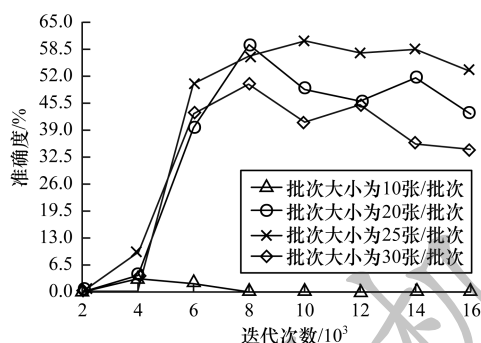


图 9 不同大小批次量的准确度  
Fig. 9 Accuracy of different batch sizes

从图 9 可以看出,批次量太大或太小均会影响系统的整体性能,当批次量为 25 张/批次时,在该网络结构条件下准确度达到最优,且最优值为 60%。

**实验 2** 由实验 1 可知,当批次量设置为 25 张/批次时,系统所能达到的性能具有最优标准。但是实验 1 随机固定了初始学习率,不能说明实验 1 的最高准确度就是该网络结构下的最高值。基于此,在实验 1 的基础上(即固定批次量为 25 张/批次),设置 4 组不同的学习率参数进行对照实验。第 1 组学习率为 0.000 2,第 2 组学习率为 0.000 3,第 3 组学习率为 0.000 4,第 4 组学习率为 0.000 5。图 10 显示了不同学习率在不同迭代次数下的准确度。

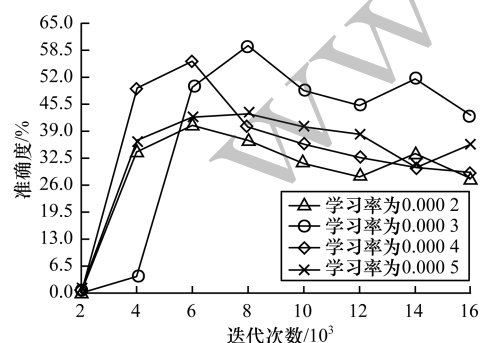


图 10 不同学习率的准确度  
Fig. 10 Accuracy of different learning rates

从图 10 可以看出,学习率过大或过小均会影响系统整体性能,当学习率为 0.000 3 时,在该网络结

构条件下,准确度达到最优,且最优值为 60% (说明实验 1 随机设置的学习率即为最佳学习率)。

基于 Petri-ANPP-net 所能达到的准确度最高为 60%,与预处理实验中 55% 相比略有提升。而此时库所中 token 资源与流关系  $F$  均已调整为最佳。由以上实验可知,参数调节能在一定范围内优化系统性能。

### 3.4 Petri-ANPS-net 实验

由 Petri-ANPS-net 可知,网结构优化后,其性能也会发生改变。为此,设置实验 3、实验 4 两组实验。

**实验 3** 为验证不同学习率下 Petri 网优化结构后模型的准确度,将批次量设置为 25 张/批次(结合实验 1、实验 2),进行了 4 组对比实验。第 1 组学习率为 0.000 1,第 2 组学习率为 0.000 2,第 3 组学习率为 0.000 3,第 4 组学习率为 0.000 4,实验结果如图 11 所示。

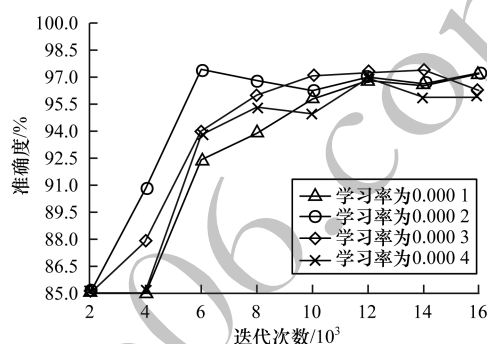


图 11 优化结构后不同学习率的准确度  
Fig. 11 Accuracy of different learning rates after optimizing structure

从图 11 可以看出,在网络结构合理的条件下,准确度将保持较高水平,当学习率为 0.000 3 时,其最高值相对平稳且略优于其他各组。

**实验 4** 为验证不同批次下 Petri 网优化结构后模型的准确度,将学习率设置为 0.000 3(由实验 3 得),调整批次大小后再进行 4 组实验。第 1 组批次大小为 20 张/批次,第 2 组为 25 张/批次,第 3 组为 30 张/批次,第 4 组为 40 张/批次,优化结构后不同批次下准确度的实验结果如图 12 所示。

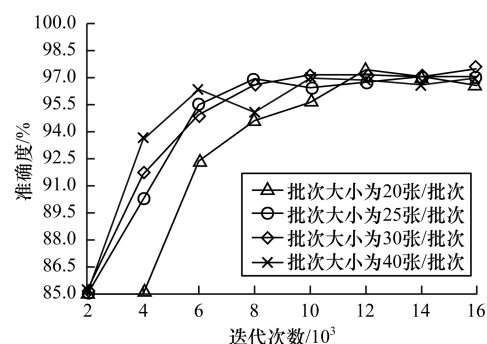


图 12 优化结构后不同批次的准确度  
Fig. 12 Accuracy of different batches after optimizing structure



从图 12 可以看出,在网络结构合理的情况下,单一参数在一定范围内变化对准确度影响较小;当学习率为 0.000 3、批次大小为 30 张/批次时,其准确度最高达到了 97.50%。

对以上各组实验中不同参数条件下的平均最高准确度进行总结,结果如表 3 所示。

表 3 不同参数下的平均最高准确度  
Table 3 Average highest accuracy under different parameters

模型	批次/张	学习率/ $10^{-4}$	准确度/%
Petri-ANPP-net	10	3	8.00
	20	3	59.60
	25	3	60.40
	30	3	50.20
	25	2	40.20
	25	3	59.80
	25	4	56.00
Petri-ANPS-net	25	5	43.20
	20	3	97.44
	25	3	97.06
	30	3	97.50
	40	3	97.00
	25	1	97.19
	25	2	97.12
	25	3	97.37
	25	4	96.94

从表 3 可以看出,批次和学习率均能影响模型性能,尤其是一些不合理的参数会导致网络模型失败,在最佳参数条件下,Petri-ANPS-net 网络相对于 Petri-ANPP-net 绝对准确度提升了 37.10%,说明网络模型设计是系统成功的关键。

表 5 不同模型的性能指标  
Table 5 Performance indicators of different models

模型	验证码大小	参数量		深度	增长率/%	准确度/%	PQ/ $10^3$	FPN/106FLOPS	$\mathfrak{N}$
		批次/张	学习率/ $10^{-4}$						
Petri-ANPP-net	224 × 224	25	3	—	—	60.40	46.70	688	14
Petri-ANPS-net	224 × 224	30	3	—	—	97.50	46.70	688	14
Petri-DNBC-net	224 × 224	64	1 000	40	12	98.44	0.17	52	280
	224 × 224	64	1 000	40	24	98.92	0.63	175	280
	224 × 224	64	1 000	100	12	99.13	0.77	207	280
	224 × 224	64	1 000	100	24	99.24	2.90	809	280

从表 5 可以看出:

1) Petri-ANPP-net 和 Petri-ANPS-net 网络的超活性  $\mathfrak{N}$  值相同,说明它们的灵敏度相同,这也验证了同构网络具有相同的 PQ、FPN 和超活性  $\mathfrak{N}$  值。同时,Petri-ANPP-net 的最高准确度为 60.40%,而 Petri-ANPS-net 的最高准确度为 97.50%,说明 Petri-ANPS-net 综合性能远高于 Petri-ANPP-net。

2) 无论在何种条件下,Petri-DNBC-net 的准确度都略高于 Petri-ANPS-net。而在超活性方面,Petri-

### 3.5 Petri-DNBC-net 实验

由 Petri-DNBC-net 可知,当网络模型改变后,则网络性质将会发生一定的改变,从而导致网络性能发生改变。

实验 5 参考 Petri-ANPP-net 以及 Petri-ANPS-net,将验证码大小设置为 224 × 224。经过综合比较后,学习率设置为 0.1,偏置值设置为 0.000 1,批次大小为 64 张/批次。在此基础上,设置 4 组对比实验,第 1 组网络深度为 40,增长率为 12%,第 2 组网络深度为 40,增长率为 24%,第 3 组网络深度为 100,增长率为 12%,第 4 组网络深度为 100,增长率为 24%。表 4 为在各参数条件下 Petri-DNBC-net 的准确度。

表 4 Petri-DNBC-net 的准确度  
Table 4 Accuracy of Petri-DNBC-net

深度	增长率/%	准确度/%
40	12	98.90
40	24	98.92
100	12	99.13
100	24	99.24

从表 4 可以看出,当深度和增长率增加后,其准确度会相应增加,当深度为 100 时,其准确度大于 99%。

### 3.6 超活性对模型的性能评价

由以上实验可知,Petri-DNBC-net 的准确度略优于 Petri-ANPS-net,且 Petri-ANPP-net 准确度最低。但是,准确度仅是验证模型综合性能的指标之一。为此,还应考虑超活性对模型性能的影响。

为测试各模型的超活性,选择在不同模型的最高准确度下进行超活性分析,实验结果如表 5 所示。

DNBC-net 的超活性  $\mathfrak{N}_D \approx 280$ ,Petri-ANPS-net 的超活性  $\mathfrak{N}_A \approx 14$ 。因此,Petri-DNBC-net 的灵敏度高于 Petri-ANPS-net,且综合性能也优于 Petri-ANPS-net 网络。

## 4 结束语

本文利用 Petri 网理论,基于 AlexNet、DenseNet-BC 网络得到 Petri-ANPP-net、Petri-ANPS-net 以及 Petri-DNBC-net 3 种优化的网络模型,并提出基于



Petri 网优化的验证码识别方法。基于模型异构性对评判标准产生的影响,提出了超活性概念,其对模型的灵敏度评判具有较高准确性,能够消除异构性从而达到公平评判模型灵敏度的目的。实验结果表明,本文方法能够在一定程度上优化网络模型的结构和参数。下一步将优化卷积核、池化核以及研究学习率与批次大小之间的关系,以提升验证码的安全性。

### 参考文献

- [1] VON A L, BLUM M, HOPPER N J, et al. CAPTCHA: using hard AI problems for security [C]//Proceedings of International Conference on Theory and Applications of Cryptographic Techniques. Berlin, Germany: Springer, 2003: 294-311.
- [2] ZHANG Zheng, WANG Shunfan, DONG Lei. CAPTCHA recognition based on deep learning [J]. Journal of Hubei University of Technology, 2018, 33(2): 5-8, 25. (in Chinese)  
张铮,王顺帆,董雷. 基于深度学习的验证码识别[J]. 湖北工业大学学报, 2018, 33(2): 5-8, 25.
- [3] CHEN Jun, LUO Xiangyang, LIU Yingying, et al. Selective learning confusion class for text-based CAPTCHA recognition [J]. IEEE Access, 2019, 7: 22246-22259.
- [4] MA Wentao, QIN Jiaohu, XIANG Xuyu, et al. Adaptive median filtering algorithm based on divide and conquer and its application in CAPTCHA recognition [J]. Computers, Materials and Continua, 2019, 58(3): 665-677.
- [5] ARAIN R H, SHAIKH R A, MAITLO A, et al. A deep learning model for recognition of complex text-based CAPTCHA [J]. International Journal of Computer Science and Network Security, 2018, 18(2): 103-107.
- [6] QIN Bo, GU Naijie, ZHANG Xiaoci, et al. Image CAPTCHA recognition based on convolutional neural network [J]. Computer Systems & Applications, 2018, 27(11): 144-150. (in Chinese)  
秦波,顾乃杰,张孝慈,等. 基于卷积神经网络的图像验证码识别[J]. 计算机系统应用, 2018, 27(11): 144-150.
- [7] ROBERTSON S, PENN G, WANG Y. Exploring spectro-temporal features in end-to-end convolutional neural networks [EB/OL]. [2019-05-10]. <https://arxiv.org/abs/1901.00072?context=cs.LG>.
- [8] LI Wenxiang. Modeling and verification of a temperature control system based on temporal Petri nets [J]. Journal of Shandong University of Technology (Natural Science Edition), 2018, 32(6): 24-28. (in Chinese)  
李文翔. 温控系统的时序 Petri 网建模与验证[J]. 山东理工大学学报(自然科学版), 2018, 32(6): 24-28.
- [9] VAN D A W, WEIJTERS T, MARUSTER L. Workflow mining: discovering process models from event logs [J]. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(9): 1128-1142.
- [10] SMIRNOV S, WEIDLICH M, MENDLING J. Business process model abstraction based on behavioral profiles [C]//Proceedings of International Conference on Service-Oriented Computing. Berlin, Germany: Springer, 2010: 1-16.
- [11] SOBOCINSKI P, STEPHENS O. Penrose: putting compositionality to work for petri net reachability [C]//Proceedings of International Conference on Algebra and Coalgebra in Computer Science. Berlin, Germany: Springer, 2013: 346-352.
- [12] ZHAO Yang, LI Tong, LIU Qing. A performance analysis method for software process modeling based on the extended Petri net [J]. Computer Engineering and Applications, 2004, 40(26): 70-72. (in Chinese)  
赵杨,李彤,柳青. 一种基于扩展 Petri 网的软件过程模型性能分析方法[J]. 计算机工程与应用, 2004, 40(26): 70-72.
- [13] TAX N, SIDOROVA N. Discovering more precise process models from event logs by filtering out chaotic activities [J]. Journal of Intelligent Information Systems, 2019, 52(1): 107-139.
- [14] XU Yuyin. Discussion on Petri net activity determination method [D]. Jinan: Shandong University of Science and Technology, 2006. (in Chinese)  
徐誉尹. Petri 网活性判定方法的探讨 [D]. 济南: 山东科技大学, 2006.
- [15] ZHANG Jianping. Application of Petri net dynamic workflow in OA system of mining company [J]. China's Manganese Industry, 2019, 37(2): 107-110. (in Chinese)  
张建平. Petri 网动态工作流在矿业公司 OA 系统中的应用 [J]. 中国锰业, 2019, 37(2): 107-110.
- [16] REISIG W. Petri nets: an introduction [M]. Berlin, Germany: Springer, 2012.
- [17] MARSAN M A, BALBO G, CONTE G, et al. Modelling with generalized stochastic Petri nets [M]. [S. l.]: John Wiley & Sons, Inc., 1994.
- [18] GUERAR M, MERLO A, MIGLIARDI M. Completely automated public physical test to tell computers and humans apart: a usability study on mobile devices [J]. Future Generation Computer Systems, 2018, 82: 617-630.
- [19] GOTO M, SHIRATO T, UDA R. Text-based CAPTCHA using phonemic restoration effect and similar sounds [C]//Proceedings of the 38th IEEE International Computer Software and Applications Conference Workshops. Washington D. C., USA: IEEE Press, 2014: 270-275.
- [20] HUANG G, LIU Z, VAN D M L, et al. Densely connected convolutional networks [C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2017: 4700-4708.
- [21] PLEISS G, CHEN D, HUANG G, et al. Memory-efficient implementation of densenets [EB/OL]. [2019-05-10]. <https://www.researchgate.net/publication>.
- [22] HE Kaiming, SUN Jian. Convolutional neural networks at constrained time cost [C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Washington D. C., USA: IEEE Press, 2015: 5353-5360.