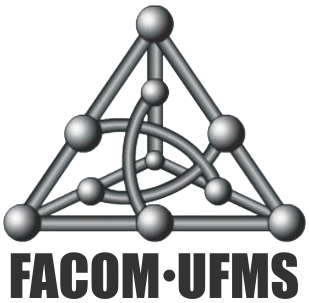
Algoritmos de Inferência de Redes de Regulação Gênica usando Deep Learning

Relatório de Atividade Orientada de Ensino

Matheus Droppa Omido

Orientação: Prof. Dr. Carlos Henrique Aguena Higa

Curso: Sistemas de Informação



Faculdade de Computação
Universidade Federal de Mato Grosso do Sul

Campo Grande, dezembro de 2025

Resumo

Nesta Atividade Orientada de Ensino (AOE) estudamos o problema da inferência de redes de regulação gênica, usando *Deep Learning* e dados de *single cell RNA-Seq* (scRNA-Seq). Para isso, estudamos os conceitos de Deep Learning a partir de um curso de *Neural Networks and Deep Learning*. Foi realizada uma revisão na literatura científica sobre metodologias de inferência de redes que utilizam modelos de Deep Learning. A partir dos códigos-fonte disponibilizados em alguns dos artigos estudados, experimentos foram parcialmente realizados.

Abstract

In this Guided Teaching Activity (GTA), we studied the problem of inferring gene regulatory networks using Deep Learning and single-cell RNA-Seq (scRNA-Seq) data. To do this, we studied the concepts of Deep Learning from a course on Neural Networks and Deep Learning. A review of the scientific literature on network inference methodologies using Deep Learning models was conducted. Based on the source code provided in some of the studied articles, experiments were partially performed.

1 Introdução

No contexto Biológico as informações dos organismos vivos podem ser interpretadas sendo armazenadas em DNA, processadas para RNA e, posteriormente, traduzidas para proteínas. Na biologia molecular esse fluxo desempenha diversas funções para a replicação e a propagação de informações por meio da transcrição do RNA [4]. No entanto nem todos os genes são expressos ao mesmo tempo, a ativação ou inibição de determinados genes depende de mecanismos de regulação que controlam quando ocorre a transcrição.

As Redes de Regulação Gênica são quem controlam esse estado transcricional de uma célula. Essas redes funcionam como sistemas de controle, nos quais os fatores de transcrição (TFs) se regulam mutuamente e coordenam a expressão de outros genes [1]. Analisar esses dados genéticos, com precisão, se torna desafiador pois, no nível de célula única, as expressões gênicas podem variar de forma estocástica devido às diferentes fontes de ruído [7].

Com o avanço dos estudos em Aprendizado Profundo (Deep Learning), as abordagens para análise de dados de expressão gênica têm se tornado cada vez mais sofisticadas, incorporando métodos de correção de ruídos mais precisos e eficientes [13]. Dessa forma, torna-se possível solucionar limitações e inconsistências que anteriormente não podiam.

Diante do discutido, e pelos interesses em desenvolver mais em relação ao assunto, durante a AOE (Atividade Orientada de Ensino), focamos em estudar alguns artigos relacionados a Redes de Regulação Genica e o sequenciamento de RNA em célula única (scRNA-seq), além de estudar o básico de Deep Learning e Redes Neurais com um curso externo do Angrew Ng oferecido na plataforma Coursera [6]. Por fim colocamos em pratica códigos em Python de alguns artigos relacionados a inferir redes de regulação gênica a partir de dados de scRNA-Seq.

2 Curso Neural Networks and Deep Learning

No curso *Neural Networks and Deep Learning*, ministrado por Andrew Ng, é apresentado conceitos teóricos e práticos das redes neurais artificiais, desde os conceitos básicos até a implementação de redes de aprendizado profundas. Ao longo dos quatro módulos, o curso aborda Inteligencia Artificial, Programação em Python, Aprendizado Supervisionado, Aprendizado de Máquina, Álgebra Linear, Aprendizado Profundo e Redes Neurais Artificiais.

2.1 Modulo 1 - Introdução à Inteligência Artificial

Na primeira parte do curso é introduzido o conceito de *Inteligência Artificial (IA)* e destaca e *Deep Learning (Aprendizado Profundo)* como uma das principais ferramentas para a extração automática de padrões em grandes volumes de dados na atualidade.

O ponto de partida é o modelo de rede neural artificial, inspirado no funcionamento do cérebro humano, no qual um neurônio artificial recebe entradas $x_1, x_2, ..., x_n$ aplica os pesos $w_1, w_2, ..., w_n$, e então é somado ponderadamente e também adiciona um viés b.

$$z = w_1x_1 + w_2x_2 + ... + w_nx_n + b$$

O resultado é então passado por uma função de ativação $a = g(z)$, que introduz não linearidade no modelo.

Essa arquitetura é aplicada em problemas de aprendizado supervisionado, onde o objetivo é aprender uma função que mapeie uma entrada $\mathbf{x} \in \mathbb{R}^n$ para uma saída $y \in \{0, 1\}$ com base em exemplos rotulados.

Andrew Ng destaca que o crescimento do Deep Learning foi impulsionado principalmente por três fatores:

- Aumento massivo de dados disponíveis;
- Avanço na capacidade computacional (GPUs);
- Melhoria nos algoritmos de treinamento.

2.2 Modulo 2 - Regressão Logística

O segundo modulo introduz a Regressão logística (Logistic Regression) como um padrão fundamental de classificação binaria e a base matemática para redes neurais. O objetivo foi compreender o funcionamento de um neurônio isolado, fundamental para a estrutura de um rede neural, e a forma que aprende os parâmetros por meio da otimização da função de custo.

Seja $\mathbf{x}$ um vetor de características, tal que $|\mathbf{x}| = n$, representado como um vetor coluna:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} .$$

Logo, uma amostra de treinamento consiste de um par $(\mathbf{x}, y)$, onde $\mathbf{x} \in \mathbb{R}^n$ e $y \in \{0, 1\}$. Dizemos que y é um *rótulo* de $\mathbf{x}$. Para fins didáticos,

dizemos que $\mathbf{x}$ é uma foto ou figura, em que $y = 1$ representa o fato de que a figura é de um gato, e que $y = 0$ significa que a figura não é de um gato. Em geral, o conjunto de dados de treinamento possui m amostras: $\{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots, (\mathbf{x}^{(m)}, y^{(m)})\}$. Em outras palavras, temos várias fotos em que sabemos, para cada uma, se ela representa um gato ou não.

Na Regressão Logística, dado $\mathbf{x}$ (figura), queremos estimar $\hat{y} = P(y = 1 \mid \mathbf{x})$ (probabilidade de representar um gato, dado a figura). Para isso, introduzimos parâmetros no modelo que são ajustados a medida em que o processo de regressão avança. Sejam os parâmetros $\mathbf{w} \in \mathbb{R}^n$ (coeficientes) e $b \in \mathbb{R}$. Temos que $\hat{y} = \sigma(\mathbf{w}^\intercal \mathbf{x} + b)$, onde

$$\sigma(z) = \frac{1}{1 + e^{-z}} \ .$$

A função sigmóide σ faz com que $0 \leq \hat{y} \leq 1$, como deve ser, pois $\hat{y}$ é uma probabilidade. Na Fig. 1 temos a ilustração de uma unidade de computação na regressão logística. Em termos de redes neurais, tal unidade corresponde a um neurônio na rede.

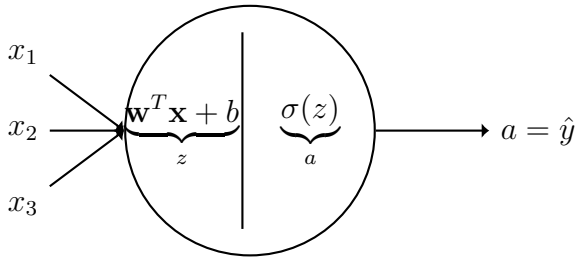


Figura 1: Esquema gráfico ilustrando a regressão logística aplicada a um vetor $\mathbf{x}$ de tamanho $n = 3$.

Para medir o erro entre a previsão $\hat{y}$ e o valor real y, utiliza-se a função de perda logarítmica (log loss), definida para um único exemplo como:

$$\mathcal{L}(\hat{y}, y) = -\left[y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})\right] \ .$$

Para um conjunto de m amostras, a *função de custo* total é dada por:

$$J(\mathbf{w}, b) = \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}(\hat{y}^{(i)}, y^{(i)}) \ .$$

O objetivo do treinamento é minimizar $J(\mathbf{w}, b)$, ajustando os parâmetros $\mathbf{w}$ e b gradativamente. Para o processo de otimização dos parâmetros utiliza-se a gradiente descendente (Gradient Descent), ajustando os parâmetros de

$\mathbf{w}$ e b para minimiza $J(\mathbf{w}, b)$:

$$\mathbf{w} \leftarrow \mathbf{w} - \alpha \frac{\partial J(\mathbf{w}, b)}{\partial \mathbf{w}}$$

$$b \leftarrow b - \alpha \frac{\partial J(\mathbf{w}, b)}{\partial b} \quad ,$$

onde α é a taxa de aprendizagem (*learning rate*) e as derivadas parciais correspondem à inclinação da função J. Os ajustes são feitos iterativamente, até que se atinja uma convergência.

Vetorização : É um processo - destacado por Andrew Ng - utilizado para eliminar loops explícitos e acelerar o cálculos em grupos de dados.

Broadcasting : É um conceito - destacado por Andrew Ng - que permite operações matemáticas entre matrizes de dimensões diferentes.

2.3 Modulo 3 - Rede Neural com uma camada oculta

Uma rede neural contendo uma única camada oculta pode ser vista na Fig. 2. Cada neurônio da camada oculta realiza o mesmo processamento da unidade apresentada na Fig. 1.

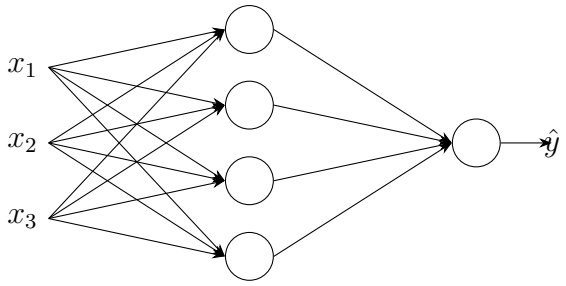


Figura 2: Um esquema gráfico ilustrando uma rede neural com apenas uma camada oculta e $n = 3$.

A estrutura matemática de uma rede com uma camada escondida é:

$$Z^{[1]} = W^{[1]} X + b^{[1]}$$
$$A^{[1]} = g^{[1]}(Z^{[1]})$$
$$Z^{[2]} = W^{[2]} A^{[1]} + b^{[2]}$$
$$\hat{Y} = A^{[2]} = g^{[2]}(Z^{[2]})$$

onde:

- $W^{[l]}$ e $b^{[l]}$ são os parâmetros da camada l,
- $g^{[l]}$ é a função de ativação aplicada em cada camada,
- e $\hat{Y}$ representa a saída predita do modelo.

O processo de *Forward Propagation* (propagação direta) consiste no cálculo das ativações da rede neural de forma sequencial, iniciando-se na camada de entrada e prosseguindo sucessivamente até a camada de saída. Para ajustar os pesos, aplica-se a *Backpropagation* (retropropagação), que utiliza o Teorema da Cadeia para calcular gradientes de forma eficiente.

A retropropagação calcula os gradientes de cada camada a partir da saída até a entrada:

$$dZ^{[2]} = A^{[2]} - Y$$
$$dW^{[2]} = \frac{1}{m} dZ^{[2]} (A^{[1]})^T$$
$$db^{[2]} = \frac{1}{m} \sum_{i=1}^{m} dZ^{[2](i)};$$
$$dZ^{[1]} = (W^{[2]})^T dZ^{[2]} * g'^{[1]}(Z^{[1]})$$
$$dW^{[1]} = \frac{1}{m} dZ^{[1]} X^T$$
$$db^{[1]} = \frac{1}{m} \sum_{i=1}^{m} dZ^{[1](i)}$$

O curso apresenta também diferentes funções de ativação, com suas vantagens e desvantagens:

Sigmoid: $g(z) = \frac{1}{1+e^{-z}}$

Tanh: $g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$

ReLU (Rectified Linear Unit): $g(z) = \max(0, z)$

Funções de ativação não lineares, como tanh e ReLU, permitem que a rede aprenda representações complexas. Além disso, esse modulo também apresenta a inicialização com pesos aleatórios — uma etapa fundamental para evitar que todos os neurônios aprendam os mesmos padrões, o que ocorre se os pesos forem inicializados com zero. Durante o treinamento, é importante monitorar o comportamento da função de custo J para garantir a convergência do modelo.

2.4 Modulo 4 - Redes Neurais Profundas

No último modulo o Andrew Ng estende os conceitos apresentados nos módulos anteriores, aplicando-os para uma rede neural com múltiplas camadas ocultas *(Neural Deep Networks)*.

No exemplo abaixo é representada uma rede neural com três camadas ocultas Fig. 3. Como visto nos módulos anteriores, o mesmo conceito se repete da Fig. 2, apenas aumentando a escala de dados, tendo a possibilidade L camadas ocultas.

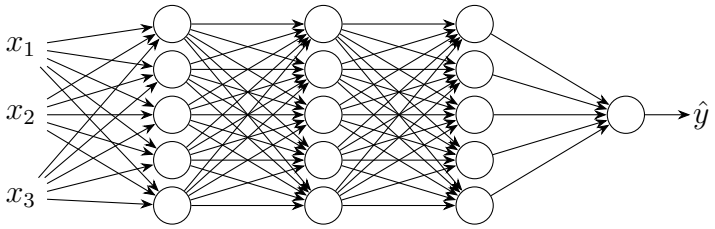


Figura 3: Rede neural com três camadas ocultas e $n = 3$.

Em uma rede profunda com L camadas temos

$$Z^{[l]} = W^{[l]} A^{[l-1]} + b^{[l]}$$

$$A^{[l]} = g^{[l]}(Z^{[l]}), \quad l = 1, 2, \ldots, L \ ,$$

onde $A^{[0]} = X$ representa a matriz de entrada.

O processo de treinamento segue o mesmo princípio das redes anteriores: *Forward propagation*, cálculo da função de custo e *Back propagation* para ajuste dos parâmetros.

Durante o treinamento, é importante distinguir entre dois tipos de variáveis fundamentais:

- Parâmetros: $W^{[l]}$ e $b^{[l]}$ — são aprendidos automaticamente durante o treinamento via otimização da função de custo.
- Hiperparâmetros: taxa de aprendizado (α), número de iterações, número de camadas e neurônios por camada, e a escolha da função de ativação — definidos manualmente pelo projetista.

A justificativa teórica para o uso de redes mais profundas está relacionada à teoria dos circuitos: determinadas funções podem ser representadas de forma muito mais eficiente (com menos unidades e conexões) por redes profundas, enquanto redes rasas exigiriam um número exponencialmente maior de neurônios para representar as mesmas relações.

O curso enfatiza que o processo de desenvolvimento de modelos de *Deep Learning* é altamente empírico. Na prática, a criação de redes neurais eficazes envolve um ciclo iterativo de experimentação, ajuste de hiperparâmetros e avaliação de desempenho. Fatores como normalização de dados, inicialização adequada e regularização também influenciam diretamente a estabilidade e a eficiência do aprendizado.

3 Artigos estudados

Durante a AOE alguns artigos sobre inferência de redes de regulação gênica e outros temas relacionados foram estudados:

- *Gene regulatory network inference in the era of single-cell multi-omics* [2];
- *Single-cell RNA sequencing technologies and applications: A brief overview* [3];
- *TRENDY: gene regulatory network inference enhanced by transformer* [8];
- *Wendy: Covariance dynamics based gene regulatory network inference* [10];
- *GRLGRN: graph representation-based learning to infer gene regulatory networks from single-cell RNA-seq data* [9];
- *A hybrid deep learning framework for gene regulatory network inference from single-cell transcriptomic data* [12];
- *LineGRN: a line graph neural network for gene regulatory network inference* [11];
- *GMFGRN: a matrix factorization and graph neural network approach for gene regulatory network inference* [5].

Estudamos artigos envolvendo os temas: (i) inferência de redes de regulação gênica; (ii) Deep Learning; e (iii) single-cell RNA-Seq. Com isso, buscamos fazer uma revisão bibliográfica sobre o estado da arte em que o problema de inferência de redes é atacado por metodologias envolvendo modelos de Deep Learning, considerando que os dados disponíveis são provenientes de scRNA-Seq.

Alguns artigos em que algoritmos baseados em Deep Learning são apresentados, o código-fonte foi disponibilizado. Em particular, foi estudado o algoritmo LineGRN [11], baseado em redes neurais e dados de scRNA-Seq. Entre as dificuldades encontradas para se realizar os experimentos, podemos

citar o hardware necessário para fazer o treinamento do modelo, e os dados fornecidos de maneira incompleta pelos autores do artigo. Ao término da AOE, utilizamos o Google Colab em uma tentativa de realizar os experimentos. Com relação aos dados, estamos buscando alternativas, como dados de outras fontes, para que os experimentos possam ser realizados.

4 Conclusão

Nesta AOE, procuramos estudar o problema da inferência de redes de regulação gênica, usando Deep Learning e dados de scRNA-Seq. Inicialmente, estudamos a tecnologia de scRNA-Seq para entender, de fato, quais são as informações que podem ser obtidas através desse tipo de dado. Em seguida, focamos em estudar e entender como os modelos de aprendizagem profunda funcionam. Para isso, acompanhamos um curso online de *Neural Networks and Deep Learning*, ministrado por Andrew Ng, professor da Stanford University e reconhecido globalmente como um dos pioneiros da IA e da Educação à Distância.

Com relação ao problema de inferência de redes, fizemos uma revisão da literatura sobre o tema, estudando os artigos descritos da Seção 3. Tentamos reproduzir alguns resultados apresentados nos artigos, porém a dificuldade de infra-estrutura adequada e consistência dos dados fornecidos dificultaram esta eta vpa. Sendo assim, a execução de experimentos para confirmar os resultados dos artigos estudados ficam para um possível estudo futuro.

Referências

[1] Sara Aibar, Carmen Bravo González-Blas, Thomas Moerman, Vân Anh Huynh-Thu, Hana Imrichova, Gert Hulselmans, Florian Rambow, Jean-Christophe Marine, Pierre Geurts, Jan Aerts, et al. SCENIC: single-cell regulatory network inference and clustering. *Nature methods*, 14(11):1083–1086, 2017.

[2] Pau Badia-i Mompel, Lorna Wessels, Sophia Müller-Dott, Rémi Trimbour, Ricardo O Ramirez Flores, Ricard Argelaguet, and Julio Saez-Rodriguez. Gene regulatory network inference in the era of single-cell multi-omics. *Nature Reviews Genetics*, 24(11):739–754, 2023.

[3] Dragomirka Jovic, Xue Liang, Hua Zeng, Lin Lin, Fengping Xu, and Yonglun Luo. Single-cell RNA sequencing technologies and applications: A brief overview. *Clinical and translational medicine*, 12(3):e694, 2022.

[4] Samuel A Lambert, Arttu Jolma, Laura F Campitelli, Pratyush K Das, Yimeng Yin, Mihai Albu, Xiaoting Chen, Jussi Taipale, Timothy R Hughes, and Matthew T Weirauch. The human transcription factors. *Cell*, 172(4):650–665, 2018.

[5] Shuo Li, Yan Liu, Long-Chen Shen, He Yan, Jiangning Song, and Dong-Jun Yu. Gmfgrn: a matrix factorization and graph neural network approach for gene regulatory network inference. *Briefings in Bioinformatics*, 25(2):bbad529, 2024.

[6] Andrew Ng. Neural Networks and Deep Learning. Coursera, Stanford University, 2024.

[7] Arjun Raj and Alexander Van Oudenaarden. Nature, nurture, or chance: stochastic gene expression and its consequences. *Cell*, 135(2):216–226, 2008.

[8] Xueying Tian, Yash Patel, and Yue Wang. TRENDY: gene regulatory network inference enhanced by transformer. *Bioinformatics*, 41(6):btaf314, 2025.

[9] Kai Wang, Yulong Li, Fei Liu, Xiaoli Luan, Xinglong Wang, and Jingwen Zhou. GRLGRN: graph representation-based learning to infer gene regulatory networks from single-cell RNA-seq data. *BMC bioinformatics*, 26(1):108, 2025.

[10] Yue Wang, Peng Zheng, Yu-Chen Cheng, Zikun Wang, and Aleksandr Aravkin. Wendy: Covariance dynamics based gene regulatory network inference. *Mathematical Biosciences*, 377:109284, 2024.

[11] Ziwei Wang, Ge Xu, Weiming Yu, and Le Ou-Yang. LineGRN: a line graph neural network for gene regulatory network inference. *IEEE Journal of Biomedical and Health Informatics*, 2025.

[12] Mengyuan Zhao, Wenying He, Jijun Tang, Quan Zou, and Fei Guo. A hybrid deep learning framework for gene regulatory network inference from single-cell transcriptomic data. *Briefings in bioinformatics*, 23(2):bbab568, 2022.

[13] Bin Zou, Tongda Zhang, Ruilong Zhou, Xiaosen Jiang, Huanming Yang, Xin Jin, and Yong Bai. deepMNN: deep learning-based single-cell RNA sequencing data batch correction using mutual nearest neighbors. *Frontiers in Genetics*, 12:708981, 2021.