XXVIII Congresso {virtual} de Iniciação Científica da Unicamp 2020

Redes Neurais Adversariais

José L. Barretto*, Romis Attux.

Resumo
Projeto de pesquisa desenvolvido com o intuito de compreender aspectos teóricos e práticos do modelo de Redes Neurais Generativas Adversariais. Os aspectos teóricos estão relacionados à formulação matemática do framework adversarial, bem como possíveis adaptações que podem ser realizadas de forma a melhorar a qualidade das amostras geradas ou reduzir o custo computacional. Quanto aos aspectos práticos, a tarefa designada foi gerar amostras sintéticas de dígitos escritos à mão, com o auxílio do *MNIST Dataset of Handwritten Digits*. As implementações consistiram em projetar, treinar e analisar modelos generativos, e foram realizadas com um grau crescente de complexidade, iniciando com o modelo adversarial inicial proposto por Ian Goodfellow, seguindo para as Redes Adversariais Convolucionais e concluindo com Redes Adversariais Condicionais. Ao longo do processo de implementação, foram utilizadas técnicas diversas visando a melhoria das amostras sintéticas produzidas pela rede geradora, bem como o controle sobre quais amostras serão produzidas.

Palavras-chave:
Redes Neurais Adversariais, GANs, Modelos Generativos

Introdução

As Redes Neurais Generativas Adversariais são um modelo de aprendizado de máquina, apresentado em 2014, por Ian Goodfellow, que revolucionou o cenário de estimação generativa com o uso de redes neurais que treinam em conjunto. A principal fonte de estudo para este projeto foi o artigo *Generative Adversarial Nets*, que propõe o *framework* adversarial, sugere estratégias de treinamento e apresenta resultados iniciais.

O *framework* adversarial consiste basicamente em duas redes neurais MLP: o modelo generativo **G** e o modelo discriminativo **D**. Para aprender a distribuição generativa $\mathbf{p_g}$ sobre os dados **x**, o modelo **G** recebe como entrada uma amostra aleatória **z** de uma distribuição de ruído $\mathbf{p_z}$, e representa um mapeamento dessa distribuição (também chamada de espaço latente) para o espaço dos dados. O modelo **D**, por sua vez, recebe como entrada uma amostra de dado e fornece como saída a probabilidade de esta amostra ser advinda de **x** (amostra genuína) e não de $\mathbf{p_g}$ (amostra gerada). Treina-se o modelo **D** para maximizar a probabilidade de ele atribuir o rótulo correto a exemplos advindos tanto de **x** como de **G**. Em contrapartida, treina-se **G**, simultaneamente, para minimizar $log(1-\mathbf{D}(\mathbf{G}(\mathbf{z})))$ - ou seja, **G** é treinado para enganar **D** a aceitar as suas amostras sintéticas como verdadeiras [1].

Resultados e Discussão

As implementações seguiram um roteiro pré-planejado, com complexidade crescente. Dessa forma, o conhecimento desenvolvido em experimentos mais básicos foi utilizado como fundação para experimentos mais complexos. Além disso, os experimentos consistiram não só em gerar amostras sintéticas realísticas do conjunto de dados, mas também buscaram manipular o espaço latente do *framework*, de forma a manipular e controlar a geração de amostras. Nesse sentido, diversas técnicas foram adotadas, de forma a otimizar o processo de treinamento, gerar amostras melhores e viabilizar o controle sobre a geração de imagens.

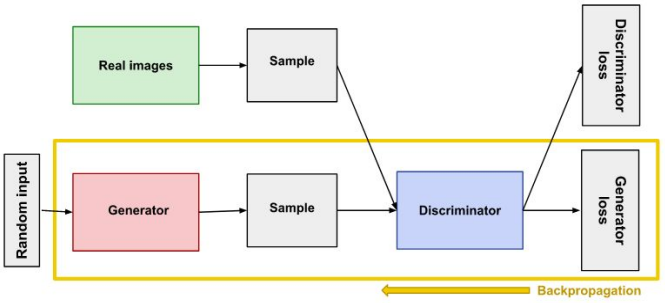

Figura 1. Esquemático do Framework Adversarial [2].

- **Implementação Inicial do Framework Adversarial:**

A ideia deste experimento foi produzir uma implementação inicial do *framework* adversarial proposto por Goodfellow originalmente - **G** e **D** são MLPs simples. A função de custo utilizada nesta e em todas as outras implementações foi a entropia cruzada binária. Essa função é numericamente estável, e apresenta valores interpretáveis para as perdas.

Ao analisar o custo médio por época para o discriminador e para o gerador (figura 2), pode-se perceber que as curvas estão bem coerentes. No início do treinamento, a perda do discriminador cai rapidamente, enquanto a do gerador aumenta. Isso ocorre pois as amostras falsas do gerador ainda são ruins. Com o tempo, o gerador produz cópias melhores, o que leva a uma diminuição na sua perda e a um aumento pequeno na perda do discriminador.

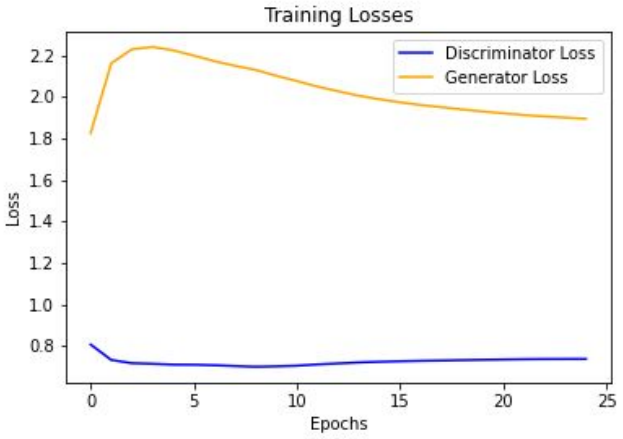

Figura 2. Média das perdas por época para o gerador e para o discriminador.

O número de épocas de treinamento foi determinado empiricamente, e escolhido de forma que a qualidade das amostras geradas já não sofre melhoria significativa com o aumento das iterações de treinamento. Após treinar o *framework* por 25 épocas, pode-se perceber que ele já consegue gerar amostras sintéticas razoavelmente similares às amostras originais do *dataset* (figura 3)

Figura 3. Amostras aleatórias produzidas pelo gerador MLP após 25 épocas de treinamento.

- **Implementação de uma DCGAN:**

O *framework* original proposto por Goodfellow utiliza redes MLP para os modelos discriminador e gerador. Todavia, é possível alterar a arquitetura original dos modelos para redes convolucionais, como proposto no artigo *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks* [3]. Os autores propõem uma arquitetura convolucional sem camadas *fully-connected*, além de outras mudanças. Algumas dessas mudanças são: adição de normalização de *batch* em ambos os modelos, camadas convolucionais com passo no discriminador e camadas convolucionais com passo fracionário no gerador.

Seguindo estas orientações, adotou-se uma arquitetura similar, com dimensões adaptadas para o conjunto de dados escolhido, ilustrada nas figuras 4 e 5.

Os resultados obtidos após 50 épocas de treinamento apresentaram uma melhora significativa: as amostras sintéticas apresentam um ruído menor e se aproximam ainda mais das amostras verdadeiras contidas no conjunto de dados (figura 6).

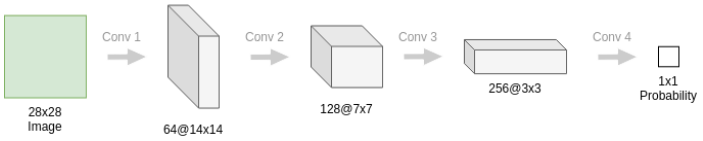

Figura 4. Arquitetura do Discriminador na DCGAN.

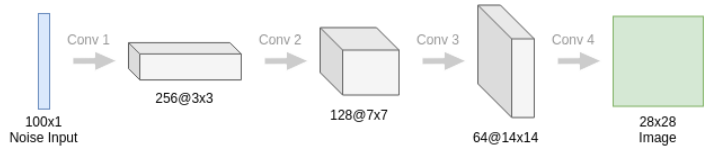

Figura 5. Arquitetura do Gerador na DCGAN.

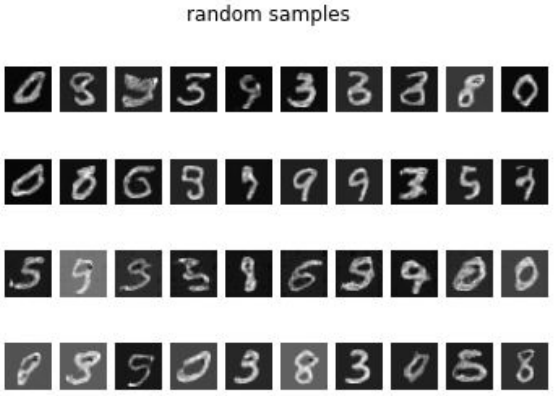

Figura 6. Amostras sintéticas aleatórias geradas pela DCGAN.

- **Experimentos com o Espaço Latente:**

De forma a compreender melhor o funcionamento das GANs, alguns experimentos foram realizados a fim de explorar o espaço latente amostrado e utilizado como sinal de entrada para a rede geradora.

O espaço latente é um espaço multidimensional que pode ser amostrado para gerar uma variável de ruído, que, por sua vez, é transformado pelo gerador em valores observáveis de dados. Isso significa que, amostras diferentes do espaço latente geram amostras sintéticas de dados diferentes. Por isso, pode-se afirmar que o gerador nada mais é do que uma representação do espaço latente para o espaço dos dados. Essa representação é aprendida durante o processo de treinamento do *framework* adversarial. Analisar o espaço latente, portanto, é de suma importância para entender como gerar amostras desejadas de dados.

Inicialmente, estudou-se o impacto da variável de ruído nas amostras produzidas pelo gerador. Uma técnica comum para visualizar este efeito é a interpolação entre dois pontos amostrados do espaço latente. A interpolação produz uma transição suave entre as imagens geradas, e demonstra como a rede transita entre pontos diferentes do espaço latente (figura 7).

Como visto anteriormente, o gerador é um mapeamento do espaço latente para o espaço dos dados. O mapeamento inverso, todavia, não é trivialmente obtido quando se treina o *framework* adversarial. Este mapeamento inverso, que consiste basicamente em recuperar o vetor latente que dá origem à um determinado dado produzido pelo gerador, pode ser

obtido através de um simples método baseado em gradiente [4].

Este método é suficiente para obter o mapeamento do espaço dos dados para o espaço latente, mas também pode ser utilizado para recuperar o vetor latente de imagens não produzidas pelo gerador. Isso é extremamente útil para reconstruir sinteticamente uma imagem desejada. Nesse caso, o vetor latente encontrado provavelmente não resultará exatamente na imagem desejada quando interpretado pelo gerador, mas sim em uma imagem similar (figura 8).

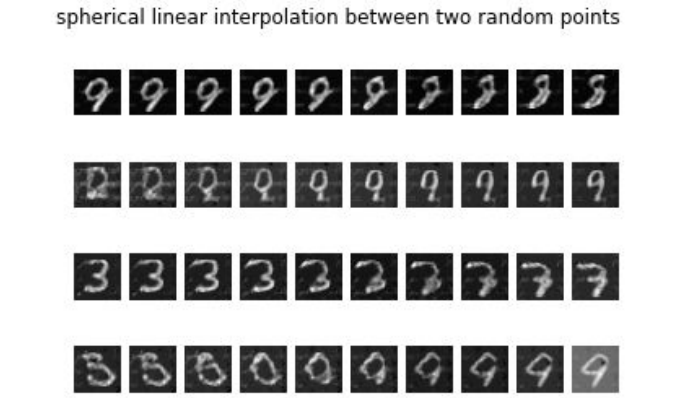

Figura 7. Interpolação linear esférica entre dois pontos amostrados aleatoriamente do espaço latente.

Figura 8. Reconstrução sintética de amostras aleatórias do *MNIST dataset* através do método de recuperação do vetor latente. Na coluna da esquerda, estão amostras retiradas aleatoriamente do conjunto de dados. Na coluna da direita estão as amostras reconstituídas, geradas a partir do vetor latente encontrado.

- **Implementação de uma c-DCGAN:**

Como visto anteriormente, o modelo adversarial não possui controle sobre as amostras geradas sem a utilização de métodos externos ao modelo. Uma forma de contornar este problema é com a utilização de um modelo generativo condicional. Neste caso, o modelo é condicionado com informações adicionais, o que viabiliza o controle sobre o processo de geração de dados. Esse condicionamento pode ser feito com as *labels* disponíveis no dataset - no caso do *MNIST*, é possível utilizar as *labels* de classe de cada imagem [5].

Este modelo condicional é bastante similar ao modelo adversarial visto anteriormente, com a diferença de que tanto o gerador quanto o discriminador serão condicionados com uma informação extra **y**: a classe a qual cada dado (gerado ou discriminado) pertence. Dessa forma, os modelos passam a ter uma nova entrada, que é a informação condicional - o gerador passa a ser **G**(**z**|**y**) e o discriminador passa a ser **D**(**x**|**y**).

Para ambos os modelos, a *label* **y** é mapeada em uma camada *embedding* e, em seguida, passa por uma camada linear com uma dimensão pré-definida. O objetivo desta camada é fornecer a dimensão desejada para o sinal condicional.

Nesse caso, as amostras geradas apresentaram a melhor qualidade dentre os modelos utilizados. Além disso, como o modelo é condicional, existe um maior controle sobre as amostras geradas, uma vez que pode-se condicionar o gerador arbitrariamente com a classe desejada, de forma a obter a utilização das *labels* uma amostra sintética específica.

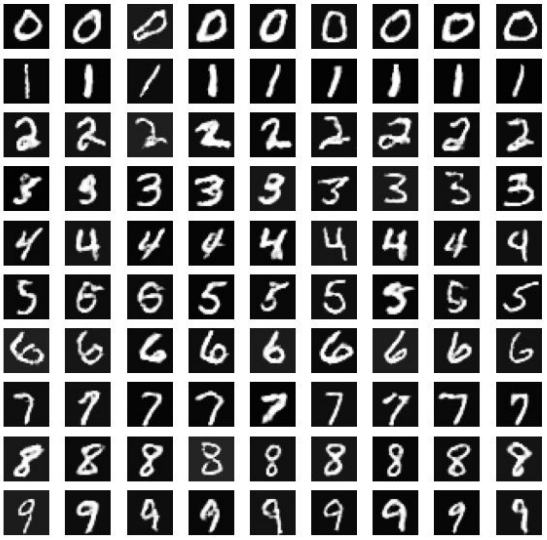

Figura 9. Amostras aleatórias produzidas pelo gerador de uma c-DCGAN, condicionado a gerar imagens de dígitos entre 0 e 9.

Este aumento na qualidade das imagens produzidas artificialmente é esperado, dado que no modelo condicional há a utilização de uma informação extra: a *label*. Sendo assim, o modelo condicional pode ser classificado como um modelo semi-supervisionado, o que o diferencia do modelo adversarial original, que é não supervisionado.

Conclusões

O projeto de iniciação científica forneceu ao aluno um conhecimento teórico e prático de diversos aspectos das Redes Adversariais, desde a sua formulação matemática até tópicos subjacentes de otimização envolvidos no treinamento das redes.

As implementações atingiram resultados de qualidade altíssima, o que reafirma o potencial deste modelo generativo. Apesar de apresentar desafios no treinamento, a utilização de diversos métodos heurísticos viabiliza o aprendizado do modelo - especialmente no caso do conjunto de dados adotado.

Por fim, é possível concluir que o projeto foi desenvolvido com sucesso, oferecendo ao aluno uma gama de técnicas e métodos essenciais para o desenvolvimento

desse tipo de modelo em tarefas que envolvam a geração de imagens.

Todas as implementações produzidas durante o projeto estão em: **github.com/lucasbarretto/MNIST_GANN**.

---

[1] Ian Goodfellow et al. “Generative adversarial nets”. Em: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[2] Google, ed. *Google Developers - GAN Anatomy*. URL: **https://developers.google.com/machine-learning/gan/generator**.

[3] Alec Radford, Luke Metz e Soumith Chintala. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. 2015. arXiv: 1511.06434 [cs.LG].

[4] Zachary C. Lipton e Subarna Tripathi. *Precise Recovery of Latent Vectors from Generative Adversarial Networks*. 2017. arXiv: 1702.04782 [cs.LG].

[5] Mehdi Mirza e Simon Osindero. *Conditional Generative Adversarial Nets*. 2014. arXiv: 1411.1784 [cs.LG].