



Aprendizado por Representação

Seminário - Aprendizado Profundo

Filipe Augusto Jesus Rodrigues

frodriguesfajr@gmail.com

Programa de Engenharia Elétrica
Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa em Engenharia
Universidade Federal do Rio de Janeiro

07/11/2024

Conteúdo da Apresentação

1. Introdução
2. *Greedy Layer-Wise Unsupervised Pretraining*
3. *Transfer Learning*
4. *Disentangling* Semi-Supervisionado de Fatores Causais
5. Representação Distribuída
6. Ganhos exponenciais da profundidade
7. Pistas para Descobrir Causas Subjacentes

Introdução

Importância da Representação

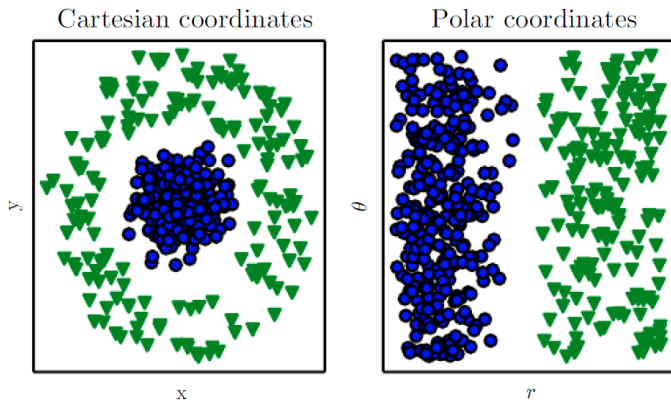


Figura 1: Diferentes representações para um problema de classificação [1].

Representação - Complexidade de Tarefas

- Executar cálculos: Algoritmos Arábicos vs Romanos
 - Divisão de $210/6$ vs CCX/VI
- Qual é uma boa representação para ML?
 - Aquela que torna o aprendizado subsequente mais fácil

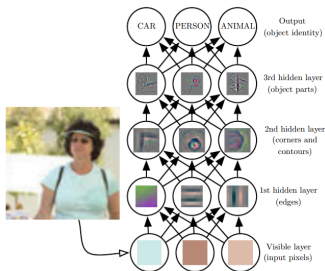


Figura 2: Ilustração de um modelo *Deep Learning* [1].

Redes FF aprendem representações

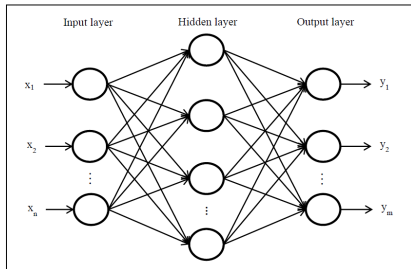


Figura 3: Estrutura de uma Rede FF [2].

- Podemos pensar em uma rede FF $f(\mathbf{x}) = f^{(3)}[f^{(2)}[f^{(1)}(\mathbf{x})]]$ treinada por aprendizado supervisionado como um tipo de Aprendizado por Representação
- A última camada da rede é um classificador linear, como o classificador de regressão softmax. O restante da rede aprende a representação para o classificador

Como especificamos a representação?

- Aprendizado supervisionado de redes feed-forward:
 - Nenhuma imposição de quaisquer condições em recursos aprendidos
- Outros algoritmos de aprendizado de representação fazem isso
 - Ex: Na estimativa de densidade, incentive h_i a ser independente
- Algoritmos de aprendizado profundo não supervisionados
 - Em aprendizado não supervisionado, o modelo também aprende representações das entradas, mas não por meio de rótulos; a representação é aprendida como um efeito colateral do objetivo principal (como a reconstrução no caso de autoencoders).
- O aprendizado de representação envolve um *trade-off* entre:
 - Preservar o máximo de informações sobre a entrada quanto possível
 - Obter boas propriedades (como independência)

Greedy Layer-Wise Unsupervised Pretraining

Pré-treinamento e Ajuste Fino

- Usando o dataset A treinamos o modelo M . :
 - No Pré-treinamento, temos o dataset B
 - Antes de treinar o modelo, iniciamos alguns dos parâmetros do modelo M treinados como o dataset A
 - No Ajuste Fino, treinamos o modelo M no dataset B
- Esta é uma forma de Aprendizado por Transferência
- O aprendizado não-supervisionado desempenhou um papel histórico fundamental no renascimento das Redes Neurais Profundas em 2006, permitindo o treinamento de uma Rede Supervisionada sem exigir estruturas como convolução ou recorrência
- Esse procedimento é conhecido como “*Greedy Layer-Wise Unsupervised Pretraining*”

Greedy Layer-Wise Unsupervised Pretraining

Pretraining

Duas fases, uma para Treinamento e outra para ajuste fino

Unsupervised

Não são utilizados dados com *label*

Layer-Wise

O Treinamento é realizado camada por camada

Greedy

Cada camada maximiza sua própria função custo, independentemente das outras

Pré-Treinamento Não Supervisionado *Greedy Layer-Wise*

- O Pré-Treinamento Não Supervisionado *Greedy Layer-Wise* depende de um aprendizado de representação de camada única
- Precisamos de um algoritmo de aprendizagem de representação de camada única, como: RBM, Autoencoder de camada única, modelo de codificação esparsa ou outro modelo que aprenda representações latentes
- Cada camada é pré-treinada usando aprendizagem não supervisionada. Tomamos a saída da camada anterior e produzimos como saída uma nova representação de dados, onde a distribuição (ou relação com categorias) é mais simples

Algoritmo

State $f \leftarrow$ Identity function

$\tilde{\mathbf{X}} = \mathbf{X}$

for $k = 1, \dots, m$ **do**

$f^{(k)} = \mathcal{L}(\tilde{\mathbf{X}})$

$f \leftarrow f^{(k)} \circ f$

$\tilde{\mathbf{X}} \leftarrow f^{(k)}(\tilde{\mathbf{X}})$

end for

if *fine – tuning* **then**

$f \leftarrow \mathcal{T}(f, \mathbf{X}, \mathbf{Y})$

end if

return f

Quando e por que o Pré-treinamento funciona?

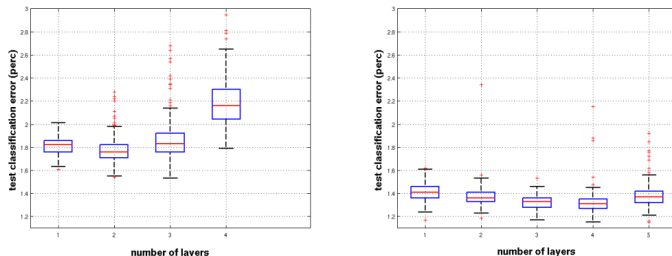


Figura 4: Taxa de erro de reconhecimento de dígitos manuscritos (dados MNIST) [3]

- O pré-treinamento não supervisionado combina duas ideias
 1. Os parâmetros iniciais têm um efeito regularizador, ou seja, se aproximam de um mínimo local em relação a outro. Mas os mínimos locais não são mais considerados um problema
 2. Aprender sobre a distribuição de entrada pode ajudar a aprender sobre o mapeamento de entradas para saídas (carros e motocicletas têm rodas)

Trajetórias de aprendizado

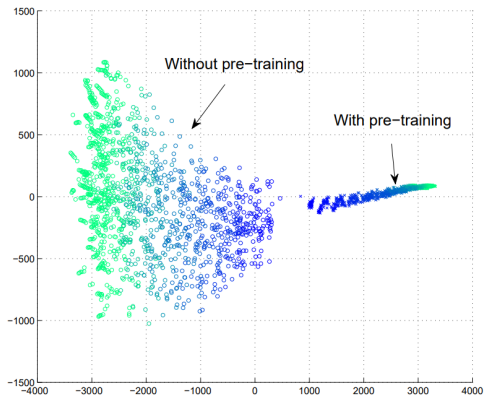


Figura 5: Visualização de funções projetadas no espaço 2D. Cada função é um vetor infinitodimensional que associa cada entrada x com saída y . A cor de azul escuro a ciano indica o tempo (o treinamento é mais longo sem pré-treinamento) [3].

Pré-Treinamento Não Supervisionado *Greedy Layer-Wise* - Atualidade

- *Greedy Layerwise Unsupervised Pretraining* não está mais em uso atualmente
 - Foi desenvolvida como uma maneira de treinar redes neurais profundas em uma época em que os métodos supervisionados modernos ainda não eram tão avançados
- Difícil de otimizar. Cada camada tem seu próprio conjunto de hiperparâmetros.
- Abordagens supervisionadas são melhores para grandes quantidades de dados.
 - Com grandes volumes de dados rotulados disponíveis, os métodos supervisionados têm mostrado um desempenho superior ao pré-treinamento não supervisionado. Não há necessidade de ajustes intermediários camada a camada

Transfer Learning

Definição de *Transfer Learning*

%

- Também conhecida como *Domain Adaptation*
- O aprendizado em um cenário com distribuição P_1 é usado para melhorar a generalização em outra distribuição P_2
- Generaliza *Greedy Unsupervised Pretraining*
 - Transferimos representações entre tarefas não supervisionada e supervisionada
- Classificação Visual



Figura 6: Exemplo *Transfer Learning* [4].

- Temos mais dados na distribuição amostrada de cães e gatos
- A partir desse aprendizado, podemos generalizar rapidamente um problema de classificação entre formigas e vespas
- Categorias visuais compartilham *features* tais como bordas, efeitos de mudanças geométricas e alterações na iluminação

Vantagens - *Transfer Learning*

- Treinar um modelo de aprendizado profundo requer muitos **dados** e, sobretudo, **muito tempo**.
- *Transfer Learning* pode ser usado para aplicar pesos pré-treinados em grandes conjuntos de dados que levam dias ou semanas para treinar, e aproveitá-los em cenários que utilizam conjunto de dados menores.

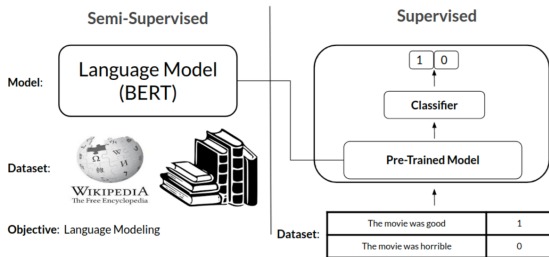


Figura 7: *Transfer Learning* em NLP [4].

Semântica compartilhada - *Input*

- *Transfer Learning*, *Multi-task Learning* e *Domain Adaptation* são alcançadas por meio de Aprendizagem por Representação, onde existem características que são úteis para diferentes tarefas

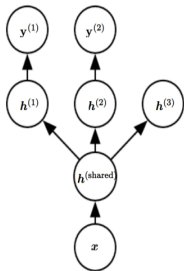


Figura 8: Semântica compartilhada - *Input* [1].

- As tarefas compartilham uma entrada comum mas envolvem diferentes variáveis aleatórias *target*
- Parâmetros específicos da tarefa (pesos) para $h^{(1)}$ e $h^{(2)}$ podem ser aprendidos em cima daqueles que produzem $h^{(shared)}$
- Parâmetros específicos da tarefa (pesos) para $h^{(1)}$ e $h^{(2)}$ podem ser aprendidos em cima daqueles que produzem $h^{(shared)}$. No contexto Não Supervisionado alguns fatores de nível superior $h^{(3)}$ não estão associados a nenhuma das tarefas

Semântica compartilhada - *Output*

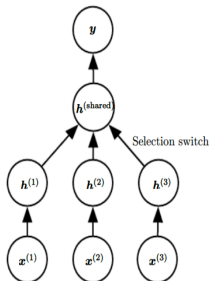


Figura 9: Semântica compartilhada - *Output* [1].

- Em alguns casos, diferentes tarefas compartilham a semântica de saída, como no ambiente de reconhecimento de fala
- Nesse caso, há a necessidade de produzir sentenças válidas na saída
- Camadas anteriores perto da entrada precisam reconhecer versões muito diferentes de fonemas de entrada, dependendo da pessoa que fala

Formas de implementar *Transfer Learning*




Training size	Illustration	Explanation
Small		Freezes all layers, trains weights on softmax
Medium		Freezes most layers, trains weights on last layers and softmax
Large		Trains weights on layers and softmax by initializing weights on pre-trained ones

Figura 10: Formas de implementação em *Transfer Learning* [5].

Domain Adaptation

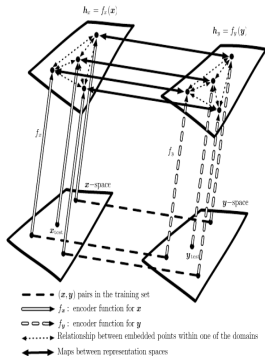
- Relacionado à *Transfer Learning*.
- O mapeamento ótimo de entrada/saída permanece o mesmo entre cada *Setting*. No entanto, a distribuição de entrada é ligeiramente diferente
- Exemplo: Em um cenário de Análise de Sentimentos, podemos nos mover do domínio de mídia (livros/música/vídeos) para o domínio de eletrônicos de consumo (TV/smartphones)
- Neste caso, temos que determinar se o comentário é positivo ou negativo, usando o preditor de sentimento treinado em avaliações de clientes sobre mídia (livros/música/vídeos) para predição de comentários sobre eletrônicos de consumo (TV/smartphones)

Transfer Learning – Formas

1. Aprendizagem *One-Shot*: apenas um exemplo de amostra *labeled* da tarefa de transferência é fornecido
 - Possível porque a representação aprendida separa as classes durante o Estágio 1
 - Durante o Estágio 2, apenas um exemplo *labeled* é necessário para inferir o *label* dos exemplos de teste que se agrupam em torno do mesmo ponto
 - Funciona na medida em que os fatores de variação correspondentes às invariâncias foram separados de outros fatores no espaço de representação aprendido
2. Aprendizagem *Zero-Shot*: nenhuma amostra *labeled* é fornecida para a tarefa de transferência
 - Um aluno lê uma grande coleção de textos e então resolve problemas de reconhecimento de objetos. Tendo lido que um gato tem quatro pernas e orelhas pontudas, o aluno supõe que uma imagem é um gato sem nunca ter visto um gato antes

Transfer Learning - Zero-Shot

- Exemplos *labeled/unlabeled* de \mathbf{x} permitem aprender uma função de representação f_x
- Da mesma forma, com exemplos de \mathbf{y} podemos aprender f_y



- Cada aplicação de f_x e f_y aparece como setas para cima
- Distâncias no espaço h_x e h_y fornecem uma métrica de similaridade
- Imagem \mathbf{x}_{test} da imagem é associado com a palavra \mathbf{y}_{test} mesmo que nenhuma imagem daquela palavra tenha sido apresentada

Figura 11: Transfer Learning habilitando Zero-Shot [1].

***Disentangling* Semi-Supervisionado de Fatores Causais**

Representação Causal

- Uma hipótese de uma representação ideal \mathbf{h} é que ela é uma representação causal
 1. A representação corresponde às causas subjacentes de um conjunto de dados observados \mathbf{x} .
 2. Se um fator causal específico muda, apenas as dimensões correspondentes da representação \mathbf{h} deveriam se modificar.
- Em uma representação causal, cada direção no *feature space* deve corresponder a uma causa distinta. Por exemplo, em um modelo de imagem, uma direção pode corresponder à iluminação e outra ao ângulo da câmera. Dessa forma, a alteração de um fator causal não interfere em outros.
- *Disentangling* os fatores causais significa que cada dimensão ou direção no *feature space* corresponde a uma causa específica e independente.
- Objetivos da Aprendizagem por Representação
 1. Gerar uma representação que seja fácil de modelar, como representações esparsas
 2. Gerar uma representação que separe fatores causais, a qual pode não ser fácil de modelar

Aprendizado Semi Supervisionado

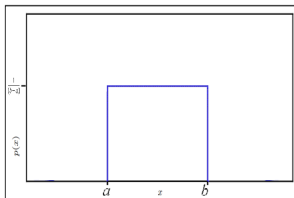


Figura 12: Distribuição $p(\mathbf{x})$ uniforme [4].

- $p(\mathbf{x})$ é uniforme e queremos aprender $f(\mathbf{x}) = \mathbb{E}[\mathbf{y}|\mathbf{x}]$
- Claramente observar o conjunto de treinamento de valores \mathbf{x} sozinho não nos dá nenhuma informação sobre $p(\mathbf{y}|\mathbf{x})$

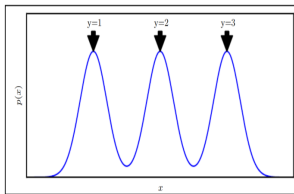


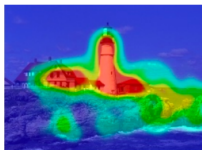
Figura 13: Distribuição $p(\mathbf{x})$ gaussiana [4].

- A modelagem de $p(\mathbf{x})$ revela $p(\mathbf{y}|\mathbf{x})$, onde um único exemplo *labeled* por classe é suficiente para aprender $p(\mathbf{y}|\mathbf{x})$
- Neste caso $p(\mathbf{y}|\mathbf{x})$ é uma Gaussiana univariada para $\mathbf{y} = 1, 2, 3$

Deteção de Saliência



Question: What have you seen?



Answer 1: Lighthouse

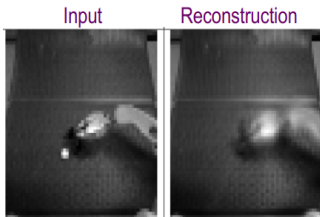
Answer 2: Lighthouse and Houses

Answer 3: Lighthouse, Houses and Rocks

Figura 14: Deteção de Saliência *Deep Learning* [4].

Falha na Detecção de Saliência

- *Autoencoder* treinado com MSE para uma tarefa de robótica falha ao reconstruir uma bola de pingue-pongue



- A existência da bola de pingue-pongue e todas as suas coordenadas espaciais são fatores causais subjacentes importantes que geram a imagem e são relevantes para a tarefa de robótica

Figura 15: Falha na Detecção de Saliência [1].

- O *Autoencoder* tem capacidade e treinamento limitados. MSE não identificou a bola como saliente o suficiente
- Talvez o mesmo robô tivesse sucesso com objetos maiores, como bolas de beisebol que são mais salientes de acordo com MSE

Outras definições de saliência

- Se um grupo de pixels segue um padrão altamente reconhecível, mesmo que esse padrão não envolva brilho ou escuridão extremos, então esse padrão pode ser considerado saliente
- Uma maneira de implementar tal definição de saliência é chamada de *Generative Adversarial Network* (GANs)
 - Um modelo generativo (rede G): é treinado para enganar um classificador *feedforward* gera imagens a partir do ruído
 - Um modelo discriminativo (rede D): um classificador *feedforward* que tenta reconhecer amostras de G como falsas e amostras do conjunto de treinamento como reais

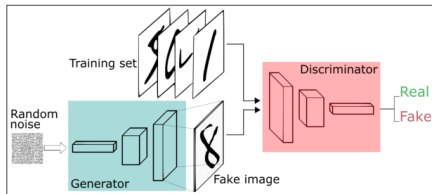


Figura 16: *Generative Adversarial Network* (GAN) [4].

MSE vs GAN

- GANs podem determinar saliência. Qualquer padrão estruturado que a rede *feedforward* (rede D) pode reconhecer é altamente saliente
 - GANs aprendem como determinar o que é saliente
- Modelos treinados para gerar cabeças humanas negligenciam a geração de orelhas quando treinados com MSE
- Mas geram orelhas quando treinados com GANs, uma vez que as orelhas não são especialmente brilhantes ou escuras em comparação com a pele ao redor
- Mas sua forma altamente reconhecível e posição consistente significa que a rede *feedforward* pode facilmente aprender a detectá-las

Redes generativas preditivas

- Os modelos foram treinados para prever a aparência de um padrão 3-D em um ângulo de visão

Ground Truth: Correct image that network should emit

MSE: Network trained with MSE alone. Considers ears to be not salient to learn to generate them

Adversarial: Trained with MSE and adversarial loss. Ears are salient since they follow a predictable pattern

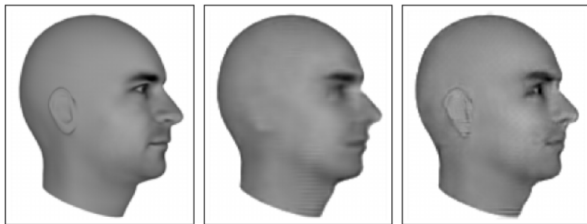


Figura 17: Modelo 3D de uma cabeça humana em um ângulo de visão específico [1].

Representação Distribuída

Representação Distribuída de Conceitos

- Representações distribuídas de conceitos são representações compostas de muitos elementos que podem ser definidos separadamente uns dos outros
- São uma das ferramentas mais importantes para aprendizado por Representação
- São poderosas porque podem usar n recursos com k valores cada para descrever k^n conceitos diferentes
 - Com $k = 2$, 2^n conceitos

Representação Distribuída – Similaridade

- Conceito que distingue distribuído vs simbólico
 - Em simbólico: gato e cachorro estão tão distantes um do outro quanto quaisquer outros dois símbolos
 - Em distribuído: generalização devido a atributos compartilhados entre conceitos. Coisas sobre gatos generalizam para gatos.
 - Entradas “has_fur” ou “no_of_legs” que têm o mesmo valor
- Representações distribuídas induzem um rico espaço de similaridade
 - Conceitos semanticamente próximos são próximos em distância
 - Propriedade ausente em representações puramente simbólicas

Ganhos exponenciais da profundidade

Fundamentos teóricos

- *Multilayer perceptrons* (MLP) são aproximadores universais
 - Podem aproximar a maioria das funções, dadas unidades ocultas suficientes até qualquer tolerância diferente de zero
- *Deep Networks* são mais eficientes que *Shallow Networks* na maioria dos casos. Em determinadas aplicações [6], *Shallow Networks* apresentam vantagens em relação às *Deep Networks*

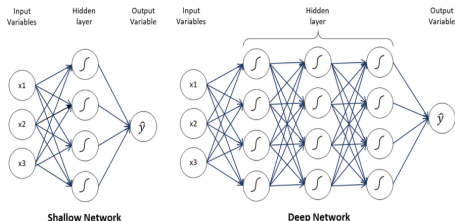


Figura 18: Estruturas de *Shallow* e *Deep Neural Network* [7].

- Em muitos casos, o número de unidades ocultas necessário pelo modelo *Shallow* é exponencial em n
- MLP: As funções são representadas por redes profundas menores em comparação com *Shallow Networks*
- O número de regiões lineares com d entradas, profundidade l e n unidades por camada oculta é exponencial em l
- A redução do tamanho do modelo leva à eficiência estatística

Eficiência Redes *Deep*

- Maior Capacidade de Generalização
 - Redes *Deep* podem ter uma generalização mais eficiente em problemas complexos, desde que treinadas com regularização adequada e quantidade suficiente de dados. A habilidade de aprender padrões multi-nível significa que redes profundas tendem a capturar as propriedades essenciais dos dados em vez de memorizar detalhes específicos (overfitting).
 - Redes *Shallow*, ao contrário, podem ficar limitadas a captar características superficiais dos dados e, muitas vezes, precisam de mais unidades para compensar sua incapacidade de criar representações hierárquicas.
- Teoria da Aproximação Universal em Redes *Deep*
 - indica que uma Rede *Shallow* com uma camada oculta suficientemente larga pode aproximar qualquer função contínua. No entanto, para realizar aproximações práticas em problemas reais de alta complexidade, isso exigiria um número excessivo de neurônios para alcançar a mesma precisão que uma rede profunda.

Pistas para Descobrir Causas Subjacentes

Representação Ideal

- O que torna uma representação melhor do que outra?
 - A representação ideal *disentangle* os fatores causais subjacentes de variação que geraram os dados, especialmente aqueles relevantes para a aplicação
 - A maioria das estratégias introduz pistas que ajudam o aprendizado a encontrar os fatores subjacentes de variação
- O aprendizado supervisionado fornece uma pista forte para encontrar os fatores de variação
 - O *label y* apresentado com cada *x* especifica um fator de variação diretamente
- Para fazer uso de dados *unlabeled*, o aprendizado de representação faz uso de outras dicas menos diretas sobre fatores subjacentes
- Objetivo do *Deep Learning*: encontrar estratégias de regularização aplicáveis a muitas tarefas de IA que as pessoas resolvem

Estratégias Genéricas para Regularização

- Incentivar algoritmos de aprendizagem a descobrirem características que correspondam a fatores de variação
 - *Smoothness*
 - Linearidade
 - Múltiplos fatores explicativos
 - Fatores causais
 - Profundidade
 - Fatores compartilhados entre tarefas
 - *Manifolds*
 - *Natural Clustering*
 - Coerência temporal/espacial
 - Esparsidade
 - Simplicidade de dependências de fatores

Referências Bibliográficas I

- [1] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
<http://www.deeplearningbook.org>.
- [2] S. Ahmadian and A. Khanteymoori, “Training back propagation neural networks using asexual reproduction optimization,” 2015.
- [3] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, “Why does unsupervised pre-training help deep learning?,” *Journal of Machine Learning Research*, vol. 11, pp. 625–660, 02 2010.
- [4] S. N. Srihari, “Lecture notes cse 676 deep learning,” 2020.
- [5] S. Amidi, “Tips and tricks cheatsheet cs 230 - deep learning,” 2020.

Referências Bibliográficas II

- [6] Y. Meir, O. Tevet, Y. Tzach, S. Hodassman, R. Gross, and I. Kanter, “Efficient shallow learning as an alternative to deep learning,” 2022.
- [7] B. Abediniangerabi, A. Makhmalbaf, and M. Shahandashti, “Deep learning for estimating energy savings of early-stage facade design decisions,” *Energy and AI*, vol. 5, 05 2021.