Uso de redes neurais artificiais para reconhecer personagens com imagens de quadros de animação

Adson M. da S. Esteves1, Alisson S. Henrique1, Augusto C. Pluschkat1

1Ciência da Computação – Universidade do Vale do Itajaí (UNIVALI)  
CEP 88302-202 – Itajaí – SC – Brazil

{shinadson,ali.steffens}@gmail.com, acpluschkat@hotmail.com

**Resumo.** Este artigo tem por objetivo o desenvolvimento de um sistema para avaliar a necessidade de ajuda de um usuário da IDE Portugol Studio enquanto o mesmo programa, utilizando a lógica difusa e formulando regras para defuzzificar variáveis que contam os erros, avisos, número de linhas, de variáveis e funções que o usuário pode fazer no código. Após realizar testes com diferentes códigos com diferentes problemas e complexidades, o sistema foi validado. Ainda há melhoras a serem feitas como uma pesquisa melhor sobre as variáveis utilizadas e seus limites e detalhes de tempo de análise, porém o uso da lógica difusa foi no Portugol Studio foi aceita dentro dos parâmetros estabelecidos.

# 1. Introdução

Há problemas que não podem ser trivialmente formulados em forma de algoritmos, pois dependem de múltiplos fatores que para serem detalhados, gerariam uma quantidade exponencial de regras a serem implementadas e ajustadas. Enquanto um computador precisa seguir algorítmos para solucionar problemas, o cérebro humano possui o que chamamos de inteligência. Pode-se definir “inteligência” como a habilidade de adquirir, aprender e aplicar conhecimentos em novas situações (INTELLIGENCE, 2017). Com o objetivo de replicar tal capacidade de aprendizado, surgiu o conceito de redes neurais artificiais (RNAs).

Um dos principais pontos de RNAs é a capacidade de generalizar e associar dados através do aprendizado de padrões dentro de uma área bem especifícada. Dentro dos possíveis ramos de aplicação, pode-se citar o reconhecimento de imagens presente na área de visão computacional, que trata-se de utilizar uma entrada de pixels como passíveis a serem categorizados ou não como determinado objeto.

Com o objetivo de aprofundar os conceitos de RNAs, foi definido o uso de imagens de personagens de um jogo para treinar e verificar a eficiência da implementação de uma rede. Este artigo tem como foco apresentar o uso de uma RNA para o reconhecimento de quadros de animação de três personagens do jogo *Pokémon Diamond and Pearl*, e possui suas seções dividas da seguinte forma: a seção 2 explicará a fundamentação teórica de base do projeto; a seção 3 irá descrever um apanhado geral da metodologia aplicada; a seção 4 focará em apresentar as particularidades da implementação e do treinamento da RNA; na seção 5 serão exibidos os resultados obtidos através de testes para verificação da taxa de acertos da rede; e por fim, a seção 6 será focada nas conclusões obtidas com base nos resultados e apresentará sugestões para o que deve ser aperfeiçoado em busca de melhorar a taxa de acertos da RNA.

# 2. Fundamentação Teórica

Nesta seção serão discutidos os principais temas de interesse deste artigo, sendo eles: o histórico de redes neurais artificias e reconhecimento de imagens.

## 2.1 Redes Neurais Artificiais

Usando neurônios criados artificialmente, a ideia de redes neurais é simular o processo de pensamento do um ser humano, ao fazer com que cada neurônio artificial se comunique uns com os outros enviando sinais, como se fossem sinapses de um cérebro humano normal. Esse sistema foi feito para a identificação de padrões, quando não se sabe exatamente o problema ocorrente, resolvendo-o a partir de uma generalização dos dados com que é treinado.

Geralmente as redes neurais feitas, são multicamadas, onde cada camada de neurônios recebe entradas de uma camada anterior a ela, sendo assim, o sinal original entra pela primeira camada, passando por cada neurônio dela e liberando sinais para camadas inferiores, até se ter uma saída na última camada.

Foi primeiramente proposta e realizada em 1943 quando Warren McCulloch do o Instituto Tecnológico de Massachusetts e Walter Pitts da Universidade de Illinois criaram um modelo de redes neurais com algoritmos e matemática a partir de eletrônicos, chamado de Lógica de limite (Threshold logic), simulando um comportamento de neurônios. A ideia trazia que a saída de um neurônio era uma função da soma de diversas entradas [MACCULLOCH e PITTS, 1943].

Em 1958 Frank Rosenblatt crio o Perceptron, um algoritmo de redes neurais que utilizava apenas uma camada de neurônios[ROSENBLATT, 1985], porém após pesquisas de Marvin Minsky e Seymour Paper em 1969, as pesquisas no campo de redes neurais ficaram estagnadas, pois de acordo com o artigo deles haviam 2 problemas com o processamento de redes neurais, perceptrons básicos não conseguiam processar XOR e computadores da época não tinham poder de processamento suficiente para executar algoritmos de redes neurais.

Assim, Redes Neurais voltou em 1975, com o algoritmo de Backpropagation após Paul Werbos apresentar a proposta para redes neurais artificiais, aonde a saída final da rede é comparada com o resultado desejado e retreinada se necessária até que fique com um erro ou diferença ao valor desejado mínimo.



# 3. Metodologia

Foi definido o uso de uma RNA Perceptron de multicamadas para o reconhecimento de imagens, tendo como tema o reconhecimento de quadros de animação de três personagens selecionados do jogo *Pokémon Diamond and Pearl*:Bulbasaur, Squirtle e Charmander. Os quadros foram retiradas de um repositório de imagens online relacionada ao jogo.

<Imagem dos três personagens>

Além desses três personagens, também foram retirados quadros de outros 148 personagens para comparar e verificar a capacidade de reconhecimento da RNA gerada.

<Imagem de todos os sprites>

# 4. Desenvolvimento

Para a implementação do projeto, foi utilizada a biblioteca de código de aberto produzida em Javascript, Neataptic. Com a utilização de tal biblioteca, as RNAs foram armazenadas como arquivos no formato JSON. As imagens utilizadas para treino foram redimensionadas para 16x16 pixels e aplicadas filtro de tons de cinza para haver apenas um valor RGB por pixel, diminuindo o número de entradas nas redes.

<Imagem do front-end do programa>

## 4.1 Arquitetura da Rede Neural

Na fase inicial do projeto planejava-se gerar apenas uma instância de rede neural para satisfazer o objetivo de reconhecimento dos três personagens, porém após testes percebeu-se que a qualidade da rede obtida era muito baixa mesmo com ajustes de parâmetros e mais de duas horas de execução por tentativa de treinamento. Por fim, optou-se por gerar três redes, uma para cada personagem, com parâmetros e treinos similares.

Cada RNA possui 256 nós de entrada para satisfazer o número de pixels das imagens de 16x16, assim como um nó de saída, com valores de 0 a 1, sendo que quanto mais próximo do 1, maior foi o reconhecimento positivo em relação à imagem testada. Foi utilizada apenas uma camada oculta de 257 nós, visto que o uso de menos nós geraram resultados não satisfatórios nas fases iniciais de testes e mais nós seriam inviáveis de se utilizar pela alta demanda de tempo para se executar uma tentativa de treinamento.

## 4.2 Regras de Treinamento

Para cada rede, foram utilizadas dez imagens para treinamento, sendo nove imagens de personagens não correspondentes ao que se gostaria de reconhecer e uma imagem de quadro do personagem para ser reconhecido. Cada execução de treinamento recebeu os seguintes parâmetros: taxa de erro de 0.0000001; taxa de aprendizado de 0.001; e número de iterações máximo de 50000.

A baixa taxa de erro foi definida para que o número de iterações máximo fosse alcançado, visando melhorar a qualidade geral da rede. O valor da taxa de aprendizado se deu por notar-se que quanto menor a taxa de aprendizado até determinado ponto, menor era a taxa de erro de erro obtida após o término das iterações. Valores menores que 0.001 para a taxa de aprendizado não mostraram impacto na taxa de erro obtida, portanto definiu-se fixar a taxa nesse valor.

# 5. Experimentos e Resultados

Fora.

# 6. Conclusões

A.

# Referencias

INTELLIGENCE. Merriam-Webster Online Dictionary. 2017. Disponível em <merriam-webster.com>. Acesso em 30 de maio de 2017.

BEAUBOUEF, Theresa; MASON, John (2005). Why the high attrition rate for computer science students: some thoughts and observations. In: ACM SIGCSE Bulletin, USA, v. 37, n. 2, p. 103-106, jun. 2005.

CASTRO, T. H. C. (2002) et al. “Arquitetura SAAP - Sistema de Apoio à Aprendizagem de Programação”. XXII Congresso da Sociedade Brasileira de Computação, volume 5 - VIII Workshop de Informática na Escola, Florianópolis.

KINNUNEN, P. and MALMI, L. (2006) “Why students drop out CS1 course?” In Proceedings of the Second international Workshop on Computing Education Research (Canterbury, United Kingdom).ICER '06.ACM, New York, NY, 97-108.

LAHTINEN, E.; ALA-MUTKA, K.; JÄRVINEN, H.-M (2005). A study of the difficulties of novice programmers. Annual Sigcse Conference On Innovation And Technology In Computer Science Education, v. 37, n. 3, p. 14–18, Caparica, Portugal.

LOFTI A. Zedah. (1964). Fuzzy Sets. Information and Control. http://www.sciencedirect.com/science/article/pii/S001999586590241X , November.

LOFTI A. Zedah. (1964). A computational approach to fuzzy quantifiers in natural languages. Computers & Mathematics with Applications. http://www.sciencedirect.com/science/article/pii/0898122183900135, November.

MANNES, Paula (2013). Integração do Portugol Core com o Bipide. http://siaibib01.univali.br/pdf/Paula%20Mannes.pdf, November.

NORVIG, Peter; RUSSELL, Stuart Jonathan (2004). Inteligência artificial. editora Campus.

NOSCHANG, Luiz Fernando (2012). Adaptação Do Portugol Core Para Permitir A Integração Com Outras Ferramentas. Universidade do Vale do Itajaí.

NOSCHANG, Luiz Fernando; PELZ, Fillipi; JESUS, Elieser A. de Jesus; RAABE, André L. A.(2014). Uma IDE para Iniciantes em Programação. In: Workshop de Informática na Escola, No prelo.

OLIVEIRA, B. C. d. S.; WANG, M., GIBBONS, J. (2008). The visitor pattern as a reusable, generic, type-safe componente OOPSLA ’08 Proceedings of the 23rd ACM SIGPLAN conference on Object-oriented programming system languages and applications.

PELZ, Fillipi.(2011). Correção Automática de Algorítmos no Ensino Introdutório de Programação. http://lite.acad.univali.br/portugol/resources/pdfs/tra02.pdf, June.

PELZ, Fillipi (2014). Um Gerador de Dicas para Guiar Novatos na Aprendizagem de Programação. http://lite.acad.univali.br/portugol/resources/pdfs/dis01.pdf, February.

RAABE, André L. A.; SILVA, Júlia M. Carvalho da. Um ambiente para atendimento as dificuldades de aprendizagem de algoritmos. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 25., São Leopoldo, Rio Grande do Sul. Anais... RS:SBC, 2005.

SCHEFFLER, P. (2008). Teaching Algorithmics – Theory and Pratice. In: Proc. 2nd Intern. Sc. Conf. “Informatics in the Scientific Knowledge.” Varna, pp. 259-269.

SHEARD (2009), Judy et al. Analysis of research into the teaching and learning of programming. In: INTERNATIONAL COMPUTING EDUCATION RESEARCH WORKSHOP, 5., 2009, Berkeley, USA. Proceedings… USA: ACM, 2009. p. 93-104.