Trabalho II do Componente Currícular de Inteligência Artificial de 2019

Redes Neurais

Éverton de A. Vieira e Gabriel H. Moro

17 de maio de 2019

Introdução

Objetivo

O objetivo do trabalho é o desenvolvimento e uso de rede neural que resolve *Captchas*.

Problema Escolhido

O problema escolhido é o problema de quebra de *Captchas*. Um *Captcha* é uma que contém um texto que deve ser digitado para se ter permissão de acesso a ou um site. Para este trabalho serão utilizados *Captchas* simples, contendo quatro caracteres alfanuméricos em um fundo branco, excluindo os caracteres *O*, *I*, *0* e *1*.

Desenvolvimento

Para o desenvolvimento do trabalho foi utilizada a linguagem *Python* 3.5+ e as bibliotecas *OpenCV*, para o processamento das imagens, *Keras* e *TensorFlow*, para a rede neural.

Base de dados

O banco de imagens contém 10000 *Captchas* simples descritos anteriormente[1]. Todas as imagens são nomeadas com o resultado esperado para

[1] Fonte: https://drive.google.com/file/d/1jDa5UE1jSGr5apotQUzjdC9Wrkj0SKSj/view

uso no treinamento da rede.

Primeiramente as imagens passam por um processo de limiarização para que só possuam tons de preto e branco para a detecção dos contornos dos caracteres utilizando a função *findContours()* da biblioteca *OpenCV*. A partir dos contornos gerados são salvas imagens dos caracteres separados, e caso haja sobreposição de caracteres isso é detectado pelas dimensões do contorno e corrigido dividindo a região contornada em duas, cada uma sendo um caractere. Cada imagem de caractere é salva em uma pasta com a *label* correspondente ao caractere e nomeado com a *label* e um número. Os caracteres separados são usados como entrada para a etapa de treinamento da rede

Treinamento da rede

A rede treinada é uma rede neural convolucional simples, com duas camadas convolucionais (*Max pooling*), uma com 20 neurônios e outra com 50, que usam uma função de ativação *Rectified Linear Unit* ou *ReLu*, uma camada oculta completamente conectada com 500 neurônios, que também usa a função *ReLu*, e uma camada de saída com 32 neurônios, um para cada caractere possível, que usa a função *SoftMax*.

Inicialmente são carregadas as letras extraídas na primeira etapa que são redimensionadas para 20x20px e são armazenadas em uma lista de dados de treino da rede junto com a label correspondente, que é o nome da pasta na qual a imagem se encontra. A lista de dados de treinamento é separada em dados de treino (75%) e dados de teste (25%), a rede é construída e treinada com dez épocas e o modelo é salvo.

Resultados

Os resultados obtidos são imagens nas quais os caracteres estão contornados e é exibido o caractere correspondente em cima do contorno. A taxa de acerto da rede foi de aproximadamente 100%, como mostrado na Figura 1. A Figura 2 mostra alguns exemplos de resultados obtidos com o modelo treinado.

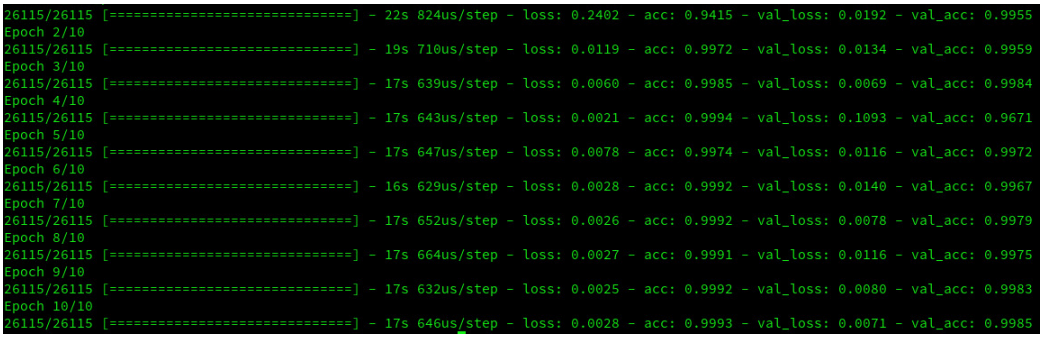

Figure 1: Erro e acerto da rede durante as épocas do treinamento

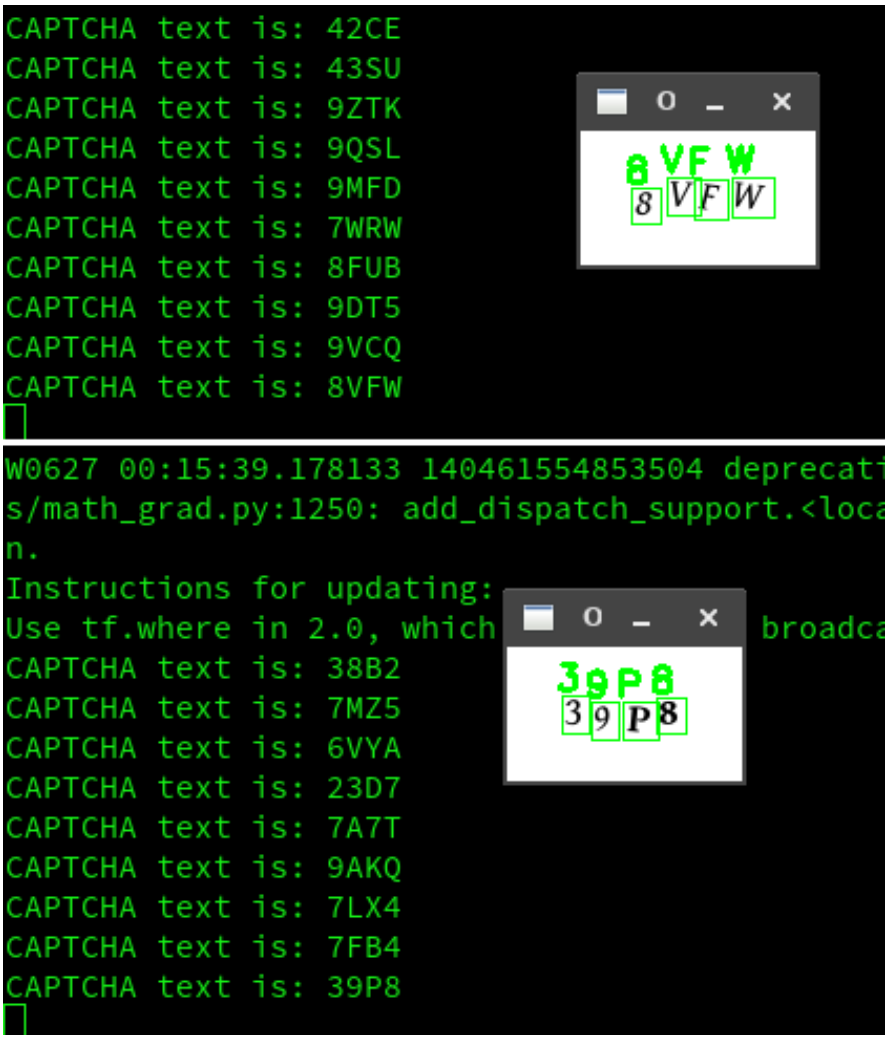

Figure 2: Exemplos de resultados obtidos