

# T319 - Introdução ao Aprendizado de Máquina: *Regressão Linear (Parte VI)*



***Inatel***

Felipe Augusto Pereira de Figueiredo  
felipe.figueiredo@inatel.br

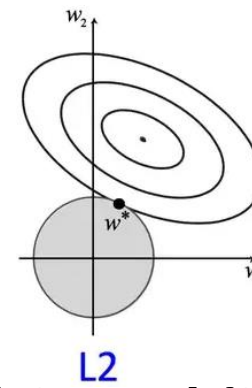
# Recapitulando

- Anteriormente, vimos como escolher o melhor modelo de regressão utilizando as técnicas de validação cruzada: holdout, k-Fold e leave-P-out.
- Escolhemos sempre o modelo menos complexo, mas que generaliza bem.
- Ou seja, escolhemos o modelo que apresenta valores baixos para ambos os erros de treinamento e de validação.
- Uma abordagem alternativa é minimizar conjuntamente o erro e a complexidade da ***função hipótese***.
- Como veremos, esta abordagem combina erro e complexidade em uma única métrica, possibilitando que encontremos a melhor hipótese de uma só vez.
- Portanto, hoje, veremos as seguintes abordagens para se escolher o melhor modelo de regressão:
  - **Regularização**: penaliza ***funções hipótese*** muito complexas, ou seja, muito flexíveis.
  - **Early-stop**: encerra o treinamento de algoritmos iterativos quando o erro de validação for o menor possível.

# Regularização: penalizando a complexidade dos modelos

- **Regularização**: deixar o modelo mais regular, ou seja, menos flexível.
- A ideia por trás da **regularização** é penalizar, explicitamente, **hipóteses** complexas.
- Técnicas de **regularização** podem reduzir o risco de **sobreajuste** do modelo ao conjunto de treinamento, aumentando sua capacidade de **generalização**.
- O **sobreajuste** pode ser evitado incorporando **penalizações** proporcionais à alguma **norma** do **vetor de pesos** ao processo de treinamento.
- As principais técnicas de **regularização** são: *ridge regression*, LASSO e *elastic-net*.

# Ridge Regression



- Ao invés de minimizarmos apenas o **erro quadrático médio**, como fizemos antes, introduzimos um **termo de penalização** proporcional à **norma Euclidiana** (ou seja, a norma L2) do vetor de pesos:

$$\min_{\mathbf{a} \in \mathbb{R}} (\|\mathbf{y} - \Phi \mathbf{a}\|^2 + \lambda \|\mathbf{a}\|_2^2) \text{ com } \|\mathbf{a}\|_2^2 = \sum_{i=1}^K a_i^2$$

Início em 1 e não em 0.

onde  $\lambda \geq 0$  é o **fator de regularização**,  $\Phi$  é a matriz de atributos e  $\mathbf{a}$  é o vetor de pesos.

- Podemos re-escrever o **problema de regularização** como um **problema de otimização** com restrições da seguinte forma

$$\begin{aligned} \min_{\mathbf{a} \in \mathbb{R}} \|\mathbf{y} - \Phi \mathbf{a}\|^2 \\ \text{s. a. } \|\mathbf{a}\|_2^2 \leq c, \end{aligned}$$

- Se  $c$  diminui,  $\|\mathbf{a}\|_2^2$  também diminui até que se  $c \rightarrow 0$ , então  $a_i \rightarrow 0$ .
- Se  $c$  aumenta,  $\|\mathbf{a}\|_2^2$  pode assumir valores maiores até que se  $c \rightarrow \infty$ , então  $a_i \rightarrow \infty$ .

onde  $c$  restringe a magnitude dos pesos e é inversamente proporcional à  $\lambda$ .

- Portanto,  $\lambda$  altera a complexidade (ou seja, a flexibilidade) da função hipótese.
- OBS.:** o peso  $a_0$  não é considerado no cálculo da **norma L2**, pois a **complexidade** se deve à ordem do modelo e  $a_0$  apenas dita o deslocamento em relação ao eixo  $y$ .

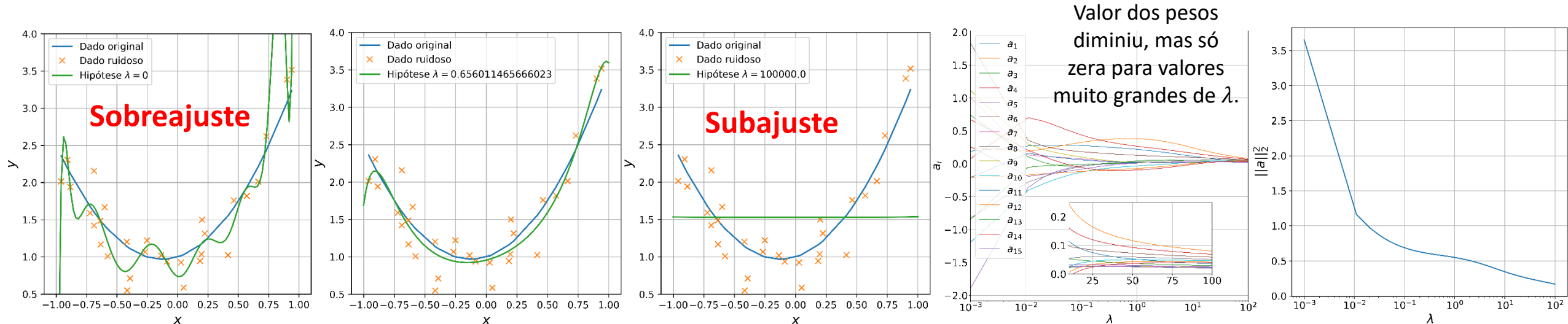
# Ridge Regression

- A equação de erro regularizado,  $\|\mathbf{y} - \Phi\mathbf{a}\|_2^2 + \lambda\|\mathbf{a}\|_2^2$ , continua sendo quadrática com relação aos pesos, e portanto, a superfície de erro continua sendo convexa.
- Desta forma, podemos encontrar uma solução de forma fechada seguindo o mesmo procedimento que usamos para encontrar a **equação normal**:

$$\mathbf{a} = (\Phi^T \Phi + \lambda \mathbf{I}')^{-1} \Phi^T \mathbf{y}, \text{ onde } \mathbf{I}' = \begin{bmatrix} 0 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

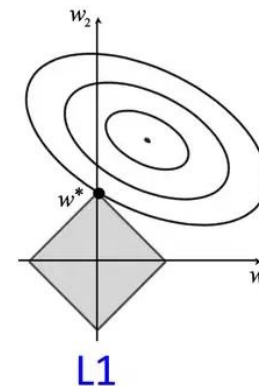
- **OBS.1:** mesmo que a matriz  $\Phi$  não possua ~~posto completo~~ (i.e., matriz singular), a inversa na equação acima sempre existirá por conta da adição do **termo de regularização** à diagonal principal da matriz quadrada  $\Phi^T \Phi$ .
- **OBS.2:** como a **norma L2** é diferenciável, os problemas de aprendizagem usando a regularização de Ridge também podem ser resolvidos iterativamente através do **algoritmo do gradiente descendente**.
- **OBS.3:** o termo de regularização deve ser adicionado apenas à função de erro durante o treinamento. Depois que o modelo é treinado, a avaliação do desempenho do modelo não utiliza a regularização.

# Ridge Regression: Exemplo



- Função hipótese polinomial de grau 15.
- Modelo treinado com 30 amostras geradas a partir de  $y_{\text{noisy}} = 1 + 0.5x + 2x^2 + w$ , onde  $x \sim U(-1,1)$  e  $w \sim N(0,1)$ .
- Com  $\lambda = 0$ , regressão de Ridge se torna uma regressão polinomial sem regularização.
- Conforme  $\lambda$  aumenta, o modelo não se “contorce” tanto e passa a se ajustar aos dados de treinamento.
- Se  $\lambda$  continuar aumentando, todos os pesos acabarão muito próximos de zero e o resultado será uma linha reta que passa pela **média dos dados de treinamento**.
- **O aumento de  $\lambda$  leva a hipóteses menos complexas.** Isso reduz a variância do modelo, mas aumenta seu bias. Ou seja, ele tende a **subajustar**.
- Conforme  $\lambda$  aumenta, os pesos e a norma L2 do vetor de pesos diminuem.
- Utiliza-se técnicas de **validação cruzada** para encontrar o valor ideal de  $\lambda$ . [Exemplo: ridge\\_regression.ipynb](https://github.com/amueller/ridge_regression.ipynb)

# LASSO Regression



- A **regressão LASSO** (*Least Absolute Shrinkage and Selection Operator*) adiciona à função de erro um **termo de penalização** proporcional à **norma L1** do vetor de pesos.

$$\min_{\mathbf{a} \in \mathbb{R}} (\|\mathbf{y} - \Phi \mathbf{a}\|^2 + \lambda \|\mathbf{a}\|_1),$$

onde  $\|\mathbf{a}\|_1 = \sum_{i=1}^K |a_i|$  e  $\lambda \geq 0$  é o **fator de regularização**.

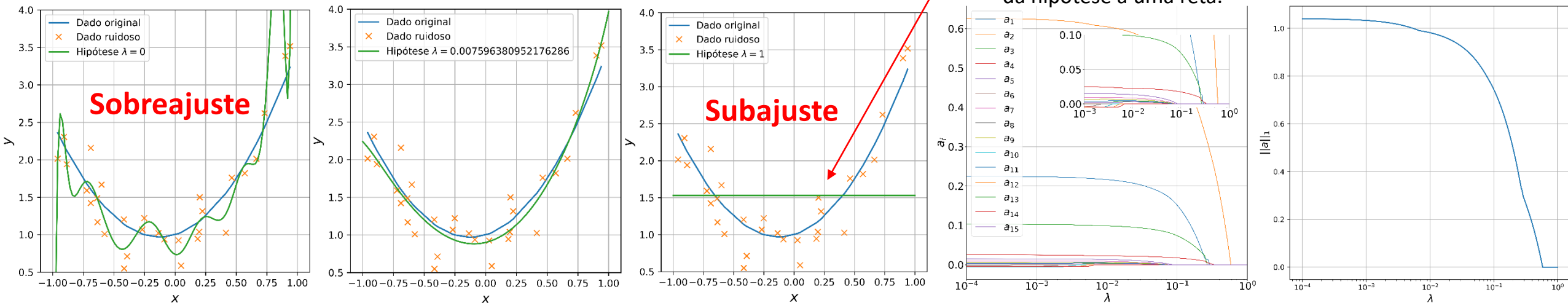
- Podemos re-escrever o **problema de regularização** acima como um **problema de otimização** com restrições da seguinte forma

$$\begin{aligned} \min_{\mathbf{a} \in \mathbb{R}} & \|\mathbf{y} - \Phi \mathbf{a}\|^2 \\ \text{s. a. } & \|\mathbf{a}\|_1 \leq c, \end{aligned}$$

onde  $c$  restringe a magnitude dos pesos e é inversamente proporcional à  $\lambda$ .

**OBS.:**  $a_0$  também não faz parte do cálculo da norma.

# LASSO Regression

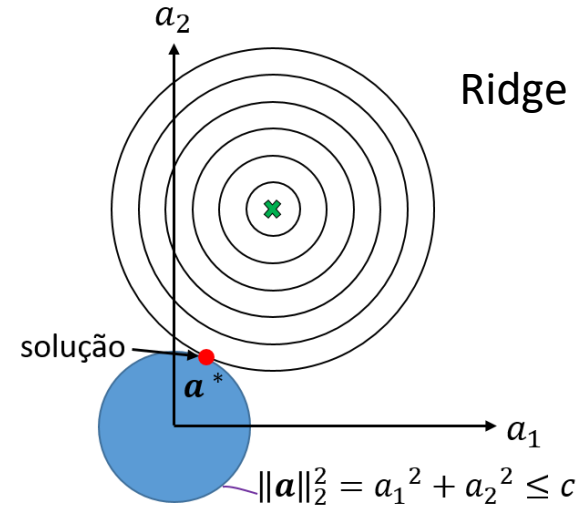
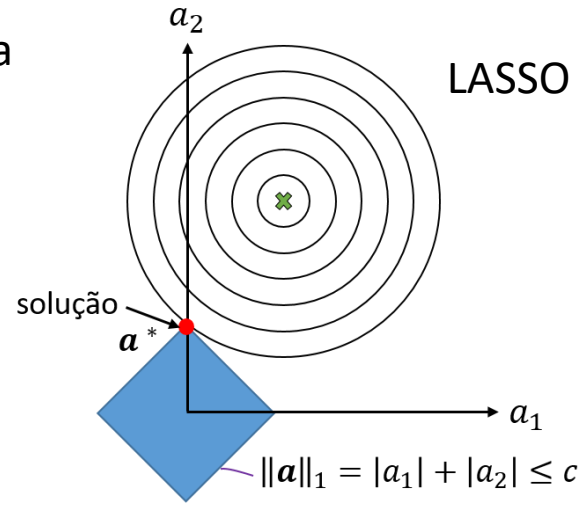


- Mesmas funções geradora e hipótese do exemplo anterior.
- Valores pequenos de  $\lambda$  fazem LASSO se comportar como regressão tradicional e valores muito grandes fazem os pesos serem anulados.
- A regularização com **norma L1** tem como vantagem a produção de **modelos esparsos**.
- Ou seja, vários elementos do vetor de pesos acabam sendo **anulados**, indicando que os atributos correspondentes são irrelevantes.
- Isso sugere a ocorrência implícita de um processo de **seleção automática de atributos**, e leva a **modelos** mais **regulares**, ou seja, **menos complexos**.
- **Desvantagem:** como a **norma L1** não possui derivada no ponto  $a_i = 0, \forall i$ , o problema da minimização não possui solução em forma fechada.
- Utiliza-se técnicas de validação cruzada para encontrar o valor ideal de  $\lambda$ .



# Vantagem do LASSO sobre Ridge

- O quadrado azul representa o conjunto de pontos  $\mathbf{a}$  no espaço de pesos bidimensional que tenham norma L1 menor do que  $c$ .
- A solução deve estar em algum lugar dentro do quadrado.



- O círculo azul representa o conjunto de pontos  $\mathbf{a}$  no espaço de pesos bidimensional que tenham norma L2 menor do que  $c$ .
- A solução deve estar em algum lugar dentro do círculo.

- **Por que a regressão LASSO tem como vantagem a produção de modelos esparsos?**
  - A figura mostra as curvas de nível da função de erro de um problema de regressão linear, bem como as regiões do **espaço de hipóteses** em que as restrições L1 (esquerda) e L2 (direita) são válidas, considerando o caso em que dois pesos ( $a_1$  e  $a_2$ ) estão sujeitos a regularização.
  - A solução para ambos os métodos corresponde ao ponto, dentro da **região de factibilidade** (área em azul), mais próximo do ponto de mínimo da função de erro.
  - É fácil ver que para uma posição arbitrária do mínimo, será comum que um **canto** (ou ponta) do quadrado seja o ponto mais próximo do ponto de mínimo da função de erro.
  - Os **cantos** na **região de factibilidade** da restrição L1 aumenta as chances de alguns pesos assumirem o valor zero.
  - E claro, os **cantos** são os pontos que possuem um valor igual a 0 em alguma das dimensões (i.e., pesos).

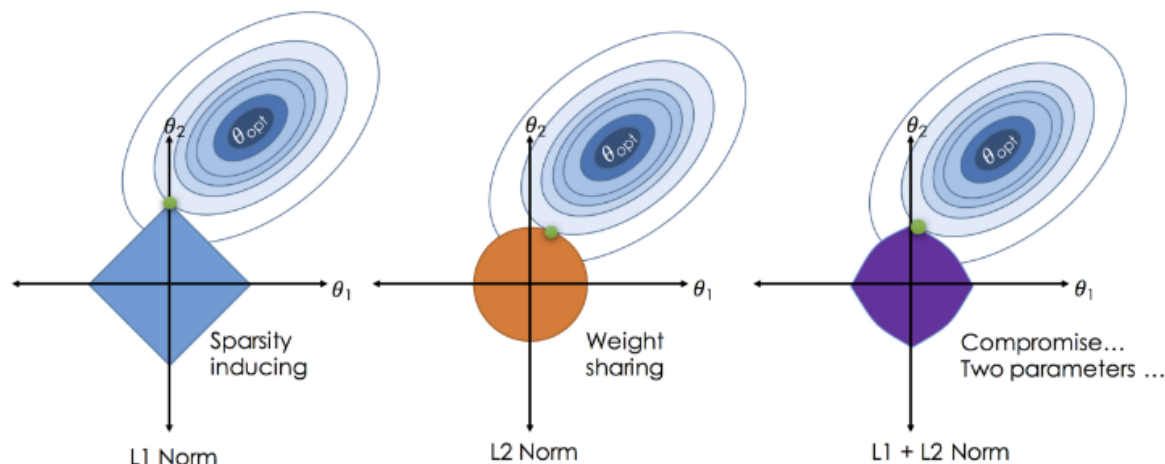
# Elastic-net

- *Elastic-net* é uma solução intermediária entre as regressões Ridge e LASSO.
- Nada mais é do que uma combinação linear entre as penalizações baseadas nas normas L1 e L2 do vetor de pesos.

$$\min_{\mathbf{a} \in \mathbb{R}} (\|\mathbf{y} - \Phi \mathbf{a}\|^2 + \lambda [\kappa \|\mathbf{a}\|_1 + (1 - \kappa) \|\mathbf{a}\|_2^2]),$$

onde  $\kappa \in [0, 1]$  é o termo de mistura ou parâmetro de elasticidade entre as duas normas.

- Quando  $\kappa = 0$ , a Elastic-net é equivalente a Ridge regression, e quando  $\kappa = 1$ , ela é equivalente a Regressão Lasso.
- Utiliza-se técnicas de validação cruzada para encontrar os valores ideais de  $\kappa$  e  $\lambda$ .



O hiperparâmetro  $\kappa$  dita a relação de compromisso entre as duas regularizações.

# Quando utilizar regressão LASSO, Ridge ou Elastic-Net?

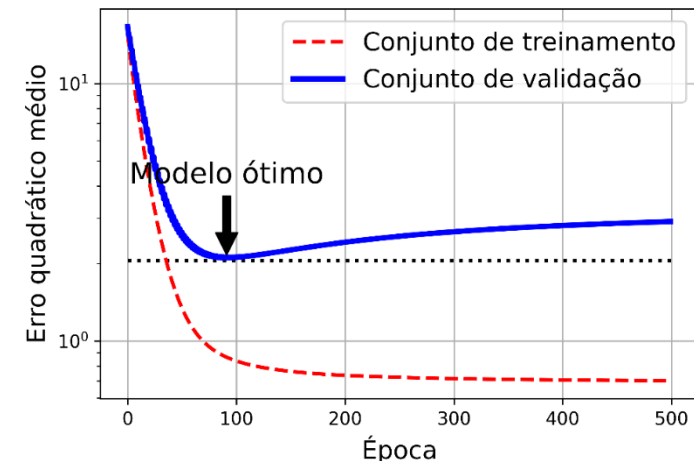
- **Regressão de Ridge:** um bom começo. No entanto, se você suspeitar que apenas alguns atributos são realmente úteis, você deve preferir LASSO ou *Elastic-Net*.
- **Regressão LASSO:** boa para *seleção automática de atributos*. No entanto, se o número de atributos,  $K$ , for maior que o número de exemplos de treinamento,  $N$ , ou quando houverem atributos fortemente correlacionados, deve-se usar a regressão *Elastic-Net*.
- **Elastic-Net:** é mais versátil que as anteriores, pois o parâmetro de elasticidade  $\kappa$  é ajustável. Uma proporção de 50% entre as penalizações L1 e L2 é uma boa escolha inicial para esse parâmetro.

# Early-stop: Parada antecipada

- Uma forma de se **regularizar** algoritmos de **aprendizado iterativo**, como o **gradiente descendente**, é interromper seu treinamento assim que o **erro de validação** comece a crescer sistematicamente.
- Essa abordagem é chamada de **early-stop** e pode ser vista como uma **regularização temporal**.
- Assim como as outras abordagens, ela tem o objetivo de evitar o **sobreajuste** de um modelo.
- Intuitivamente, o algoritmo do **gradiente descendente** tenderá a aprender modelos cada vez mais complexos à medida que o número de épocas aumenta.
- Ao se regularizar no **tempo**, a complexidade do modelo pode ser controlada, melhorando sua **generalização**.
- Mas como saber quando interromper o treinamento? Ou seja, qual é o critério de parada?

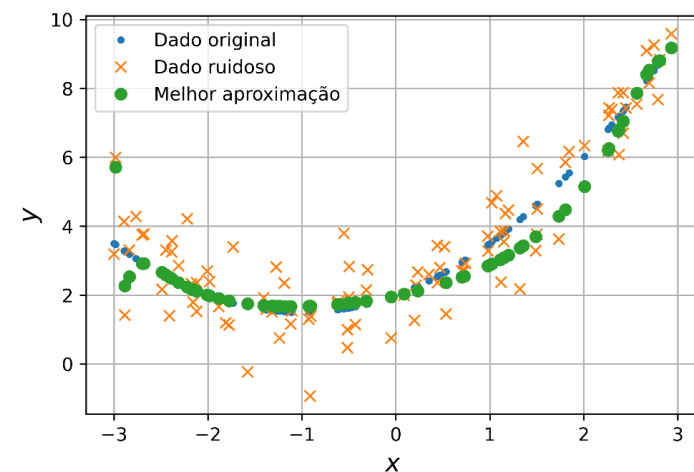
# Early-stop: Exemplo

- Existem duas estratégias para se definir o critério de parada:
  - Interromper o treinamento quando o erro de validação aumenta por ***P*** (paciência) épocas sucessivas.
    - **Problema:** como o erro de validação pode oscilar bastante (e.g., SGD), nem sempre é fácil desenvolver detectores automáticos de mínimos e encerrar o treinamento.
  - Permitir que o treinamento prossiga por um determinado número de épocas, mas sempre armazenando os pesos associados ao ***menor erro de validação***.
- A figura mostra um modelo de regressão polinomial com grau igual a 90 sendo treinado usando o ***gradiente descendente estocástico*** e apenas 100 amostras de treinamento.
- À medida que as épocas passam, o algoritmo aprende e seu erro quadrático médio no conjunto de treinamento diminui, juntamente com o erro de predição no conjunto de validação.
- No entanto, após algumas épocas, o erro de validação para de diminuir e começa a crescer.
- Isso indica que o modelo começou a ***sobreajustar*** aos dados de treinamento.



$$y_{\text{noisy}} = 2 + x + 0.5x^2 + w,$$

onde  $x \sim U(-3,3)$  e  $w \sim N(0,1)$



[Exemplo: early\\_stopv2.ipynb](#)

# Tarefas

- **Quiz:** “*T319 - Quiz - Regressão: Parte VI (1S2021)*” que se encontra no MS Teams.
- **Exercício Prático:** [Projeto Prático](#).
  - Pode ser baixado do MS Teams ou do GitHub.
  - Pode ser respondido através do link acima (na nuvem) ou localmente.
  - [Instruções para resolução e entrega dos laboratórios](#).
  - **Projeto pode ser feito em grupo de no máximo 3 alunos.**
  - **Enviar email com nomes e matrículas dos integrantes do grupo.**
  - **Atentem-se ao prazo de entrega definido na tarefa do MS Teams.**
  - **Entregas fora do prazo não serão aceitas.**
  - **Leiam os enunciados atentamente.**

Obrigado!



