

Redes convolucionais para segmentação semântica

Convolutional networks for semantic segmentation

RESUMO

Paulo Sérgio Avila Junior
pauloj.2018@alunos.utfpr.edu.br
Universidade Tecnológica Federal do Paraná,
Curitiba, Paraná, Brasil

Myriam Regattieri De Biase da Silva Delgado
myriamdelg@utfpr.edu.br
Universidade Tecnológica Federal do Paraná,
Curitiba, Paraná, Brasil

Rodrigo Minetto
rminetto@dainf.ct.utfpr.edu.br
Universidade Tecnológica Federal do Paraná,
Curitiba, Paraná, Brasil

O aprendizado profundo fornece atualmente uma estrutura poderosa para o aprendizado supervisionado. Ao adicionar mais camadas e mais unidades dentro de uma camada, uma rede profunda pode representar funções complexas. Com isso é possível solucionar problemas desafiadores como é o caso da segmentação semântica que pode ser entendida como a tarefa de agrupar partes de imagens que pertencem à mesma classe de objeto. Neste trabalho utilizou-se um conjunto de redes U-net para segmentação semântica, tendo por base o conceito da Hydra para colaborações de modelos individuais. A Hydra foi originalmente desenvolvida para classificação de imagens e seu uso no contexto de segmentação de imagens é inovador. O trabalho foi desenvolvido inteiramente em ambientes de computação em nuvem e programado em Python, usando Keras com backend de Tensorflow. Como estudo de caso utilizou-se uma competição do Kaggle que envolve imagem de células de diferentes tipos. Nos experimentos realizados, foram obtidos resultados que permitem concluir que a Hydra é um ensemble poderoso e com potencial para ser aplicada também em segmentação.

PALAVRAS-CHAVE: Redes neurais (Computação). Aprendizado do computador. Computação em nuvem. Computação gráfica.

Recebido: 04 set. 2020.

Direito autorial: Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.



ABSTRACT

Deep learning currently provides a powerful framework for supervised learning. By adding more layers and more units within a layer, a deep network can represent complex functions. Solving some challenging problems is now possible, such as semantic segmentation, which can be understood as the task of grouping parts of images that belong to the same object class. In this work, an ensemble of U-net networks is used for semantic segmentation, based on the Hydra concept for collaborations of individual models. Hydra was originally developed for image classification and its use in the context of image segmentation is innovative. The work was developed entirely in cloud computing environments and programmed in Python, using Keras with Tensorflow backend. As a case study, we used a Kaggle competition that involves imaging cells of different types. In the experiments, results allow us to conclude that Hydra is a powerful ensemble and presents a potential to be applied also in segmentation.

KEYWORDS: Neural networks. Machine learning. Cloud computing. Computer graphics.



INTRODUÇÃO

Aprendizado Profundo (AP) é um tipo de aprendizado de máquina, em geral associado a modelos de Redes Neurais Artificiais (RNAs) profundas, ou seja, RNAs formadas por uma quantidade maior de camadas de neurônios que são adicionadas para se obter diferentes níveis de abstração. O AP foi viabilizado nos últimos anos pelos avanços na tecnologia de placas de vídeo, possibilitando a aplicação de modelos profundos em diversos problemas que até então não haviam sido solucionados de forma adequada por modelos rasos (DARGAN et al., 2019). Alguns exemplos envolvem reconhecimento de fala (Hinton et al., 2012), detecção de pedestres (OUYANG; WANG, 2013), reconhecimento facial (DEEP..., 2019) e detecção de objetos (ZHAO et al., 2019). Apesar do sucesso no contexto de classificação, recentemente modelos profundos têm sido usados também em tarefas de segmentação semântica de imagens com resultados estado-da-arte (GIRSHICK et al., 2014; LAI, 2015).

As redes profundas apresentam inúmeras vantagens em termos de aplicabilidade. No entanto, possuem a desvantagem de necessitar de uma quantidade maior de amostras de treinamento do que os modelos rasos (LECUN; BENGIO; HINTON, 2015). A U-net é um exemplo de rede convolucional voltada para segmentação e capaz de operar em situações onde poucas imagens de treino estão disponíveis. O seu funcionamento considera camadas dispostas sequencialmente mas com objetivos diferentes: (i) a primeira parte da sequência de camadas (convolução) reduz o tamanho da matriz de informações à medida que se avança com o fluxo de informação na rede e busca extrair contexto (características) da informação fornecida na entrada; (ii) a segunda parte da sequência (deconvolução) busca fornecer a localização específica dos pixels segmentados e, ao contrário da convolução, aumenta a dimensão da matriz de informações à medida que avança, até que se alcance o tamanho da camada de entrada (RONNEBERGER; FISCHER; BROX, 2015).

A pesquisa envolveu uma estrutura baseada no modelo da Hydra (MINETTO; SEGUNDO; SARKAR, 2019) que pode ser entendida como um *ensemble* de redes onde ocorre inicialmente o treinamento da rede base denominada de "corpo". Após um processo de treinamento com um número pequeno de épocas (menor do que o necessário para se atingir a convergência), o corpo é dividido em "cabeças" especializadas e treinadas de modo a cobrir a maior variedade de casos possível, visando a uma boa generalização. A Hydra foi desenvolvida para a tarefa de classificação e, até onde é de conhecimento dos autores, nunca foi usada com o intuito de segmentar imagens. Portanto, este trabalho tem um componente de inovação importante ao avaliar as vantagens e desvantagens do uso da Hydra no contexto de segmentação semântica.

Neste relatório são descritos os passos adotados na pesquisa para a criação de uma rede profunda baseada no modelo Hydra para segmentação de imagens, tendo a U-net como modelo base e considerando-se duas etapas: (a) treinamento do corpo e (b) treinamento das cabeças. Como estudo de caso, considera-se a competição do Kaggle para segmentação de núcleo de células de tipos variados. Considerada atualmente uma das maiores comunidades de ciência de dados na internet, o Kaggle disponibiliza uma plataforma para a obtenção de bancos de imagens e desenvolvimento de técnicas inteligentes para problemas reais. Sua plataforma é voltada para discussão, aprendizado e compartilhamento de dados associados ao tema de aprendizado de máquina de uma forma geral, e permite ainda que novas técnicas sejam avaliadas mediante participação de seus autores em competições.

O relatório está estruturado da forma descrita a seguir. Após essa seção introdutória, a seção **MATERIAIS E MÉTODOS** traz os passos metodológicos adotados para a criação do modelo (corpo e cabeça) da Hydra, sendo dividida nas seguintes subseções: **Ambiente de Trabalho; Banco de imagens; Pré-processamento das imagens; Treinamento da rede**. A seção **RESULTADOS E DISCUSSÕES** discute os resultados obtidos com o modelo da Hydra quando comparado com as cabeças treinadas separadamente, e por fim a Seção **CONCLUSÕES** finaliza o trabalho e discute perspectivas de trabalhos futuros.

MATERIAIS E MÉTODOS

Neste trabalho foi desenvolvida uma rede baseada no modelo da Hydra (MINETTO; SEGUNDO; SARKAR, 2019) que teve que sofrer algumas adaptações da proposta original, de modo a explorar os elementos necessários para o funcionamento de uma rede convolucional para segmentação semântica de imagens.

A metodologia para a criação de um ensemble de redes profundas baseado no modelo Hydra para segmentação de imagens consiste em duas etapas: (a) treinamento do corpo e (b) treinamento das cabeças. Para isso foi necessário definir um ambiente de trabalho (Kaggle + Tensorflow + Keras) e um banco de imagens (Data Bowl da competição do Kaggle de 2018, que contém 670 imagens de núcleos de células rotuladas e 3084 imagens de teste) e os seguintes métodos: pré-processamento das imagens; *data augmentation*; *design* da rede; treinamento, incluindo a fase de treinamento do corpo e das cabeças; e por fim validação dos resultados. As seções a seguir detalham cada um desses elementos e métodos.

Ambiente de trabalho

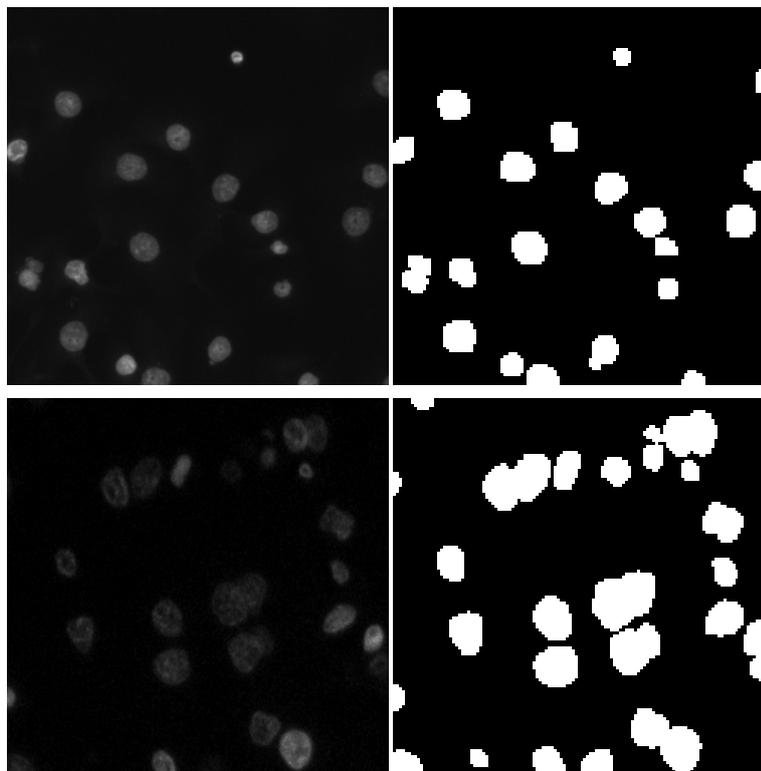
Foi utilizado o ambiente de computação em nuvem do Kaggle para realizar todo o trabalho de programação e execução da rede. O ambiente já vem com a maioria das bibliotecas mais populares para aprendizado de máquina instaladas, sendo possível adicionar outras bibliotecas. As bibliotecas usadas neste trabalho foram:

- Tensorflow: Foi utilizado em conjunto com o Keras para a modelagem, treino e validação da rede.
- Scikit-Image: Foi usado em conjunto com o NumPy e o OpenCV para manipular as imagens durante o pré-processamento e os testes de sanidade.
- Matplotlib: Foi utilizado para verificar a integridade das imagens durante o processo e durante os testes de sanidade.
- NumPy: Foi utilizado para manipular as matrizes de imagens e resultados de modo a possibilitar o treinamento e os testes de sanidade.
- OpenCV: Foi utilizado durante o pré-processamento em alguns casos onde o NumPy não funcionou devido à falta de compatibilidade com o ambiente do Kaggle.
- Pandas: Foi utilizada para analisar os dados resultantes das redes e criar os arquivos .csv com resultados e análises.

Banco de imagens

O banco de imagens utilizado para o treinamento da rede foi obtido da competição do Data Science Bowl de 2018. Ele contém 670 imagens de núcleos de células com máscara (vide Figura 1) e 3084 imagens para teste cego — tipo de teste onde as máscaras não estão disponíveis para os usuários e a performance é calculada através da submissão dos resultados na plataforma do Kaggle. Essa competição teve 3634 equipes e atualmente contém diversas soluções e discussões disponíveis para serem visualizadas. O primeiro lugar na competição utilizou um ensemble de 32 redes U-Net para atingir uma pontuação F1 de 0.82 no limiar IoU de 0,7, atingindo uma pontuação de 0,63164 no placar da competição. As imagens de treino (670 imagens com máscara) foram divididas em 80% imagens de treino e 20% de teste (dados não utilizados no treinamento).

Figura 1 - Imagens de células e suas respectivas máscaras que compõem o banco de dados utilizado no trabalho.



Fonte: Kaggle - Data Science Bowl (2018).

Pré-processamento das imagens

O pré-processamento das imagens envolveu duas etapas: redimensionamento e *data augmentation*:

- Redimensionamento: Tendo em vista que a U-Net utilizada aceita como entrada apenas imagens com tamanho de 128x128 pixels e 3 canais de cor, todas as imagens foram comprimidas de maneira simples para esse tamanho usando as funções do NumPy, OpenCV e Scikit-image.
- *Data augmentation*: Após o redimensionamento as imagens passam pelo processo de *data augmentation* de três maneiras diferentes, de modo a definir diferentes conjuntos de dados a serem fornecidos como entrada para as cabeças da Hydra:
 - cabeça 1: realização de zoom nas imagens de treino, aproximando ou afastando as imagens de 50% a 150%;
 - cabeça 2: alteração aleatória da luminosidade das imagens, escurecendo-as em até 20% da luminosidade original;
 - cabeça 3: rotação aleatória nas imagens em um ângulo de até 60 graus.
 - cabeça 4: inversão vertical aleatória das imagens.
 - cabeça 5: inversão horizontal aleatória das imagens..

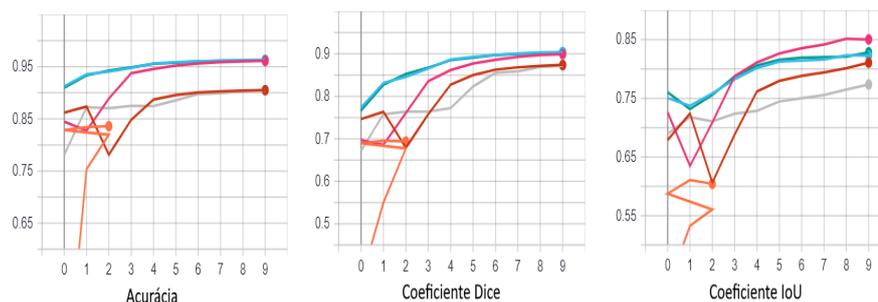
Treinamento da rede

A rede foi modelada utilizando as ferramentas do Keras tendo como base uma U-Net tradicional e treinada a partir da função de perda Dice, que é mais adequada para tarefas de segmentação com grande desbalanço entre o background e os objetos. Esta função de perda (a ser minimizada no treinamento) considera informação tanto local quanto global. Como algoritmo de treinamento optou-se pelo Adam, uma extensão do SGD que é muito utilizado para tarefas de visão computacional e processamento de linguagens naturais. O Adam tem como principal característica suas taxas de aprendizado adaptativas (KINGMA; BA, 2014). O corpo da Hydra foi treinado por 3 épocas e as cabeças por mais 10 épocas a partir dos pesos do corpo, de modo que o corpo não consiga, por si só, alcançar um ponto de mínimo na função de perda, mas sim com o auxílio das cabeças que a partir do ponto inicial melhoram o desempenho do *ensemble*. Após treinadas as cabeças, o resultado da Hydra é composto pela votação da maioria das suas cabeças, ou seja, se um pixel é classificado por três ou mais cabeças, ele é classificado como pixel do objeto pela Hydra;

RESULTADOS E DISCUSSÃO

Foram feitos testes com 3 e 6 cabeças e os melhores resultados (com 6 cabeças) são mostrados a seguir. A Figura 2 ilustra a evolução do treinamento do corpo (laranja) e das cabeças (demais cores) em termos de acurácia, coeficiente Dice e IoU, respectivamente.

Figura 2 - Gráficos de treinamento das redes.



Fonte: Autoria própria.

Após o treinamento da Hydra, foram realizados dois tipos de testes: (i) Teste Tipo1 com dados não utilizados no treinamento (ii) Teste Tipo2 com teste cego no Kaggle.

Resultados do Teste Tipo1: Os resultados obtidos com 20% dos dados rotulados disponíveis e que não foram utilizados no treinamento podem ser vistos na Tabela 1. Destaca-se que, ao invés de se escolher o resultado do modelo de melhor desempenho, optou-se por mostrar os resultados de uma combinação específica de hiperparâmetros mas que representa um comportamento médio de desempenho da Hydra. É possível observar que Hydra apresenta acurácia superior à maioria de suas cabeças funcionando de forma isolada. No entanto, quando se considera IoU e Dice, o seu desempenho não se sobressai.

Tabela 1 - Resultado médio do teste tipo 1.

Modelos	Acurácia	IoU	Coefficiente Dice
Hydra	0.9613	0.7820	0.9037
Modelo 1	0.9409	0.8106	0.9048
Modelo 2	0.9409	0.7332	0.8594
Modelo 3	0.9595	0.7657	0.9044
Modelo 4	0.9631	0.8230	0.9136
Modelo 5	0.9620	0.8270	0.9117

Fonte: Autoria Própria.

Resultados do Teste Tipo2: Quando se realiza o teste Tipo 2 (o teste da competição do Kaggle) envolvendo maior desafio quanto à capacidade de generalização, os resultados na Tabela 2 mostram que a Hydra se comportou melhor que as duas melhores cabeças.

Tabela 2 - Resultados do Teste Tipo 2 (teste cego) realizado na melhor performance obtida.

Modelos	Resultado
Hydra	0.19557
Modelo 1	0.18785
Modelo 2	< 0.16
Modelo 3	< 0.16
Modelo 4	0.18399
Modelo 5	< 0.16

Fonte: Autoria Própria.

Os resultados dos experimentos realizados mostram que o treinamento das cabeças a partir de um ponto de parada do corpo é benéfico para o modelo, e que o ensemble é capaz de prover maior capacidade de generalização à Hydra. Acredita-se que isso ocorra devido à informação complementar (diferentes mínimos locais atingidos por suas cabeças), fazendo com que o desempenho do conjunto seja melhor do que o das partes individuais.

CONCLUSÕES

Este trabalho apresentou os resultados de um ensemble de modelos Unet, que se baseia no conceito da Hydra (um corpo e múltiplas cabeças). O trabalho envolveu o conhecimento em diversos conceitos incluindo computação em nuvem e programação em Python, usando Keras com backend de Tensorflow. O estudo de caso foi baseado na competição do Kaggle considerando-se a segmentação semântica em imagens de células de diferentes tipos. Foi possível perceber que a Hydra pode ser um tipo de ensemble interessante por possuir boa capacidade de generalização em especial quando os problemas são mais desafiadores (como o teste Tipo 2 que foi realizado envolvendo o teste cego do Kaggle). No entanto, para problemas mais simples a relação custo/benefício pode não justificar seu uso. Os maiores desafios envolveram a definição de hiperparâmetros como o número de cabeças, o tipo de *Data Augmentation* para cada cabeça e a definição do momento de parar o treinamento, tanto do corpo quanto das cabeças, em especial quando não há disponibilidade de dados de validação. Os experimentos envolvendo o teste tipo 2 permitem concluir que a Hydra tem potencial para segmentação de imagens, mas novos testes são necessários para se delimitar melhor o alcance dos benefícios e limitações do modelo.

Em trabalhos futuros pretende-se investigar formas de se automatizar a definição de hiperparâmetros, incluindo quais os tipos de *Data Augmentation* são mais adequados para a Hydra quando aplicada a problemas de segmentação semântica, quantas cabeças e quais diferenças devem ser adotadas em relação a cada cabeça.

AGRADECIMENTOS

Os autores agradecem ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pela bolsa de Iniciação Científica (Edital PROPPG 2019/2020 PIBIC) concedida ao acadêmico Paulo Sergio Avila Junior e também pelo financiamento parcial obtido pela orientadora Myriam Regattieri De Biase da Silva Delgado através dos projetos 309935/2017-2 e 439226/2018-0.

REFERÊNCIAS

- DARGAN, S. et al. A survey of deep learning and its applications: A new paradigm to machine learning. **Archives of Computational Methods in Engineering**, Springer, p. 1–22, 2019.
- DEEP Sparse Representation Classifier for facial recognition and detection system. **Pattern Recognition Letters**, v. 125, p. 71 – 77, 2019. ISSN 0167-8655.
- GIRSHICK, R. et al. Rich feature hierarchies for accurate object detection and semantic segmentation. In: IEEE. **Proceedings of the Conference on computer vision and pattern recognition**, 2014. p. 580–587.
- Hinton, G. et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. **IEEE Signal Processing Magazine**, v. 29, n. 6, p. 82–97, 2012.
- KINGMA, D. P.; BA, J. **Adam: A Method for Stochastic Optimization**. 2014.
- LAI, M. **Deep Learning for Medical Image Segmentation**. 2015.
- LECUN, Y.; BENGIO, Y.; HINTON, G. Review. **NATURE**, v. 521, 2015.
- MINETTO, R.; SEGUNDO, M. P.; SARKAR, S. Hydra: An ensemble of convolutional neural networks for geospatial land classification. **IEEE Transactions on Geoscience and Remote Sensing**, IEEE, v. 57, n. 9, p. 6530–6541, 2019.
- OUYANG, W.; WANG, X. Joint deep learning for pedestrian detection. In: IEEE. **Proceedings of the International conference on computer vision**, 2013. p. 2056–2063.
- RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: SPRINGER. **Proceedings of International Conference on Medical image computing and computer-assisted intervention**, 2015. p. 234–241.
- ZHAO, Z.-Q. et al. Object detection with deep learning: A review. **IEEE transactions on neural networks and learning systems**, IEEE, v. 30, n. 11, p. 3212–3232, 2019.