Regularização para redução da sobreestimação (*overfitting*)

Onde estamos e para onde vamos

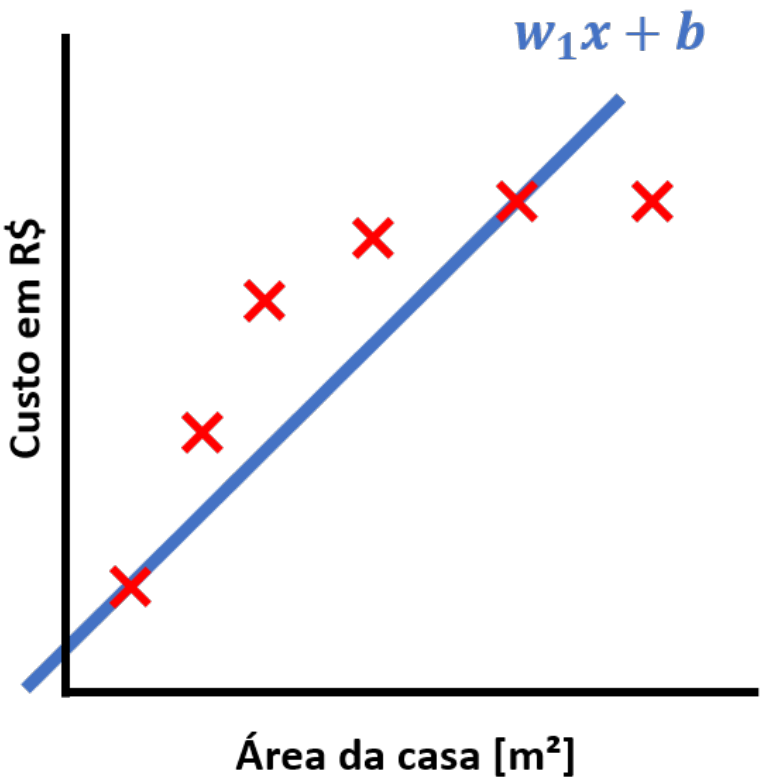
Na aula anterior, implementamos o algoritmo de **Regressão Logística**.

Nesta aula, vamos aprender sobre **sobreestimação**, também conhecida pelo termo *overfitting*, sendo esse um problema comum que nosso modelo pode apresentar em algumas situações.

Pergunta:

Mas afinal, o que é *overfitting*?

Um exemplo vindo da Regressão



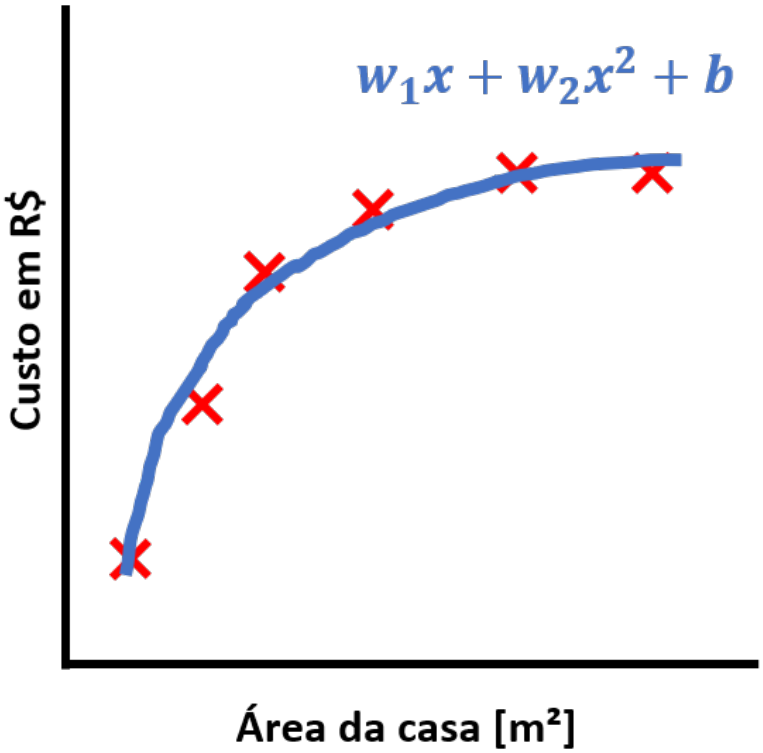
Perguntas

1. O modelo acima se ajusta bem aos dados?
2. O modelo subestima ou sobreestima os dados?

Termos:

underfit = high bias $\rightarrow$ O modelo não é capaz de explicar o comportamento presente nos dados.

Um exemplo vindo da Regressão



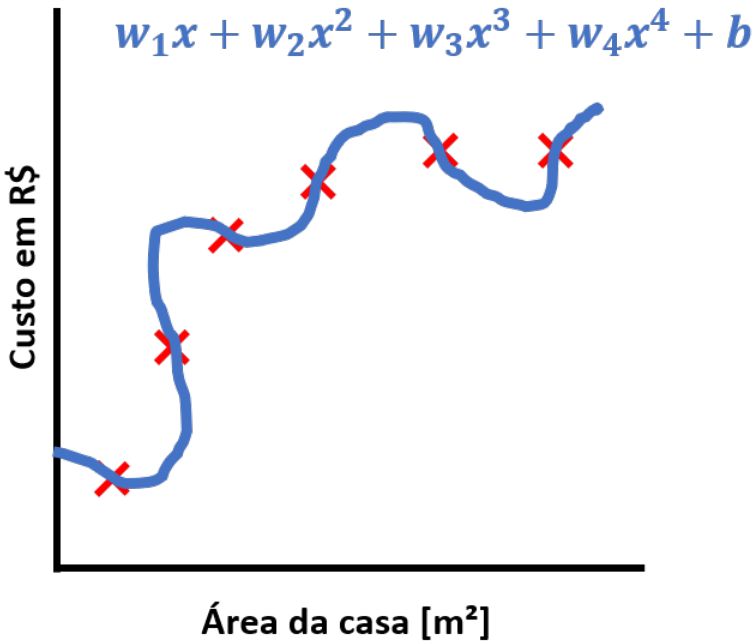
Perguntas

1. Esse segundo modelo se ajusta bem aos dados?
2. O modelo subestima ou sobreestima os dados?

Termos:

Generalização → é a capacidade que um modelo tem (ou não) de performar bem para dados não usados durante seu treinamento.

Um exemplo vindo da Regressão



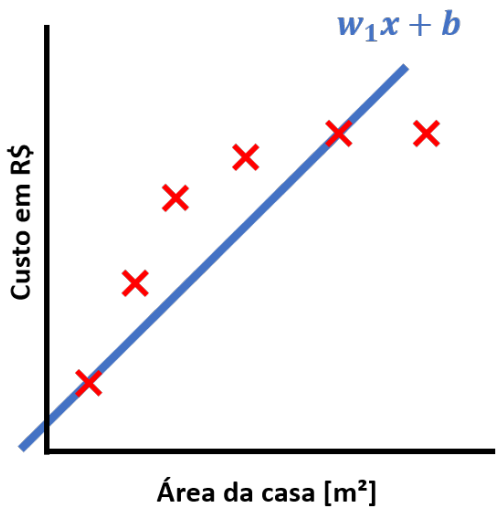
Perguntas

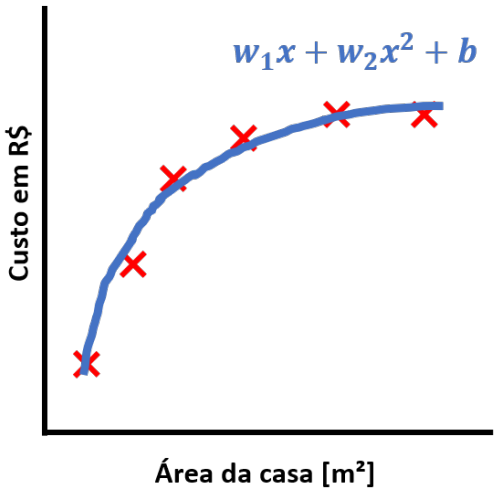
1. Esse terceiro modelo se ajusta perfeitamente aos dados de treinamento?
2. Qual seria o valor da função custo $J(\vec{w}, b)$ para esse caso?
3. Qual é o problema com esse modelo então?
4. O modelo subestima ou sobreestima os dados?

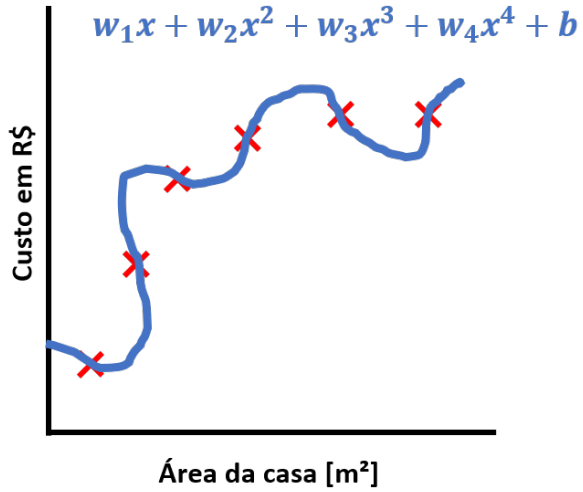
Termos:

overfit = **high variance** $\rightarrow$ O modelo se ajustou mais do que deveria aos dados que lhe foram mostrados.

Um exemplo vindo da Regressão







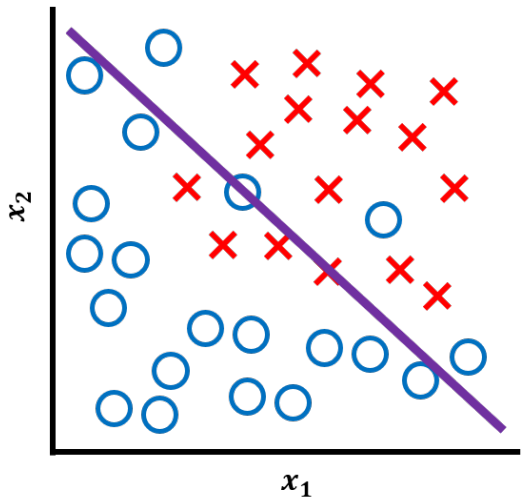
Pergunta

Qual dos três modelos acima você escolheria?

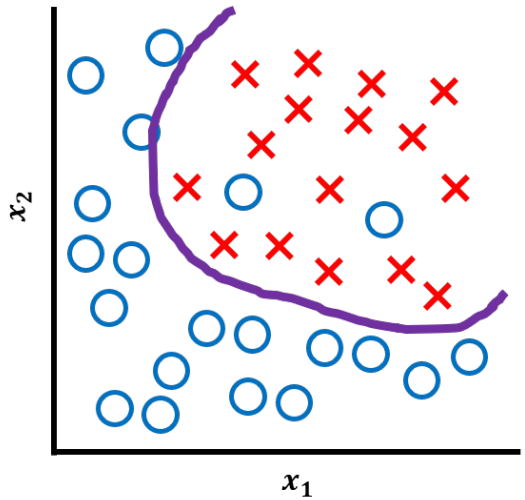
Conclusões

- Modelos muito simples, com poucos parâmetros, podem não ser suficientes para explicar o comportamento presente nos dados
- Por outro lado, modelos muito complexos, com um excesso de parâmetros, podem explicar perfeitamente bem os dados de treinamento, mas não generalizar bem para novos dados

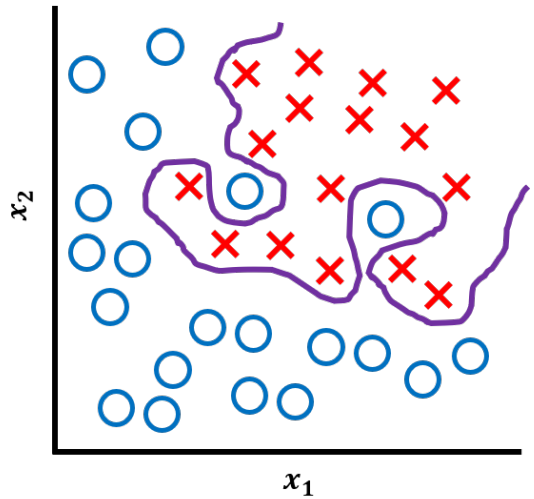
As mesmas conclusões se aplicam para problemas de **classificação**



$$z = w_1x_1 + w_2x_2 + b$$
$$f_{\vec{w},b}(\vec{x}) = g(z)$$



$$z = w_1x_1 + w_2x_2 + w_3x_1^2 + w_4x_2^2 + w_5x_1x_2 + b$$

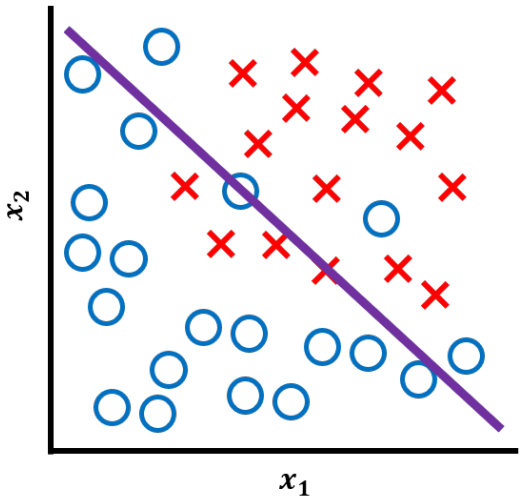


$$z = w_1x_1 + w_2x_2 + w_3x_1^2x_2 + w_4x_1^2x_2^2 + w_5x_1^2x_2^3 + w_6x_1^3x_2 + \cdots + b$$

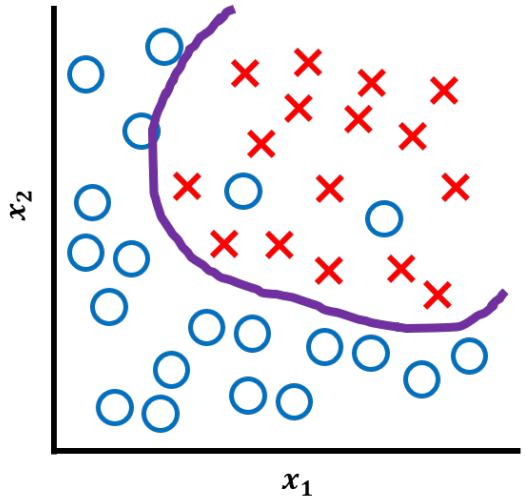
Perguntas

1. Qual modelo subestima os dados?
2. Qual modelo parece super ok?
3. Qual modelo sobreestima os dados?

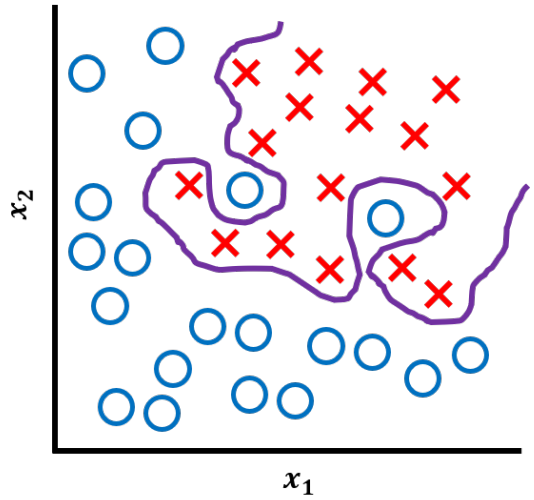
As mesmas conclusões se aplicam para problemas de **classificação**



$$z = w_1x_1 + w_2x_2 + b$$
$$f_{\vec{w},b}(\vec{x}) = g(z)$$



$$z = w_1x_1 + w_2x_2 + w_3x_1^2 + w_4x_2^2 + w_5x_1x_2 + b$$



$$z = w_1x_1 + w_2x_2 + w_3x_1^2x_2 + w_4x_1^2x_2^2 + w_5x_1^2x_2^3 + w_6x_1^3x_2 + \cdots + b$$

Pergunta

Seja x_1: diâmetro do tumor e x_2: idade do paciente.

Qual dos três modelos acima você escolheria para estimar a probabilidade de um novo paciente estar ou não com um tumor maligno?

Our goal when creating a model is to be able to use the model to predict outcomes correctly for **new examples**. A model which does this is said to **generalize** well.

When a model fits the training data well but does not work well with new examples that are not in the training set, this is an example of:

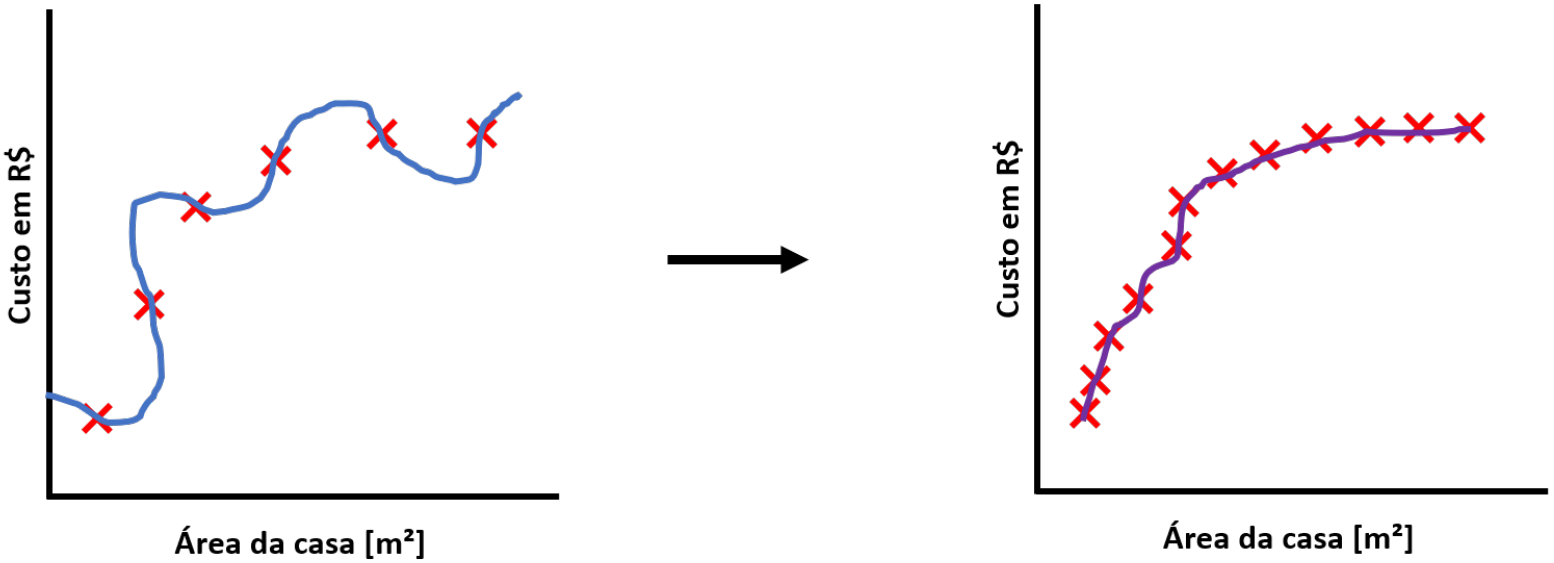
- ◯ Overfitting (high variance)
- ◯ A model that generalizes well (neither high variance nor high bias)
- ◯ Underfitting (high bias)
- ◯ None of the above

Fonte: **Machine Learning Specialization**, *deeplearning.ai*, Stanford Online, Coursera.org.

Como resolver o problema de overfitting?

Opção 1

Colete e utilize mais dados durante o treinamento:



Observação

Infelizmente, coletar mais dados nem sempre é uma opção.

Como resolver o problema de overfitting?

Opção 2

Selecione características que podem ser incluídas ou excluídas

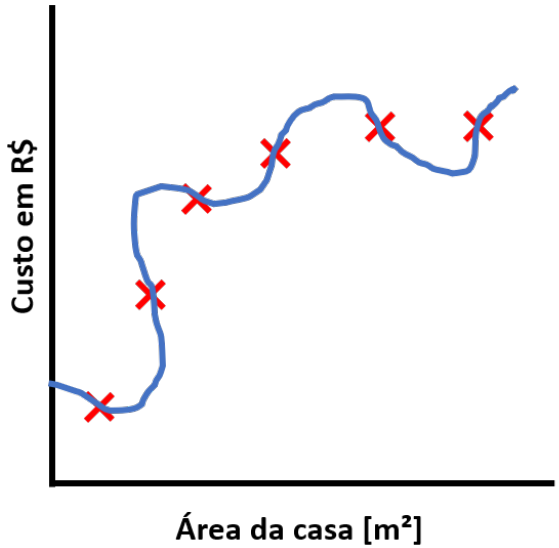
Observações

- Muitas características + poucos dados podem levar à sobreestimação.
- Use a intuição para selecionar: distância até o mercado mais próximo é de fato importante?
- **Desvantagem**: Características relevantes podem ser ignoradas (informação relevante perdida).

Como resolver o problema de overfitting?

Opção 3

Regularização



$$f(x) = 28x - 385x^2 + 39x^3 - 174x^4 + 100$$

Observações

- Em muitos casos, o overfitting ocorre pois alguns parâmetros do modelo assumem valores muito elevados (exemplo: $w_4 = -174$)
- Regularização permite que os parâmetros existam, mas gera uma penalização elevada caso eles sejam excessivamente elevados.
- Geralmente, regularizamos apenas os parâmetros w_j do modelo.
- Regularizar também o parâmetro b geralmente não gera muito impacto.

Como resolver o problema de overfitting?

Opção 1

Coletar mais dados

Opção 2

Selecionar as características

Opção 3

Regularização $\rightarrow$ Estudaremos agora com mais detalhes como implementar!

Applying regularization, increasing the number of training examples, or selecting a subset of the most relevant features are methods for...

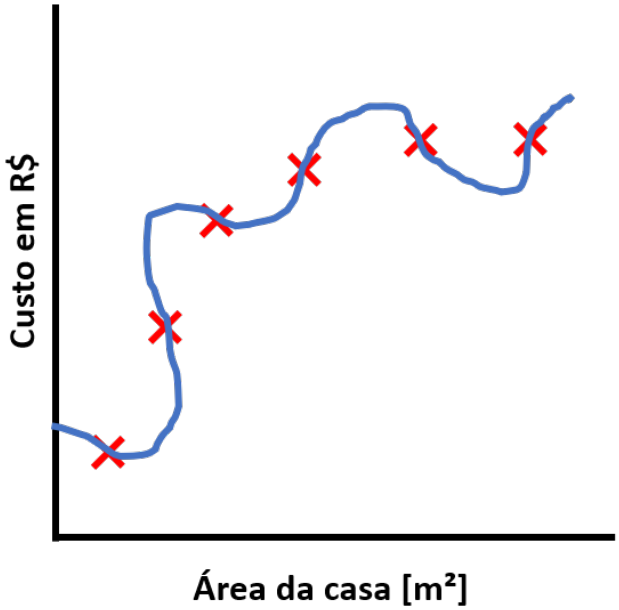
- [x] Addressing overfitting (high variance)
- [ ] Addressing underfitting (high bias)

Fonte: **Machine Learning Specialization**, *deeplearning.ai*, Stanford Online, Coursera.org.

Veremos agora como implementar a Regularização na prática

Implementando a Regularização

No exemplo abaixo, se escolhermos valores excessivamente grandes para w_3 e w_4 podemos ter overfitting.



$$f(x) = w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4 + b$$

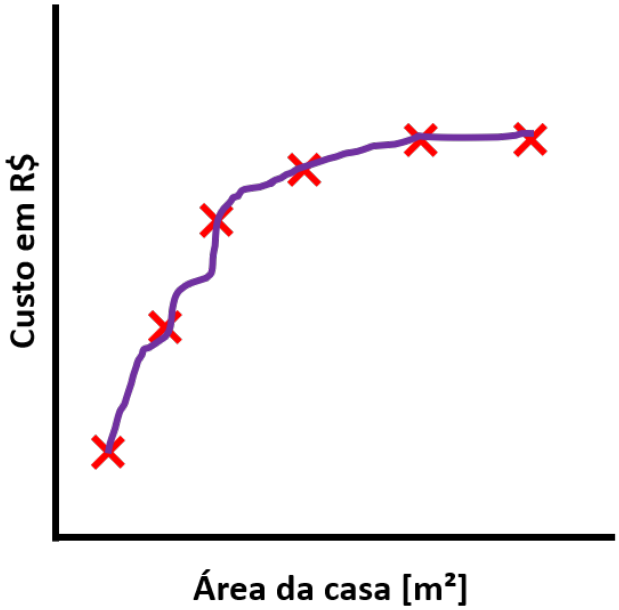
Pergunta:

O que acontece se estimarmos os parâmetros $\vec{w}, b$ por meio da função custo modificada:

$$J(\vec{w}, b) = \frac{1}{2m} \sum_{i=1}^{m} \left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right) - y^{(i)} \right)^2 + 1000 w_3^2 + 1000 w_4^2 \quad ?$$

OBS: Note que estamos penalizando valores elevados para w_3 e w_4 multiplicando ambos por um valor escalar elevado e adicionando esses termos à função custo $J(\vec{w}, b)$.

Implementando a Regularização



$$f(x) = w_1x + w_2x^2 + w_3x^3 + w_4x^4 + b$$

Resposta:

- Os parâmetros w_3 e w_4 serão garantidamente pequenos, 0.001 e 0.002 por exemplo.
- Com isso, a chance de overfitting é drasticamente reduzida.
- Por outro lado, w_3 e w_4 ainda permanecem presentes no modelo, contribuindo para que o modelo explique bem os dados.
- Na prática, penalizamos todos os parâmetros w_j do modelo, para $j = 1, \cdots, n$
- Geralmente isso leva a modelos mais simples, mais suaves e que não sobreestimam os dados.

Implementando a Regularização (caso geral)

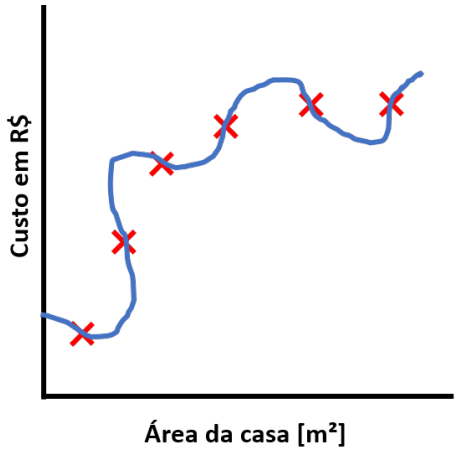
No caso geral, como não sabemos quais características são mais importantes, penalizamos todos os parâmetros w_j, usando a função custo:

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

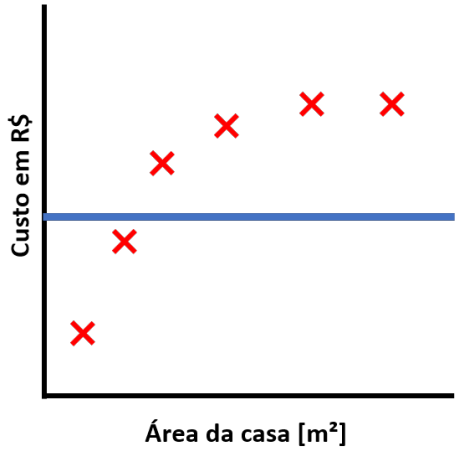
- λ é chamado de **parâmetro de regularização**, e é ≥ 0.
- Ao escolhermos $\lambda = 0$, eliminamos completamente o efeito da regularização.
- Note que, o primeiro termo da função custo busca adequar o modelo aos dados.
- Enquanto o segundo termo busca manter os parâmetros w_j pequenos.

Implementando a Regularização (caso geral)

Extremos:



$$f(x) = w_1x + w_2x^2 + w_3x^3 + w_4x^4 + b$$



$$f(x) = w_1x + w_2x^2 + w_3x^3 + w_4x^4 + b$$

Função custo com regularização

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

Perguntas

- Em qual caso acima foi escolhido $\lambda = 0$?
- Em qual caso acima foi escolhido $\lambda = 10^{10}$?

For a model that includes the regularization parameter λ (lambda), increasing λ will tend to...

- [x] Increases the size of the parameters $w_1, w_2, ...w_n$
- [ ] Decrease the size of parameters $w_1, w_2, ...w_n$.
- [ ] Decrease the size of the parameter b.
- [ ] Increase the size of parameter b.

Fonte: **Machine Learning Specialization**, *deeplearning.ai*, Stanford Online, Coursera.org.

Vamos agora resumir como implementar o método do gradiente com regularização tanto para Regressão Linear como também para Regressão Logística

Apenas lembrando que:

- Regressão Linear $\rightarrow$ Problemas de Regressão (y pode assumir infinitos valores possíveis)
- Regressão Logística $\rightarrow$ Problemas de Classificação (y assume apenas um pequeno conjunto de valores)

Regressão Linear com Regularização (resumo)

Função custo:

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

Método do Gradiente: repetir

$$w_j = w_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right) - y^{(i)} \right) x_j^{(i)} + \frac{\lambda}{m} w_j \right]$$

$$b = b - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{w},b}\left(\vec{x}^{(i)}\right) - y^{(i)} \right) \right]$$

Modelo

$$f_{\vec{w},b}\left(\vec{x}^{(i)}\right) = \vec{w} \cdot \vec{x} + b$$

Tarefa para casa: Deduzir as derivadas.

Regressão Logística com Regularização (resumo)

Função custo:

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

Método do Gradiente: repetir

$$w_j = w_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{w},b} \left(\vec{x}^{(i)} \right) - y^{(i)} \right) x_j^{(i)} + \frac{\lambda}{m} w_j \right]$$

$$b = b - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} \left(f_{\vec{w},b} \left(\vec{x}^{(i)} \right) - y^{(i)} \right) \right]$$

Modelo

$$f_{\vec{w},b}(\vec{x}) = \frac{1}{1 + e^{-(\vec{w} \cdot \vec{x} + b)}}$$

Tarefa para casa: Deduzir as derivadas.

De olho no código!

Buscando consolidar nosso conhecimento acerca de regularização, vamos agora implementar novamente o método de regressão logística fazendo as modificações necessárias.

Nome do arquivo que trabalharemos agora:

codigo - Regressão Logística com Regularização.ipynb

Tarefa para casa: Fica como tarefa para casa implementar a regularização no contexto da Regressão Linear.

Atividade de aula

Parte 1

Rode todo o "codigo - Regressão Logística com Regularização.ipynb" sem fazer qualquer tipo de alteração. Certifique-se de que você o compreendeu.

Parte 2

1. Explique, com as suas próprias palavras, o conceito de overfitting e as possibilidades de resolução desse problema.
2. Explique, com as suas próprias palavras, como implementar a regularização no método do gradiente.
3. Qual foi a taxa de acerto obtida com o modelo treinado? Explique, com as suas próprias palavras, o que significa essa taxa de acerto.
4. Qual seria a taxa de acerto esperada para um modelo com saída 0/1 aleatória?