Reconhecimento em conjuntos abertos com Intra-Class Splitting

João Victor Mergner Scaravonatti
Ciência da computação
Instituto Federal Catarinense
Videira, Brasil
jvscaravonatti@gmail.com

***Abstract*—Neste trabalho será feito a reprodução de um método para resolver problemas de reconhecimento em conjuntos abertos (OSR). Em cenários de classificação/reconhecimento tradicionais, todas as classes são conhecidas a priori, ou seja, na etapa de treinamento. Já em cenários de conjuntos abertos, nem todas as classes são conhecidas nesta fase, portanto, classes desconhecidas podem surgir na fase de teste. Como solução para este problemas foram desenvolvidas diversas técnicas, no entanto, o método apresentado neste trabalho utilizará deep learning. Para abordar o problema em questão, o método proposto transforma um problema com K classes em K + 1 classes, subdividindo o conjunto de dados em classes típicas e atípicas, sendo que a última será responsável por modelar as classes desconhecidas. Desta forma, o problema de conjunto aberto será transformado em um problema de conjunto fechado.**

***Palavras-chave*—Reconhecimento em conjuntos abertos, deep learning, intra-class splitting**

I. Introdução

Dentro de cenários onde se utilizam conjuntos fechados, ou seja, tem-se conhecimento de todas as classes utilizadas na fase de treinamento e de teste, algoritmos de machine learning tradicional (SVM, por exemplo), conseguiram obter bons resultados [1]. No entanto, quando se trata de uma aplicação no mundo real, muitas vezes não é possível ter conhecimento sobre todas as possíveis classes de um conjunto de dados. Consequentemente, nestes cenários que se aproximam de cenários realísticos, o que temos de informações são números finitos de classes/objetos que conhecemos, em meio a diversos outros objetos que não temos pouca ou nenhuma informação a respeito [2].

De acordo com Geng et al. [1], existem quatro categorias básicas de classes:

1) *classes conhecidas conhecidas (KKCs)*: classes de amostras de treino classificadas positivamente ou como negativas para outros KKCs
2) *classes conhecidas desconhecidas (KUCs)*: classes rotuladas como amostras negativas
3) *classes desconhecidas conhecidas (UKCs)*: classes não vistas no treinamento, mas com informações adicionais que podem ser utilizadas para o aprendizado
4) *classes desconhecidas desconhecidas (UUCs)*: classes não vistas no treinamente e sem informações adicionais, são apenas vistas na fase de teste

Em um problema de reconhecimento de conjuntos abertos, os classificadores lidam com apenas duas das categorias de classes: KKCs e UUCs [1]. As potenciais soluções para OSR devem ser capazes de classificar KKCs corretamente, tanto quanto classificar UUCs como desconhecidas. Isto é o que difere classificadores de conjuntos abertos com os tradicionais que, por sua vez, irão classificar as classes desconhecidas em alguma das classes vistas no treinamento [2]. Uma comparação entre conjuntos abertos e fechados pode ser observado na Figura 1.

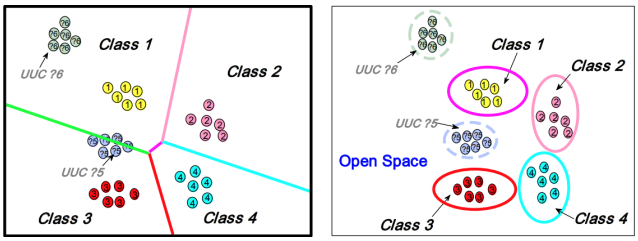

Fig. 1. Comparação entre conjuntos abertos e fechados. Na primeira imagem pode-se observar os limites de decisão de um classificador tradicional e à direita os limites feitos por um classificador de conjuntos abertos.

Nos anos recentes, deep learning se demonstrou uma técnica muito efetiva na classificação de imagens. Porém, a maioria destes métodos atuam na perspectiva de conjuntos fechados. Por conta disso, foram desenvolvidos trabalhos onde são feitas algumas adaptações em redes para a serem utilizadas no contexto de OSR [3].

II. Trabalhos correlatos

Os métodos tradicionais que tratam o problema de conjuntos abertos são variantes das máquinas de vetores de suporte, portanto, as amostras desconhecidas são classificadas dessa maneira levando em consideração thresholds [2], [4].

Em relação ao métodos que utilizam deep learning, o primeiro estudo propôs a utilização do OpenMax, uma camada de modelagem que extende a função SoftMax. A camada OpenMax tem como objetivo reduzir os erros que uma rede profunda convencional comete, e estimar a probabilidade de uma amostra ser desconhecida [5]. Além deste método, também foi realizado um estudo com redes adversárias generativas (GAN), onde imagens fake são utilizadas para modelar as classes desconhecidas [6].

III. Metodologia

A. Ideia geral

O método proposto tem como objetivo reformular um problema que, originialmente, possui um número de K classes, em um problema que contém K + 1 classes. Para fazer isso, um conjunto *Xi* de amostras, onde *i* indica o número de classes conhecidas, é dividido em dois subconjuntos de classes normais: *Xi,típico* e *Xi,atípico*. Após, os conjuntos de amostras atípicas normais são consideradas como uma classe adicional, que será designada a cada amostra desses conjuntos durante o treinamento [3].

Com essa divisão de dados, subentende-se que uma rede profunda com K + 1 neurônios de saída é o suficiente para um classificador de conjuntos abertos. No entanto, as amostras dos conjuntos atípicos normais possuem uma classificação diferente do que são na realidade, isto é, podem ser classificadas incorretamente no treinamento. Para evitar essa situação, uma subrede de regularização de conjunto fechado será utilizada para classificar estas amostras corretamente [3].

Na fase de teste, a rede profunda com a camada de regularização de conjunto fechado será deixada de lado, e apenas a rede de conjunto aberto será utilizada [3]. Na Figura 2 pode-se observar a arquitetura resultante da rede do método proposto.

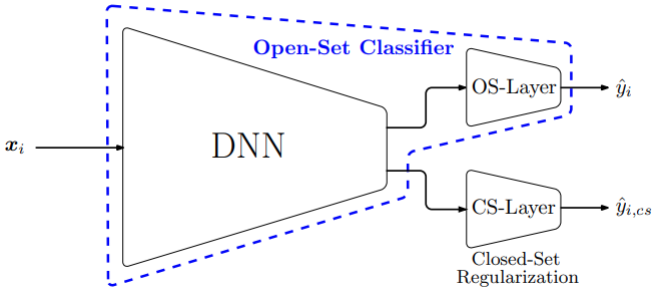

Fig. 2. Arquitetura do método proposto. Possui um classificador de conjunto aberto com K + 1 saídas e um classificador de conjunto fechado com K saídas [3].

B. Intra-Class Spliting

Ao observar as amostras de uma determinada classe, pode-se dizer que algumas amostras possuem melhor qualidade na representação da classe que elas pertencem. Com base nisso, assume-se que um conjunto de amostras de uma base dados comum é composto por duas partes: amostras típicas normais e amostras atípicas normais. As amostras atípicas normais são as que melhor representam suas respectivas classes e também compõe o maior número de amostras dentro de uma bases de dados. Em contrapartida, as amostras restantes são as consideradas atípicas e podem acabar induzindo o classificador a erros [7].

Segundo uma hipótese idealizada por Schlachter et al. (2019), dentro de um espaço latente, os dados desconhecidos ou anormais (como são tratados inicialmente pelos autores) são mais similares com dados atípicos do que com dados típicos [7]. Por conta disso, os dados que representam amostras atípicas normais serão utilizados para a modelagem das classes desconhecidas. Para realizar essa modelagem, durante a etapa de treinamento as amostras atípicas receberão uma nova classe, que será compartilhada com as amostras desconhecidas, conforme pode ser observado na Figura 3 [3].

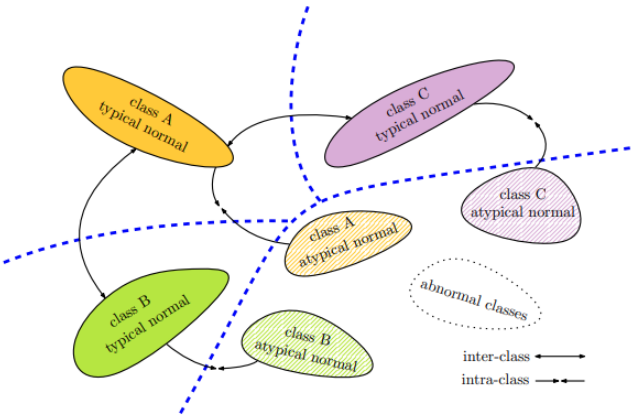

Fig. 3. Exemplo de divisão entre classes (ICS). As linhas azuis representam os limites entre cada classe. Note que, por exemplo, mesmo classe A possuindo duas amostras, uma delas é atípica, portanto ela compartilhará a mesma classe que as todas as outras amostras atípicas e também com as anormais (desconhecidas) [3].

Para a divisão dos dados (Intra-Class Splitting), um classificador é treinado com K classes de dados normais. Em seguida, as amostras que foram classificadas incorretamente e as amostras com baixa probabilidade são selecionadas como amostras normais atípicas. Esse método é formulado da seguinte maneira: $f(.)$ é o mapeamento de uma rede neural de K classes. x representa uma amostra do conjunto de dados do treinamento. Deste modo, as probabilidades previstas no mapeamento são $\hat{\mathbf{y}}prob = f(x)$, onde $\hat{\mathbf{y}}prob \in \mathbb{R}^{K\times 1}$. $\hat{\mathbf{y}}$ é a classe resultante da previsão e é apresentado no valor de one-hot encoding. $\mathbf{y} \in \mathbb{R}^{K\times 1}$ indica o valor real da classificação dentro do one-hot encoding e $\odot$ é o valor do produto Hadamard. Por fim, o score para o Intra-Class Splitting é denotado como

$$score = (\hat{\mathbf{y}}prob \odot \hat{\mathbf{y}} \odot \mathbf{y})^T * \mathbf{1} \quad (1)$$

onde $score \in \mathbb{R}$, $\mathbf{1} \in \mathbb{R}^{N\times 1}$, sendo que $\mathbf{1}$ é um vetores de números 1. Com uma taxa de splitting predefinida ρ, $\rho\%$ amostras com os scores mais baixos serão selecionadas para compor o grupo de amostras atípicas normais, enquantos a restantes serão consideradas típicas normais [3].

C. Regularização do conjunto fechado

A subrede de regularização é formada por uma camada com K neurônios, que será utilizada para manter a precisão alta. Ela será responsável por forçar a classificação correta das amostras atípicas normais somente durante a fase de treinamento [3].

D. Funções de perda

O objetivo da rede toda é transformado em um problema de otimização composto por dois termos de perda, um pertencendo a camada de conjunto aberto e a outro do conjunto fechado, e pode ser denotado da seguinte maneira:

$$\mathcal{L} = \mathcal{L}os + \gamma * \mathcal{L}cs \tag{2}$$

onde $\mathcal{L}os$ é a função de perda da camada de conjunto aberto e $\mathcal{L}cs$ corresponde à função de perda da camada de conjunto fechado. γ é um hiperparâmetro que sintoniza a taxa dos dois termos. Para as duas camadas será utilizada a função conhecida como categorical cross-entropy loss [3].

E. Bases de dados

Os experimentos serão realizados com as bases de dados MNIST [8] e CIFAR-10 [9]. Cada base possui 10 classes, sendo que 6 delas serão selecionads aleatoriamente para as clases conhecidas durante o treinamento. Na fase de teste existirão, além das 6 vistas no treinamento, as 4 classes restantes serão definidas como desconhecidas.

F. Configurações das redes neurais

As implementações de todas as redes neurais serão feitas utilizando scikit-learn, TensorFlow e o wrapper Keras. O método irá utiliza uma versão modificada da rede VGG-16, uma rede neural convolucional proposta por Simonyan e Zisserman em 2014 [10], com o objetivo de reduzir o número de parâmetros da rede [3]. Para camada camada convolucional será utilizada a regularização L2 com decaimento de 10^-3. O valor γ utilizado na função de perda será 1 e taxa de splitting ρ será 10 para a base MNIST e 20 para a CIFAR-10. Por fim, o batch size será 64 e as redes responsáveis pela classificação e divisão dos dados serão treinados em 50 épocas.

G. Métricas para avaliação

A acurácia balanceada (BACC) [11] será a medida principal para avaliar o desempenho da rede, pois ela fornece uma comparação justa entre bases de dados balanceadas e desbalanceadas, o que é muito comum no contexto de conjuntos abertos. As classes conhecidas serão consideradas como positivas e as desconhecidas serão negativas. Com essas variáveis definidas, a acurácia balanceada para conjuntos abertos pode ser calculada por:

$$BACC = \frac{1}{2} * (\frac{TP}{TP + FN} + \frac{TN}{FP + TN}) \tag{3}$$

onde TP representa os verdadeiros positivos, FN os falsos negativos, TN os verdadeiros negativos e FP os falso positivos.

Para complementar a avaliação dos resultados, será utilizada a área abaixo da curva ROC para o classificador de conjunto aberto e também para o de conjunto fechado. A acurácia será utilizada para analisar os resultados obtidas pela rede de conjunto fechado.

IV. Resultados

A. MNIST

Nesta base de dados, a rede neural responsável pelo Intra-Class Splitting obteve uma acurácia de 99,91% no treinamento. O score de similaridade com as classes calculado ficou em 7,6266. Após a comparação com a taxa de similaridade, o conjunto foi dividido em 33162 amostras típicas e 3685 amostras atípicas. A melhor acurácia balanceada para a rede de conjunto aberto foi de 84,27% e a área abaixo da curva ROC ficou em 92,05%. Em relação à rede de conjunto fechado, a melhor acurácia foi 86,51%.

B. CIFAR-10

Em relação à base de dados CIFAR-10, a rede neural do Intra-Class Splitting obteve uma acurácia de 98,95% ao final do treinamento. O valor do score mínimo de similaridade das amostras com as classes ficou em 3,1189. Depois das amostras serem comparadas com a taxa de similaridade, o conjunto foi dividido em 33162 amostras típicas e 3685 amostras atípicas. A melhor acurácia balanceada para a rede de conjunto aberto foi de 63,31% e a área abaixo da curva ROC ficou em 76,93%. Para a rede de conjunto fechado, a melhor acurácia foi 69,80%.

V. Conclusão

Os experimentos reproduzidos conseguiram performances interessantes em relação a outros métodos que foram comparados por [3]. No entanto, o trabalho de Schlachter et al. (2019) ainda obteve uma melhor performance na classificação, principalmente na base de dados MNIST. Os resultados para esta base no experimento conduzido ficou com a BACC de 84,27%, enquanto a dos autores do método alcançou 94,3%. Apesar da diferença ser grande para a base MNIST, a acurácia balanceada alcançada pelo experimento reproduzido na base CIFAR-10 foi muito semelhante ao valor encontrado pelos autores. No experimento conduzido neste trabalho o valor da BACC chegou a 69,8%, enquanto a BACC dos autores ficou em 71,2%.

References

[1] C. Geng, S.-J. Huang, and S. Chen, "Recent advances in open set recognition: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[2] W. J. Scheirer, A. Rocha, A. Sapkota, and T. E. Boult, "Towards open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, vol. 35, July 2013.

[3] Y. L. Patrick Schlachter and B. Yang, "Open-set recognition using intra-class splitting," in *2019 IEEE European Signal Processing Conference (EUSIPCO)*, September 2019.

[4] W. J. Scheirer, L. P. Jain, and T. E. Boult, "Probability models for open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, vol. 36, nov 2014.

[5] A. Bendale and T. E. Boult, "Towards open set deep networks," *CoRR*, vol. abs/1511.06233, 2015. [Online]. Available: http://arxiv.org/abs/1511.06233

[6] L. Neal, M. Olson, X. Fern, W.-K. Wong, and F. Li, "Open set learning with counterfactual images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, set 2018.

[7] P. Schlachter, Y. Liao, and B. Yang, "One-class feature learning using intra-class splitting," *CoRR*, vol. abs/1812.08468, 2018.

[8] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
[9] A. Krizhevsky, "Learning multiple layers of features from tiny images," 2009.
[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.
[11] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, "The balanced accuracy and its posterior distribution," in *2010 20th International Conference on Pattern Recognition*, 2010, pp. 3121–3124.