MINISTÉRIO DA CIÊNCIA E TECNOLOGIA
INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS

Teste Metamórfico para Produtos de Software relacionados à Definição de Características Hidrológicas

RELATÓRIO DE RENOVAÇÃO DE PROJETO DE INICIAÇÃO CIENTÍFICA

(PIBIC/CNPq/INPE)

Aluna: Renata Bitencourt Vaz (UNESP, Bolsista PIBIC/CNPq)

E-mail: renatabitencourtvaz@gmail.com

Orientadores: Dr. Valdivino Alexandre de Santiago Júnior (LABAC/ COCTE/INPE)

E-mail: valdivino.santiago@inpe.br

Dr. Alan James Peixoto Calheiros (LABAC/COCTE/INPE)

E-mail: alan.calheiros@inpe.br

Julho de 2019

RESUMO

O INPE tem vários exemplos de softwares científicos usados para, por exemplo, implementar modelos de sistemas de controle de atitude e órbita de satélites, analisar a radiação X e Gama oriundas de objetos astrofísicos a bordo de plataformas espaciais, estudar a interação bidirecional entre a biosfera terrestre e o sistema climático, implementar modelos numéricos de previsão de tempo assim como modelos de estudos climáticos, entre outros. É muito importante que tais produtos sejam criados com o mais alto nível de qualidade para que seus resultados possam ser confiáveis, e, dessa forma, ser uma fonte adequada para a tomada de decisão de usuários que fazem uso de tais softwares. No entanto, software científico é bastante complexo e garantir a sua qualidade é algo bastante desafiador. Teste de software é um processo que justamente é aplicado para aumentar a qualidade de sistemas de software. Porém, determinados tipos de software, tais como os softwares científicos, podem padecer do chamado problema do oráculo, ou seja, quando não é possível determinar com certeza os resultados esperados de um conjunto de dados de entrada de teste. Teste metamórfico é uma técnica que se adequa a esse tipo de sistema de software. Mas, a principal limitação do teste metamórfico é a identificação das Relações Metamórficas (RMs), que é uma tarefa manual que requer um bom conhecimento do problema em questão. Uma das alternativas é tentar identificar, automaticamente, as RMs de um Sistema Sob Teste. Esse projeto de pesquisa possui dois objetivos específicos: a.) investigar a viabilidade de teste metamórfico para realizar o teste de software científico, particularmente abordando o problema do oráculo de teste de software. Para identificar automaticamente as RMs será investigada a possibilidade de uso de Aprendizado Profundo (e.g. Redes Neurais Convolucionais); b.) aplicar a abordagem proposta a produtos de software relacionados à definição de características hidrológicas, com os quais o INPE vêm trabalhando, com o intuito de melhorar a qualidade dos mesmos. Esse relatório apresenta as atividades desenvolvidas no período de 01 de agosto de 2018 a 30 de junho de 2019.

1.) INTRODUÇÃO

Software científico pode ser definido, simplificadamente, como software usado para propósitos científicos [Kanewala e Bieman 2014]. Diversos produtos de software criados para as indústrias nuclear e aeroespacial, medicina e sistemas militares se enquadram nessa categoria. O INPE tem vários exemplos de softwares científicos usados para, por exemplo, implementar modelos de sistemas de controle de atitude e órbita de satélites, analisar a radiação X e Gama oriundas de objetos astrofísicos a bordo de plataformas espaciais, estudar a interação bidirecional entre a biosfera terrestre e o sistema climático, implementar modelos numéricos de previsão de tempo assim como modelos de estudos climáticos, entre outros. É relevante mencionar que os resultados (saídas) fornecidos pela execução de tais softwares científicos são tomados como base para tomadas de decisão importantes sejam elas relacionadas a saúde da população, relacionadas ao desenvolvimento de projetos científicos e de inovação, ou relacionadas a políticas sociais ou industriais. Portanto, é extremamente importante que tais produtos sejam criados com o mais alto nível de qualidade para que seus resultados possam ser confiáveis.

No entanto, software científico é bastante complexo e, devido ao conhecimento no domínio exigido, usualmente o desenvolvimento acaba sendo realizado por cientistas [Pipitone e Easterbrook 2012] [Kanewala e Bieman 2014]. Porém, de um modo geral, cientistas não estão acostumados, ou mesmo desconhecem, as melhores práticas usadas por engenheiros de software para a criação de ferramentas. E isso pode impactar consideravelmente na qualidade do software científico. Alguns exemplos de problemas relatados em softwares científicos são destacados a seguir.

Cinco artigos tiveram que ser retratados (ou seja, foram reconhecidos que possuíam informações notoriamente erradas), sendo três deles após terem sido publicados na revista Science e mais dois em outros periódicos, devido a defeitos no software científico que gerou os resultados para os trabalhos [Miller 2006]. Os problemas estavam relacionados à estruturas de certas proteínas descritas nos trabalhos que mostraram estar erradas. A causa foi um programa de análise de dados que trocou, de forma equivocada, duas colunas de dados invertendo o mapa de densidade eletrônica a partir do qual foi derivada a estrutura proteica final. Somente o artigo mais influente dos que foram retratados tinha recebido, por volta de 2006, em torno de 364 citações.

Para aplicações de processamento de dados sísmicos, um experimento foi realizado que demonstrou que diversos pacotes de software científicos comerciais, que implementavam algoritmos científicos baseados em especificações idênticas ou similares e

usando a mesma linguagem de programação (Fortran), produziram resultados consideravelmente diferentes, quando submetidos aos mesmos conjuntos de dados de entrada e configurados com os mesmos parâmetros [Hatton e Roberts 1994]. Os autores concluíram que a causa primordial dessas diferenças era devido a defeitos nos pacotes de software.

Em outros artigos, cientistas afirmaram que pensavam ter que modificar o modelo que representava o fenômeno físico ou desenvolver novos algoritmos, mas depois chegaram a conclusão que as causas dos problemas eram defeitos no código-fonte [Dubois 2012]. Esses são apenas alguns dos exemplos que mostram a importância de aplicar as diretivas da Engenharia de Software para desenvolver software científico com alta qualidade.

Portanto, as metodologias, técnicas, atividades e processos da Engenharia de Software podem contribuir para melhorar a qualidade de um produto. A área de Verificação e Validação (V&V) da Engenharia de Software almeja contribuir para essa melhoria. Teste de Software é, provavelmente, o processo mais adotado, na prática, entre todos relacionadas à V&V. O objetivo de testar um produto de software é encontrar defeitos no código-fonte do mesmo. Inúmeras teorias, metodologias, abordagens têm sido propostas e/ou usadas para as diversas atividades do processo de Teste de Software. Uma das atividades do processo de Teste mais estudada, mas que ainda apresenta diversos desafios, é a geração de casos/dados de teste. No fundo, dado que a execução de teste exaustivo não é viável, a ideia é utilizar de formas para selecionar, de infinitas possibilidades, um conjunto de dados de entrada de teste do domínio de entrada de um programa P, de forma a detectar o maior número possível de defeitos. Existem diversas abordagens para esse propósito, tais como teste aleatório [Anand et al. 2013], teste de interação combinatória [Balera e Santiago Júnior 2016][Balera e Santiago Júnior 2017], teste baseado em modelos [Utting e Legeard 2007][Santiago Júnior e Vijaykumar 2012][Santiago Júnior e Silva 2017], teste metamórfico [Chen et al. 2009][Troya et al. 2018], entre outras.

A atividade de avaliação de resultados de teste é também conhecida como o problema do oráculo. É outra atividade que tem chamado bastante atenção da comunidade de Teste de Software. Um oráculo de teste é portanto um mecanismo para dizer se um determinado caso de teste passou ou falhou. No caso de falha, pode-se inferir que existe(m) defeito(s) no código-fonte. Detalhadamente, um oráculo de teste gera os resultados esperados de um determinado conjunto de dados de entrada de teste, e compara se os resultados reais (produzidos pelo Software Sob Teste (SST)) são iguais aos resultados esperados.

O problema é que existem programas que são considerados não testáveis [Weyuker 1982]. Um programa não testável é aquele onde uma das duas condições a seguir são satisfeitas: i) não existe um oráculo disponível; ii) é teoricamente possível, mas na prática muito difícil, determinar qual seria a saída correta para uma certa entrada. Software científico se

enquadra perfeitamente nessa categoria onde, por exemplo, a implementação de um modelo baseado em um conjunto de equações matemáticas (diferenciais, …) pode perfeitamente ter como resultado uma saída não determinística para uma certo conjunto de dados de entrada de teste.

Uma das técnicas de teste interessantes para amenizar o problema do oráculo, quando não é possível determinar com certeza os resultados esperados de um conjunto de dados de entrada de teste, é o teste metamórfico [Chen et al. 2009][Troya et al. 2018]. Em vez de verificar a saída de uma execução de programa individual, o teste metamórfico verifica se várias execuções do SST preenchem certas propriedades chamadas de Relações Metamórficas (RMs). Porém, a principal limitação do teste metamórfico é a identificação das RMs, que é uma tarefa manual que requer um bom conhecimento do problema em questão. Portanto, uma das alternativas é tentar identificar, automaticamente, as RMs de um SST, onde muito poucos trabalhos têm sido publicados nesse sentido [Kanewala e Bieman 2013][Troya et al. 2018]. No entanto, até onde se sabe, nenhum dos trabalhos fez uso de Aprendizado Profundo [Guo et al. 2016] para automaticamente identificar as Relações Metamórficas. Aprendizado Profundo é uma subárea do Aprendizado de Máquina que tenta inferir conhecimento, com altos níveis de abstração, a partir de dados e têm sido bastante usado em aplicações clássicas da Inteligência Artificial tais como Visão Computacional e Processamento em Linguagem Natural. Portanto, é interessante investigar a viabilidade de usar Aprendizado Profundo para automaticamente identificar as RMs.

Entre os diversos softwares científicos que o INPE vêm trabalhando, é interessante mencionar aqueles que tentam definir as características hidrológicas, como a quantidade de água precipitada. Isso é de suma importância para o desenvolvimento sócio-econômico do país. A agricultura, defesa civil, aviação e geração de energia são áreas essenciais que precisam de tal informação em tempo quase real para definir suas estratégias e decisões a curto, médio, e longo prazo. No entanto, no Brasil existe uma baixa densidade de pluviômetros e sensores ativos como radares meteorológicos para definir precisamente a precipitação. De modo a suprir essa lacuna, técnicas físico-empíricas foram desenvolvidas ao longo dos anos para recuperar essa informação a partir do uso de dados de satélites meteorológicos geoestacionários, pois tais satélites fornecem imagens com alta resolução espacial e temporal do topo das nuvens sobre toda a América do Sul.

Entre os modelos criados e que o INPE vêm trabalhando está o Hydro-Estimator, que é uma versão empiricamente melhorada do Auto-Estimator, desenvolvido por [Vicente et al. 1998] na National Environmental Satellite, Data, and Information Service (NESDIS) dos EUA. Esse é um algoritmo que produz estimativas instantâneas de precipitação de modo automático a cada passo de tempo do satélite geoestacionário GOES, a partir do uso de imagens no canal

infravermelho. Tal técnica parte da suposição de que as nuvens de topo mais frio e espessas proporcionam maiores taxas de chuva que aquelas com topo mais quente.

Outro algoritmo que é vastamente utilizado para determinar as características das nuvens é o chamado Classificador de Nuvens [Bottino e Ceballos 2015]. Esse algoritmo é um método para identificação de diferentes tipos de nuvens a partir do uso de diversos canais, também do satélite GOES, principalmente as bandas espectrais do visível e infravermelho. Essa metodologia utiliza a análise de agrupamento de 30 classes de nuvens que são determinadas por quatro variáveis definidas pelas medidas do satélite: refletância do canal visível, temperatura de brilho do infravermelho, e as texturas de ambos os canais. Tais variáveis definem padrões sobre um conjunto de pixels em uma imagem de satélite, a mesma imagem que é utilizada como entrada no algoritmo do Hydro-Estimator. A partir da medida da distância euclidiana em análise de agrupamento é possível definir a pertinência de um pixel a um centroide e, assim, determinar a classificação do tipo de nuvem e, consequentemente, relacionar a taxa de chuva à nuvem classificada. Lembrando que o intuito inicial do Hydro-Estimator é estimar a chuva de nuvens convectivas e diminuir o erro decorrente das nuvens estratiforme e cirrus adjacente a esses eventos. Sendo assim, o Classificador de Nuvens é um algoritmo complementar para melhorar a definir uma das componentes do balanço hidrológico, a chuva, a partir das estimativas via satélite.

Desde que está se propondo uma nova metodologia, é também muito relevante realizar uma avaliação experimental rigorosa, via experimento controlado ou quasiexperimento [Zannier et al. 2006], para avaliar a viabilidade da metodologia no contexto de teste de software científico. A ausência de estudos de avaliação mais rigorosos na área de Teste de Software é um fato comprovado por recentes publicações, não somente na comunidade brasileira mas também na comunidade internacional de Engenharia de Software [Lemos et al. 2013].

Portanto, os objetivos específicos desse projeto são:

a.) Investigar a viabilidade de teste metamórfico para realizar o teste de software científico, particularmente abordando o problema do oráculo de teste de software. Para identificar automaticamente as RMs será investigada a possibilidade de uso de Aprendizado Profundo (e.g. Redes Neurais Convolucionais);

b.) Aplicar a abordagem proposta a produtos de software relacionados à definição de características hidrológicas, com os quais o INPE vêm trabalhando, com o intuito de melhorar a qualidade dos mesmos.

Esse relatório apresenta as atividades desenvolvidas no período de **01 de agosto de 2018 a 30 de junho de 2019.** Esse é o relatório de renovação do projeto.

2.) CRONOGRAMA DE ATIVIDADES E ETAPAS CONCLUÍDAS

A metodologia a ser empregada para atender aos objetivos do projeto está descrita a seguir, conforme o formulário de solicitação da bolsa do Programa Institucional De Bolsas (INPE-CNPq):

1. Estudar a fundamentação teórica relativa ao projeto. Especificamente, se familiarizar com os conceitos relacionados à Teste de Software (com particular ênfase em teste metamórfico), Aprendizado Profundo, metodologias e produtos de software relacionados à definição de características hidrológicas;

2. Estudar a forma de uso de produtos relacionados à definição de características hidrológicas, com os quais o INPE vêm trabalhando: Hydro-Estimator e Classificador de Nuvens;

3. Propor um método que identifique, automaticamente, as Relações Metamórficas que os produtos de software devem satisfazer. Para isso, existe a possibilidade de fazer uso de Aprendizado Profundo (e.g. Redes Neurais Convolucionais);

4. Integrar o método para identificar as Relações Metamórficas automaticamente em uma metodologia para gerar casos/dados de teste e detectar defeitos nos produtos de software. A metodologia será apoiada por uma ferramenta de software que será desenvolvida;

5. Realizar avaliação experimental rigorosa (experimento controlado ou quasiexperimento) para avaliar a viabilidade da metodologia proposta;

6. Submeter artigo para conferência e/ou workshop e/ou simpósio na área de Engenharia de Software e/ou Meteorologia, e elaborar relatório final de atividades.

O cronograma para desenvolvimento das atividades da metodologia está mostrado na tabela 1 a seguir. O número das atividades está de acordo com os números mostrados acima. Cada uma das colunas de Ano I e II representa um mês.

Atividade	Ano I												Ano II						
	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7
1																			
2																			
3																			
4																			
5																			
6																			

Tabela 1 – Cronograma de atividades

Esse relatório compreende o **mês 1 do Ano I (agosto/2018) ao mês 11 do Ano I (junho/2019)**. Ou seja, é um relatório de renovação do projeto. Considerando as atividades previstas para serem desenvolvidas, mostradas na Tabela 1, a Tabela 2 a seguir mostra o desenvolvimento das atividades do projeto, no que se refere a esse relatório de acompanhamento.

Tabela 2 – Etapas Concluídas e a Concluir

	Atividades da Metodologia	Previsão	Realização
1	Estudar a fundamentação teórica relativa ao projeto. Especificamente, se familiarizar com os conceitos relacionados à Teste de Software (com particular ênfase em teste metamórfico), Aprendizado Profundo, metodologias e produtos de software relacionados à definição de características hidrológicas;	100%	100%
2	Estudar a forma de uso de produtos relacionados à definição de características hidrológicas, com os quais o INPE vêm trabalhando: Hydro-Estimator e Classificador de Nuvens;	100%	100%
3	Propor um método que identifique, automaticamente, as Relações Metamórficas que os produtos de software devem satisfazer. Para isso, existe a possibilidade de fazer uso de Aprendizado Profundo (e.g. Redes Neurais Convolucionais);	66,7%	25%
4	Integrar o método para identificar as Relações Metamórficas automaticamente em uma metodologia para gerar casos/dados de teste e detectar defeitos nos produtos de software. A metodologia será apoiada por uma ferramenta de software que será desenvolvida;	41,7%	5%
5	Realizar avaliação experimental rigorosa (experimento controlado ou quasiexperimento) para avaliar a viabilidade da metodologia proposta;	0%	0%
6	Submeter artigo para conferência e/ou workshop e/ou simpósio na área de Engenharia de Software e/ou Meteorologia, e elaborar relatório final de atividades.	0%	0%

Na Tabela 2 acima, a coluna **Previsão** mostra a porcentagem prevista para a realização da atividade, e a coluna **Realização** mostra a porcentagem realmente realizada da atividade, considerando o período a que se refere esse relatório de acompanhamento.

Comentários gerais sobre o desenvolvimento das atividades são dados a seguir e detalhados mais adiante nesse relatório.

Na atividade 1, foram estudados os fundamentos teóricos relacionados ao projeto, os conceitos relacionados à Teste de Software, Teste Metamórfico e também à hidrologia. Foram estudados artigos sobre produtos meteorológicos, como o Auto-Estimator (similar ao Hydro-Estimator) e o Classificador de Nuvens (Bottino, Ceballos 2014). Essa atividade foi, portanto, 100 % concluída.

Na atividade 2, o estudo foi focado em dois produtos utilizados pelo INPE, o Hydro-Estimator, entendendo o produto como um todo e o executando para se familiarizar com o mesmo, e o Classificador de Nuvens. Portanto, essa atividade foi também 100% concluída.

A atividade 3 é a mais desafiadora do projeto, pois é necessário propor um método para, automaticamente, identificar as RMs que os produtos de software Hydro-Estimator e Classificador de Nuvens devem satisfazer. Esse método está sendo feito com aprendizado profundo, utilizando Redes Neurais Convolucionais (CNNs), criando uma maneira de testar esses sofwares automaticamente. Até o momento, foram feitos estudos aprofundados sobre a arquitetura e o funcionamento das CNNs, além da implementação de uma rede neural simples de reconhecimento de imagens.

A atividade 4 não foi muito desenvolvida no período a que se refere esse relatório devido a complexidade para entendimento das CNNs (atividade 3), dado que esse é um assunto relativamente novo, se comparado à outras abordagens de Machine Learning.

As atividades 5 e 6 não estavam previstas para serem desenvolvidas no período a que se refere esse relatório de acompanhamento.

Os detalhamentos do desenvolvimento das atividades estão apresentados a seguir.

3.) ATIVIDADE 1: FUNDAMENTAÇÃO TEÓRICA

A fundamentação teórica deste projeto foi dividida em duas partes: o estudo da parte dos softwares utilizados pelo INPE (Auto-Estimator e o Classificador de Nuvens, detalhados na **seção 4**) e também sobre testes metamórficos, o problema do oráculo e as aplicações de machine learning nesse tipo de teste.

3.1)Testes Metamórficos

O Teste Metamórfico (MT) é uma técnica de testes de software que pode ser uma abordagem eficaz para resolver o problema do Oráculo e o problema de geração de casos de teste. O problema do Oráculo é a dificuldade de determinar os resultados esperados de casos de teste selecionados ou para determinar se os resultados reais concordam com os resultados esperados.

Relações Metamórficas (MRs) são propriedades necessárias da funcionalidade pretendida do software e devem envolver múltiplas execuções do software. Por exemplo, um programa que implementa *sen(*x*)* correto para 100 algarismos significativos; uma relação metamórfica para funções senoidais é "*sen(*π - x*) = sen(*x*)*". Portanto, mesmo que o valor esperado de *sen(*x_1*)* para o caso de teste de origem x_1 = 1.234 correto para a precisão requerida não seja conhecido, um caso de teste de acompanhamento x_2 = π - 1.234 pode ser construído. É possível verificar se as saídas reais produzidas pelo programa em teste a partir do caso de teste de origem e o caso de teste de acompanhamento são consistentes com o MR em questão. Qualquer inconsistência (após considerar os erros de arredondamento) indica uma falha na implementação.

O teste metamórfico foi reportado por Chen (1998), desde então, mais de 150 pesquisadores e profissionais internacionais aplicaram a técnica a aplicações da vida real. Alguns exemplos incluem serviços web, computação gráfica, sistemas embarcados, simulação e modelagem, machine learning, suporte à decisão, bioinformática, componentes, análise numérica e compiladores.

Embora o MT tenha sido inicialmente proposto como uma técnica de verificação de software, mais tarde foi desenvolvido em um paradigma que abrange a verificação, validação e outros tipos de avaliação de qualidade de software. O MT pode ser aplicado independentemente e também pode ser combinado com outras técnicas de análise de software estáticas e dinâmicas, como a prova e depuração .

4.) ATIVIDADE 2: ANÁLISE DO HYRO-ESTIMATOR E O CLASSIFICADOR DE NUVENS

Esta atividade é dividida no estudo de dois produtos usados pelo INPE para a área da meteorologia. Até o momento, só foi possível o estudo do Hydro-Estimator devido a sua complexidade e a dificuldade de encontrarmos as relações metamórficas e como testar o programa, deixando a pesquisa do código do Classificador de Nuvens para um momento posterior.

Abaixo, as atividades relativas ao Hydro-Estimator serão descritas.

4.1) O Hydro-Estimator

A ferramenta chamada Hydro-Estimator tem seu código-fonte baseado no Auto-Estimator apresentado em Vicente *et al* (1998). Com base nessa publicação apresentada, iremos fazer uma breve revisão sobre o "auto-estimador". A ferramenta utiliza um algoritmo de regressão não-linear de lei de potência para calcular precipitação em tempo real baseadas em imagens na banda infra-vermelho de 10.7μm do satélite geoestacionário GOES. Além das imagens de satélite também são utilizados dados de modelo para estimativa de umidade e precipitação. Essa técnica é muito útil para

monitoramento das chamas enchentes relâmpago devido a obtenção de novas imagens a cada 15 minutos e também para monitorar grandes sistemas de tempestades, quando inúmeras células de chuva precisam ser monitoradas simultaneamente, o que é possível graças a grande abrangência da área observada pelas imagens do satélite.

A técnica empregada pelo Auto-Estimador pode ser dividida em cinco partes:

1.) Taxa de chuva pela temperatura de topo de nuvem: Considerando apenas sistemas convectivos, as taxas de precipitação são inicializadas usando uma curva de ajuste de lei de potência entre estimativas instantâneas de precipitação coletadas de dados de radar pelas medições por satélite de temperaturas de brilho infravermelho nos topos de nuvem. Essa curva dá a relação entre temperaturas mais baixas no topo das nuvens e as maiores médias de precipitação naquele local.

2.)Fator de Correção de Umidade: A curva baseada nas temperaturas de topo de nuvem por si só não e suficiente para uma estimativa muito realista, sendo comum ter uma super-estimativa de chuva para regiões secas por exemplo. Logo, é proposto uma camada com dados de chuva esperada e umidade relativa como ajuste à curva de regressão.

3.)Fator de Correção de taxa de crescimento da nuvem: Chuvas são mais intensas em nuvens que tem seu topo se esfriando e quase nulas quando o topo está se esquentando. Com base em duas imagens infra-vermelhas consecutivas, é possível detectar os pixeis que têm temperatura inferior na segunda imagem e classificá-los como uma tempestade em crescimento e associar o valor 1 para esse pixel, enquanto um pixel que sofreu aquecimento na segunda imagem é dito como uma tempestade enfraquecida e é associado ao valor 0. Assim é feito uma máscara de chuva/não-chuva com a progressão da temperatura de topo.

4.)Correção do gradiente de temperatura no topo da nuvem: Na ausência de imagens consecutivas por um período maior que 0.5h, é calculado um gradiente da temperatura do topo de nuvem a partir de uma única imagem por meio de um recorte de 3x3 pixeis ou 5x5 pixeis, de forma a conseguir uma máscara binária de precipitação.

5.)Cálculo da Taxa de Precipitação: É calculado o acumulado de chuva em um período por meio da média das precipitações estimadas em três imagens consecutivas, seja por estimação por taxa de crescimento ou por correção de gradiente.

5.) ATIVIDADES 3: ESTUDO DE UMA REDE NEURAL CONVOLUCIONAL E A CRIAÇÃO DO MÉTODO DE TESTES

De forma simplificada, uma rede neural convolucional (CNN) é uma classe de redes neurais profundas, mais comumente aplicada à análise de imagens visuais. Nesta atividade o foco é em como as CNN funcionam e propor um método de testes automatizado.

5.1) Design de uma CNN.

As CNNs usam uma variação de perceptrons multi-camadas projetados para requerer um pré-processamento mínimo. Eles também são conhecidos como redes neurais artificiais invariantes a mudanças ou invariantes no espaço (SIANN), com base em suas características de invariância de arquitetura e arquitetura de pesos compartilhados.

Elas usam relativamente pouco pré-processamento em comparação com outros algoritmos de classificação de imagem. Isso significa que a rede aprende os filtros que, nos algoritmos tradicionais, foram manipulados à mão. Essa independência do conhecimento prévio e do esforço humano no design de recursos é uma grande vantagem.

Uma rede neural convolucional consiste em uma camada de entrada e uma de saída, além de várias camadas ocultas. As camadas ocultas de uma CNN consistem tipicamente em camadas convolucionais, camada RELU, isto é, função de activação, camadas de agrupamento, camadas completamente ligadas e camadas de normalização.

A descrição do processo como uma convolução em redes neurais é por convenção. Matematicamente, é uma correlação cruzada em vez de uma convolução. Isso só tem significado para os índices na matriz e, portanto, quais pesos são colocados em qual índice.

Camadas convolucionais aplicam uma operação de convolução à entrada, passando o resultado para a próxima camada. A convolução emula a resposta de um neurônio individual a estímulos visuais.

Cada neurônio convolucional processa os dados apenas por seu campo receptivo. Embora redes neurais totalmente conectadas possam ser usadas para aprender recursos e classificar dados, não é prático aplicar essa arquitetura a imagens. Um número muito alto de neurônios seria necessário, mesmo em uma arquitetura superficial (oposta à profundidade), devido aos tamanhos de entrada muito grandes associados às imagens, em que cada pixel é uma variável relevante.

As CNNs podem incluir camadas de agrupamento locais ou globais, que combinam as saídas de agrupamentos de neurônios em uma camada em um único neurônio na camada seguinte.

As camadas totalmente conectadas conectam todos os neurônios em uma camada a todos os neurônios em outra camada. Em princípio, é o mesmo que a rede neural perceptron multicamada tradicional (MLP).

Nas redes neurais, cada neurônio recebe informações de alguns locais na camada anterior. Em uma camada totalmente conectada, cada neurônio recebe entrada de cada elemento da camada anterior. Em uma camada convolucional, os neurônios recebem entrada de apenas uma subárea restrita da camada anterior. Tipicamente, a subzona é de uma forma quadrada (por exemplo, tamanho 5 por 5). A área de entrada de um neurônio é chamada de campo receptivo. Então, em uma camada totalmente conectada, o campo receptivo é a camada anterior inteira. Em uma camada convolucional, a área receptiva é menor que a camada anterior inteira.

Cada neurônio em uma rede neural calcula um valor de saída aplicando alguma função aos valores de entrada vindos do campo receptivo na camada anterior. A função que é aplicada aos valores de entrada é especificada por um vetor de pesos e um viés (geralmente números reais). O aprendizado em uma rede neural progride, fazendo ajustes incrementais nos vieses e pesos. O vetor de pesos e o viés são chamados de filtro e representam alguma característica da entrada (por exemplo, uma forma particular). Uma característica distintiva das CNNs é que muitos neurônios compartilham o mesmo filtro. Isso reduz o consumo de memória porque um único viés e um único vetor de pesos são usados em todos os campos receptivos que compartilham esse filtro, em vez de cada campo receptivo ter seu próprio viés e vetor de pesos.

6. ATIVIDADES 4, 5 e 6:

A atividade 4 está extremamente ligada à atividade 3, pois é necessário fazer um estudo sobre redes neurais e seu funcionamento para então criar o método que irá identificar as Relações Metamórficas automaticamente em uma metodologia para gerar casos/dados de teste e detectar defeitos nos produtos de software, integrar esse método é o maior desafio dessa pesquisa.

A atividade 5 é dependente da atividade 4, pois o método proposto precisa estar pronto para ser testado, tendo início somente no mês 4 do Ano II.

Na última atividade esperamos submeter artigo para conferência e/ou workshop e/ou simpósio na área de Engenharia de Software e/ou Meteorologia, e elaborar relatório final de atividades.

7. CONCLUSÃO

Esse relatório apresentou as atividades desenvolvidas, no período de 01 de agosto de 2018 a 31 de janeiro de 2019, relacionadas ao projeto “Teste Metamórfico para Produtos de Software relacionados à Definição de Características Hidrológicas”. Pode-se concluir que as atividades realizadas, nesse período a que se refere esse relatório, foram tais que os objetivos, para esse período, foram alcançados de forma satisfatória, embora parcial.

8. REFERÊNCIAS

[Kanewala e Bieman 2014] KANEWALA, U.; BIEMAN, J. M. Testing scientific software: A systematic literature review, Information and Software Technology, Volume 56, Issue 10, 2014, Pages 1219-1232, ISSN 0950-5849,

https://doi.org/10.1016/j.infsof.2014.05.006.

[Pipitone e Easterbrook 2012] PIPITONE, J.; EASTERBROOK, S.: Assessing climate model software quality: a defect density analysis of three models, Geosci. Model Dev., 5, 1009-1022, https://doi.org/10.5194/gmd-5-1009-2012, 2012.

[Miller 2006] MILLER, G. A scientist’s nightmare: software problem leads to five retractions, Science 314 (5807) (2006) 1856–1857.

[Hatton e Roberts 1994] HATTON, L.; ROBERTS, A. 1994. How Accurate is Scientific Software?. IEEE Trans. Softw. Eng. 20, 10 (October 1994), 785-797. DOI=http://dx.doi.org/10.1109/32.328993

[Dubois 2012] DUBOIS, P. Testing scientific programs, Comput. Sci. Eng. 14 (4) (2012), 69–73.

[Anand et al. 2013] ANAND, S.; BURKE, E. K.; CHEN, T. Y.; CLARK, J.; COHEN, M. B.; GRIESKAMP, W.; HARMAN, M.; HARROLD, M. J.; McMINN, P.; BERTOLINO, A.; LI, J. J.; ZHU, H. An orchestrated survey of methodologies for automated software test case generation, The Journal of Systems and Software, Volume 86, Issue 8, 2013, p. 1978-2001.

[Balera e Santiago Júnior 2016] BALERA, J. M.; SANTIAGO JÚNIOR, V. A. 2016. A Controlled Experiment for Combinatorial Testing. In Proceedings of the 1st Brazilian Symposium on Systematic and Automated Software Testing (SAST). ACM, New York, NY, USA, Article 2, 10 pages. https://doi.org/10.1145/2993288.2993289

[Balera e Santiago Júnior 2017] BALERA, J. M.; SANTIAGO JÚNIOR, V. A. 2017. An algorithm for combinatorial

interaction testing: definitions and rigorous evaluations. Journal of Software Engineering Research and Development 5, 1 (28 Dec 2017), 41. https://doi.org/10.1186/s40411-017-0043-z

[Utting e Legeard 2007] UTTING, M.; LEGEARD, B. Practical Model-Based Testing: A tools Approach. Waltham, MA, USA: Morgan Kaufmann Publishers, 2007.

[Santiago Júnior e Vijaykumar 2012] SANTIAGO JÚNIOR, V. A.; VIJAYKUMAR, N. L. Generating model-based test cases from natural language requirements for space application software. Software Quality Journal, v. 20, n. 1, p. 77-143, 2012. DOI: 10.1007/s11219-011-9155-6.

[Santiago Júnior e Silva 2017] SANTIAGO JÚNIOR, V. A.; SILVA, F. E. C. 2017. From Statecharts into Model

Checking: A Hierarchy-based Translation and Specification Patterns Properties to Generate Test Cases. In Proceedings of the 2nd Brazilian Symposium on Systematic and Automated Software Testing (SAST). ACM, New York, NY, USA, Article 2, 10 pages. https://doi.org/10.1145/3128473.3128475

[Chen et al. 2009] CHEN, T. Y.; HO, J. W. K.; LIU, H.; XIE, X. 2009. An innovative approach for testing bioin- formatics programs using metamorphic testing. BMC Bioinformatics, 10 (1), 24. doi: 10.1186/1471-2105- 10- 24 .

[Troya et al. 2018] TROYA, J.; SEGURA, S.; RUIZ-CORTÉS, A. Automated inference of likely metamorphic relations for model transformations, Journal of Systems and Software, Volume 136, 2018, Pages 188-208, ISSN 0164-1212, https://doi.org/10.1016/j.jss.2017.05.043.

[Weyuker 1982] WEYUKER, E .J. 1982. On testing non-testable programs. The Computer Journal, 25(4), 465–470.

[Kanewala e Bieman 2013] KANEWALA, U.; BIEMAN, J. M. Using machine learning techniques to detect metamorphic relations for programs without test oracles, 2013 IEEE 24th International Symposium on Software Reliability Engineering (ISSRE), Pasadena, CA, 2013, pp. 1-10. doi: 10.1109/ISSRE.2013.6698899

[Zannier et al. 2006] ZANNIER, C.; MELNIK, G.; MAURER, F. On the success of empirical studies in the international conference on software engineering. In Proceedings of the 28th International

Conference on Software Engineering, ICSE '06, p. 341-350, New York, NY, USA, 2006. ACM.

[Lemos et al. 2013] LEMOS, O. A. L.; FERRARI, F. C.; ELER, M. M.; MALDONADO, J. C.; MASIERO, P. C. Evaluation studies of software testing research in Brazil and in the World: A survey of two premier software engineering conferences. J. Syst. Softw., 86(4):951-969, April 2013.

[Guo et al. 2016] GUO, Y.; LIU, Y.; OERLEMANS, A.; LAO, S.; WU, S.; LEW, M. S. Deep learning for visual understanding: A review, Neurocomputing, Volume 187, 2016, Pages 27-48, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2015.09.116.

[Vicente et al. 1998] VICENTE, G. A.; SCOFIELD, R. A.; MENZEL, W. P. The operational GOES infrared rainfall estimation technique. Bulletin of the American Meteorological Society, v. 79, n. 9, pp. 1883-1898, 1998.

[Bottino e Ceballos 2015] Bottino, M. J.; Ceballos, J. C. Daytime cloud classification over South American region using multispectral GOES-8 imagery. International Journal of Remote Sensing, 36(1), 1-19, 2015.