

Priorização de Casos de Teste Utilizando Redes Neurais Artificiais

Erik Lopes da Silva, André Luiz Alves

Escola Politécnica - Pontifícia Universidade Católica de Goiás

eriklopes562@gmail.com, andre.luiz@pucgoias.edu.br

Abstract. *Software testing is one of the most arduous steps in software systems development. For the product to present the expected quality, several test cases are executed. A single requirement can generate a large amount of test cases, and due to this high number, the manual test often becomes very costly to the software development, making it essential to prioritize those test cases where there is a greater possibility of occurrence incidence of failures. This work presents a way to automate the prioritization of test cases using artificial neural networks.*

Keywords: *Software testing, Artificial Intelligence.*

Resumo. *O teste de software é uma das etapas mais árduas no desenvolvimento de sistemas de software. Para que o produto apresente a qualidade esperada, diversos casos de teste são executados. Um único requisito pode gerar uma grande quantidade de casos de teste, e devido a este número elevado, o teste manual se torna por muitas vezes muito oneroso, fazendo com que seja fundamental priorizar aqueles casos de teste onde há maior possibilidade de incidência de falhas. Este trabalho apresenta uma forma de se automatizar a priorização de casos de teste utilizando redes neurais artificiais.*

Palavras-Chave: *Teste de Software, Inteligência Artificial.*

1. Introdução

A demanda por *softwares* que sejam produzidos em um curto espaço de tempo e ao mesmo tempo confiáveis, tem motivado esforços que possam auxiliar de forma eficiente o teste de *software* [Bajaj e Sangwan, 2019]. O teste de *software* evidencia a presença de erros, mas não garante a ausência deles, a meta do teste de software é mostrar que o sistema está pronto para uso operacional, sendo este um processo que visa demonstrar sua confiabilidade [Dijkstra 1972].

Um único requisito pode gerar uma grande quantidade de casos de teste, tornando sua priorização uma tarefa bastante árdua para a equipe envolvida no projeto. Este grande conjunto de dados torna por muitas vezes inviável a verificação em sua totalidade, surge então a necessidade de se priorizar os casos de teste [Miranda et. al, 2018]. A priorização de casos de teste os ordena em algumas propriedades de modo que aqueles casos de teste altamente classificados sejam executados em uma prioridade mais alta [Yoo e Harman 2012].

O *Machine Learning* (ML) é uma subcategoria da Inteligência Artificial (IA) utilizada no processo de reconhecimento de padrões pelos computadores. O ML fornece recursos muito eficientes para a extração e classificação de informações, como as Redes Neurais Artificiais (RNAs) [Hart et. al, 2021].

As RNAs são algoritmos computacionais que apresentam um modelo matemático inspirado na estrutura neural de organismos inteligentes tentando emular em computadores o funcionamento do cérebro humano [Spörl et. al, 2011]. Diante do exposto, este artigo apresenta uma forma de se realizar a priorização de casos de teste utilizando RNAs.

2. Trabalhos Relacionados

[Ashraf et al, 2017] propuseram uma técnica em que classificaram os casos de teste em seis fatores e utilizaram o algoritmo *Particle swarm optimization* (PSO) para realizar sua priorização. O algoritmo proposto utiliza os valores da função *fitness* e os valores da equação de velocidade para calcular a ordem mais otimizada dos casos de teste. Segundo os autores, nos testes realizados a solução proposta prioriza até cerca de 78% dos casos de teste de forma correta.

No método de priorização proposto no trabalho de [Silveira et al 2020], utiliza-se a associação entre um algoritmo de ordenação e um comparador de atividades. O algoritmo de ordenação recebe um conjunto de atividades não priorizadas e por meio da comparação de pares realizada por uma RNA, produzindo um conjunto ordenado de atividades em relação a sua prioridade.

Os dois trabalhos citados acima contribuíram diretamente para o desenvolvimento da solução proposta. Os fatores de classificação apresentados por [Ashraf et al, 2017] mostraram ser uma forma interessante de se realizar o treinamento de algoritmos de ML. A utilização de RNA conforme apresentada por [Silveira et al 2020] é um excelente método classificação.

3. Metodologia

A solução proposta para realizar a priorização de casos de teste faz uso de RNA utilizando uma função matemática para converter um vetor numérico em um vetor de probabilidades. A função recebe um conjunto de casos de teste não priorizado com pesos variando de 1 a 10 em quatro categorias e, retorna um conjunto ordenado de casos de teste de acordo com sua probabilidade de ser classificado como prioritário.

O conjunto de dados é composto por n casos de teste $CT = [ct_1, ct_2, \dots, ct_n]$ de modo que cada caso de teste é constituído por 4 atributos, onde $A_x = [a_1, a_2, a_3, a_4]$. Os atributos A_x são detalhados a seguir:

- a₁: Se refere a **imprescindibilidade** do caso de teste especificado.
- a₂: Se refere a **complexidade** de se implementar esta funcionalidade no sistema.
- a₃: Se refere a **volatilidade** do requisito que deu origem ao caso de teste.
- a₄: Se refere ao **impacto da ausência do caso de teste**.

O subconjunto de treinamento consiste em 80% do conjunto original de casos de teste, os 20% remanescentes são utilizados para validar o treinamento realizado. Após o conjunto de dados passar pelo modelo de regressão logística, dois novos valores são atribuídos a cada caso de teste, estes valores são conhecidos como *logits*. Os *logits* passam pela função de ativação retornando à probabilidade do caso de teste pertencer a cada uma das classes.

Tabela 1. Parâmetros utilizados no treinamento da RNA

Propriedade	Valor
Função de ativação	<i>Softmax</i>
Otimizador	<i>Stochastic Gradient Descent</i>
<i>Learning rate</i>	0,001 a 0,05
Número de épocas	1000
<i>Momentum</i>	0,00

A tabela 1 apresenta os parâmetros específicos utilizados no treinamento da RNA, descritos a seguir:

Função de ativação *Softmax* converte um vetor de K números reais em uma distribuição de probabilidade de K resultados possíveis;

O *Stochastic Gradient Descent* é um algoritmo de otimização utilizado para encontrar os parâmetros do modelo que correspondem ao melhor ajuste entre as saídas previstas e reais;

Learning rate é um parâmetro de ajuste em um algoritmo de otimização que determina o tamanho da etapa em cada iteração enquanto se move para um mínimo de uma função de perda;

Número de épocas representa a quantidade de iterações que a RNA será executada para aprender o modelo;

O *Momentum* é uma constante que determina o efeito das mudanças passadas dos pesos na direção atual do movimento no espaço de pesos.

4. Resultados Obtidos

Nas figuras 1 e 2, o primeiro gráfico representa a taxa de perda a cada época. O segundo gráfico representa a classificação dos casos de teste, onde a linha azul representa os casos de teste classificados como prioritários e, a linha amarela representa os casos de teste classificados como não prioritários. Ao fixar o *learning rate* em 0,001 a probabilidade média para que determinado caso de teste fosse classificado como prioritário ficou em cerca de 60%. Nas épocas iniciais a taxa de perda se manteve acima de 3,0, porém, após a época 200 esta taxa se estabilizou em cerca de 0,6.

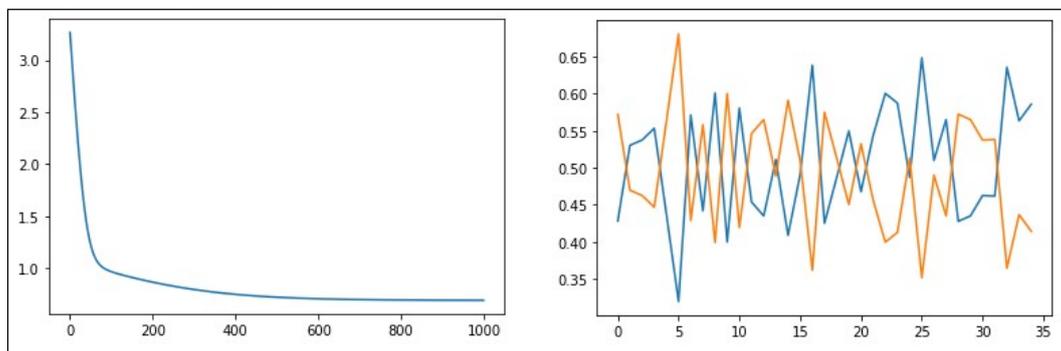


Figura 1. Loss e percentual ao priorizar lr 0,001

Quando o *learning rate* foi fixado em 0,05 a taxa de perda se estabilizou em cerca 0,6 em apenas 100 épocas e, a probabilidade alcança seus maiores resultados com cerca de 90%, porém, o *loss* também sobe de forma significativa alcançando taxas de perda próximas de 1,5. Desta forma, durante os testes se observou que uma taxa de *learning rate* satisfatório esta localizada entre 0,001 e 0,03.

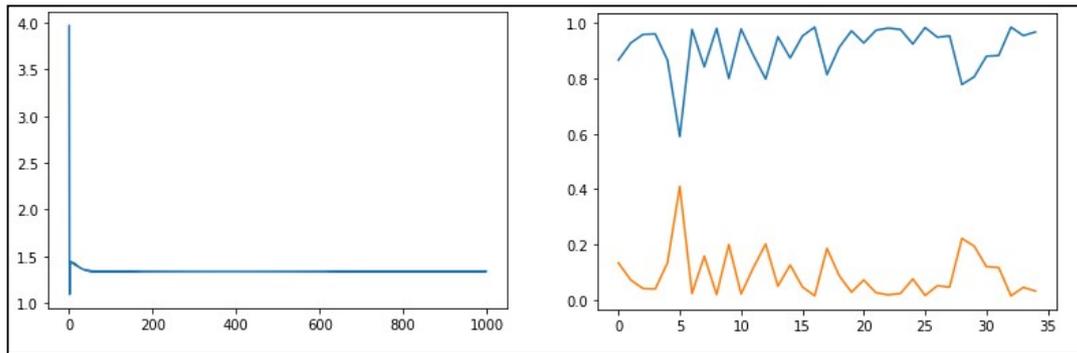


Figura 2. Loss e percentual ao priorizar lr 0,05

5. Considerações Finais

Este trabalho apresenta uma forma de se realizar a priorização de casos de teste utilizando recursos proporcionados pela IA, em especial a aprendizagem de máquina. No contexto considerado, o estudo apresenta resultados promissores, apontando que as RNAs podem contribuir com a priorização de casos de teste e, a solução apresentada não está sujeita às limitações de escalabilidade que cercam a priorização de forma manual.

Sugere-se como trabalhos futuros a implementação do modelo em ambiente de produção com um conjunto de dados mais robusto, bem como testes com diferentes números de épocas e *learning rate* e, comparar o desempenho desta solução com outros algoritmos de ML.

Referências

- Ashraf, E., Mahmood, K., Khan, T. A., & Ahmed, S. (2017). **Value based PSO test case prioritization algorithm**. International Journal of Advanced Computer Science and Applications.
- Bajaj, A., & Sangwan, O. P. (2019). **A systematic literature review of test case prioritization using genetic algorithms**. IEEE
- Dijkstra, E. W. (1972). **The humble programmer**. Communications of the ACM.
- Hart, G. L., Mueller, T., Toher, C., & Curtarolo, S. (2021). **Machine learning for alloys**. Nature Reviews Materials.
- Miranda, B., Cruciani, E., Verdecchia, R., & Bertolino, A. (2018). **FAST approaches to scalable similarity-based test case prioritization**. International Conference on Software Engineering (ICSE).
- Silveira, M. J., Marasca, A. L., Marcon, M., & Ortoncelli, A. R. (2020). **Priorização de Atividades de Desenvolvimento de Software com uma Rede Neural Artificial**. Anais do Computer on the Beach.
- Spörl, Christiane, Emiliano Castro, and Aílton Luchiari (2011). **Aplicação de redes neurais artificiais na construção de modelos de fragilidade ambiental**. Revista do Departamento de Geografia
- Yoo, S., & Harman, M. (2012). **Regression testing minimization, selection and prioritization: a survey**. Software testing, verification and reliability.