Cloud computing para *deploy* de modelos de *deep learning* para a classificação de Retinopatia Diabética *

Matheus W. Camargo[1], **Cristiano A. Künas**[1], **Philippe O. A. Navaux**[1]

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre, RS – Brasil

{mwcamargo, cakunas, navaux}@inf.ufrgs.br

Resumo. *A retinopatia diabética (RD) é uma doença que vem crescendo a ritmos alarmantes. A falta de mão de obra especializada para diagnóstico, essencial para o tratamento bem-sucedido da doença, traz a necessidade de estudo de alternativas para o diagnóstico via meios computacionais. Neste trabalho, avaliamos o desempenho e custo de alternativas para o deploy de modelos de Deep Learning para classificação de RD. Através da escolha da melhor arquitetura, foi possível melhorar o desempenho em até 1,71 vezes, com redução de custo de 4,17%.*

1. Introdução e motivação

A diabetes no mundo vem crescendo em ritmos alarmantes. Conforme Federation (2021), em 2021 estima-se que cerca de 537 milhões de pessoas sofreram com essa doença mundialmente, das quais é esperado que 1/3 dessas pessoas possuam retinopatia diabética (RD). A RD é uma complicação consequente da diabetes, no qual o alto nível de açúcar no sangue causa danos à retina, prejudicando a visão e podendo em seu nível mais grave levar até mesmo à cegueira.

Muito embora a realização de exames preventivos seja reconhecida mundialmente como a principal forma de combate à doença, essa prática ainda não é amplamente difundida, por consequência da escassez de profissionais especializados capazes de realizar a avaliação das imagens de fundo de olho. Na China, a razão entre pacientes com diabetes e oftalmologistas chega a 1:3000 [Dai et al. 2021].

Frente a esse cenário, um método muito promissor para a classificação de RD é o uso de *Deep Learning*. Nesse método, redes neurais são treinadas de maneira supervisionada para receber imagens de fundo de olho e realizarem a classificação da imagem em diferentes estágios de retinopatia diabética, conforme realizado em Moreira et al. (2020) e Voets et al. (2019). Tal tecnologia, se disponibilizada para profissionais da saúde, poderia ser utilizada para priorizar os casos mais graves, permitindo uma utilização mais eficiente de mão de obra especializada.

Partindo desta conjuntura, o presente trabalho se propõe a avaliar o uso de diferentes arquiteturas para o *deploy* de dois modelos de *deep learning*, treinados para a detecção de Retinopatia Diabética, na *cloud*. Serão avaliados tanto custo como desempenho de diferentes infraestruturas utilizadas para a fase de inferência dos modelos.

2. Trabalhos Relacionados

O aprendizado de máquina vem ganhando destaque nas últimas décadas, mostrando-se cada vez mais promissor para tarefas de classificação, como o reconhecimento de objetos ou o di-

*O presente trabalho foi apoiado pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001, pelo edital CNPq/MCTI/FNDCT - Universal 18/2021 sob número 406182/2021-3 e pelos projetos CIARS RITEs/FAPERGS e CI-IA FAPESP-MCTIC-CGI-BR.

agnóstico de doenças. Conforme mostrado em Dai et al. (2021), a partir do sistema *DeepDR* é possível detectar a presença de diferentes graus de diabetes com área sob a curva de característica operacional do receptor de 0,916 até 0,970 para cada grau de RD.

A fim de proporcionar uma maior acessibilidade desses modelos para o usuário final, é imprescindível pensar em formas de provisionar os recursos de uma maneira performática e de baixo custo. Nesse contexto, o uso de *cloud computing* para o provisionamento da infraestrutura de um sistema de análise de imagens médicas já foi estudado em [Ivanova et al. (2018). Neste trabalho, foi utilizado uma plataforma baseada na provedora de nuvem AWS, em conjunto da linguagem *terraform* para descrever a infraestrutura seguindo os princípios de IaC (*Infrastructure as Code*). Neste contexto, o presente trabalho visa explorar o uso da plataforma *Amazon Sagemaker* para o *deploy* de modelos de *deep learning* capazes de classificar RD, avaliando o custo e desempenho do serviço *real-time inference*.

3. Implementação e resultados

Primeiramente, foi necessário treinar e selecionar modelos para a classificação de retinopatia diabética. Utilizou-se a plataforma Kaggle, usando como *dataset* um subconjunto do *dataset* original da competição APTOS 2019[1]. De modo a verificar o comportamento de modelos de diferentes tamanhos frente a diferentes arquiteturas para a inferência, escolheram-se os modelos **VGG19** e **MobileNet**. A relação entre os modelos, tamanho de artefato em disco e número de parâmetros é apresentado na tabela 1.

Modelo	Número de parâmetros	Tamanho do artefato (SavedModel)
MobileNet	3.377.221	15,7 MB
VGG19	20.107.205	77,7 MB

Tabela 1. Relação entre modelo utilizado no trabalho e número de parâmetros e tamanho do artefato de modelo (SavedModel) armazenado em disco.

Em seguida, os modelos e o código necessário para a realização da inferência foi definido em uma imagem Docker, seguindo a interface definida em Hudgeon and Nichol (2020), necessária para a utilização do serviço *Amazon Sagemaker real-time inference*. O Amazon Sagemaker é uma plataforma para a construção, treinamento e deploy de modelos de aprendizado de máquina. Dentre os diversos serviços ofertados, o serviço *real-time inference*, utilizado neste trabalho, permite o *deploy* de modelos de aprendizado de máquina utilizando *containers* em diversos tipos de instâncias distintos.

A fim de avaliar diferentes arquiteturas para a realização da inferência, utilizaram-se diferentes tipos de máquina, sumarizadas na tabela 2. Ao todo, foram avaliadas quatro máquinas, tendo duas a *ISA* x86-64 e outras duas a *ISA* ARMv8. Para avaliar o desempenho das arquiteturas utilizadas neste trabalho, realizou-se um experimento composto de 25 requisições em sequência, com 5 segundos de intervalo entre cada requisição. Essa configuração foi escolhida para avaliar o tempo de predição dos modelos nas respectivas arquiteturas em uma única execução, sem concorrência de requisições em um mesmo *container*. Deste experimento foi extraído o tempo de predição de cada modelo, como exibido nas Figuras 1 e 2.

Conforme mostrado nas Figuras 1 e 2, há uma diferença considerável de desempenho entre diferentes tipos de máquina, mesmo que todas as instâncias tenham a mesma quantidade

[1]https://kaggle.com/competitions/aptos2019-blindness-detection

de vCPUs, como mostrado na tabela 2. Considerando o tipo de instância que apresenta o melhor desempenho, `m5.xlarge`, com a máquina de pior desempenho, `m6g.xlarge`, houve um *speed-up* de $1,92\times$ para o modelo **VGG19** e um *speed-up* de $1,42\times$ para o modelo **MobileNet**.

Figura 1. Tempo de predição para o modelo VGG19.

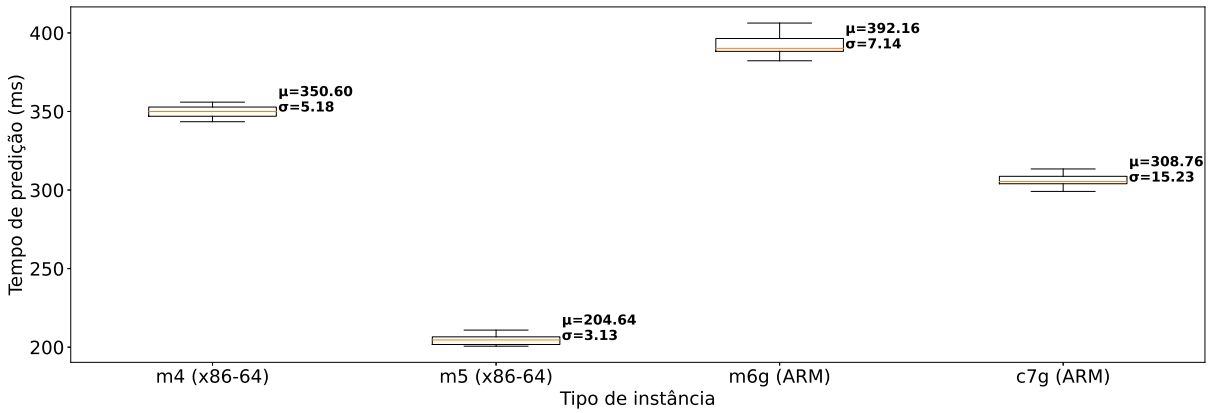
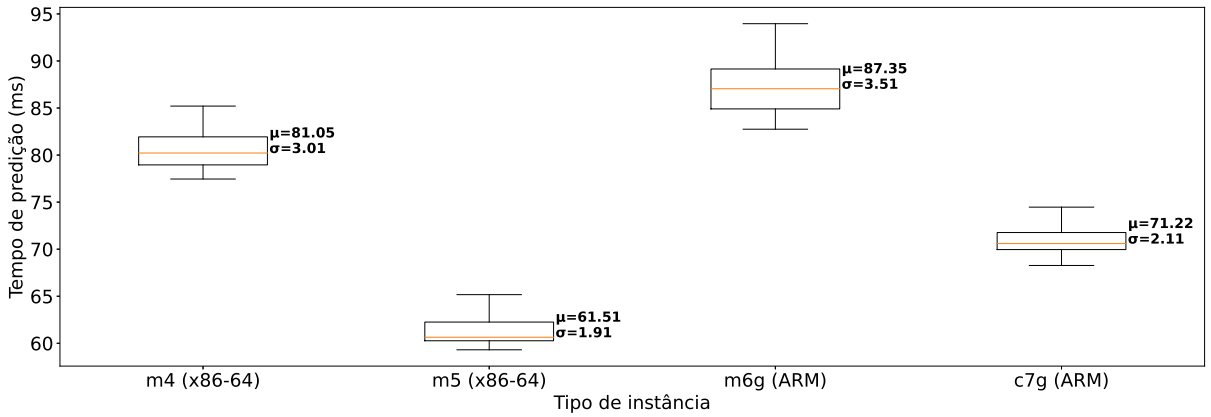


Figura 2. Tempo de predição para o modelo MobileNet.



Nota-se que dentro de cada arquitetura as máquinas de geração mais atual apresentaram o melhor desempenho. Para o caso da arquitetura x86, comparando a instância `m6` frente à instância `m5`, houve um *speed-up* de $1,71\times$ para o modelo **VGG19** e $1,32\times$ para o modelo **MobileNet**. De maneira similar para a arquitetura ARM, quando comparamos uma instância `c7g` frente a uma instância `m6g`, há um *speed-up* de $1,27\times$ para o modelo **VGG19** e $1,23\times$ para o modelo **MobileNet**.

Quando se considera o custo de cada tipo de instância, expresso na tabela 2, percebe-se que, em geral, é vantajoso utilizar a versão mais recente de cada arquitetura, já que há tanto ganhos de desempenho como também redução de custo. Para o caso da arquitetura x86 existe uma redução de custo de $4,17\%$ ao utilizar a instância mais recente, e para a arquitetura ARM existe uma redução de custo de $5,84\%$.

Embora exista uma redução de desempenho de $33,72\%$ para o modelo VGG19 quando comparamos a utilização da instância `c7g` com a instância `m5`, essa redução diminui para $13,63\%$ quando analisamos o modelo menor, MobileNet. Nesse cenário, também pode ser

Tipo de instância	ISA	Processador	Memória	vCPUs	Custo por hora
`ml.m4.xlarge`	x86-64	Intel Xeon E5-2676	16 GB	4	0,24 USD
`ml.m5.xlarge`	x86-64	Intel Xeon Platinum	16 GB	4	0,23 USD
`ml.m6g.xlarge`	ARMv8.2	Graviton 2	16 GB	4	0,1848 USD
`ml.c7g.xlarge`	ARMv8.4	Graviton 3	8 GB	4	0,174 USD

Tabela 2. Relação de tipos de instância, arquitetura do processador, memória, vCPUs e custo por hora.

interessante a utilização da instância ARM, visto que ao utilizar uma instância `c7g` o custo para inferência é reduzido em $24,34\%$.

4. Conclusões e trabalhos futuros

Neste trabalho foi verificado o impacto em termos de custo e desempenho do uso de diferentes arquiteturas para *deployment* na *cloud* de modelos para classificação de retinopatia diabética. Apresentou-se um primeiro passo no sentido de explorar a relação entre tamanho de modelos de redes neurais e o custo e desempenho de alternativas para a execução da inferência de tais modelos. Os próximos passos consistem em estudar o impacto de técnicas de quantização de modelos de rede neural no tempo de predição e custo das arquiteturas utilizadas neste trabalho, avaliando a relação entre tamanho de modelo e ganhos de desempenho e custo. Outro passo importante a ser realizado é o *profiling* da aplicação, com o objetivo compreender as razões por trás da diferença de desempenho observada entre as arquiteturas avaliadas.

Referências

Dai, L., Wu, L., Li, H., Cai, C., Wu, Q., Kong, H., Liu, R., Wang, X., Hou, X., Liu, Y., et al. (2021). A deep learning system for detecting diabetic retinopathy across the disease spectrum. *Nature communications*, 12(1):1–11.

Federation, I. D. (2021). *IDF Diabetes Atlas*. International Diabetes Federation, Brussels, Belgium.

Hudgeon, D. and Nichol, R. (2020). Amazon sagemaker: Use your own inference code with hosting services. Acesso em: 10 de mar de 2023.

Ivanova, D., Borovska, P., and Zahov, S. (2018). Development of paas using aws and terraform for medical imaging analytics. *AIP Conference Proceedings*, 2048(1):060018.

Moreira, F., Schaan, B., Schneiders, J., Reis, M., Serpa, M., and Navaux, P. (2020). Impacto da resolução na detecção de retinopatia diabética com uso de deep learning. In *Anais do XX Simpósio Brasileiro de Computação Aplicada à Saúde*, pages 494–499, Porto Alegre, RS, Brasil. SBC.

Voets, M., Møllersen, K., and Bongo, L. A. (2019). Reproduction study using public data of: Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *PloS one*, 14(6):e0217541.