Uso de redes neurais artificiais para reconhecimento de 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 apresentar o desenvolvimento de um sistema de reconhecimento de imagens a partir do algoritmo de redes neurais artificiais chamado backpropagation. Foram escolhidos como imagens para reconhecimento quadros de animação de três personagens do jogo Pokémon Diamond and Pearl, sendo realizados treinamentos com regras iguais em três redes, uma para cada personagem, e tendo dez imagens de treinamento com apenas uma correspondente do personagem alvo da rede neural artificial. Nos experimentos as redes apresentaram facilidade em encontrar seus pares exatos dentro do conjunto de treinamento, porém para o conjunto exclusivo de experimentos, a rede encontrou dificuldade moderada em reconhecer os personagens, o que provavelmente se deve ao fato de que o número de imagens utilizadas no treinamento foi consideravelmente pequeno, além de que as imagens precisaram ser reduzidas e descoloridas para reduzir o tempo de treinamento. Conclui-se que para se obter melhores resultados nos reconhecimentos propostos, seria necessário uma rede com um número maior de neurônios e imagens de treinamentos, ou abordagem mais especializada com o uso redes convolutivas.

# 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 algoritmos 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 especificada. Dentro dos possíveis ramos de aplicação, pode-se citar o reconhecimento de imagens presente na área de visão computacional, que se trata 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

Com a utilização de 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 entre si enviando sinais, como se fossem sinapses de um cérebro humano natural. Esse sistema foi feito para a identificação de padrões ou 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 [MCCULLOCH e PITTS, 1943].

Em 1958 Frank Rosenblatt criou 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, o conceito de RNAs voltou em 1975 com o algoritmo de *backpropagation* após Paul Werbos apresentar a proposta de que 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[WERBOS, 1975].



Figura 1. Representação de uma Rede Neural Artificial de backpropagation

Conforme Figura 1, as entradas vão para a camada de entrada (Input Layer), em seguida para a camada oculta (Hidden Layer) e após ir para a saída(Output) um erro é calculado (Error Calculation), e se não estiver de acordo a saída é recolocada na camada de entrada, caso contrário, esta é a resposta da rede.

## 2.2 Reconhecimento de Imagens

O cérebro humano, por padrão, não apresenta dificuldade em reconhecer formas e objetos a partir da visão. Como por exemplo distinguir animais ou objetos, identificar símbolos, e até mesmo identificar faces humanas. Tais tarefas, porém, só aparentam ser simples pelo fato do cérebro humano ser altamente eficiente em entender imagens, por outro lado, tais problemas são extremamente difíceis de serem resolvidos computacionalmente e são estudados abordados no ramo da visão computacional.

Pesquisadores vem demonstrando um avanço significativo em visão computacional através do modelo de redes neurais convolutivas(ver Figura 2), [OLAH, 2014], [SURYANI, 2017], que tem demonstrado performance razoável em tarefas difíceis de reconhecimento visual, muitas vezes se igualando ou até mesmo superando a visão humana.

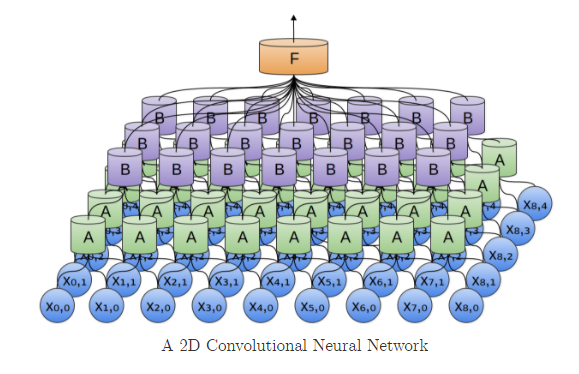


Figura 2. Representação de uma Rede Neural Artificial Convolutiva

Tais redes tem sido validadas através de comparações de atividades feitas com imagens da ferramenta ImageNet, um banco de dados de imagens para pesquisadores, trazendo um benchmark acadêmico para visão computacional, e os modelos vêm sucessivamente demonstrando melhorias, como podem ser vistos nos modelos: QuocNet descrito em *Building High-level Features Using Large Scale Unsupervised Learning* (2012), AlexNet descrito em *ImageNet Classification with Deep Convolutional Neural Network* (2012), Inception (GoogLeNet) descrito em *Going Deeper with Convolutions* (2014) , BN-Inception-v2 descrito em *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift* (2015) e finalmente Inception-v3, o atual modelo utilizado pela Google e que é descrito em *Rethinking the Inception Architecture for Computer Vision* (2015).

# 3. Metodologia

Foi definido o uso de uma RNA Perceptron de multicamadas com algoritmo de *backpropagation* 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 retirados de um repositório de imagens online relacionada ao jogo.

Figura 3. Imagens originais dos personagens Bulbasaur, Charmander e Squirtle respectivamente, retiradas do repositório online

Além desses três personagens, também foram retirados dois quadros, um para treinamento e outro exclusivo para experimentos, dos outros 148 personagens para comparar e verificar a capacidade de reconhecimento da RNA gerada.

# 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.

Figura 4. Imagens tratadas dos personagens Bulbasaur, Charmander e Squirtle respectivamente, modificadas a partir das originais no repositório online



## 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 é 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

Pretendia-se utilizar todas as imagens do primeiro conjunto, entretanto a demanda de tempo para executar uma instância de teste deixou a proposta inviável. Por fim, foram utilizadas dez imagens do primeiro conjunto para treinamento de cada rede, 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

Para a realização dos testes, cada imagem foi nomeada sequencialmente de 1 até 151, havendo dois conjuntos de imagens, sendo o primeiro o mesmo que foi utilizado para o treinamento e o segundo exclusivo para testes, tendo nesse segundo conjunto os personagens em posições diferentes.

Figura 5. Imagens do Squirtle no conjunto 2 e 1 respectivamente

As imagens 1, 4 e 7 nas duas listas correspondem aos personagens alvo do reconhecimento das RNAs geradas. As saídas geradas foram transformadas em valores percentuais, e considerou-se valores acima de 50% como um reconhecimento positivo do personagem alvo da RNA.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Reconhecimento | | Falso Positivos |
| Conjunto 1 | Conjunto 2 |
| Bulbasaur | 87.8% | 51.2% | 6.7% |
| Charmander | 98.7% | 62.5% | 10% |
| Squirtle | 99.3% | 18.3% | 2% |

Tabela 1. Percentuais de reconhecimento dos quadros alvo da RNA e o percentual de falso positivos encontrados no total dos dois conjuntos

O conjunto 1, que foi utilizado para treinar as RNAs, apresentou um resultado final satisfatório de reconhecimento dos personagens alvo conforme exibido pela tabela 1. No conjunto 2, dois personagens foram reconhecidos com sucesso, porém houve um reconhecimento baixo para o terceiro personagem.

Os falsos positivos exibidos na tabela 1 consideram os dois conjuntos juntos, possuindo 300 quadros no total após desconsiderar os dois quadros do personagem alvo em cada contagem. Para a RNA do personagem Bulbasaur, houveram 6 falso positivos no conjunto 1 e 14 no conjunto 2; no caso do personagem Charmander, houveram 16 falso positivos no conjunto 1 e 14 no conjunto 2; e para o personagem Squirtle, o conjunto 1 obteve 2 falso positivos e o conjunto 2 obteve 3 falso positivos.

# 6. Conclusões

Comparando os resultados da pesquisa, que utilizou de redes Perceptron de multicamadas padrões, e os modelos descritos na literatura com a utilização de redes neurais convolutivas, é possível afirmar que muitos aspectos influenciaram diretamente na taxa de convergência dos experimentos neste artigo descritos, sendo os principais: amostra, modelo de rede, pré-processamento e linguagem de programação.

Neste experimento foram utilizados para treino apenas 1 exemplar de cada *Pokémon*, sendo que desta forma, os resultados dependem muito da diferença de posicionamento de um quadro para o outro, como é o caso do personagem Squirtle (ver Figura 5), que com o quadro de treino possuiu uma convergência de 99,3%, porém somente 18,3% com o quadro de testes. Neste caso em específico ficam evidentes as diferenças entre o quadro de treino e o de testes. A utilização de mais casos poderia diminuir o impacto da movimentação do personagem.

Caso fossem utilizadas mais imagens de entrada, outra melhoria possível seria uma melhor escolha de métodos de pré-processamento de imagens, visto que o efetuado no experimento envolvia apenas o redimensionamento do quadro e a descoloração do mesmo (ver Figuras 3 e 4). Redimensionamento este que foi empregado dada necessidade de diminuir o tempo de treinamento da rede, que mesmo reduzida ainda necessita de aproximadamente duas horas para o treino.

Tal demora no treinamento da rede deve-se por conta principalmente do tipo de rede utilizada, visto esse tipo de rede não ser o mais indicado para reconhecimento de imagens devido a fatores de complexidade, e a linguagem de programação escolhida, Javascript, que mesmo facilitando em muitos aspectos a programação, não apresenta desempenho otimizado para uso das tecnologias propostas no projeto.

# Referencias

INTELLIGENCE. Merriam-Webster Online Dictionary. 2017. https://www.merriam-webster.com/.

MUCCULLOCH, WARREN; WALTER PITTIS. (1943). A Logical Calculus of the Ideas Immanent in Nervous Activity. Bullettin of Mathematical Biophysics. https://link.springer.com/article/10.1007%2FBF02478259.

ROSENBLATT, F. (1943). The Perceptron: A Probabilistc Model For Information Storage And Organization In The Brain. Psychological Review. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.588.3775.

WERBOS, P.J. (1975). Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences. https://www.researchgate.net/publication/35657389\_Beyond\_regression\_new\_tools\_for\_prediction\_and\_analysis\_in\_the\_behavioral\_sciences.

OLAH, CHRISTOPHER (2014). Conv Nets: A Modular Perspective. http://colah.github.io/posts/2014-07-Conv-Nets-Modular/.

SURYANI, DEWI (2017). Convolutional Neural Network. BINUS UNIVERSITY School of Computer Science. http://socs.binus.ac.id/2017/02/27/convolutional-neural-network/.

LE, QUOC V.; RANZATO, MARC’AURELIO; MONGA, RAJAT; DEVIN, MATTHIEU; CHEN, KAI; CORRADO, GREG S.; DEAN, JEFF; NG ANDREW Y.; Building High-level Features Using Large Scale Unsupervised Learning. http://static.googleusercontent.com/media/research.google.com/en//archive/unsupervised\_icml2012.pdf.

KRIZHEVSKY, ALEX; SUTSKEVER, ILYA; HINTON GEOFFREY E. ImageNet Classification with Deep Convolutional Neural Networks. http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf .

SZEGEDY, CHRISTIAN; LIU, WEI; JIA, YANGQING; SERMANET, PIERRE; REED, SCOTT; ANGUELOV, DRAGOMIR; ERHAN, DUMITRU; VANHOUKE, VINCENT; RABINOVICH, ANDREW. Going Deeper with Convolutions. https://arxiv.org/abs/1409.4842.

LOFFE, SERGEY; SZEGEDY, CHRISTIAN; Batch Normalization: Accelerating Deeper Network Training by reducing Interval Covariate Shift. https://arxiv.org/abs/1502.03167.

SZEGEDY, CHRISTIAN; VANHOUKE, VINCENT; LOFFE, SERGEY; SHLENS, JONATHON; WOJNA, ZBIGNIEW. Rethinking the Inception Architecture for Computer Vision. https://arxiv.org/abs/1512.00567 .