

Como implementar/testar métodos e modelos (redes neurais)



Suponha que você está implementado uma Regressão Linear com regularização para o problema de estimação do módulo da tensão num determinado ponto de um sistema elétrico complexo:

$$J(\vec{w}, b) = \frac{1}{2m} \sum_{i=1}^m \left(f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)} \right)^2 + \frac{\lambda}{2m} \sum_{j=1}^n w_j^2$$

Você percebeu que as previsões feitas pelo modelo atual são inaceitáveis. **O que fazer?**
Podemos tentar o seguinte:

- Conseguir mais dados → (em caso de sobrestimação)

Suponha que você está implementado uma Regressão Linear com regularização para o problema de estimação do módulo da tensão num determinado ponto de um sistema elétrico complexo:

$$J(\vec{w}, b) = \frac{1}{2m} \sum_{i=1}^m \left(f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)} \right)^2 + \frac{\lambda}{2m} \sum_{j=1}^n w_j^2$$

Você percebeu que as previsões feitas pelo modelo atual são inaceitáveis. **O que fazer?**

Podemos tentar o seguinte:

- Conseguir mais dados → (em caso de sobrestimação)
- Excluir características (testar modelo mais simples) → (em caso de sobrestimação)

Suponha que você está implementado uma Regressão Linear com regularização para o problema de estimação do módulo da tensão num determinado ponto de um sistema elétrico complexo:

$$J(\vec{w}, b) = \frac{1}{2m} \sum_{i=1}^m \left(f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)} \right)^2 + \frac{\lambda}{2m} \sum_{j=1}^n w_j^2$$

Você percebeu que as previsões feitas pelo modelo atual são inaceitáveis. **O que fazer?**

Podemos tentar o seguinte:

- Conseguir mais dados → (em caso de sobrestimação)
- Excluir características (testar modelo mais simples) → (em caso de sobrestimação)
- Incluir características (testar modelo mais complexo) → (em caso de subestimação)

Suponha que você está implementado uma Regressão Linear com regularização para o problema de estimação do módulo da tensão num determinado ponto de um sistema elétrico complexo:

$$J(\vec{w}, b) = \frac{1}{2m} \sum_{i=1}^m \left(f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)} \right)^2 + \frac{\lambda}{2m} \sum_{j=1}^n w_j^2$$

Você percebeu que as previsões feitas pelo modelo atual são inaceitáveis. **O que fazer?**

Podemos tentar o seguinte:

- Conseguir mais dados → (em caso de sobrestimação)
- Excluir características (testar modelo mais simples) → (em caso de sobrestimação)
- Incluir características (testar modelo mais complexo) → (em caso de subestimação)
- Adicionar características polinomiais ao modelo $(x_1^2, x_2^2, x_1 x_2, \dots)$ → (em caso de subestimação)

Suponha que você está implementado uma Regressão Linear com regularização para o problema de estimação do módulo da tensão num determinado ponto de um sistema elétrico complexo:

$$J(\vec{w}, b) = \frac{1}{2m} \sum_{i=1}^m \left(f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)} \right)^2 + \frac{\lambda}{2m} \sum_{j=1}^n w_j^2$$

Você percebeu que as previsões feitas pelo modelo atual são inaceitáveis. **O que fazer?**

Podemos tentar o seguinte:

- Conseguir mais dados → (em caso de sobrestimação)
- Excluir características (testar modelo mais simples) → (em caso de sobrestimação)
- Incluir características (testar modelo mais complexo) → (em caso de subestimação)
- Adicionar características polinomiais ao modelo $(x_1^2, x_2^2, x_1 x_2, \dots)$ → (em caso de subestimação)
- Aumentar λ → (em caso de sobrestimação)

Suponha que você está implementado uma Regressão Linear com regularização para o problema de estimação do módulo da tensão num determinado ponto de um sistema elétrico complexo:

$$J(\vec{w}, b) = \frac{1}{2m} \sum_{i=1}^m \left(f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)} \right)^2 + \frac{\lambda}{2m} \sum_{j=1}^n w_j^2$$

Você percebeu que as previsões feitas pelo modelo atual são inaceitáveis. **O que fazer?**

Podemos tentar o seguinte:

- Conseguir mais dados → (em caso de sobrestimação)
- Excluir características (testar modelo mais simples) → (em caso de sobrestimação)
- Incluir características (testar modelo mais complexo) → (em caso de subestimação)
- Adicionar características polinomiais ao modelo $(x_1^2, x_2^2, x_1 x_2, \dots)$ → (em caso de subestimação)
- Aumentar λ → (em caso de sobrestimação)
- Reduzir λ → (em caso de subestimação)

Suponha que você está implementado uma Regressão Linear com regularização para o problema de estimação do módulo da tensão num determinado ponto de um sistema elétrico complexo:

$$J(\vec{w}, b) = \frac{1}{2m} \sum_{i=1}^m \left(f_{\vec{w}, b}(\vec{x}^{(i)}) - y^{(i)} \right)^2 + \frac{\lambda}{2m} \sum_{j=1}^n w_j^2$$

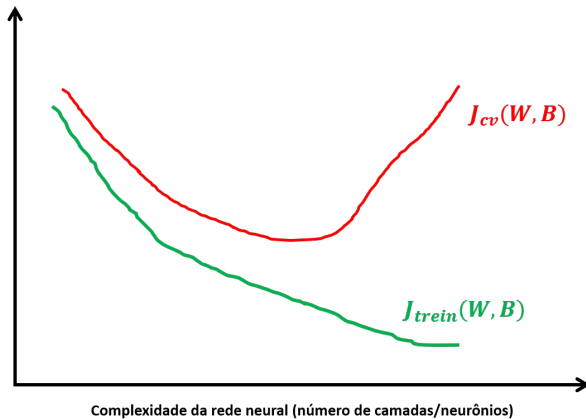
Você percebeu que as previsões feitas pelo modelo atual são inaceitáveis. **O que fazer?**

Podemos tentar o seguinte:

- Conseguir mais dados → (em caso de sobrestimação)
- Excluir características (testar modelo mais simples) → (em caso de sobrestimação)
- Incluir características (testar modelo mais complexo) → (em caso de subestimação)
- Adicionar características polinomiais ao modelo $(x_1^2, x_2^2, x_1 x_2, \dots)$ → (em caso de subestimação)
- Aumentar λ → (em caso de sobrestimação)
- Reduzir λ → (em caso de subestimação)

Pergunta: Essas mesmas considerações se aplicam às **redes neurais**?

Underfitting e overfitting usando J_{train} e J_{cv} (redes neurais)



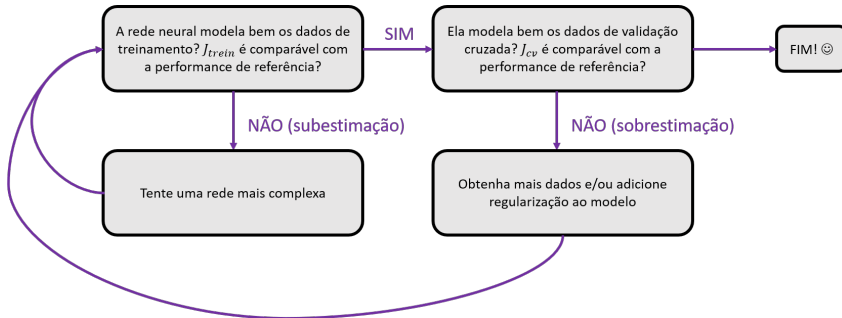
Pergunta

Onde queremos estar nessa figura?

Regra prática:

Teste desde modelos mais simples até modelos altamente complexos. Selecione aquele que resulta no menor J_{cv} .

Um passo-a-passo muito útil



Dica 1:

Desde que você escolha um valor adequado para o parâmetro de regularização λ , é possível usar uma rede neural bastante complexa (muitas camadas e neurônios) sem ter o problema de sobrestimação. **OBS:** Veremos no código dessa aula como implementamos regularização ao treinar uma rede neural (para facilitar, escolheremos um λ igual para todas as camadas).

Dica 1:

Desde que você escolha um valor adequado para o parâmetro de regularização λ , é possível usar uma rede neural bastante complexa (muitas camadas e neurônios) sem ter o problema de sobrestimação. **OBS:** Veremos no código dessa aula como implementamos regularização ao treinar uma rede neural (para facilitar, escolheremos um λ igual para todas as camadas).

Dica 2:

Na dúvida, teste uma rede neural mais complexa e faça uma busca pelo valor de λ ótimo para aquela estrutura. Isso simplifica muito o processo de escolha da arquitetura de uma rede. Os resultados podem ser muito melhores do que aqueles obtidos por meio de uma rede simples. Porém, um dos efeitos colaterais disso será o aumento no tempo computacional gasto para treinar o modelo.

De olho no código!

De olho no código!

Iremos agora verificar como avaliar a qualidade de uma rede neural na prática.

Acesse o Python Notebook usando o QR code ou o link abaixo:



https://colab.research.google.com/github/xaximpvp2/master/blob/main/codigo_aula21_Avaliando_a_performance_de_uma_rede_neural.ipynb

Parte 1

Rode todo o código. Certifique-se de que você o compreendeu.

Parte 2

- 1 Refaça todas as análises presentes no código usando a **taxa de acerto** ao invés da **taxa de erro**.