

Título em Português: Classificação de autômatos celulares através de redes neurais

Título em Inglês: Cellular automata classification using neural networks

Autor: Vitor Lopes Fabris

Instituição: Universidade de São Paulo

Unidade: Instituto de Física de São Carlos

Orientador: Odemir Martinez Bruno

Área de Pesquisa / SubÁrea: Física Geral

Agência Financiadora: CNPq - PIBIC

Classificação de autômatos celulares através de redes neurais

Vitor Lopes Fabris

Odemir Martinez Bruno

Universidade de São Paulo, São Carlos

fabrisvitor@estudante.ufscar.br

Objetivos

A tarefa de classificar autômatos visualmente pode ser comprometida pela subjetividade e desgaste humano de quem a realiza, sem contar que à medida que a configuração dos autômatos se torna mais complexa, o número de regras cresce exponencialmente, dificultando ainda mais esta tarefa. Além disso, manter a classificação de S. Wolfram [1] para autômatos celulares elementares (ECAs) têm sido a norma, mas com uma maior complexidade na configuração dos diagramas espaço-tempo (STD), pode ser que novas classificações tornem-se necessárias.

Por isso, objetivamos o desenvolvimento de um *pipeline* automático de classificação que se inicie na geração de STDs de autômatos e finalize no agrupamento de características visuais extraídas dessas imagens por meio de redes neurais convolucionais (CNNs). Testamos esse fluxo de classificação apenas para os ECAs.

Métodos e Procedimentos

Nosso conjunto de dados consiste de um gerador dinâmico de STDs de autômatos. A dinamicidade do nosso conjunto permite que não ocupamos memória ostensivamente e interage perfeitamente com o *framework* escolhido, Keras [2]. Este gerador consegue simular diferentes configurações de autômatos celulares, mas nos concentramos nos ECAs para os fins desse trabalho. A Figura 1 mostra exemplos de STDs possivelmente pertencentes ao nosso conjunto.

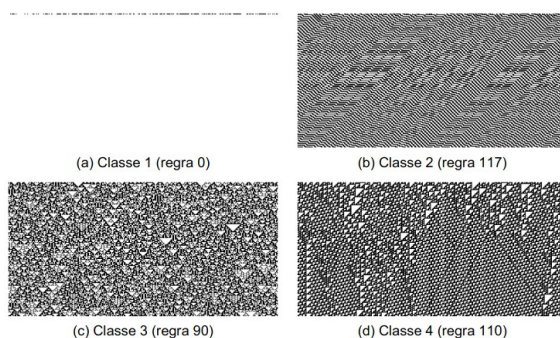


Figura 1: um STD de cada classe Wolfram

Utilizamos duas CNNs para extração de características visuais das imagens. A primeira consiste de uma arquitetura nossa (Figura 2) e a segunda, de uma ResNet50 [3] pré-treinada na ImageNet [4]. A arquitetura de nossa autoria é treinada em nosso conjunto para classificação de STDs no número da regra deles por 50 épocas. Normalizamos nossos dados antes de passá-los pelas CNNs.

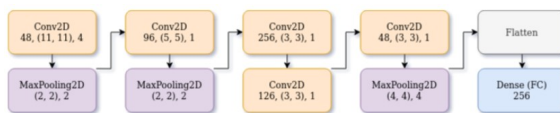


Figura 2: Arquitetura desenvolvida

Reduzimos a dimensionalidade dos resultados da extração realizada pela nossa arquitetura com a técnica de *Principal Component Analysis* (PCA) [5]. Para a extração com a ResNet50 pré-treinada, utilizamos tanto PCA quanto um Autoencoder (AE).

Por fim, realizamos um K-Means [6] com K variando de 2 a 15. Aplicamos o método do cotovelo para avaliar o melhor valor de K e analisamos os agrupamentos resultantes. Na próxima seção, trazemos os resultados do nosso *pipeline*.

Resultados

A CNN elaborada por nós com PCA não apresentou resultados significativos para o agrupamento. Os melhores valores de k para o *pipeline* ResNet50 + PCA foram $k = 4$ e $k = 6$; com Autoencoder, foi $k = 3$ e $k = 7$. Dentre estas, analisando *a posteriori*, a classificação mais significativa se deu com ResNet50 + Autoencoder para $k = 7$ (vide figura abaixo).

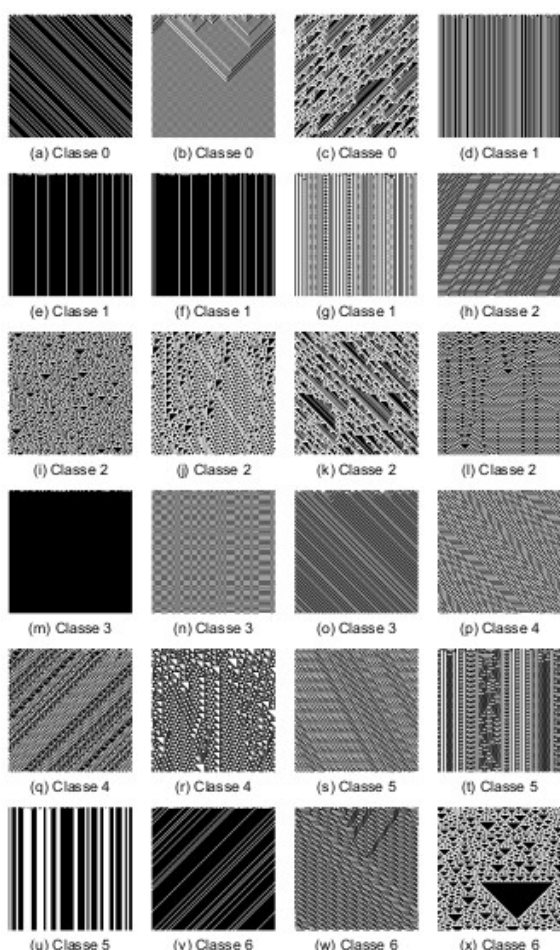


Figura 3: classificações para $k = 7$, ResNet50 + AE

Pode-se ver STDs com comportamentos similares ocupando classes distintas.

Conclusões

A elaboração do *pipeline* foi bem sucedida, com a classificação sendo gerada a partir apenas de STDs e resultando nos agrupamentos. Ademais, para estender esta aplicação para autômatos não elementares (totalísticos, outer-totalísticos, etc) basta gerar os diagramas destes CAs – funcionalidade que nosso sistema de geração de STDs possui.

Contudo, os agrupamentos estão poluídos: ECAs com comportamentos similares ocupam classes diferentes (vide Figura 3). Suspeitamos que isto é devido à ausência de algumas técnicas em nosso *pipeline*: o *fine-tuning* da ResNet50, por meio de *transfer learning*, para a rede aprender características específicas de autômatos; treinamento mais intenso no AE, que alcançou 76.95% em seu conjunto de teste. O K-Means, por sua vez, realiza suposições sobre o formato dos *clusters* que pode não ser interessante para nossa extração; utilizar algoritmos mais robustos, como o HDBSCAN, pode trazer melhores resultados a trabalhos futuros.

Referências Bibliográficas

- [1] WOLFRAM, S. **A New Kind of Science**. Champaign, IL: Wolfram Media, 2002.
- [2] CHOLLET, F. et al. Keras. **GitHub**, 2015. Disponível em: github.com/fchollet/keras.
- [3] HE, K. et al. Deep residual learning for image recognition. In: **2016 IEEE Conference on Computer Vision and Pattern Recognition**, 2016, p. 770-778.
- [4] DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: **2009 IEEE Conference on Computer Vision and Pattern Recognition**, 2009, p. 248-255.
- [5] MISHRA, S. et al. Multivariate statistical data analysis – principal component analysis (PCA). In: **International Journal of Livestock Research**, 2017, v. 7, n. 5, p. 60-78.
- [6] MACQUEEN, J. Classification and analysis of multivariate observations. In: **5th Berkeley Symp. Math. Statist. Probability**, 1967, p. 281-297.

Cellular automata classification using neural networks

Vitor Lopes Fabris

Odemir Martinez Bruno

University of São Paulo, São Carlos

fabrisvitor@estudante.ufscar.br

Objectives

The task of classifying automata visually may be compromised by human subjectivity and wear-and-tear from those who engage in it, and as the automata configuration becomes more and more complex, the number of rules grows exponentially, further hampering this task. Beyond that, using S. Wolfram's classification [1] for elementary cellular automata (ECAs) has been the norm, but higher complexity in the space-time diagrams (STDs) may require new classification schemes.

Therefore, we intend to develop an automatic pipeline for classification that begins in STDs generation and finishes grouping their visual characteristics extracted by convolutional neural networks (CNNs). We test this classification flow only for the ECAs.

Materials and Methods

Our dataset consists of a dynamic generator of automata STDs. The dynamicity of our dataset allows us to not occupy memory ostensibly and interacts perfectly with the chosen framework, Keras [2]. This generator is able to simulate different configurations of cellular automata, but we shall focus on ECAs for the purposes of this work. Figure 1 shows examples of STDs that could possibly be a part of our data.

We have used two CNNs for extraction of visual characteristics from images. The first is an architecture from us (Figure 2) and the second one, a ResNet50 [3] pretrained on ImageNet [4]

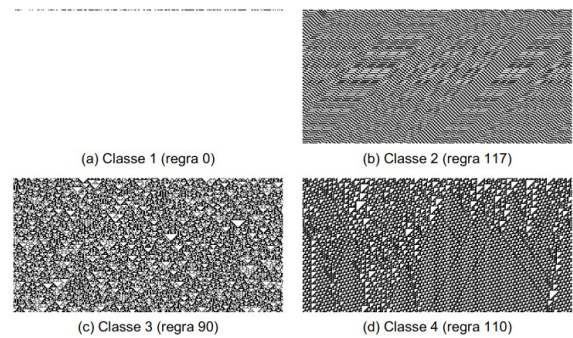


Figure 1: an STD for every Wolfram class

Our architecture is trained on the aforementioned dataset for the STD classification in its rule number for 50 epochs. We normalize our data before feeding to the CNNs.

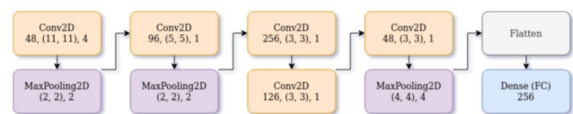


Figure 2: developed architecture

We reduce the dimensionality of our architecture's extracted features with the Principal Components Analysis (PCA) [5] technique. For the pretrained ResNet50's extraction, we used both PCA and an Autoencoder (AE).

In the end, we apply K-Means [6] with K varying between 2 and 15. The elbow method is applied to evaluate the best value for K and we analyze the resulting clustering. In the next section, we show the results of our pipeline.

Results

The CNN developed by us, alongside PCA, didn't have significative results for the clustering. The best values for K with ResNet50 + PCA pipeline were $K = 4$ and $K = 6$; with the Autoencoder, $K = 3$ and $K = 7$. Evaluating *a posteriori*, the most significative classification resulted from the ResNet50 + Autoencoder pipeline (see below figure).

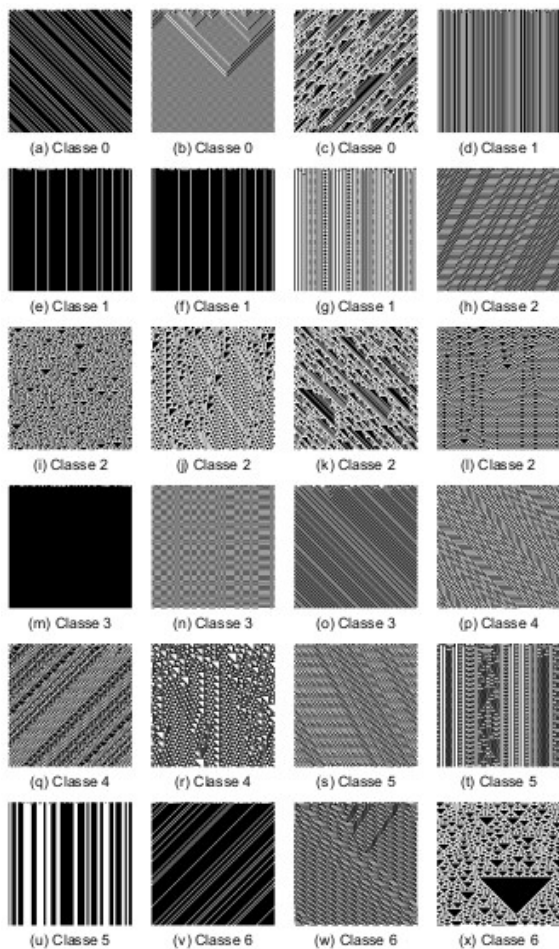


Figure 3: classifications for $K = 7$, ResNet50 + AE

Still, it can be seen that there are STDs with different behaviours occupying distinct classes.

Conclusions

The pipelines' elaboration was successful, with classification starting with STDs and resulting, in the end, in clusters. Besides that, to extend this application to non-elementary automata (totalistics, outer-totalistics, etc), it's enough to

simply generate such STDs – a present functionality in our STD generation system.

However, the clusterings are polluted: ECAs with similar behaviours occupy different classes (see Figure 3). We suspect that this is due to the absence of some techniques in our pipeline: the fine-tuning of ResNet50's model via transfer learning, so that the network can learn characteristics specific to automata; more intense training of our AE, which peaked 76.95% in the test set. K-Means, on the other hand, makes assumptions regarding the clusters shape that can be problematic with our extraction; applying more robust algorithms, such as HDBSCAN, can bring better results to future works.

References

- [1] WOLFRAM, S. **A New Kind of Science**. Champaign, IL: Wolfram Media, 2002.
- [2] CHOLLET, F. et al. Keras. **Github**, 2015. Disponível em: github.com/fchollet/keras.
- [3] HE, K. et al. Deep residual learning for image recognition. In: **2016 IEEE Conference on Computer Vision and Pattern Recognition**, 2016, p. 770-778.
- [4] DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: **2009 IEEE Conference on Computer Vision and Pattern Recognition**, 2009, p. 248-255.
- [5] MISHRA, S. et al. Multivariate statistical data analysis – principal component analysis (PCA). In: **International Journal of Livestock Research**, 2017, v. 7, n. 5, p. 60-78.
- [6] MACQUEEN, J. Classification and analysis of multivariate observations. In: **5th Berkeley Symp. Math. Statist. Probability**, 1967, p. 281-297.