Aprendizagem Automática II
Generative Adversarial Networks para Geração de Moléculas

Grupo 9
Bruno Santos[PG44414], Nelson Estevão[A76434], Rui Reis[A84930] e Susana Marques[A84167]

Universidade do Minho, Departamento de Informática, 4710-057 Braga, Portugal

1 Introdução

1.1 Motivação

Apesar dos modelos de *Generative Adversarial Networks* (GANs) terem sido introduzidos recentemente, em 2014, têm usufuído de um grande sucesso e aplicabilidade em diversas áreas, sendo considerada "a ideia mais interessante dos últimos 10 anos em *machine learning* - Yann LeCun " [5].

O que faz estes modelos serem interessantes e intriguistas é a ideia de treinamento adversarial, representando um verdadeiro progresso conceitual na literatura da área e em particular nos modelos generativos.

De uma forma ampla, as GANs pertencem ao conjunto dos modelos generativos, desto modo, são capazes de produzir/gerar novos dados dispares, mas que seguem a distribuição original dos dados. Por conseguinte, é possível criar imagens de rostos de pessoas tão reais quanto possíveis e que não pertencem a nenhuma pessoa. Este nível de realismo é possível emparelhando um gerador, que aprende a produzir a saída desejada, com um discriminador, que aprende a distinguir os dados verdadeiros dos dados da saída do gerador.

São variadas as aplicações que as GANs possuem êxito, nomeadamente, na geração de textos, na segurança de redes, gerando tráfego semelhante a ciberataques e na geração de novas moléculas, este último é o caso de estudo do presente trabalho e que será aprofundado nas decorrentes secções.

1.2 Identificação do Projeto

No âmbito da unidade curricular "Aprendizagem Automática II" do perfil "Ciência de Dados", foi-nos proposto a seleção de um projeto a ser desenvolvido no decurso do presente semestre em acompanhamento com um orientador. O código-fonte desenvolvido durante a elaboração deste projeto encontra-se no repositório GitLab acessível em *https://gitlab.com/mieiuminho/ds/aa2/tp.git*.

O projeto escolhido, D1, incide na utilização de modelos generativos na descoberta de novas moléculas com propriedades específicas para a produção de fármacos. Tradicionalmente a descoberta de novos medicamentos é um processo

demoroso, pois requer a otimização de compostos químicos em relação a muitas propriedades complexas, sendo isto feito manualmente. Os modelos generativos vieram otimizar e simplificar o processo, dado que utilizam recursos computacionais para as variadas verificações e manipulações de representações de moléculas.

Foram várias as decisões tomadas no decurso do projeto, desde a forma usada para a representação de moléculas até aos modelos generativos a usar, e dentro de cada modelo generativo, a estrutura e o tipo das camadas presentes nos modelos.

2 Objectivos e Contextualização

O cerne do projeto é, dado um conjunto de dados de treino, utilizar modelos generativos, em particular GANs, para produzir novas moléculas com propriedades semelhantes às encontradas no conjunto de dados de treino original.

A representação das moléculas de forma útil para modelos de aprendizagem máquina podem seguir várias abordagens que se podem dividir em dois grupos, representações moleculares gráficas (imagens) ou através de sequências de caracteres (*strings*). Existem diferentes estruturas de sequências para representar uma molécula, entre as quais, a mais popular representação para modelos generativos é a *Simplified molecular input lineentry system* (SMILES) [4]. Existem outras alternativas, tais como DeepSMILES que são uma extensão de SMILES com o objetivo de reduzir o número de sequências inválidas [3], ou até SELFIES *SELF-referencIng Embedded Strings* (SELFIES) que são uma nova abordagem à representação de moléculas com sequências [2] mas com resultados piores no que toca a modelos generativos [1].

Assim, partindo de um *dataset* contendo representações, o objetivo é, através de modelos generativos apresentados na secção 5, produzir novas moléculas válidas com alta similaridade estrutural com as moléculas presentes no *dataset* original. Ou seja, conseguimos gerar uma nova molécula, diferente das existentes no *dataset* original e com propriedades úteis para diferentes propósitos na produção de novos fármacos. O *dataset* utilizando é apresentado na secção 3.

3 Conjunto de dados MOSES

O conjunto de dados utillizado no presente projeto é o conjunto de dados MOSES. O MOSES é baseado no *dataset* ZINC e é constituído por cerca de dois milhões de moléculas que podem ser divididas, em subconjuntos de treino, validação e teste, como exemplificado na imagem seguinte.

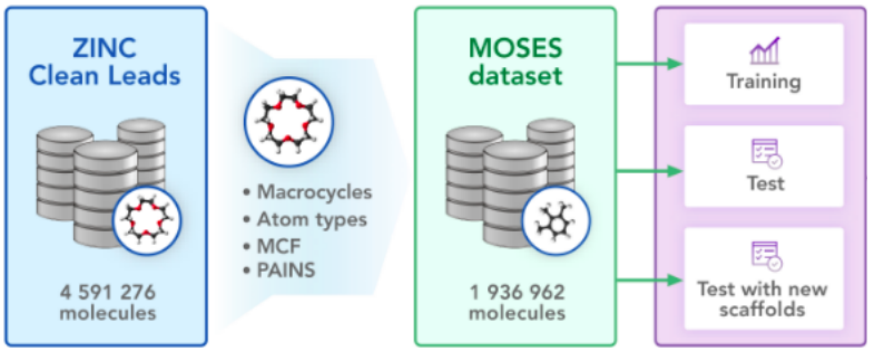


Figura 1: Conjunto de dados MOSES

Os exemplos presentes no MOSES contêm apenas um coluna, que é a representação em forma de *string* de moléculas. A representação utilizada é denominada de SMILES apresentada na secção 2. A partir de um conjunto de ferramentas disponibilizadas pelo RDKit, é possível extrair múltiplas informações de uma molécula, a partir do seu SMILES.

4 Plano de trabalhos

De modo a atingir os objetivos apresentados na secção 2, construiu-se um conjunto de etapas com metas individuais para alcançar esses fins.

A primeira fase consistiu em perceber as diferentes categorias de representações moleculares existentes e as suas características específicas. Esta fase está intimamente ligada ao tratamento e preparação dos dados para utilização nas fases de treino, validação e treino dos modelos.

Apesar das *Generative Adversarial Networks* (GANs) serem recentes, foi possível consolidar algum conhecimento sobre algumas variações existentes. Estas variações são apresentadas na secção 5.

Após o estudo das já existentes GANS, passou-se a experimentação de diferentes modelos e utilizando diferentes representações moleculares de forma a fornecer os dados mais adequados para cada modelo.

Por fim, treinaram-se os modelos e foi feita a análise de resultados apresentada na secção 6.

5 Modelos Abordados

Ao longo do semestre o grupo experimentou diferentes abordagens para o problema exposto. Numa primeira fase para compreender o conceito de Gan, o grupo optou por experimentar diversos modelos a partir de imagens de moléculas. Numa segunda fase, após os resultados não se demonstrarem significativos, foi

se dado ênfase ao uso das moléculas do *dataset* já previamente explicado (SMILES) e de impressões digitais num espaço latente.

5.1 Vanilla GAN

Um modelo mais simples que dando o *dataset* de treino com imagens de moléculas gera novas imagens com as mesmas estatísticas, sem camadas profundas de aprendizagem que serviu como ponto de partida para o grupo se ambientar com o conceito.

A ideia principal deste modelo baseia-se no treino indireto das imagens através de um discriminador que também está dinamicamente a ser treinado. Assim o gerador de imagem não está a ser treinado de forma habitual, mas sim está a ser treinado para enganar o discriminador, permitindo que o modelo aprenda de forma não-supervisionada.

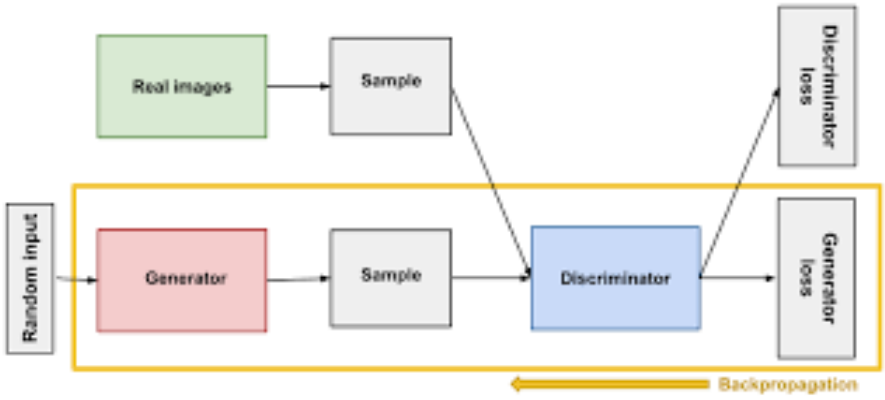


Figura 2: Workflow de uma Gan simples

Como podemos ver pela imagem a GAN inclui dois modelos, um que gera novos exemplos e outro que tenta discriminar se exemplos são novos ou originais (usados no treino), treinados para otimizar funções *loss* opostas. Um exemplo intuitivo visto nas aulas será um falsificador de quadros que vai mostrando o seu trabalho a um especialista de arte; à medida que o falsificador vai melhorando a capacidade de criar melhores falsificações, o especialista vai-se tornando melhorar a reconhecê-las.

5.2 Deep Convolutional GAN

A seguir o grupo experimentou com uma Deep Convolutional Gan que permitiu, através de existência de camadas mais profundas tirar partido do Kernel usado.

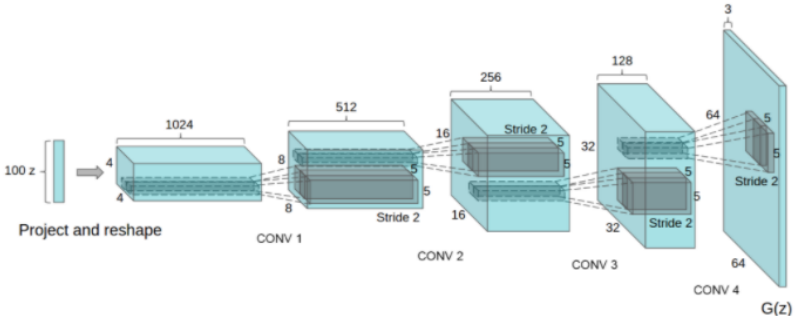


Figura 3: Estrutura das camadas do *Generator* de uma DCGAN

Uma Deep Convolutional Gan é composta principalmente de camadas de convolução sem agrupamento máximo ou camadas totalmente conectadas. Ela usa passos convolucionais e convolução transposta para o *downsampling* e o *upsampling.* Tal como a *Vanilla Gan* o *Generator* recebe como entrada um vetor aleatório (ponto gerado no espaço latente), descodificando numa imagem nova. E o *Discriminator* (adversário) recebe como entrada uma imagem (real ou gerada) e classifica-a como sendo original (do conjunto de dados de treino) ou gerada pelo modelo *Generator*.

5.3 Latent GAN

Estes dois modelos anteriores foram um bom ponto de partida para se entender e trabalhar com GANs embora os resultados ficassem aquém das espectativas. No treino de uma GAN tem que se ir alternando entre o treino do Generator e do Discriminator, uma ou mais *epochs* cada um (enquanto se treina um modelo, o outro fica inalterado) mas a convergência é difícil de identificar pois as *loss functions* são opostas (adversariais) e por isso não há melhoria absoluta.

Neste último modelo, como o treino de GANs até ao momento foi bastante difícil, o grupo concordou que representações de imagens de moléculas são inapropriadas para geração de moléculas e focou-se em usar SMILES para o desenvolvimento do próximo modelo.

Neste introduziu-se aleatoriedade adicionando ruído aleatório aos rótulos do discriminador e fez-se uma amostragem de pontos (impressões digitais mapeadas) do espaço latente já pré-treinado usando uma distribuição normal (distribuição gaussiana) em vez de uma distribuição uniforme.

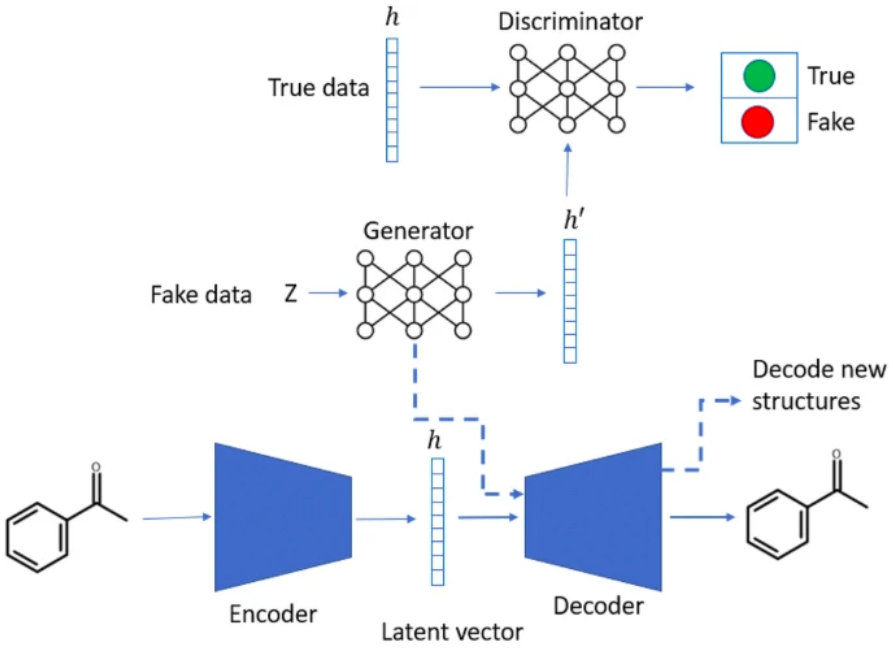


Figura 4: Workflow de uma LatentGAN

Uma GAN Wasserstein foi escolhida como modelo GAN. Primeiro, o discriminador é formado por três camadas *feed-forward* de 256 dimensões, cada uma com a função de ativação *ReLU* exceto para a última camada onde nenhuma função de ativação foi usada. Em segundo lugar, o gerador consiste em cinco camadas *feed-forward* de 256 dimensões cada uma com normalização e função de ativação ReLU.

O modelo do *heteroencoder* foi primeiro pré-treinado no *dataset* ChEMBL para mapear estruturas para vetores latentes. Para treinar o modelo GAN completo, primeiro o vetor latente h do conjunto de treinamento foi gerado usando a parte codificadora do *heteroencoder*. Em seguida, ele foi usado como a entrada de dados verdadeira para o discriminador, enquanto um conjunto de vetores aleatórios amostrados de uma distribuição uniforme foram tomados como entrada de dados falsos para o gerador. Assim que o treino da GAN foi concluído, os vetores latentes resultantes foram alimentados no *Decoder* para obter as sequências SMILES das moléculas subjacentes.

6 Resultados

6.1 Pré-Processamento

Representações Gráficas Numa primeira fase, foi gerada uma base de dados com o objetivo de servir de conjunto de treino para os modelos generativos que tiram partido de representações gráficas. Isto resultou num *data set* de milhões de imagens, tais como as da figura 5.

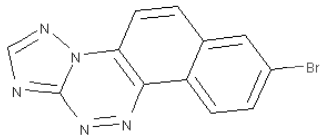


Figura 5: Exemplo de representação gráfica gerada

Propriedades Moleculares Ademais, no intuito de analisar agrupamentos naturais de moléculas com base em determinados atributos, foi gerada uma base de dados referente a atributos como: peso molecular, solubilidade e área polar de cada molécula. A figura 6 representa alguns destes dados gerados.

```
SMILES|molWt|Solubility|PolarSurfaceArea
CCCS(=O)c1ccc2[nH]c(=NC(=O)OC)[nH]c2c1|1.686|1.689|87.31
CC(C)(C)C(=O)C(Oc1ccc(Cl)cc1)n1ccnc1|3.729|3.729|44.125
Cc1c(Cl)cccc1Nc1ncccc1C(=O)OCC(O)CO|2.297|2.297|91.68
Cn1cnc2c1c(=O)n(CC(O)CO)c(=O)n2C|-2.213|-2.213|102.28
CC1Oc2ccc(Cl)cc2N(CC(O)CO)C1=O|0.807|0.807|70.0
CCOC(=O)c1cncn1C1CCCc2ccccc21|2.985|2.985|44.12
COc1ccccc1OC(=O)Oc1ccccc1OC|3.281|3.281|53.991
O=C1Nc2ccc(Cl)cc2C(c2ccccc2Cl)=NC1O|3.101|3.101|0.69
CN1C(=O)C(O)N=C(c2ccccc2Cl)c2cc(Cl)ccc21|3.125|3.125|52.906
CCC(=O)c1ccc(OCC(O)CO)c(OC)c1|1.019|1.01|75.99
Cc1nc2c([nH]1)c(=O)n(C)c(=O)n2CC1CC=CCC1|1.08|1.08|72.68
```

Figura 6: Exemplos de dados gerados

6.2 Modelos Generativos

Vanilla GAN O modelo de Vanilla GAN, como esperado, não gera os resultados pretendidos. As representações gráficas geradas são demasiadas simplistas e demasiado esparsas, devido a linhas com grossura de 1 pixel, para que alguma informação possa ser aprendida pelo modelo Vanilla GAN. A figura 7 representa um dos resultados obtidos, onde se observa facilmente que o resultado é imperceptível, sendo composto por uma nuvem mal definida de pontos, cujo significado difícil de obter.

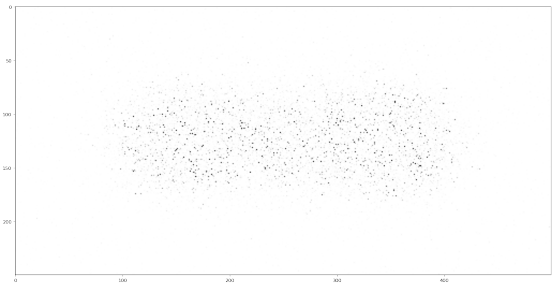


Figura 7: Resultado do modelo Vanilla GAN

DCGAN Uma solução ponderada para o resultado negativo do modelo Vanilla GAN foi a falta de capacidade dos modelos aprenderem as relações espaciais em seu redor, i.e. linhas entre compostos. Neste sentido, achamos adequado aplicar um modelo de DCGAN, que consegue, através do uso de kerneis, ter em consideração a vizinhança de cada pixel.

No entanto, os resultados, mesmo utilizando DCGAN, mostraram-se insignificantes, tal como pode ser evidenciado pela figura 8, um resultado exemplo. Com isto em mente, o grupo decidiu que o uso de representações gráficas é inadequado para a aplicação em modelos generativos.

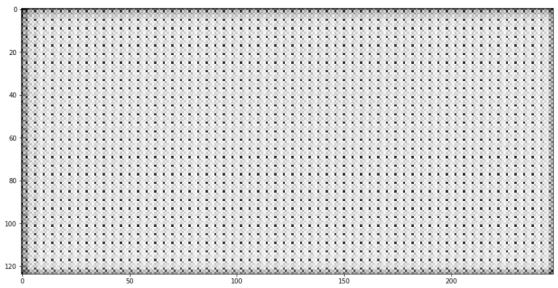
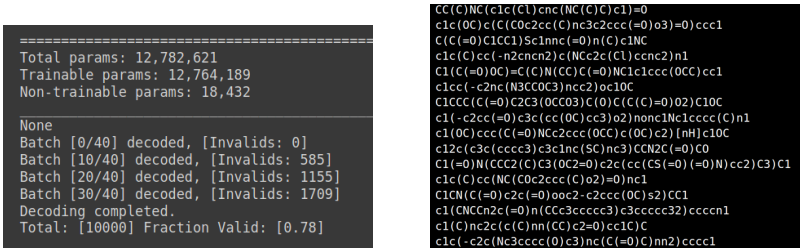


Figura 8: Resultado do modelo DCGAN

Latent GAN Finalmente, com a aplicação do modelo Latent GAN, foi possível obter resultados interessantes e com significado. Através da aplicação deste modelo foi possível obter um modelo que consegue gerar 78% de moléculas válidas. Sendo que a maioria destas não pode ser encontrada no *data set* de treino original, o que indicia um grande sucesso na capacidade de gerar novas moléculas não existentes no treino e teste. A figura 9 representa estatísticas dos dados gerados bem como, à direita, alguns exemplos de novas moléculas.



(a) Estatísticas gerais **(b)** Amostras geradas

Figura 9: Resultado do modelo Latent GAN

7 Conclusão

Finalizando o trabalho, podemos concluir que o mesmo foi bastante desafiante tendo em conta o tema escolhido.

O desenvolvimento de GANs é algo bastante prematuro principalmente referente à criação de novas moléculas na área da Bioquímica.

Neste trabalho, o desenvolvimento cognitivo ao nível de aprendizagem de trabalho na área de *SMILES* foi bastante positivo e deveras interessante. Entender o funcionamento dos *datasets* e dos *packages e librarys* foi algo trabalhoso e enriquecedor a nível do conhecimento de moléculas e da sua composição.

Chegando ao cerne do problema e como usar GANs para a geração das moléculas o grupo enfrentou diversas complicações devido à escassez de informação na *Internet* obrigando a uma grande pesquisa na fase inicial do trabalho e a implementação dos modelos levou a bons resultados no final após várias tentativas onde os resultados não foram os pretendidos (ao usar imagens de moléculas). Mas foi através desses resultados e de várias tentativas para enquiquecer os modelos criados e treinados que o grupo conseguiu entender a fundo todo o *workflow* de uma GAN e como aplica-la em diversas situações futuras não só referentes a conjuntos de moléculas mas a outras áreas também.

Assim, o grupo entende que este trabalho serviu para enriquecer o seu conhecimento sobre a área retratada, levando consigo experiências que podem ser bastante úteis no futuro na área de *Machine Learning*.

Referências

1. Kadurin, A., Aliper, A., Kazennov, A., Mamoshina, P., Vanhaelen, Q., Khrabrov, K., Zhavoronkov, A.: The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology. Oncotarget **8**(7), 10883–10890 (2017). https://doi.org/https://doi.org/10.18632/oncotarget.14073, `https://www.oncotarget.com/article/14073/`
2. Krenn, M., Häse, F., Nigam, A., Friederich, P., Aspuru-Guzik, A.: SELFIES: a robust representation of semantically constrained graphs with an example application in chemistry. CoRR **abs/1905.13741** (2019), `http://arxiv.org/abs/1905.13741`

3. Kusner, M.J., Paige, B., Hernández-Lobato, J.M.: Grammar variational autoencoder. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 70, pp. 1945–1954. PMLR (06–11 Aug 2017), http://proceedings.mlr.press/v70/kusner17a.html
4. Polykovskiy, D., Zhebrak, A., Sanchez-Lengeling, B., Golovanov, S., Tatanov, O., Belyaev, S., Kurbanov, R., Artamonov, A., Aladinskiy, V., Veselov, M., Kadurin, A., Nikolenko, S.I., Aspuru-Guzik, A., Zhavoronkov, A.: Molecular sets (MOSES): A benchmarking platform for molecular generation models. CoRR **abs/1811.12823** (2018), http://arxiv.org/abs/1811.12823
5. Rocca, J.: Understanding generative adversarial networks (gans), https://towardsdatascience.com/understanding-generative-adversarial-networks-gans-cd6e4651a29