UNIVERSIDADE
FEDERAL RURAL
DE PERNAMBUCO

Diogo Felipe Félix de Melo

Aprendizado profundo com capacidade computacional reduzida: uma aplicação à quebra de captchas.

Recife

2015

Diogo Felipe Félix de Melo

Aprendizado profundo com capacidade computacional reduzida: uma aplicação à quebra de captchas.

Monografia apresentada ao Curso de Bacharelado em Ciências da Camputação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Ciências da Camputação.

Universidade Federal Rural de Pernambuco – UFRPE

Departamento de Computação

Bacharelado em Ciências da Camputação

Orientador: Pablo de Azevedo Sampaio

Recife

2015

Agradecimentos

Meus pais, familiares e amigos.

Ao meu orientador, por toda a paciência e dedicação.

Resumo

Palavras-chave: Aprendizado de Maquina, Aprendizado Profundo, CAPTCHA.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

CAPTCHA	Completely Automated Public Turing tests to tell Computers and Humans ApartUser Datagram Protocol
OCR	Optical Character Recognition

Sumário

1 Introdução

Modelos de aprendizado baseados em neurologia são conhecidos desde meados do século passado(1). Das proposições iniciais até os dias de hoje, essa classe modelos tem evoluído em complexidade e técnicas de forma contínua, culminando em modelos com muitas camadas e níveis cada vez mais abstratos de representações (ver (2) para uma breve revisão histórica). Apesar dos avanços na área, foi apenas recentemente que modelos neurais começaram a redefinir o estado da arte, superando outras classes de algoritmos de aprendizado de máquina(3) e até mesmo alcançando performances sobre humanas(4). Tais avanços foram possíveis devido a três fatores chaves: a viabilização de bases de treino cada vez maiores o aumento do poder computacional e o desenvolvimento de novas arquiteturas neurais, como redes convolucionais e redes recursivas.

A crescente melhoria de performance dos modelos de aprendizado profundo tem motivado estudos em áreas onde se é preciso distinguir computadores e humanos. CAPTCHAs (5) (do inglês Completely Automated Public Turing tests to tell Computers and Humans Apart) definem uma coleção de técnicas que tem como objetivo bloquear a ação de agentes autônomos na rede mundial de computadores. O subconjunto mais conhecido dessas técnicas talvez seja o de captchas baseados em texto(6). Nesse tipo de desafio, uma imagem contendo uma sequência de caracteres é exibida. A validação é feita pela comparação entre o texto informado pelo usuário e a resposta correta. Em trabalhos recentes, foram relatadas acurácias próximos à humana em sequências formadas exclusivamente por números(7) ou por uma única fonte(8). Para o problema geral de quebrar captchas baseados em texto, entretanto, modelos de aprendizado profundo ainda mostram desempenho inferior ao humano. Contudo, pesquisas recentes apontam para avanços claros nos próximos anos(9). Em comum, esses modelos possuem a necessidade de muito poder computacional e/ou bases de dados extensivas. O treino dessas redes é tipicamente executado em clusters e/ou sistemas de computação sob demanda, com alto poder de paralelização e utilizando hardware de alto poder de processamento como GPUs e TPUs. Adicionalmente, As bases de treino comumente alcançam alguns terrabytes e envolvem grandes operações de aquisição e/ou geração.

2 Fundamentação

2.1 CAPTCHAS

CAPTCHAS podem ser formulados com um desafio sobre um conjunto de domínio cuja a resposta é um token. O domínio pode ser um trecho de áudio, uma sequencia de imagens ou até mesmo o histórico de navegação ou ambiente de desafiado. O token pode ser constituído de um conjunto de ações, o texto extraído do áudio ou imagem, ou possuir um histórico de navegação de baixo risco.

CAPTCHAS de texto podem ser vistos como um problema de extração de texto em imagens, sendo assim uma generalização para o problema de OCR (optical character recognition). Entretanto, CAPTCHAS são especialmente desenvolvidos para serem de difícil solução para computadores e preferencialmente fáceis para seres humanos. Assim, algoritmos usuais de OCR tendem a demonstrar baixo desempenho na solução desses desafios.

Matematicamente, uma imagem com altura H, largura W e C canais pode ser representada como um tensor $x \in \Re^{HxWxC}$, onde H, C e C são, respectivamente, a altura o comprimento e o número de canais da imagem. O token é uma sequencia w sob um alfabeto Σ. O desafiante passa na tarefa de acertar cada elemento w_i da sequencia.

2.2 Redes Neurais

De forma geral, aprendizado de máquina supervisionado pode ser descrito como, dado um conjunto de exemplos $D = \{(x, y)\}$, onde x pertence ao domínio do treino e y o rótulo associado, desejamos encontrar a função $\hat{y} = f(x)$, de tal modo que $\hat{y}$ seja o mais similar o possível à y dado x. Por 'mais similar o possível' entende-se que conhecemos uma função de erro que é tão menor quanto melhor for a aproximação dada por $f(x)$. Formalmente, desejamos encontrar f^* tal que

$$f^* = \min_f \langle err(y, f(x)) \rangle_D. \tag{2.1}$$

onde $\langle \ldots \rangle_D$ representa o valor esperado no conjunto D. $J_D = \langle err(y, f(x)) \rangle_D$ é usualmente referido como o *custo*.

Redes neurais são um conjunto de técnicas inspiradas em processos cognitivos desempenhados pelo sistema nervoso que fornecem uma maneira de descrever famílias de funções. Dada uma família de funções $f^\Theta : x \leftarrow y$ definida por uma rede neural e parametrizadas por Θ, podemos vasculhar o espaço de busca induzido por $\{\Theta\}$ para

encontrar um função que satisfaça alguma propriedade de interesse. Em particular, no caso de aprendizado de máquina, estamos interessados em encontrar o parâmetro Θ tal que:

$$\Theta^* = \min_{\Theta} \langle err(y, f^{\Theta}(x)) \rangle_D. \tag{2.2}$$

Quando munidos de um algoritmo de busca, podemos vasculhar a família de funções descritas por uma rede neural em

Redes Neurais

Camadas Totalmente conectadas.

3 Modelagem

definição da nomenclatura

definição das redes

4 Metodologia

4.1 Geração dos CAPTCHAS

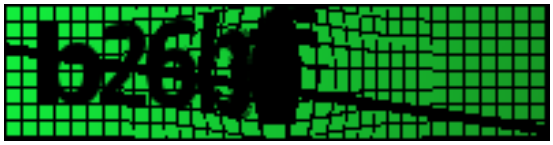
foram geradas 30000 imagens usando diferentes efeitos e cores com tokens de comprimento fixo em 5 utilizando-se a biblioteca (10)

um exemplo constitui de um tensor $(200, 50, 3)$ e de um token $(5, 36)$

exemplos normalizados entre 0-1, token codificado one-hot, de modo que

$$p(y[i]|x) = \begin{cases} 1, & \text{if } y[i] = w_i \\ 0, & \text{caso contrário.} \end{cases} \tag{4.1}$$

Os 30000 exemplos foram separados, de forma aleatória, em dois conjuntos: o conjunto de treino, D_{tr}, e o de validação, D_{va}, com 20000 e 10000 exemplos, respectivamente.



(a) b26bf

(b) ep8nb

(c) b7rw8

(d) 74wf6

(e) dnyny

(f) g4cxh

Figura 1 – Exemplos de CAPTCHAS gerados e seus respectivos tokens.

4.2 Grandezas de interesse

$$S_i^{(D)} = \sum_{(x,y)\in D} p(y[i]|x) \log \hat{p}(y[i]|x) \tag{4.2}$$

$$acc_w^{(D)} = \frac{N_w}{|D|} \tag{4.3}$$

$$\hat{p}_i^{(D)} = acc_i^{(D)} = \frac{N_i}{|D|} \tag{4.4}$$

$$\hat{p_w}^{(D)} = \prod_i \hat{p}_i^{(D)} \tag{4.5}$$

$$loss_i^{(D)} = \frac{S_i}{|D|} \tag{4.6}$$

$$loss^{(D)} = \sum_i loss_i^{(D)} \tag{4.7}$$

t tempo total de execução de uma época (treino + validação) $\tilde{t}$ tempo de treino em uma época.

4.3 Treino e Validação

Todos as redes foram treinadas em um mesmo computador com pentium core i5, 8gb de ram usando tensorflow.

Redes inicializadas segundo critério de (11)

etapa de treino consiste em mini batch: sortear $D_{batch} \subset D_{tr}$, com $|D_{batch}| = 10$ e minimizar S_i's nesse conjunto usando com (12) com taxa de aprendizado l_r.

O treino em uma época consiste em repetir $|D_{tr}|/|D_{batch}|$ vezes.

calcular as grandezas de interesse em D_{tr} e D_{va}

experimentos realizados com diferentes taxas de aprendizado durante 10 épocas. Limite superior e inferior escolhidos manualmente baseados em melhor desempenho aprendizado rápido para a superior e estável para a inferior. Decaimento linear.

Treinado usando critério de parada: mínimo de 10 épocas, máximo de 50 épocas, custo no conjunto de validação na época atual não maior que o .10 do menor valor encontrado até agora, custo no conjunto de treinamento atual menor que .03 da média dos últimos 5 valores. Inspirado em (13)

5 Resultados

Referências

1 ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, p. 65–386, 1958. Citado na página 8.

2 GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. Citado na página 8.

3 KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: PEREIRA, F. et al. (Ed.). *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., 2012. p. 1097–1105. Disponível em: ⟨http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf⟩. Citado na página 8.

4 MNIH, V. et al. Human-level control through deep reinforcement learning. *Nature*, Nature Publishing Group, a division of Macmillan Publishers Limited. All Rights Reserved., v. 518, n. 7540, p. 529–533, fev. 2015. ISSN 00280836. Disponível em: ⟨http://dx.doi.org/10.1038/nature14236⟩. Citado na página 8.

5 AHN, L. von et al. Captcha: using hard ai problems for security. In: *Advances in Cryptology, Eurocrypt*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003. v. 2656, p. 294–311. Citado na página 8.

6 CHOW, Y. W.; SUSILO, W. Text-based captchas over the years. *IOP Conference Series: Materials Science and Engineering*, v. 273, n. 1, p. 012001, 2017. Disponível em: ⟨http://stacks.iop.org/1757-899X/273/i=1/a=012001⟩. Citado na página 8.

7 GOODFELLOW, I. J. et al. Multi-digit number recognition from street view imagery using deep convolutional neural networks. *CoRR*, abs/1312.6082, 2013. Disponível em: ⟨http://arxiv.org/abs/1312.6082⟩. Citado na página 8.

8 GEORGE, D. et al. A generative vision model that trains with high data efficiency and breaks text-based captchas. v. 358, p. eaag2612, 10 2017. Citado na página 8.

9 BURSZTEIN, E. et al. The end is nigh: Generic solving of text-based captchas. In: *WOOT*. [S.l.: s.n.], 2014. Citado na página 8.

10 SIMPLECAPTCHA. *A CAPTCHA Framework for Java*. 2018. Disponível em: ⟨http://simplecaptcha.sourceforge.net/⟩. Acesso em: 23/06/2018. Citado na página 12.

11 GLOROT, X.; BENGIO, Y. Understanding the difficulty of training deep feedforward neural networks. In: *In Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10). Society for Artificial Intelligence and Statistics*. [S.l.: s.n.], 2010. Citado na página 13.

12 KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. Disponível em: ⟨http://arxiv.org/abs/1412.6980⟩. Citado na página 13.

13 PRECHELT, L. Automatic early stopping using cross validation: Quantifying the criteria. v. 11, p. 761–767, 06 1998. Citado na página 13.