

Machine Learning

Algumas aplicações possíveis

Eduardo Palhares Júnior



24 de outubro de 2019

- 1 Apresentação pessoal
- 2 Definição
- 3 Aprendizagem supervisionada
 - Regressão
 - Função hipótese
 - Função custo
 - Gradiente descendente
 - Regressão linear multivariada
 - Normalização da escala
 - Convergência
 - Regressão polinomial
 - Classificação
 - Função hipótese logística
 - Função custo logística
 - Otimização Avançada
 - Classificação multiclasse
 - Regularização

- 4 Redes neurais
 - Classificação não-linear
 - História
 - Propagação direta
 - Propagação reversa
 - Exemplo 1 - reconhecimento de escrita
 - Exemplo 2 - direção autonoma

- 5 Tecnicas Avançadas
 - Avaliação do problema
 - Treino e validação cruzada
 - Viés e variância
 - Análise de erro
 - Precisão e revocação
 - SVM - máquina de vetores de suporte
 - Exemplo 3 - Classificador de SPAM

Formação acadêmica

- Formação Técnica
 - ETEC Jorge Street - 2005 - Eletrônica
- Graduação
 - UFABC - 2010 - Ciência e Tecnologia
 - UFABC - 2014 - Engenharia Aeroespacial
 - UFABC - 2017 - Licenciatura em Matemática
- Pós-Graduação Lato Sensu
 - UNINOVE - 2011 - Gestão Empresarial (Especialização)
- Pós-Graduação Stricto Sensu
 - UFABC - 2015 - Engenharia Mecânica (Mestrado)
 - USP - em curso - Engenharia Elétrica (Doutorado)

Formação acadêmica

- Área técnica
 - IBGE - 2009 à 2010 - agente de pesquisa
 - UFABC - 2010 até agora - técnico de TI
- Docência
 - UFABC - 2008 à 2016 - monitor (matemática/aeroespacial)
 - UFABC - 2015 - professor voluntário (controle)
 - FATEC - 2016 à 2017 - professor universitário (engenharia)
 - UFABC - 2017 - professor de Excel
 - UFABC - 2018 até agora - professor de extensão (educação financeira)
 - IFSP - 2019 - professor universitário (matemática)
- Área financeira
 - Criptomoedas - 2017 (Swing e Day Trade)
 - Opções de ações - 2018 (Swing trade com estruturas)
 - Dólar futuro - 2018 (Day trade com tape reading)
 - Ações e BDR's - 2018 até agora (Position)

DEFINIÇÃO

Machine Learning: O campo de estudo que dá aos computadores a capacidade de aprender sem serem explicitamente programados



"Diz-se que um programa de computador aprende com a experiência E com relação a alguma classe de tarefas T e com a medida de desempenho P , se seu desempenho nas tarefas em T , medido por P , melhorar com a experiência E ."

Exemplo: Jogando damas.



E = a experiência de jogar muitos jogos de damas.

T = a tarefa de jogar damas.

P = a probabilidade de o programa vencer o próximo jogo.

Aprendizagem supervisionada

No aprendizado supervisionado, recebemos um conjunto de dados e já sabemos como deve ser nossa saída correta, tendo a ideia de que existe uma relação entre a entrada e a saída.

Os problemas de aprendizado supervisionado são classificados em:

- Regressão

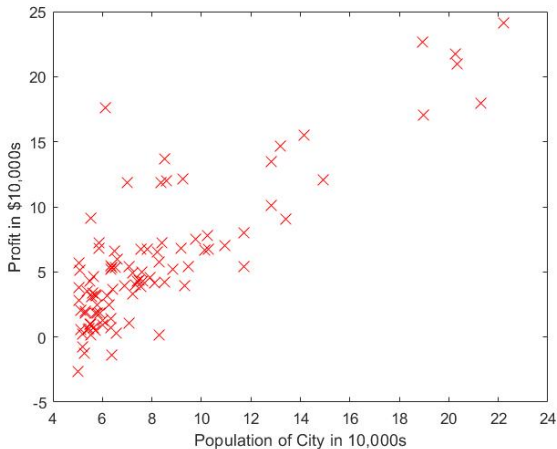
Quando estamos trabalhando com variáveis contínuas

- Classificação

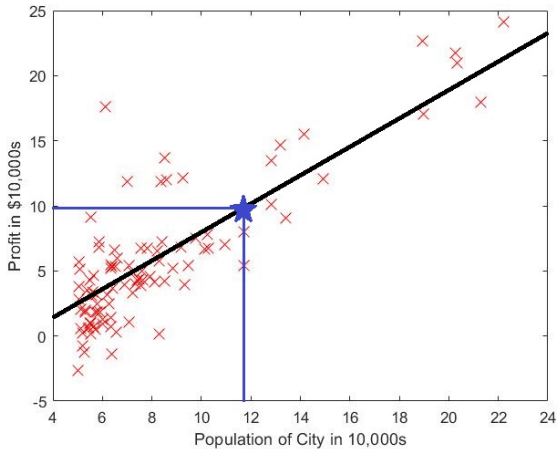
Quando estamos trabalhando com variáveis discretas

REGRESSÃO

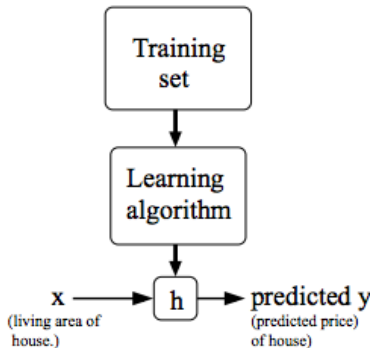
Lucratividade por densidade demográfica



Regressão Linear



Representação do modelo



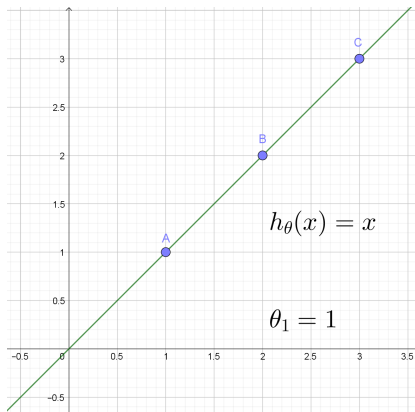
Função hipótese



$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

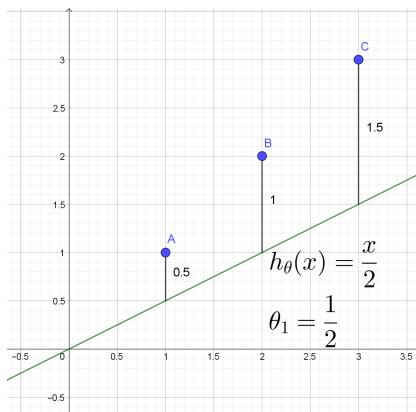
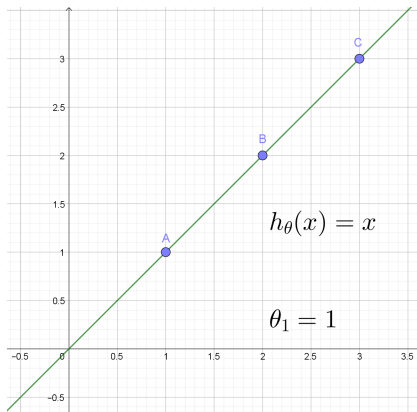
Função hipótese

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$



Função hipótese

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$



Função custo

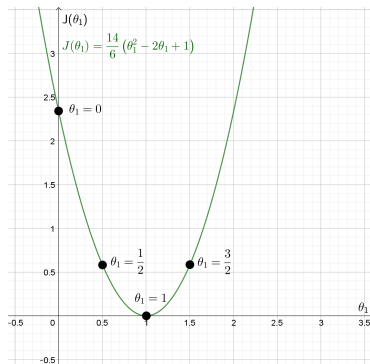


$$J(\theta_1, \theta_2) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x_i) - y_i)^2$$

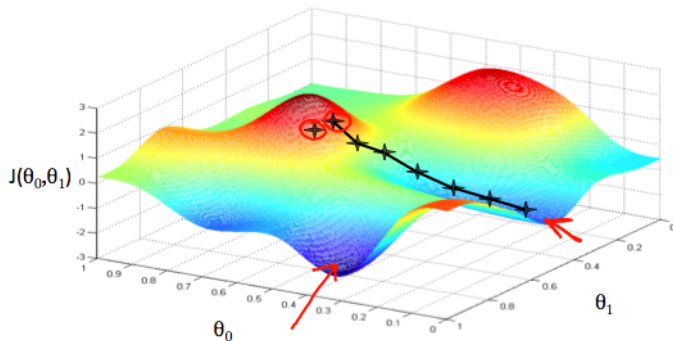
Função custo



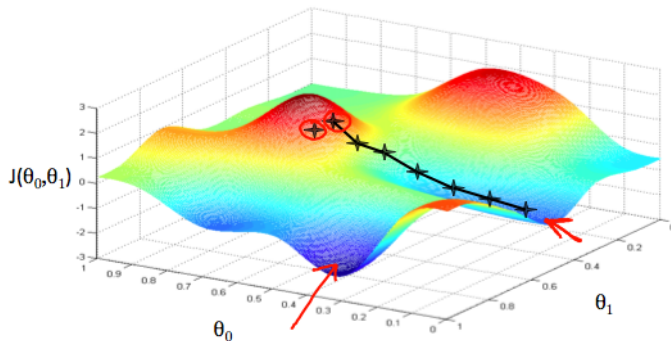
$$J(\theta_1, \theta_2) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x_i) - y_i)^2$$



Gradiente descendente



Gradiente descendente



$$\frac{\partial}{\partial \theta_j} J(\theta) = (h_{\theta}(x) - y) x_j$$

Gradiente descendente

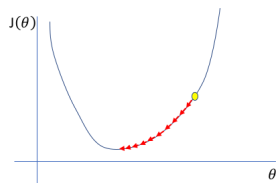


$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$$

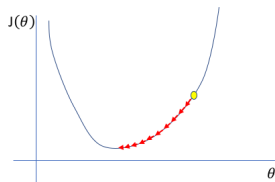
Gradiente descendente



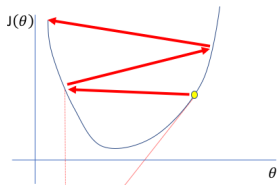
$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$$



Gradiente descendente



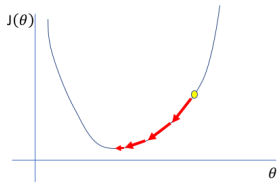
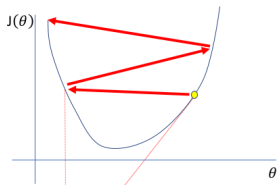
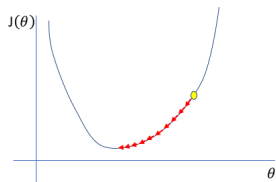
$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$$



Gradiente descendente



$$\theta_1 := \theta_1 - \alpha \frac{d}{d\theta_1} J(\theta_1)$$

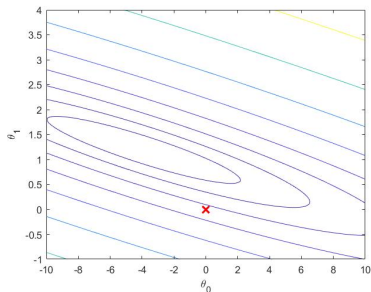
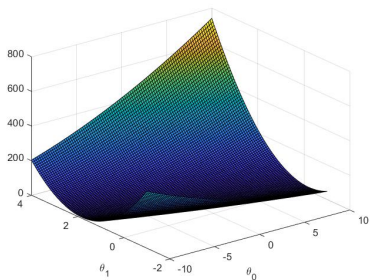


Regressão linear - Função custo

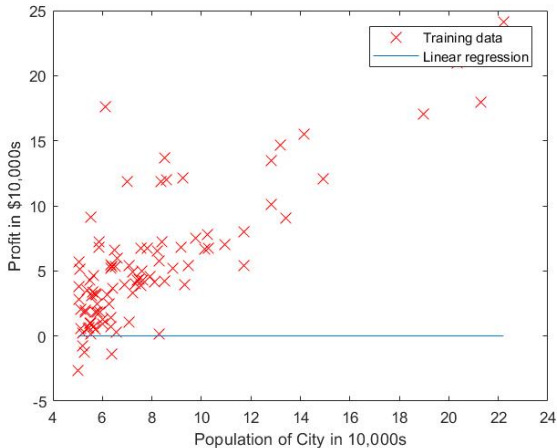


$$h = X * \theta;$$

$$J = (\text{sum}((h-y).^2)) / (2 * m);$$



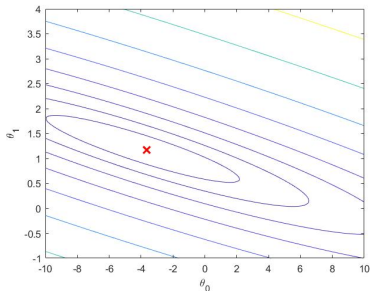
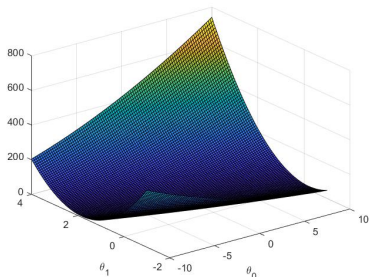
Regressão linear - Função custo



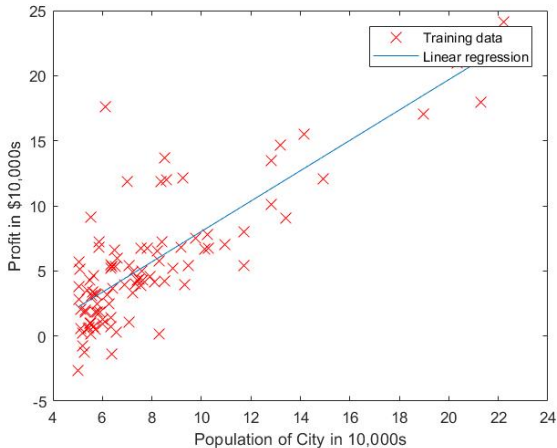
Regressão linear - Gradiente descendente



```
for iter = 1:num_iters  
  
    h = X*theta;  
  
    theta = theta - (alpha/m)*((h-y)'*X)';  
  
    J_history(iter) = computeCost(X, y, theta);  
  
end
```



Regressão linear - Gradiente descendente



Regressão linear multivariada

Se o problema depende de vários parâmetros, a função hipótese vai depender de várias variáveis.

$$h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \dots + \theta_n x_n$$

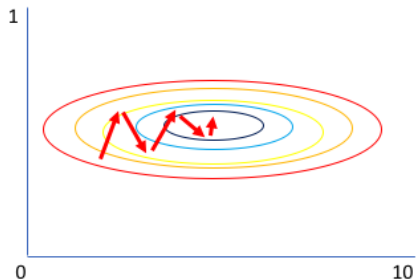
Computacionalmente é vantajoso utilizar vetores

$$h_{\theta} = [\theta_0 \quad \theta_1 \quad \dots \quad \theta_n] \begin{bmatrix} x_0 \\ x_1 \\ \dots \\ x_n \end{bmatrix} = \theta^T x$$

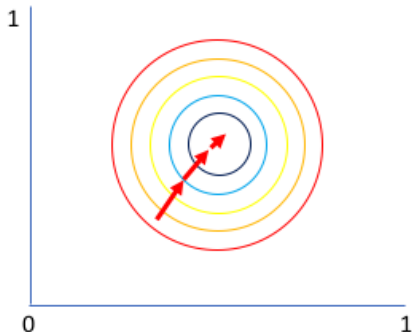
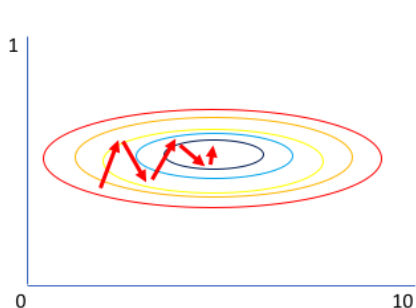
Basta adicionar um índice referente ao parâmetro

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m \left(h_{\theta} \left(x^{(i)} \right) - y^{(i)} \right) x_j^{(i)}$$

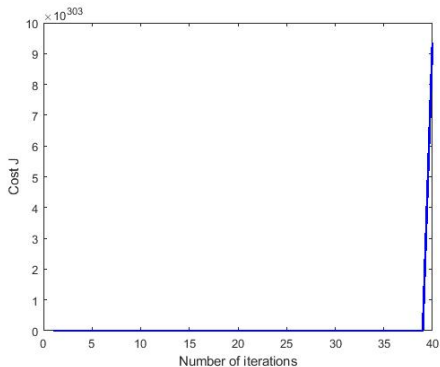
Normalização da escala



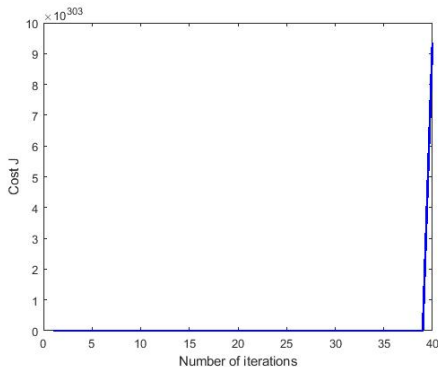
Normalização da escala



Regressão multilinear - Normalização



Regressão multilinear - Normalização



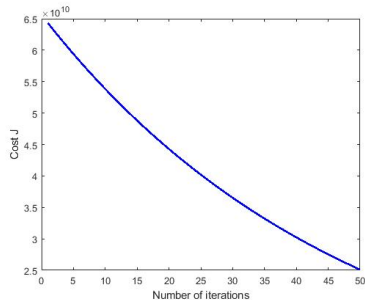
Considerando μ como sendo a média e σ o desvio padrão

$$x_i = \frac{x_i - \mu_i}{\sigma_i}$$

Em matlab, já temos essas funções prontas.

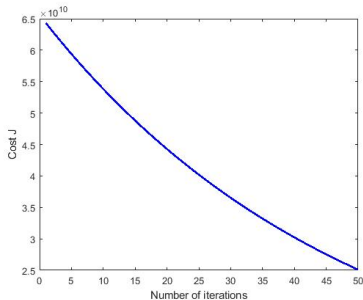
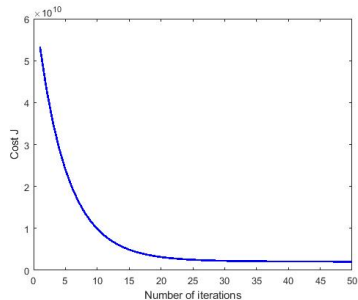
```
mu = mean(X);  
sigma = std(X);  
  
X_norm = (X-mu)./sigma;
```

Regressão multilinear - convergência

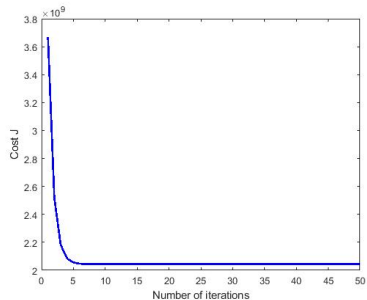


$$\alpha = 0,01$$

Regressão multilinear - convergência

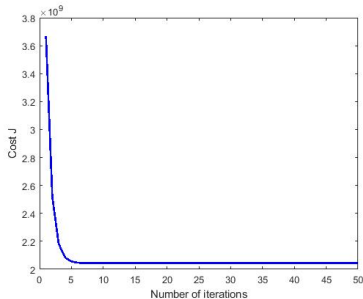
 $\alpha = 0,01$  $\alpha = 0,1$

Regressão multilinear - convergência

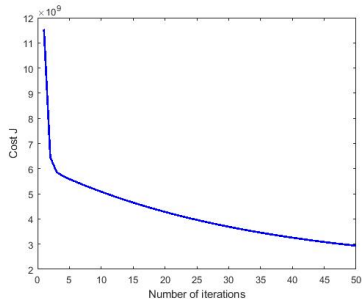


$$\alpha = 1$$

Regressão multilinear - convergência

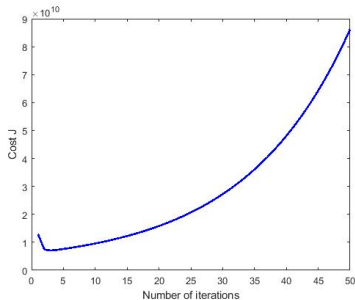


$$\alpha = 1$$



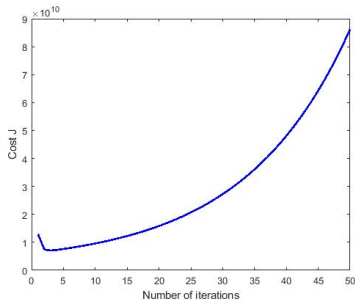
$$\alpha = 1,3$$

Regressão multilinear - convergência

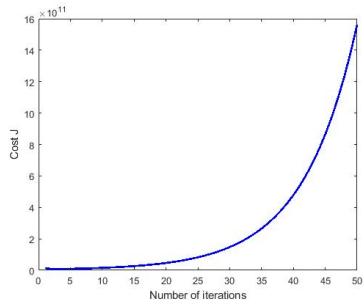


$$\alpha = 1,33$$

Regressão multilinear - convergência

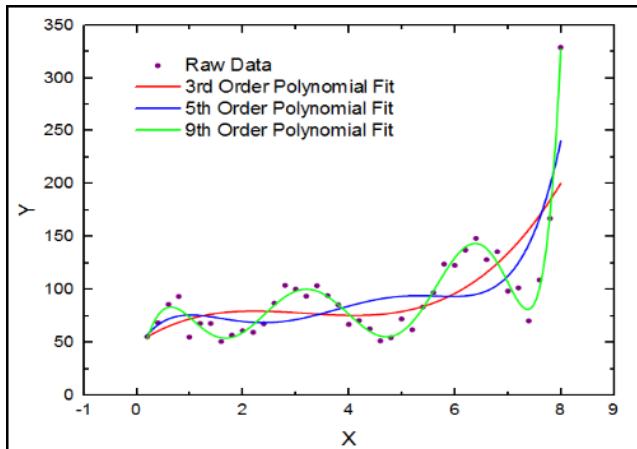


$$\alpha = 1,33$$



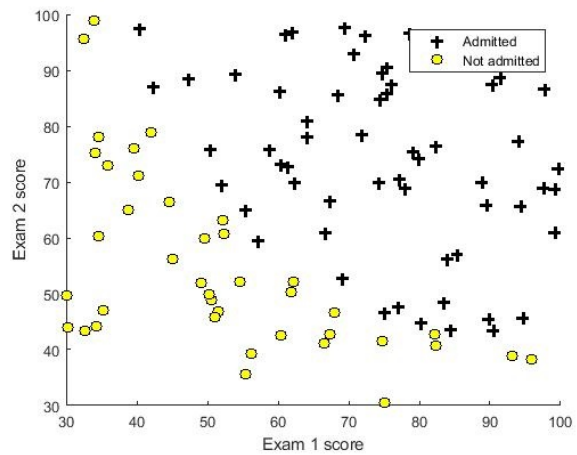
$$\alpha = 1,35$$

Regressão polinomial

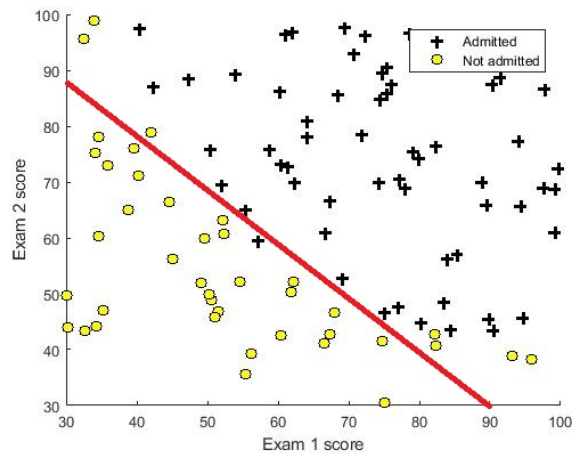


CLASSIFICAÇÃO

Aprovação escolar



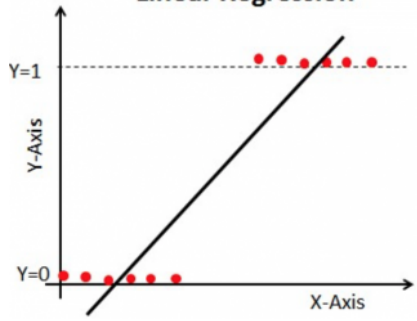
Aprovação escolar



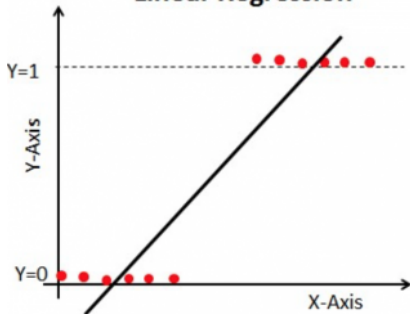
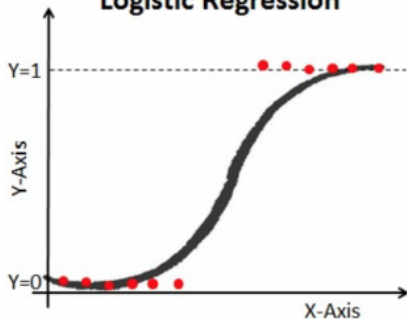
Função hipótese logística - função ativadora



Linear Regression



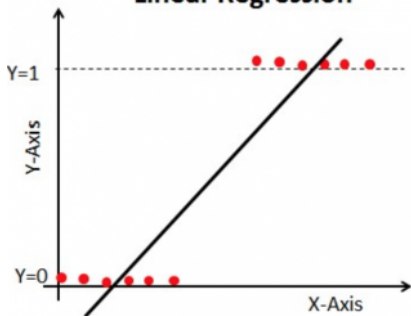
Função hipótese logística - função ativadora

**Linear Regression****Logistic Regression**

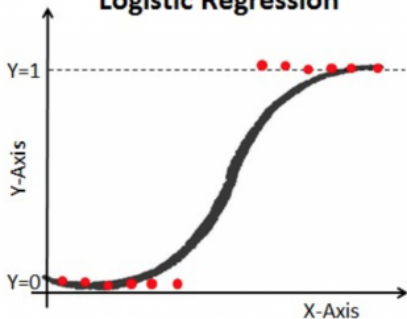
Função hipótese logística - função ativadora



Linear Regression



Logistic Regression



$$h_{\theta}(x) = \frac{1}{1 + e^{-\theta^T x}}$$

Função custo logística

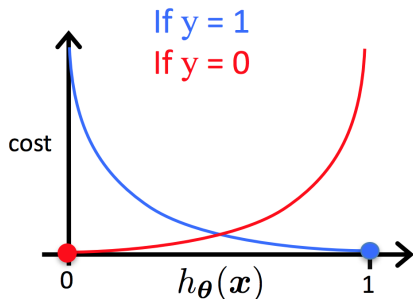
$$J(\theta) = \frac{1}{m} \sum_{i=1}^m C(h_{\theta}(x^{(i)}), y^{(i)})$$

$$\begin{cases} C = -\log(h_{\theta}(x)) & \text{se } y = 1 \\ C = -\log(1 - h_{\theta}(x)) & \text{se } y = 0 \end{cases}$$

Função custo logística

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m C(h_{\theta}(x^{(i)}), y^{(i)})$$

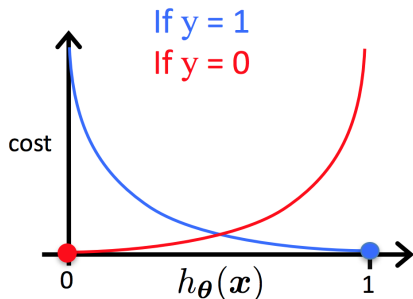
$$\begin{cases} C = -\log(h_{\theta}(x)) & \text{se } y = 1 \\ C = -\log(1 - h_{\theta}(x)) & \text{se } y = 0 \end{cases}$$



Função custo logística

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m C(h_{\theta}(x^{(i)}), y^{(i)})$$

$$\begin{cases} C = -\log(h_{\theta}(x)) & \text{se } y = 1 \\ C = -\log(1 - h_{\theta}(x)) & \text{se } y = 0 \end{cases}$$



$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m \left[y^{(i)} \log(h_{\theta}(x^{(i)})) - (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)})) \right]$$

Otimização Avançada



Algoritmos de otimização

- Gradiente descendente
- Gradiente conjugado
- BFGS
- L-BFGS

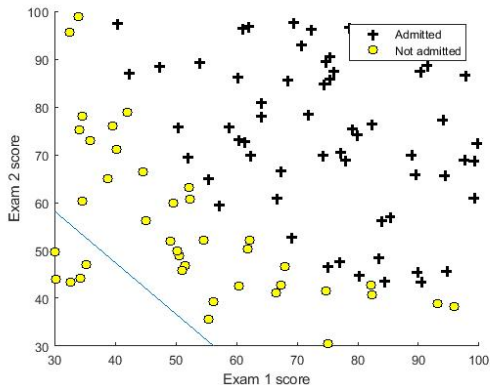
Vantagens

- O parâmetro α é calculado automaticamente (de forma otimizada)
- Convergem mais rápido que o gradiente descendente

Desvantagens

- São mais complexos

Classificação linear



```
% Função Custo Linear
```

```
h = X*theta;
```

```
J = (sum(h-y).^2)/(2*m);
```

```
% Gradiente
```

```
Sum = 0;
```

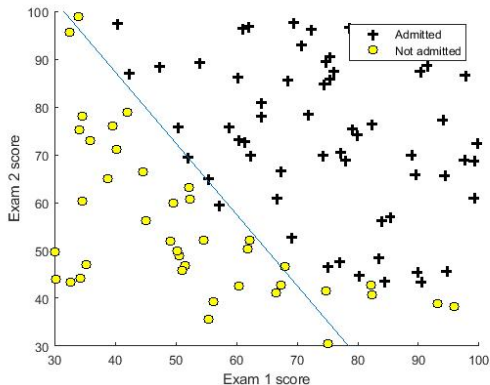
```
for i=1:m
```

```
    Sum = Sum + (h(i) - y(i)) * X(i,:);
```

```
end
```

```
grad = Sum/m;
```

Classificação logística



```
% Função Custo Logística
```

```
h = sigmoid(X*theta);
```

```
J = (sum(-y.*log(h) - (1-y).*log(1-h)))/m;
```

```
% Gradiente
```

```
Sum = 0;
```

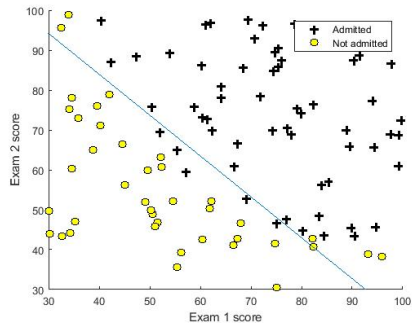
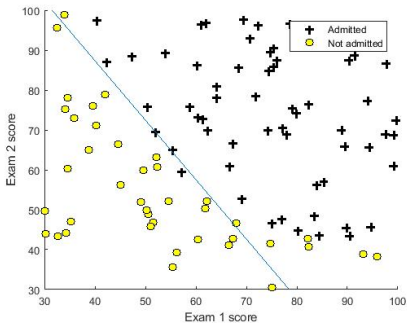
```
for i=1:m
```

```
    Sum = Sum + (h(i) - y(i))* X (i,:);
```

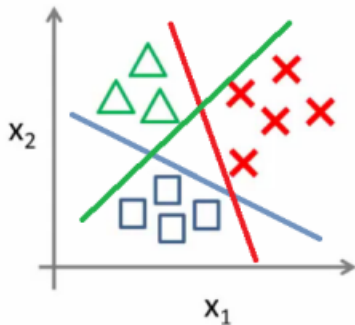
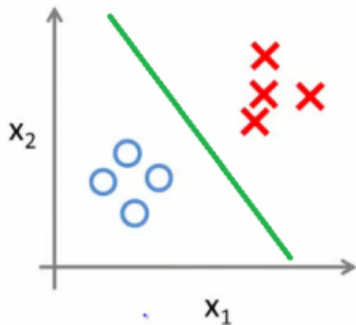
```
end
```

```
grad = Sum/m;
```

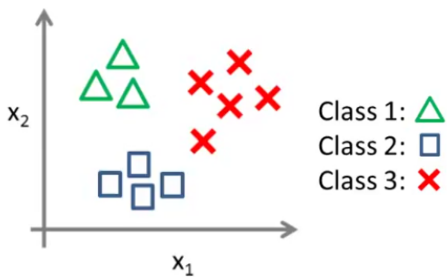
Classificação logística



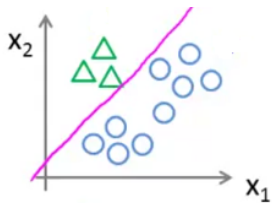
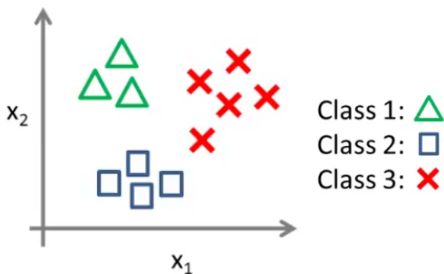
Classificação multiclasse



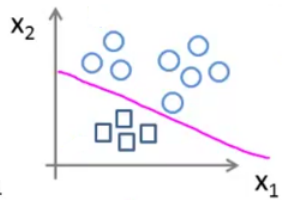
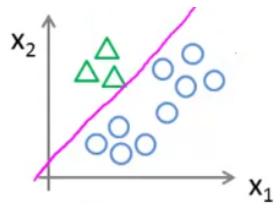
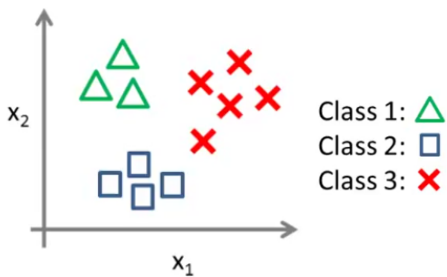
Classificação multiclasse - 1 contra todos



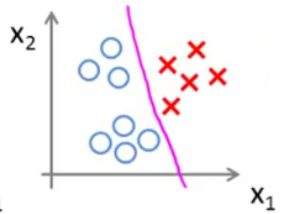
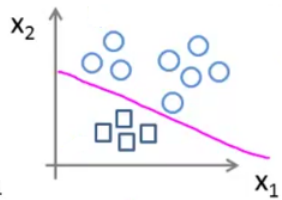
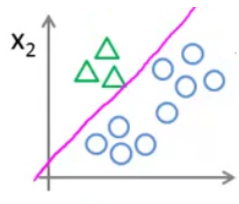
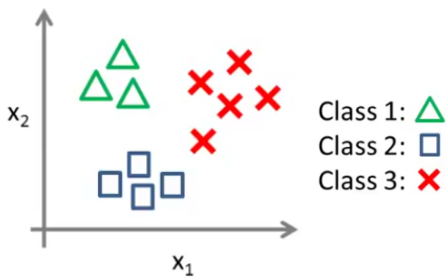
Classificação multiclasse - 1 contra todos



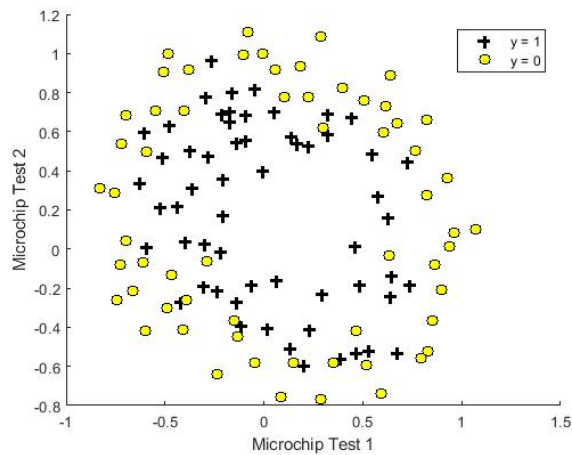
Classificação multiclasse - 1 contra todos



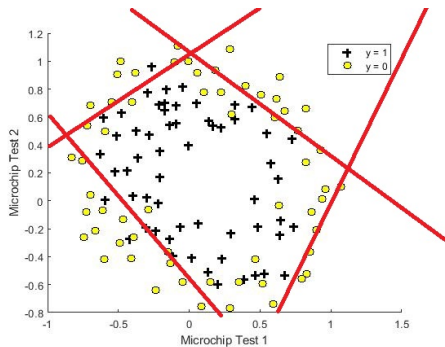
Classificação multiclasse - 1 contra todos



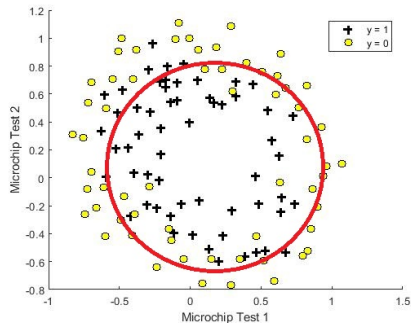
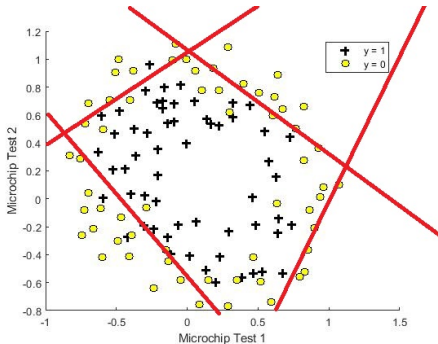
Teste de qualidade



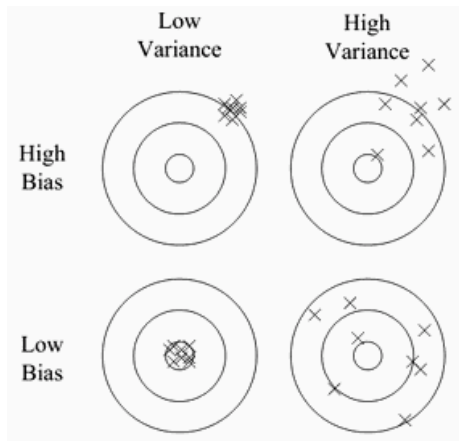
Classificação logística



Classificação logística



Viés vs variância



Bias (Viés)

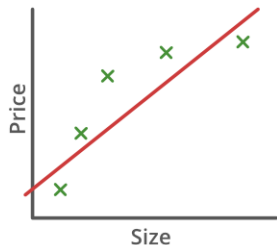
Tendência a aprender constantemente algo que está errado

Variance (Variância)

Tendência a aprender coisas aleatórias, independente dos dados de entrada

Viés vs variância - Regressão

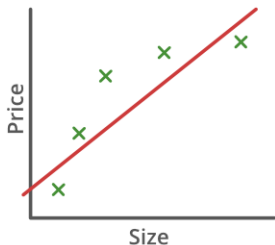
Alto viés
Sub-ajustado



$$\theta_0 + \theta_1 x$$

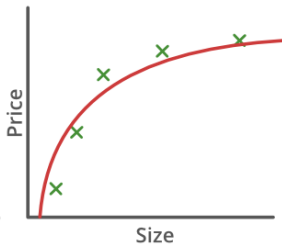
Viés vs variância - Regressão

Alto viés
Sub-ajustado



$$\theta_0 + \theta_1 x$$

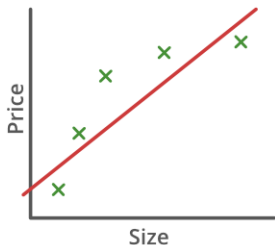
Ajuste adequado
Robusto



$$\theta_0 + \theta_1 x + \theta_2 x^2$$

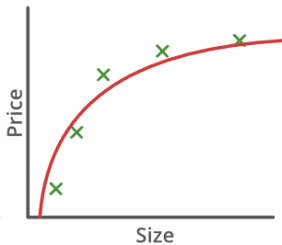
Viés vs variância - Regressão

Alto viés
Sub-ajustado



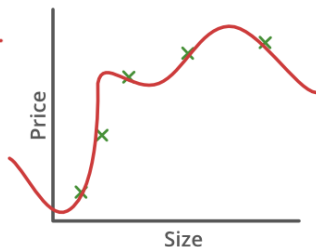
$$\theta_0 + \theta_1 x$$

Ajuste adequado
Robusto



$$\theta_0 + \theta_1 x + \theta_2 x^2$$

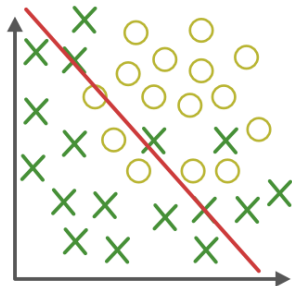
Alta variância
Sobre-ajustado



$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4 + \theta_5 x^5$$

Viés vs variância - Classificação

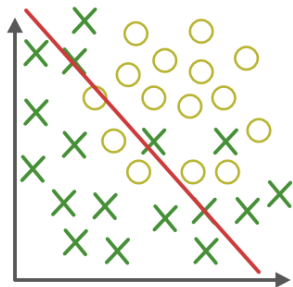
High Bias
Underfitting



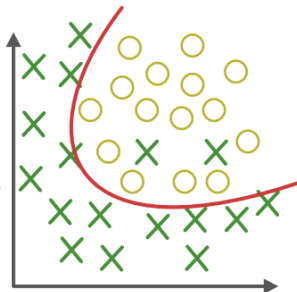
$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

Viés vs variância - Classificação

High Bias
Underfitting



Nice Fitting
Robust

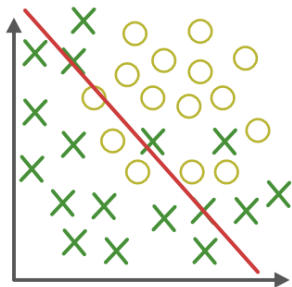


$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$$

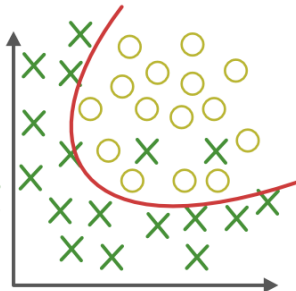
Viés vs variância - Classificação

High Bias
Underfitting



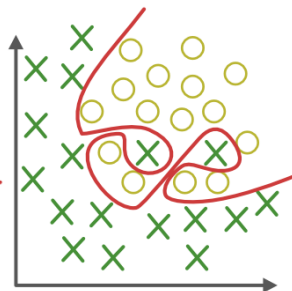
$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

Nice Fitting
Robust



$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$$

High Variance
Overfitting



$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \dots)$$

Contornando o Overfitting



1 Reduza o número de parâmetros

- Selecione manualmente quais recursos manter.
- Use um algoritmo de seleção de modelo.

2 Regularização

- Mantenha todos os recursos, mas reduza a magnitude dos parâmetros θ_j .
- A regularização funciona bem quando temos muitos recursos pouco relevantes.

Contornando o Overfitting

Regressão linear

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}), y^{(i)})^2 + \lambda \sum_{j=1}^n \theta_j^2$$

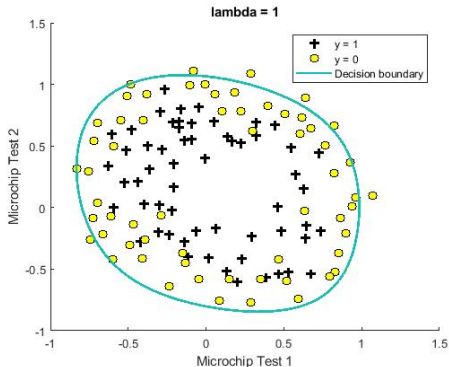
Regressão logística

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m \left[y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)})) \right] + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2$$

Gradiente descendente

$$\theta_j := \theta_j \left(1 - \alpha \frac{\lambda}{m}\right) - \alpha \frac{1}{m} \sum_{i=1}^m (h_{\theta}(x^{(i)}), y^{(i)}) x_j^{(i)}$$

Regularização Linear



```
% Função Custo Linear Regularizada
```

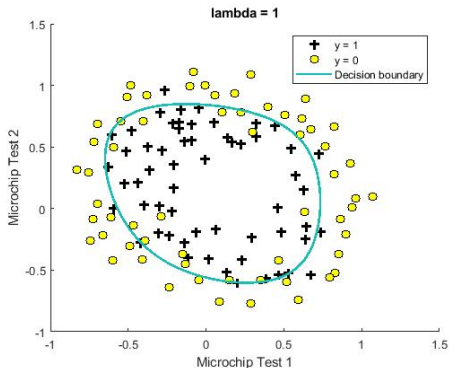
```
h = X*theta;
```

```
J = (sum(h-y).^2) / (2*m) + ...  
      (lambda / (2*m)) * sum(theta'*theta);
```

```
% Gradiente Regularizado
```

```
grad = (X'*(h-y) + lambda*theta) / m;
```

Regularização Logística



```
% Função Custo Logística Regularizada
```

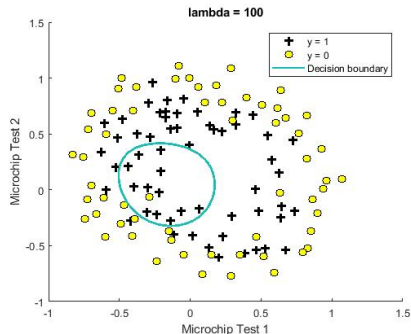
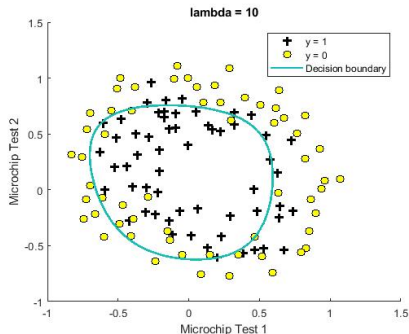
```
h = sigmoid(X*theta);
```

```
J = (sum(-y.*log(h) - (1-y).*log(1-h)))/m + ...  
      (lambda/(2*m))*sum(theta'*theta);
```

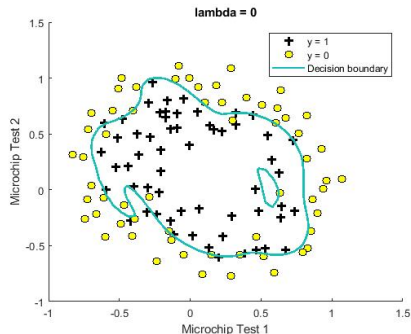
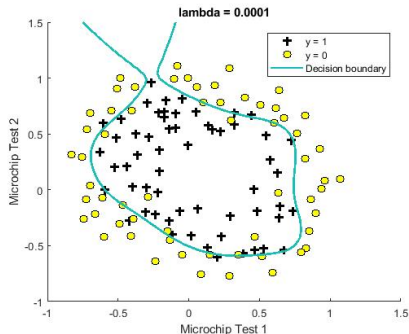
```
% Gradiente Regularizado
```

```
grad = (X'*(h-y) + lambda*theta)/m;
```

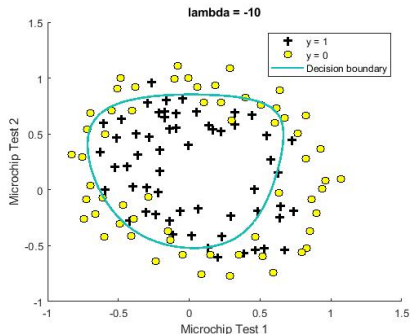
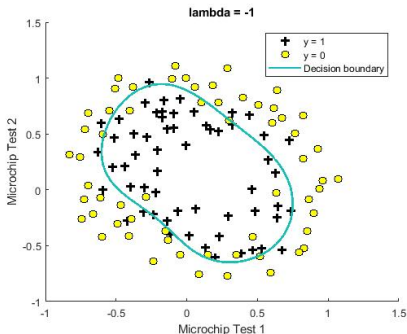

Alto Viés (Underfitting)



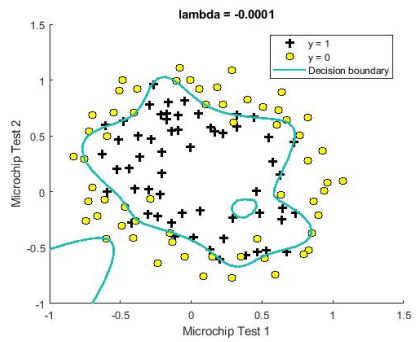
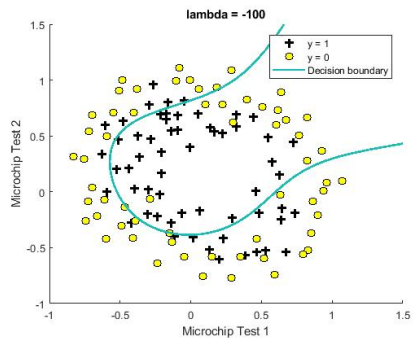
Alta Variância (Overfitting)



Regularização absurda (curiosidade)



Regularização absurda (curiosidade)



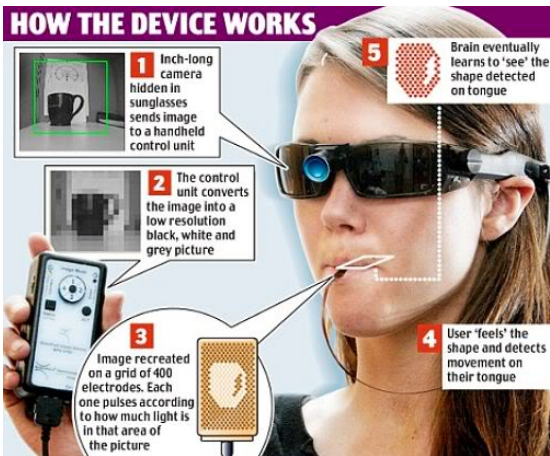
REDES NEURAIS

Reconhecimento de escrita



Enxergar com a língua

HOW THE DEVICE WORKS



- 1** Inch-long camera hidden in sunglasses sends image to a handheld control unit
- 2** The control unit converts the image into a low resolution black, white and grey picture
- 3** Image recreated on a grid of 400 electrodes. Each one pulses according to how much light is in that area of the picture
- 4** User 'feels' the shape and detects movement on their tongue
- 5** Brain eventually learns to 'see' the shape detected on tongue

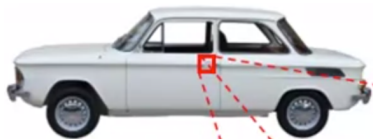
Classificação não-linear



0	2	15	0	0	11	10	0	0	0	0	9	9	0	0	0
0	0	0	4	60	157	236	255	255	177	95	61	32	0	0	29
0	10	16	119	238	255	244	245	243	250	249	255	222	103	10	0
0	14	170	255	255	244	254	255	253	245	255	249	253	251	124	1
2	98	255	228	255	251	254	211	141	116	122	215	251	238	255	49
13	217	243	255	155	33	226	52	2	0	10	13	232	255	255	36
16	229	252	254	49	12	0	0	7	7	0	70	237	252	235	62
6	141	245	255	212	25	11	9	3	0	115	236	243	255	137	0
0	87	252	250	248	215	60	0	1	121	252	255	248	144	6	0
0	13	113	255	255	245	255	182	181	248	252	242	208	36	0	19
1	0	5	117	251	255	241	255	247	255	241	162	17	0	7	0
0	0	0	4	58	251	255	246	254	253	255	120	11	0	1	0
0	0	4	97	255	255	255	248	252	255	244	255	182	10	0	4
0	22	206	252	246	251	241	100	24	113	255	245	255	194	9	0
0	111	255	242	255	158	24	0	6	39	255	232	230	56	0	0
0	218	251	250	137	7	11	0	0	2	62	255	250	125	3	0
0	173	255	255	101	9	20	0	13	3	13	182	251	245	61	0
0	107	251	241	255	230	98	55	19	118	217	248	253	255	52	4
0	18	146	250	255	247	255	255	255	249	255	240	255	129	0	5
0	0	23	113	215	255	250	248	255	255	248	248	118	14	12	0
0	0	6	1	0	52	153	233	255	252	147	37	0	0	4	1
0	0	5	5	0	0	0	0	0	14	1	0	6	6	0	0

0	2	15	0	0	11	10	0	0	0	0	9	9	0	0	0
0	0	0	4	60	157	236	255	255	177	95	61	32	0	0	29
0	10	16	119	238	255	244	245	243	250	249	255	222	103	10	0
0	14	170	255	255	244	254	255	253	245	255	249	253	251	124	1
2	98	255	228	255	251	254	211	141	116	122	215	251	238	255	49
13	217	243	255	155	33	226	52	2	0	10	13	232	255	255	36
16	229	252	254	49	12	0	0	7	7	0	70	237	252	235	62
6	141	245	255	212	25	11	9	3	0	115	236	243	255	137	0
0	87	252	250	248	215	60	0	1	121	252	255	248	144	6	0
0	13	113	255	255	245	255	182	181	248	252	242	208	36	0	19
1	0	5	117	251	255	241	255	247	255	241	162	17	0	7	0
0	0	0	4	58	251	255	246	254	253	255	120	11	0	1	0
0	0	4	97	255	255	255	248	252	255	244	255	182	10	0	4
0	22	206	252	246	251	241	100	24	113	255	245	255	194	9	0
0	111	255	242	255	158	24	0	6	39	255	232	230	56	0	0
0	218	251	250	137	7	11	0	0	2	62	255	250	125	3	0
0	173	255	255	101	9	20	0	13	3	13	182	251	245	61	0
0	107	251	241	255	230	98	55	19	118	217	248	253	255	52	4
0	18	146	250	255	247	255	255	255	249	255	240	255	129	0	5
0	0	23	113	215	255	250	248	255	255	248	248	118	14	12	0
0	0	6	1	0	52	153	233	255	252	147	37	0	0	4	1
0	0	5	5	0	0	0	0	0	14	1	0	6	6	0	0

Classificação não-linear



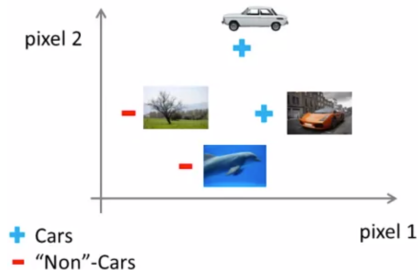
But the camera sees this:

194	210	201	212	199	213	215	195	178	158	182	209
180	189	190	221	209	205	191	167	147	115	129	163
114	126	140	188	176	165	152	140	170	106	78	88
87	103	115	154	143	142	149	153	173	101	57	57
102	112	106	131	122	138	152	147	128	84	58	66
94	95	79	104	105	124	129	113	107	87	69	67
68	71	69	98	89	92	98	95	89	88	76	67
41	56	68	99	63	45	60	82	58	76	75	65
20	43	69	75	56	41	51	73	55	70	63	44
50	50	57	69	75	75	73	74	53	68	59	37
72	59	53	66	84	92	84	74	57	72	63	42
67	61	58	65	75	78	76	73	59	75	69	50

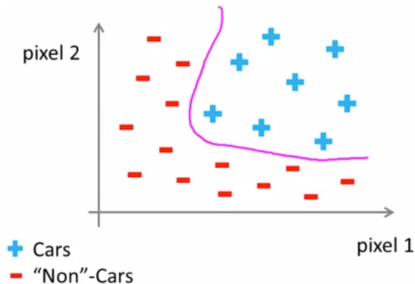
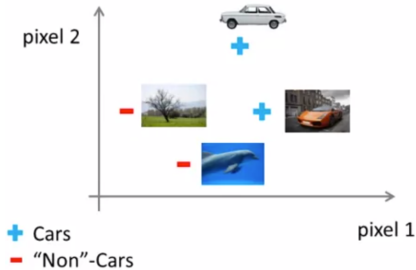
Classificação não-linear



Classificação não-linear

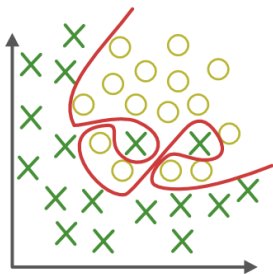


Classificação não-linear



Custo computacional

Considerando que a função hipótese possui 100 parâmetros



$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1 x_2 + \theta_4 x_1 x_3 + \dots + \theta_p x^3 + \theta_{p+1} x_1^2 x_2 + \theta_{p+2} x_1^2 x_3 + \dots)$$

Termos quadráticos $O(n^2)$

$$x_1^2, x_1 x_2, x_1 x_3, \dots, x_1 x_{100}, x_2^2, x_2 x_3, \dots \approx 5.000$$

Termos cúbicos $O(n^3)$

$$x_1^3, x_1^2 x_2, x_1 x_2 x_3, x_1 x_2^2, \dots \approx 170.000$$

Custo computacional

Imagem com 50x50 pixels.

- 2.500 parâmetros (escala de cinza)

Termos quadráticos $\approx 3,1 \cdot 10^6$ ou 3.125.000,00

Termos cúbicos $\approx 2,6 \cdot 10^9$ ou 2.604.166.666,67

- 7.500 parâmetros (RGB)

Termos quadráticos $\approx 28,1 \cdot 10^6$ ou 28.125.000,00

Termos cúbicos $\approx 70,3 \cdot 10^9$ ou 70.312.500.000,00

Imagem com 100x100 pixels.

- 10.000 parâmetros (escala de cinza)

Termos quadráticos $\approx 50 \cdot 10^6$ ou 50.000.000,00

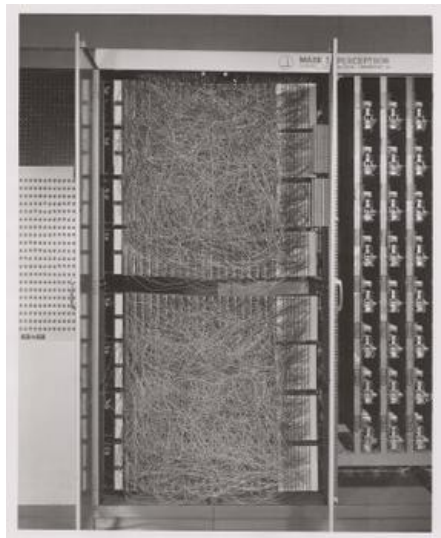
Termos cúbicos $\approx 166,7 \cdot 10^9$ ou 166.666.666.666,67

- 30.000 parâmetros (RGB)

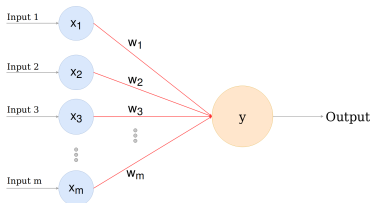
Termos quadráticos $\approx 450,1 \cdot 10^6$ ou 450.000.000,00

Termos cúbicos $\approx 4,5 \cdot 10^{12}$ ou 4.500.000.000.000,00

Neurocomputação - Perceptron



O perceptron é um tipo de rede neural artificial inventada em 1957 por Frank Rosenblatt no Cornell Aeronautical Laboratory.



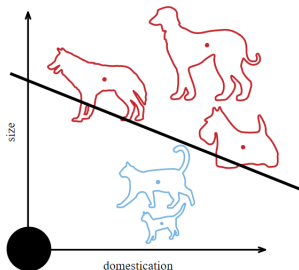
Ele pode ser visto como o tipo mais simples de rede neural feedforward: um classificador linear.

Neurocomputação - Perceptron



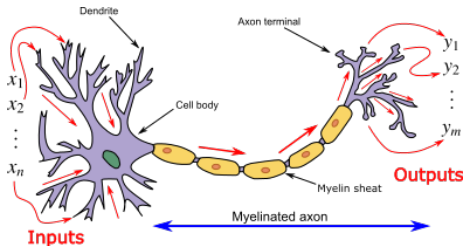
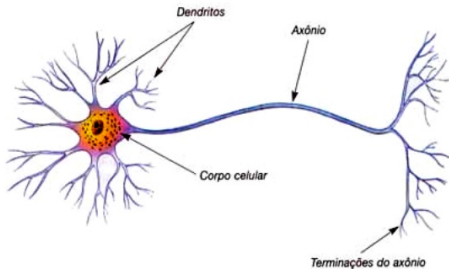
O perceptron é um classificador binário que mapeia sua entrada x (um vetor de valor real) para um valor de saída

$$f(x) = \begin{cases} 1 & \text{se } \omega \cdot x + b \leq 0 \\ 0 & \text{senão} \end{cases}$$

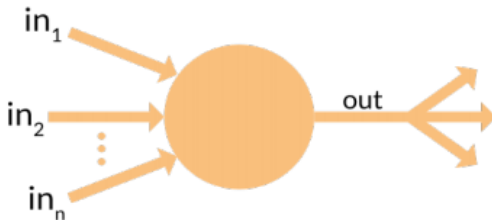
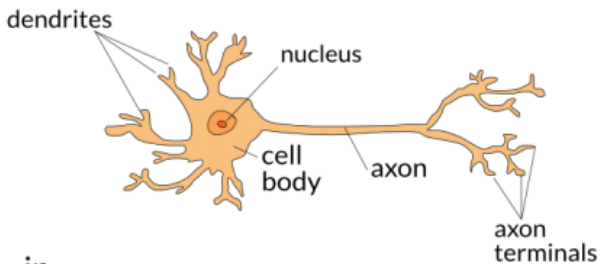


onde ω é um vetor de peso real e $\omega \cdot x$ é o produto escalar (que computa uma soma com pesos) e b é o viés (do inglês "bias"), um termo constante que não depende de qualquer valor de entrada.

Neurônio - visão geral



Rede neural - representação do neurônio



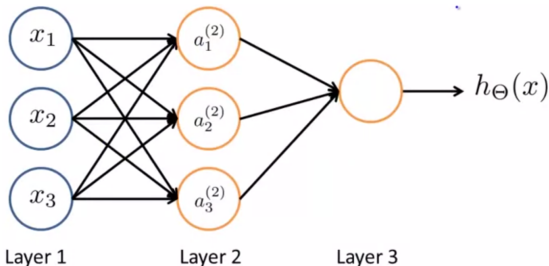
Propagação direta vetorizada

$$a_1^{(2)} = g \left(\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3 \right)$$

$$a_2^{(2)} = g \left(\Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{23}^{(1)} x_3 \right)$$

$$a_3^{(2)} = g \left(\Theta_{30}^{(1)} x_0 + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3 \right)$$

$$h_{\Theta}(x) = a_1^{(3)} = g \left(\Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)} \right)$$



Propagação direta vetorizada

Podemos representar tanto a entrada quanto os pesos da primeira camada de forma vetorial

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$\Theta^{(1)} = \begin{bmatrix} \Theta_{10}^{(1)} & \Theta_{11}^{(1)} & \Theta_{12}^{(1)} & \Theta_{13}^{(1)} \\ \Theta_{20}^{(1)} & \Theta_{21}^{(1)} & \Theta_{22}^{(1)} & \Theta_{23}^{(1)} \\ \Theta_{30}^{(1)} & \Theta_{31}^{(1)} & \Theta_{32}^{(1)} & \Theta_{33}^{(1)} \end{bmatrix}$$

Propagação direta vetorizada

Podemos representar tanto a entrada quanto os pesos da primeira camada de forma vetorial

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \Theta^{(1)} = \begin{bmatrix} \Theta_{10}^{(1)} & \Theta_{11}^{(1)} & \Theta_{12}^{(1)} & \Theta_{13}^{(1)} \\ \Theta_{20}^{(1)} & \Theta_{21}^{(1)} & \Theta_{22}^{(1)} & \Theta_{23}^{(1)} \\ \Theta_{30}^{(1)} & \Theta_{31}^{(1)} & \Theta_{32}^{(1)} & \Theta_{33}^{(1)} \end{bmatrix}$$

Podemos calcular um meta estado entre a primeira e a segunda camada, com a expressão

$$z^{(2)} = \Theta^{(1)}x$$

$$z^{(2)} = \begin{bmatrix} z_1^{(2)} \\ z_2^{(2)} \\ z_3^{(2)} \end{bmatrix}$$

Propagação direta vetorizada

Podemos representar tanto a entrada quanto os pesos da primeira camada de forma vetorial

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \Theta^{(1)} = \begin{bmatrix} \Theta_{10}^{(1)} & \Theta_{11}^{(1)} & \Theta_{12}^{(1)} & \Theta_{13}^{(1)} \\ \Theta_{20}^{(1)} & \Theta_{21}^{(1)} & \Theta_{22}^{(1)} & \Theta_{23}^{(1)} \\ \Theta_{30}^{(1)} & \Theta_{31}^{(1)} & \Theta_{32}^{(1)} & \Theta_{33}^{(1)} \end{bmatrix}$$

Podemos calcular um meta estado entre a primeira e a segunda camada, com a expressão

$$z^{(2)} = \Theta^{(1)}x$$

$$z^{(2)} = \begin{bmatrix} z_1^{(2)} \\ z_2^{(2)} \\ z_3^{(2)} \end{bmatrix}$$

Aplicando a função ativação no meta estado $z^{(2)}$, propagamos a regressão logística para a segunda camada

$$a^{(2)} = g\left(z^{(2)}\right) = \frac{1}{1 + e^{-z^{(2)}}}$$

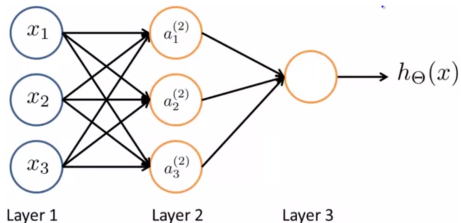
$$a^{(2)} = \begin{bmatrix} g\left(z_1^{(2)}\right) \\ g\left(z_2^{(2)}\right) \\ g\left(z_3^{(2)}\right) \end{bmatrix}$$

Propagação direta vetorizada

Adicionando o termo de "viés" $a_0^{(2)} = 1$ e podemos aplicar novamente a mesma propagação, desde que tenhamos os pesos referentes a segunda camada.

$$a^{(2)} = \begin{bmatrix} 1 \\ a_1^{(2)} \\ a_2^{(2)} \\ a_3^{(2)} \end{bmatrix}$$

$$\Theta^{(2)} = \begin{bmatrix} \Theta_{10}^{(2)} & \Theta_{11}^{(2)} & \Theta_{12}^{(2)} & \Theta_{13}^{(2)} \\ \Theta_{20}^{(2)} & \Theta_{21}^{(2)} & \Theta_{22}^{(2)} & \Theta_{23}^{(2)} \\ \Theta_{30}^{(2)} & \Theta_{31}^{(2)} & \Theta_{32}^{(2)} & \Theta_{33}^{(2)} \end{bmatrix}$$



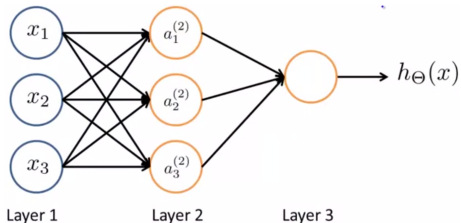
Propagação direta vetorizada

Adicionando o termo de "viés" $a_0^{(2)} = 1$ e podemos aplicar novamente a mesma propagação, desde que tenhamos os pesos referentes a segunda camada.

$$a^{(2)} = \begin{bmatrix} 1 \\ a_1^{(2)} \\ a_2^{(2)} \\ a_3^{(2)} \end{bmatrix}$$

$$\Theta^{(2)} = \begin{bmatrix} \Theta_{10}^{(2)} & \Theta_{11}^{(2)} & \Theta_{12}^{(2)} & \Theta_{13}^{(2)} \\ \Theta_{20}^{(2)} & \Theta_{21}^{(2)} & \Theta_{22}^{(2)} & \Theta_{23}^{(2)} \\ \Theta_{30}^{(2)} & \Theta_{31}^{(2)} & \Theta_{32}^{(2)} & \Theta_{33}^{(2)} \end{bmatrix}$$

Entradas \Rightarrow camada oculta



$$z^{(2)} = \Theta^{(1)} x$$

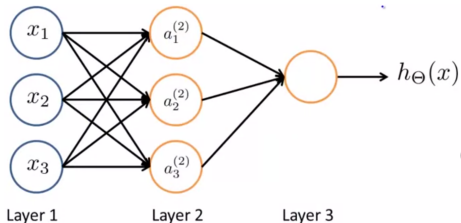
$$a^{(2)} = \begin{pmatrix} z^{(2)} \end{pmatrix}$$

Propagação direta vetorizada

Adicionando o termo de "viés" $a_0^{(2)} = 1$ e podemos aplicar novamente a mesma propagação, desde que tenhamos os pesos referentes a segunda camada.

$$a^{(2)} = \begin{bmatrix} 1 \\ a_1^{(2)} \\ a_2^{(2)} \\ a_3^{(2)} \end{bmatrix}$$

$$\Theta^{(2)} = \begin{bmatrix} \Theta_{10}^{(2)} & \Theta_{11}^{(2)} & \Theta_{12}^{(2)} & \Theta_{13}^{(2)} \\ \Theta_{20}^{(2)} & \Theta_{21}^{(2)} & \Theta_{22}^{(2)} & \Theta_{23}^{(2)} \\ \Theta_{30}^{(2)} & \Theta_{31}^{(2)} & \Theta_{32}^{(2)} & \Theta_{33}^{(2)} \end{bmatrix}$$



Entradas \Rightarrow camada oculta

$$z^{(2)} = \Theta^{(1)} x$$

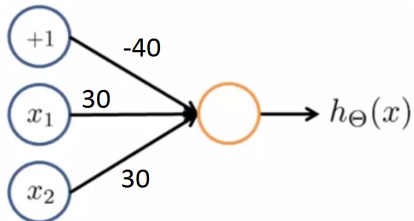
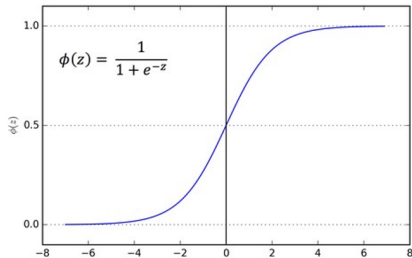
$$a^{(2)} = \begin{pmatrix} z^{(2)} \end{pmatrix}$$

Camada oculta \Rightarrow saída

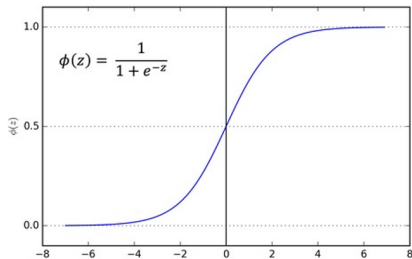
$$z^{(3)} = \Theta^{(2)} a^{(2)}$$

$$h_{\Theta}(x) = a^{(3)} = \begin{pmatrix} z^{(3)} \end{pmatrix}$$

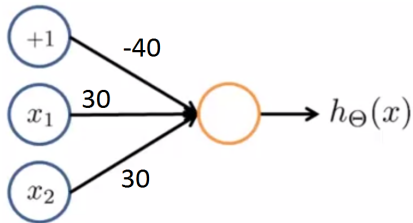
Porta AND



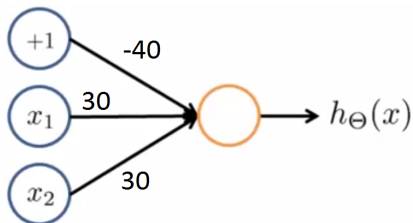
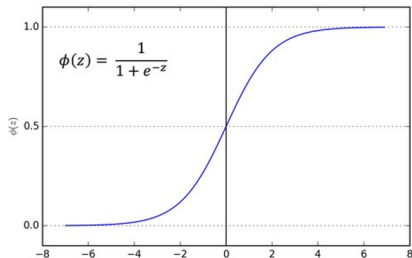
Porta AND



$$h_{\Theta}(x) = g(-40 + 30x_1 + 30x_2)$$



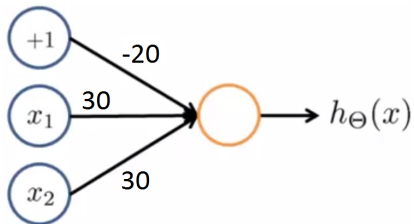
Porta AND



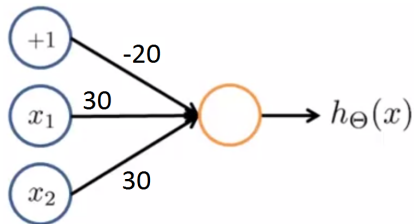
$$h_{\Theta}(x) = g(-40 + 30x_1 + 30x_2)$$

x_1	x_2	$z^{(2)}(x)$	$h_{\Theta}(x)$
0	0	-40	$g(-40) \approx 0$
0	1	-10	$g(-10) \approx 0$
1	0	-10	$g(-10) \approx 0$
1	1	20	$g(20) \approx 1$

Porta OR e NOT

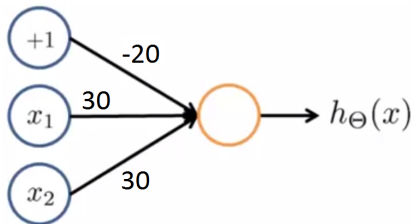


Porta OR e NOT



$$h_{\Theta}(x) = g(-20 + 30x_1 + 30x_2)$$

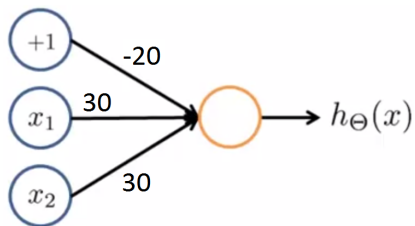
Porta OR e NOT



$$h_{\Theta}(x) = g(-20 + 30x_1 + 30x_2)$$

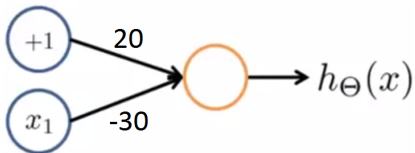
x_1	x_2	$z^{(2)}(x)$	$h_{\Theta}(x)$
0	0	-20	$g(-20) \approx 0$
0	1	10	$g(10) \approx 1$
1	0	10	$g(10) \approx 1$
1	1	20	$g(40) \approx 1$

Porta OR e NOT

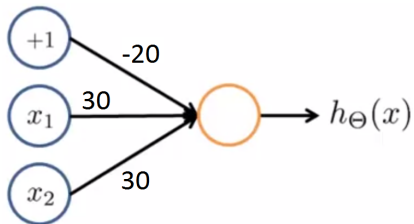


$$h_{\Theta}(x) = g(-20 + 30x_1 + 30x_2)$$

x_1	x_2	$z^{(2)}(x)$	$h_{\Theta}(x)$
0	0	-20	$g(-20) \approx 0$
0	1	10	$g(10) \approx 1$
1	0	10	$g(10) \approx 1$
1	1	20	$g(40) \approx 1$

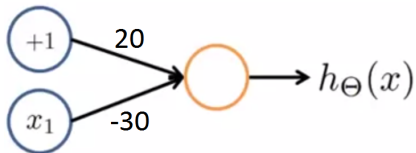


Porta OR e NOT



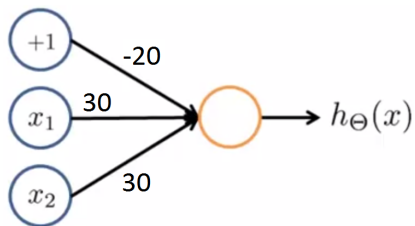
$$h_{\Theta}(x) = g(-20 + 30x_1 + 30x_2)$$

x_1	x_2	$z^{(2)}(x)$	$h_{\Theta}(x)$
0	0	-20	$g(-20) \approx 0$
0	1	10	$g(10) \approx 1$
1	0	10	$g(10) \approx 1$
1	1	20	$g(40) \approx 1$



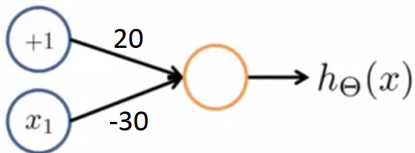
$$h_{\Theta}(x) = g(20 - 30x_1)$$

Porta OR e NOT



$$h_{\Theta}(x) = g(-20 + 30x_1 + 30x_2)$$

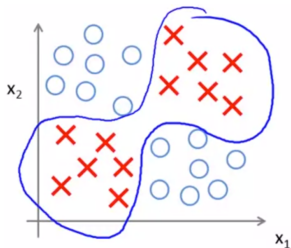
x_1	x_2	$z^{(2)}(x)$	$h_{\Theta}(x)$
0	0	-20	$g(-20) \approx 0$
0	1	10	$g(10) \approx 1$
1	0	10	$g(10) \approx 1$
1	1	20	$g(40) \approx 1$



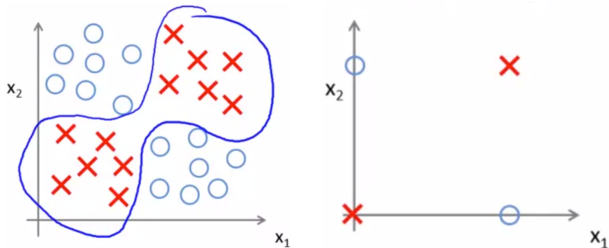
$$h_{\Theta}(x) = g(20 - 30x_1)$$

x_1	$z^{(2)}(x)$	$h_{\Theta}(x)$
0	10	$g(10) \approx 1$
1	-10	$g(-10) \approx 0$

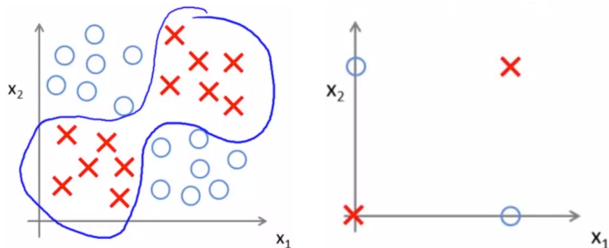
Classificação utilizando a porta XOR



Classificação utilizando a porta XOR

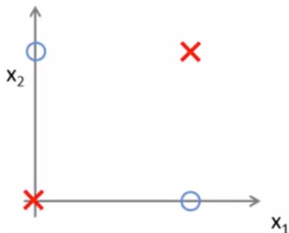
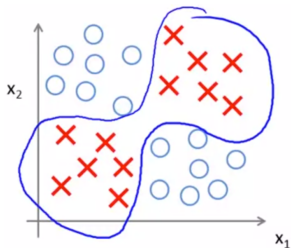


Classificação utilizando a porta XOR

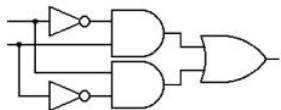


x_1	x_2	$h_{\Theta}(x)$
0	0	0
0	1	1
1	0	1
1	1	0

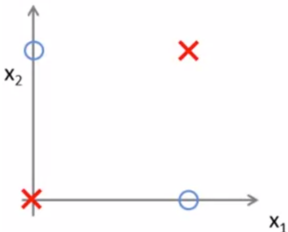
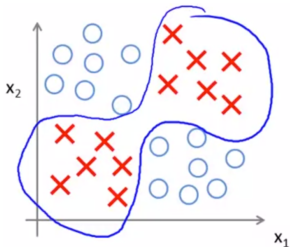
Classificação utilizando a porta XOR



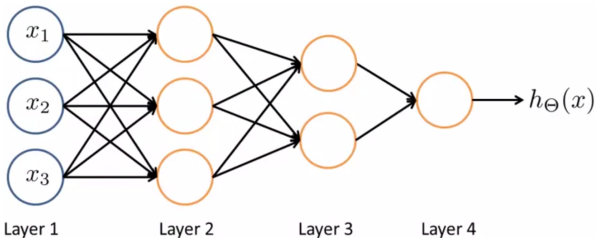
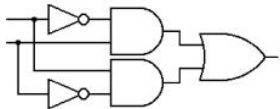
x_1	x_2	$h_{\Theta}(x)$
0	0	0
0	1	1
1	0	1
1	1	0



Classificação utilizando a porta XOR



x_1	x_2	$h_{\Theta}(x)$
0	0	0
0	1	1
1	0	1
1	1	0



Reconhecimento de escrita - classificação



```
z = sigmoid(X*theta);  
  
J = (sum(-y'*log(z) - (1-y)*log(1-z)))/m + ...  
    (lambda/(2*m))*(theta'*theta);  
  
grad = (X*(z-y) + lambda*theta)/m;
```


Reconhecimento de escrita - classificação

```

z = sigmoid(X*theta);

J = (sum(-y'*log(z) - (1-y)'*log(1-z)))/m + ...
    (lambda/(2*m))*(theta'*theta);

grad = (X'*(z-y) + lambda*theta)/m;
  
```

Testing lrCostFunction() with regularization

Cost: 2.534819

Expected cost: 2.534819

Gradients:

-0.573439

-0.308558

0.484722

0.918003

Expected gradients:

0.146561

-0.548558

0.724722

1.398003

$\lambda = 1.8$

Testing lrCostFunction() with regularization

Cost: 3.734819

Expected cost: 2.534819

Gradients:

-1.053439

-0.548558

0.724722

1.398003

Expected gradients:

0.146561

-0.548558

0.724722

1.398003

$\lambda = 3$

Reconhecimento de escrita - classificação

Otimização avançada

```

initial_theta = zeros(n + 1, 1);
options = optimset('GradObj', 'on', 'MaxIter', 50);

for c = 1:num_labels
    theta = fmincg (@(t)(lrCostFunction(t, X, (y == c), lambda)),...
        zeros(n + 1, 1), options);
    all_theta(c, :) = theta';
end
  
```

Classificação 1 contra todos

```

all_p = sigmoid(X*all_theta');

[V,P] = max(all_p, [], 2);
  
```

Treinamento - 500 interações

Training One-vs-All Logistic Regression...

```

Iteration 1 | Cost: 2.802129e-01
Iteration 2 | Cost: 9.454449e-02
Iteration 3 | Cost: 5.704679e-02
Iteration 4 | Cost: 4.688236e-02
Iteration 5 | Cost: 3.759228e-02
Iteration 6 | Cost: 3.522187e-02
Iteration 7 | Cost: 3.234838e-02
Iteration 8 | Cost: 3.145679e-02
Iteration 9 | Cost: 3.009885e-02
Iteration 10 | Cost: 2.995611e-02
  
```

```

Iteration 46 | Cost: 1.077699e-02
Iteration 47 | Cost: 1.071336e-02
Iteration 48 | Cost: 1.069749e-02
Iteration 49 | Cost: 1.064590e-02
Iteration 50 | Cost: 1.057657e-02
  
```

Program paused. Press enter to continue.

Training Set Accuracy: 94.920000



Reconhecimento de escrita - redes neurais



Para classificar utilizando redes neurais, basta incluir o termo de "viés" e propagar para a próxima camada através da função ativação.

O algoritmo é muito simples e roda muito mais rápido.

```
X = [ones(m,1) X];  
a1 = sigmoid(X*Theta1');  
  
a1 = [ones(m,1) a1];  
a2 = sigmoid(a1*Theta2');  
  
[V,p] = max(a2,[],2);
```

Além de limpo e intuitivo, tem uma acurácia superior.

```
Loading Saved Neural Network Parameters ...
```

```
Training Set Accuracy: 97.520000
```

Reconhecimento de escrita - redes neurais



Displaying Example Image

Neural Network Prediction: 5 (digit 5)

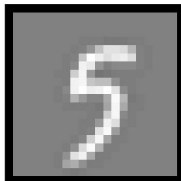
Paused - press enter to continue, q to exit:

Reconhecimento de escrita - redes neurais



Displaying Example Image

Neural Network Prediction: 5 (digit 5)
Paused - press enter to continue, q to exit:



Displaying Example Image

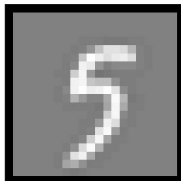
Neural Network Prediction: 5 (digit 5)
Paused - press enter to continue, q to exit:

Reconhecimento de escrita - redes neurais



Displaying Example Image

Neural Network Prediction: 5 (digit 5)
Paused - press enter to continue, q to exit:



Displaying Example Image

Neural Network Prediction: 5 (digit 5)
Paused - press enter to continue, q to exit:



Displaying Example Image

Neural Network Prediction: 3 (digit 3)
Paused - press enter to continue, q to exit:

Reconhecimento de escrita - redes neurais



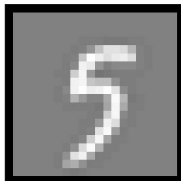
Displaying Example Image

Neural Network Prediction: 5 (digit 5)
Paused - press enter to continue, q to exit:



Displaying Example Image

Neural Network Prediction: 3 (digit 3)
Paused - press enter to continue, q to exit:



Displaying Example Image

Neural Network Prediction: 5 (digit 5)
Paused - press enter to continue, q to exit:



Displaying Example Image

Neural Network Prediction: 8 (digit 8)
Paused - press enter to continue, q to exit:8

Função custo e gradiente

Regressão logística

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m \left[y^{(i)} \log \left(h_{\theta} \left(x^{(i)} \right) \right) + \left(1 - y^{(i)} \right) \log \left(1 - h_{\theta} \left(x^{(i)} \right) \right) \right] \\ + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2$$

Função custo e gradiente

Regressão logística

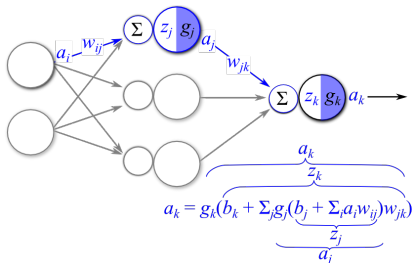
$$\begin{aligned}
 J(\theta) = & -\frac{1}{m} \sum_{i=1}^m \left[y^{(i)} \log \left(h_{\theta} \left(x^{(i)} \right) \right) + \left(1 - y^{(i)} \right) \log \left(1 - h_{\theta} \left(x^{(i)} \right) \right) \right] \\
 & + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2
 \end{aligned}$$

Rede neural

$$\begin{aligned}
 J(\Theta) = & -\frac{1}{m} \sum_{i=1}^m \sum_{k=1}^K \left[y_k^{(i)} \log \left(h_{\Theta} \left(x^{(i)} \right) \right)_k + \left(1 - y_k^{(i)} \right) \log \left(1 - \left(h_{\Theta} \left(x^{(i)} \right) \right)_k \right) \right] \\
 & + \frac{\lambda}{2m} \sum_{l=1}^{L-1} \sum_{i=1}^{S_l} \sum_{j=1}^{S_{l+1}} \left(\Theta_{ij}^{(l)} \right)^2
 \end{aligned}$$

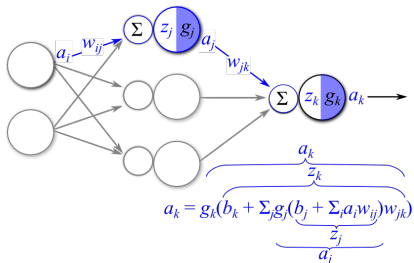
Propagação Reversa

Propagação direta do sinal de entrada

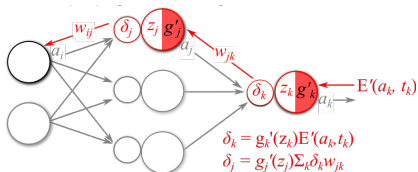


Propagação Reversa

Propagação direta do sinal de entrada

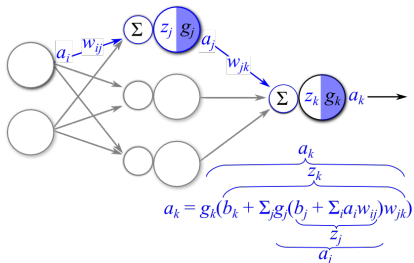


Propagação reversa do sinal de erro

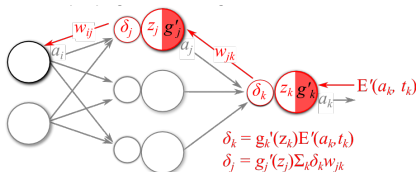


Propagação Reversa

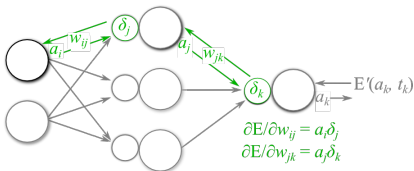
Propagação direta do sinal de entrada



Propagação reversa do sinal de erro

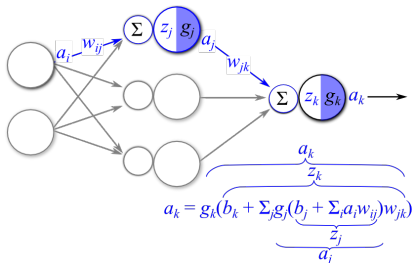


Calculo dos parâmetros do gradiente

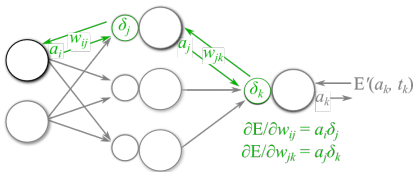


Propagação Reversa

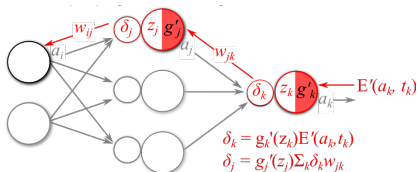
Propagação direta do sinal de entrada



Calculo dos parâmetros do gradiente



Propagação reversa do sinal de erro



Atualização dos parâmetros

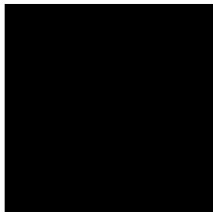
$$w_{ij} = w_{ij} - \alpha \left(\frac{\partial E}{\partial w_{ij}} \right)$$

$$w_{jk} = w_{jk} - \alpha \left(\frac{\partial E}{\partial w_{jk}} \right)$$

RECONHECIMENTO DE ESCRITA

Treinando a rede

Propagação reversa



```
Iteration 15 | Cost: 3.250830e+00  
Iteration 16 | Cost: 3.250830e+00  
Iteration 17 | Cost: 3.250830e+00
```

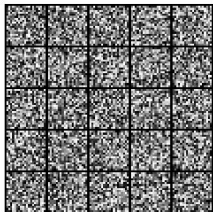
Program paused. Press enter to continue.

Visualizing Neural Network...

Program paused. Press enter to continue.

Training Set Accuracy: 10.000000

Inicialização aleatória



```
Iteration 16 | Cost: 2.123056e+00  
Iteration 17 | Cost: 2.120358e+00  
Iteration 18 | Cost: 2.120180e+00
```

Program paused. Press enter to continue.

Visualizing Neural Network...

Program paused. Press enter to continue.

Training Set Accuracy: 72.560000

Treinando a rede

Gradiente sigmoide



```
Iteration 198 | Cost: 8.949807e-01
Iteration 199 | Cost: 8.945852e-01
Iteration 200 | Cost: 8.939404e-01
```

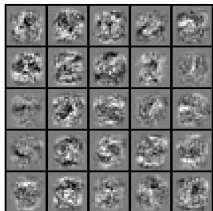
Program paused. Press enter to continue.

Visualizing Neural Network...

Program paused. Press enter to continue.

Training Set Accuracy: 86.940000

Inicialização aleatória + gradiente sigmoide



```
Iteration 198 | Cost: 3.339045e-01
Iteration 199 | Cost: 3.338426e-01
Iteration 200 | Cost: 3.337986e-01
```

Program paused. Press enter to continue.

Visualizing Neural Network...

Program paused. Press enter to continue.

Training Set Accuracy: 99.020000

Treinando a rede

Ajustando o termo de regularização λ



```
Iteration 198 | Cost: 2.631786e-02
Iteration 199 | Cost: 2.599913e-02
Iteration 200 | Cost: 2.583754e-02
```

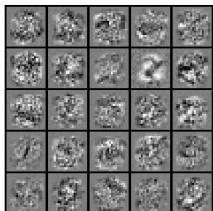
Program paused. Press enter to continue.

Visualizing Neural Network...

Program paused. Press enter to continue.

Training Set Accuracy: 99.960000

Ajustando o número de passos de treinamento



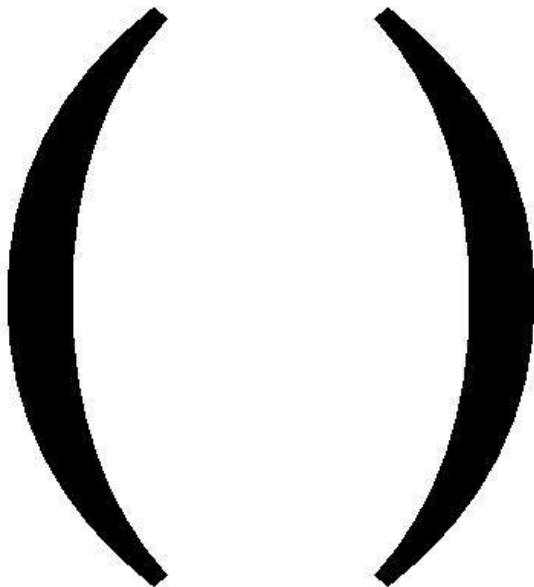
```
Iteration 398 | Cost: 1.368959e-02
Iteration 399 | Cost: 1.368683e-02
Iteration 400 | Cost: 1.368494e-02
```

Program paused. Press enter to continue.

Visualizing Neural Network...

Program paused. Press enter to continue.

Training Set Accuracy: 100.000000



História do CAPTCHA



As a protection against automated spam, you'll need to type in the words that appear in this image to register an account:
(What is this?)

sepalbeam

História do CAPTCHA



As a protection against automated spam, you'll need to type in the words that appear in this image to register an account:
(What is this?)



facebook

SEARCH BY NAME

SIMILAR NAMES
Donald Trumps

Security Check
Please enter the text below

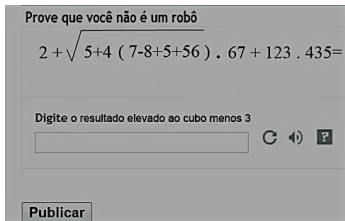


Can't read the text above?
Try another text or an audio captcha

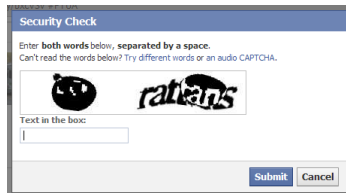
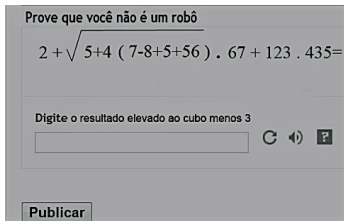
Text in the box:

What's this?

História do CAPTCHA



História do CAPTCHA



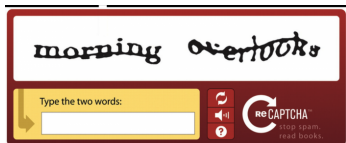
História do CAPTCHA



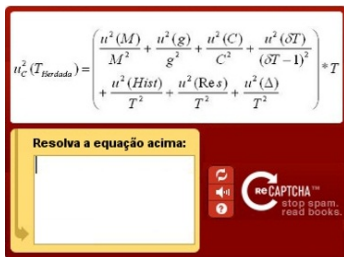
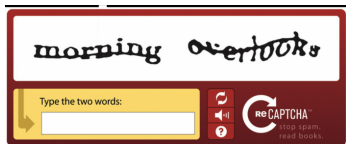
História do CAPTCHA



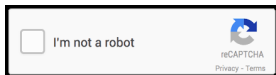
História do CAPTCHA



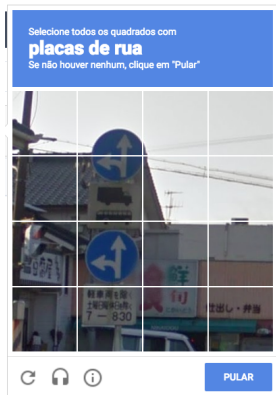
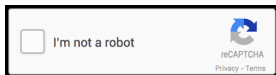
História do CAPTCHA



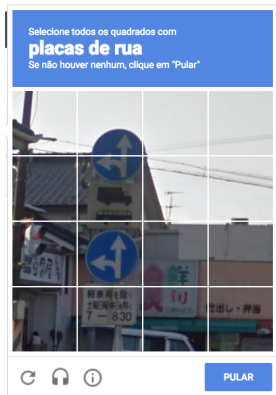
História do CAPTCHA



História do CAPTCHA



História do CAPTCHA



Tráfego de dados na internet



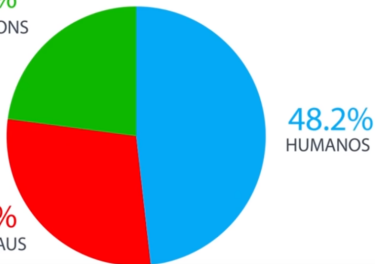
Tráfego de dados na internet



Tráfego de dados na internet



22.9%
ROBÔS BONS



28.9%
ROBÔS MAUS

CAPTCHA - significado



CAPTCHA - significado



Teste de Turing

T Turing test

CAPTCHA - significado



Teste de Turing público

P Public

T Turing test

CAPTCHA - significado



Teste de Turing público completamente

C Completely

P Public

T Turing test

CAPTCHA - significado



Teste de Turing público completamente automatizado

- C Completely
- A Automated
- P Public
- T Turing test

CAPTCHA - significado



Teste de Turing público completamente automatizado

- C Completely
- A Automated
- P Public
- T Turing test to tell

CAPTCHA - significado



Teste de Turing público completamente automatizado para diferenciar

- C Completely
- A Automated
- P Public
- T Turing test to tell

A Apart



CAPTCHA - significado

Teste de Turing público completamente automatizado para diferenciar computadores e

- C Completely
- A Automated
- P Public
- T Turing test to tell
- C Computers and

- A Apart



CAPTCHA - significado

Teste de Turing público completamente automatizado para diferenciar computadores e humanos.

- C Completely
- A Automated
- P Public
- T Turing test to tell
- C Computers and
- H Humans
- A Apart



CAPTCHA - significado

Teste de Turing público completamente automatizado para diferenciar computadores e humanos.

- C Completely
- A Automated
- P Public
- T Turing test to tell
- C Computers and
- H Humans
- A Apart

Também conhecido como teste de Turing reverso.

Patente US20050114705A1



US 20050114705A1

(19) **United States**(12) **Patent Application Publication** (10) **Pub. No.: US 2005/0114705 A1****Reshef et al.** (43) **Pub. Date: May 26, 2005**(54) **METHOD AND SYSTEM FOR
DISCRIMINATING A HUMAN ACTION
FROM A COMPUTERIZED ACTION**(76) Inventors: **Eran Reshef, Lehavim (IL); Gil
Raanan, Zoran (IL); Eilon Solan, Petah
Tikva (IL)**

Correspondence Address:

**BROWN, RAYSMAN, MILLSTEIN, FELDER
& STEINER LLP
900 THIRD AVENUE
NEW YORK, NY 10022 (US)**(21) Appl. No.: **10/790,611**(22) Filed: **Mar. 1, 2004****Related U.S. Application Data**

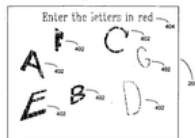
(63) Continuation of application No. 09/209,727, filed on Dec. 11, 1998, now abandoned.

(60) Provisional application No. 60/069,202, filed on Dec. 11, 1997.

Publication Classification(51) **Int. Cl.⁷** **H04L 9/00; H04L 9/32;
G06F 11/30; G06F 12/14**(52) **U.S. Cl.** **713/201**(57) **ABSTRACT**

A method and system are disclosed for discriminating automatic computerized action from a human performed action. The invention is based on applying human advantage in applying sensory and cognitive skills to solving simple problems that prove to be extremely hard for computer software. Such skills include, but are not limited to processing of sensory information such as identification of objects and letters within a noisy graphical environment, signals and speech within an auditory signal, patterns and objects within a video or animation sequence. Human skills also include higher level cognitive processing such as understanding natural language and logical assignments. The method for discriminating between humans and computerized actions can be used during authentication, to limit access by automated agents, and for confirmation of actions.

Fig. 7



Patente US6195698B1



US006195698B1

(12) **United States Patent**
Lillibridge et al.

(10) **Patent No.:** US 6,195,698 B1
(45) **Date of Patent:** Feb. 27, 2001

(54) **METHOD FOR SELECTIVELY
RESTRICTING ACCESS TO COMPUTER
SYSTEMS**

5,784,464 * 7/1998 Akiyama et al. 713/201
5,889,860 * 3/1999 Eller et al. 713/202
5,953,419 * 9/1999 Lohstroh et al. 380/21
5,966,445 * 10/1999 Park et al. 380/25

(75) **Inventors:** Mark D. Lillibridge, Mountain View;
Martin Ahadi, Palo Alto; Krishna
Bharat, Santa Clara; Andrei Z.
Broder, Menlo Park, all of CA (US)

* cited by examiner

Primary Examiner—Zarni Maung
Assistant Examiner—Philip B. Tran

(73) **Assignee:** Compaq Computer Corporation,
Houston, TX (US)

(74) *Attorney, Agent, or Firm*—Pennie & Edmonds LLP

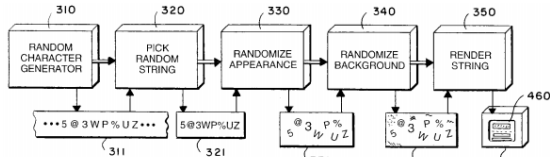
(*) **Notice:** Subject to any disclaimer, the term of this
patent is extended or adjusted under 35
U.S.C. 154(b) by 0 days.

(57) **ABSTRACT**

A computerized method selectively accepts access requests from a client computer connected to a server computer by a network. The server computer receives an access request from the client computer. In response, the server computer generates a predetermined number of random characters. The random characters are used to form a string in the server computer. The string is randomly modified either visually or

(21) **Appl. No.:** 09/059,193

(22) **Filed:** Apr. 13, 1998



CAPTCHA: Using Hard AI Problems For Security

Luis von Ahn¹, Manuel Blum¹, Nicholas J. Hopper¹, and John Langford²

¹ Computer Science Dept., Carnegie Mellon University, Pittsburgh PA 15213, USA

² IBM T.J. Watson Research Center, Yorktown Heights NY 10598, USA

Abstract. We introduce CAPTCHA, an automated test that humans can pass, but current computer programs can't pass: any program that has high success over a CAPTCHA can be used to solve an unsolved Artificial Intelligence (AI) problem. We provide several novel constructions of CAPTCHAs. Since CAPTCHAs have many applications in practical security, our approach introduces a new class of hard problems that can be exploited for security purposes. Much like research in cryptography has had a positive impact on algorithms for factoring and discrete log, we hope that the use of hard AI problems for security purposes allows us to advance the field of Artificial Intelligence. We introduce two families of AI problems that can be used to construct CAPTCHAs and we show that solutions to such problems can be used for steganographic communication. CAPTCHAs based on these AI problem families, then, imply a win-win situation: either the problems remain unsolved and there is a way to differentiate humans from computers, or the problems are solved and there is a way to communicate covertly on some channels.

1 Introduction

A CAPTCHA is a program that can generate and grade tests that: (A) most humans can pass, but (B) current computer programs can't pass. Such a program can be used to differentiate humans from computers and has many applications for practical security, including (but not limited to):

- Evitar a criação de milhares de e-mails por "bots"
- Evitar ataques para descobrir senhas de usuários em sites
- Evitar SPAM de e-mail

Processamento via CAPTCHA em 2007



200 milhões de CAPTCHAs por dia

Processamento via CAPTCHA em 2007



200 milhões de CAPTCHAs por dia

10 segundos para identificar e escrever cada palavras

Processamento via CAPTCHA em 2007



200 milhões de CAPTCHAs por dia

10 segundos para identificar e escrever cada palavras

500.000 horas de processamento humano por dia

Processamento via CAPTCHA em 2007

200 milhões de CAPTCHAs por dia

10 segundos para identificar e escrever cada palavras

500.000 horas de processamento humano por dia

This aged portion of society were distinguished from
↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
"niis aged pntkm at society were distinguished frow."

Processamento via CAPTCHA em 2007

200 milhões de CAPTCHAs por dia

10 segundos para identificar e escrever cada palavras

500.000 horas de processamento humano por dia

This aged **portion** of society were distinguished from

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

"niis aged pntkm at society were distinguished frow."

portion

Processamento via CAPTCHA em 2007

200 milhões de CAPTCHAs por dia

10 segundos para identificar e escrever cada palavras

500.000 horas de processamento humano por dia

This aged **portion** of society were distinguished from

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

niis aged pntkm at society were distinguished frow."

portion



Processamento via CAPTCHA em 2007

200 milhões de CAPTCHAs por dia

10 segundos para identificar e escrever cada palavras

500.000 horas de processamento humano por dia

This aged **portion** of society were distinguished from

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

"niis aged pntkm at society were distinguished frow."

portion



Processamento via CAPTCHA em 2007

200 milhões de CAPTCHAs por dia

10 segundos para identificar e escrever cada palavras

500.000 horas de processamento humano por dia

This aged **portion** of society were distinguished from

↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

"niis aged pntkm at society were distinguished frow."

portion




portion conhecimento

Type the two words:

reCAPTCHA™
stop spam.
read books.

fortion portion porcion

portion portion portion

partion portion portion

Aquisição pela Google em 2009



Official Blog

Insights from Googlers into our products, technology, and the Google culture

Teaching computers to read: Google acquires reCAPTCHA

September 16, 2009

Google Acquires reCAPTCHA

The image above is a [CAPTCHA](#) – you can read it, but computers have a harder time interpreting the letters. We tried to make it hard for computers to recognize because we wanted to give humans the scoop first, but we're happy to announce to everybody now that Google has acquired [reCAPTCHA](#), a company that provides CAPTCHAs to help protect more than 100,000 websites from spam and fraud.

Labels ▼

Archive ▼

Feed

Google on

YouTube 7M



Dados impressionantes

100 milhões de palavras digitalizadas por dia (2011)
Mínimo de 1 bilhão de reCAPTCHAs por dia



Dados impressionantes

100 milhões de palavras digitalizadas por dia (2011)
Mínimo de 1 bilhão de reCAPTCHAs por dia

Acervo do New York Times (1851-2011)
Mais de 13 milhões de artigos



Dados impressionantes

100 milhões de palavras digitalizadas por dia (2011)
Mínimo de 1 bilhão de reCAPTCHAs por dia

Acervo do New York Times (1851-2011)
Mais de 13 milhões de artigos

Google Books (2002)
Digitalizar todos os livros do mundo

Total de exemplares únicos
134.021.533

Total digitalizado até 2015
25.000.000

Dados impressionantes

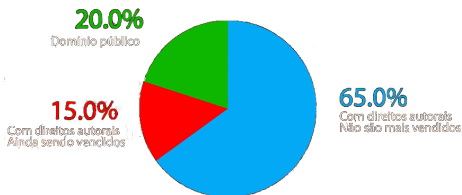
100 milhões de palavras digitalizadas por dia (2011)
Mínimo de 1 bilhão de reCAPTCHAs por dia

Acervo do New York Times (1851-2011)
Mais de 13 milhões de artigos

Google Books (2002)
Digitalizar todos os livros do mundo

Total de exemplares únicos
134.021.533

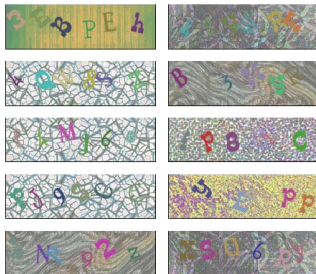
Total digitalizado até 2015
25.000.000



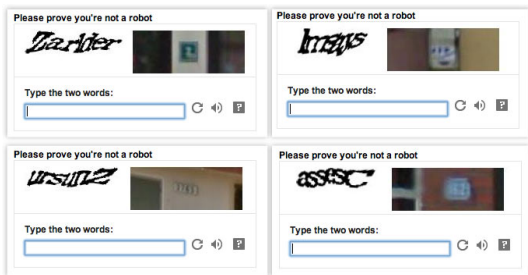
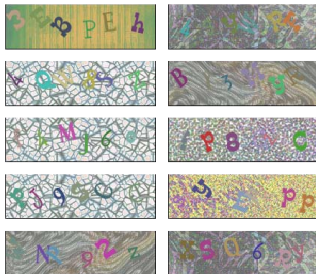
Variações do reCAPTCHA



Variações do reCAPTCHA



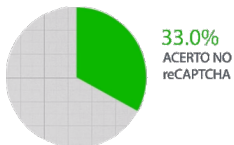
Variações do reCAPTCHA



Eficiência reCAPTCHA 2014



Seres humanos



Eficiência reCAPTCHA 2014

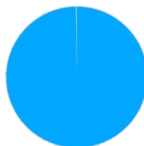


Seres humanos



33.0%
ACERTO NO
reCAPTCHA

Inteligência Artificial



99.8%
ACERTO NO
reCAPTCHA

Eficiência reCAPTCHA 2014

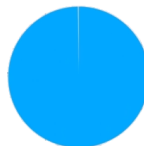


Seres humanos

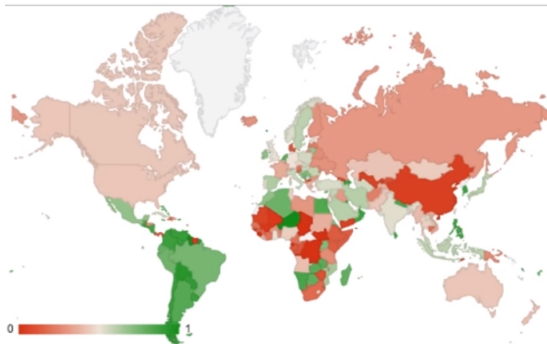


33.0%
ACERTO NO
reCAPTCHA

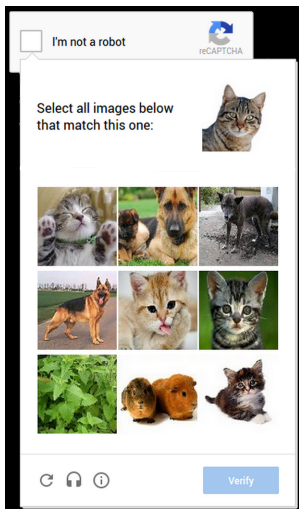
Inteligência Artificial



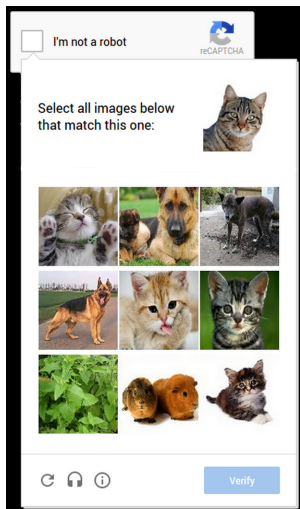
99.8%
ACERTO NO
reCAPTCHA



noCAPTCHA reCAPTCHA



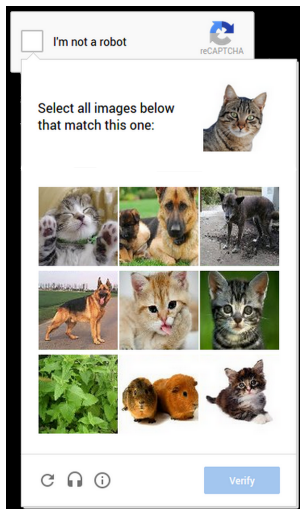
noCAPTCHA reCAPTCHA



Parâmetros de verificação

- Endereço de IP
- Movimentos do mouse
- Locais de clique
- Velocidade das ações
- Rolagem da pagina
- Cookies do navegador

noCAPTCHA reCAPTCHA



Parâmetros de verificação

- Endereço de IP
- Movimentos do mouse
- Locais de clique
- Velocidade das ações
- Rolagem da pagina
- Cookies do navegador

Utilização do processamento

- Busca reversa por imagens
- Google maps
- Google fotos
- Carro autônomo

Eficiência noCAPTCHA reCAPTCHA 2014



Seres humanos



96.0%
ACERTO NO
reCAPTCHA

Eficiência noCAPTCHA reCAPTCHA 2014



Seres humanos



96.0%
ACERTO NO
reCAPTCHA

Inteligência Artificial



5.0%
ACERTO NO
reCAPTCHA

Eficiência noCAPTCHA reCAPTCHA 2014



Seres humanos



96.0%
ACERTO NO
reCAPTCHA

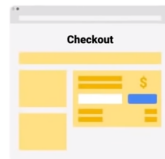
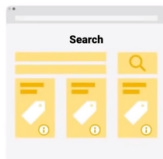
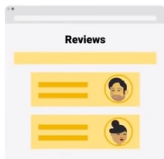
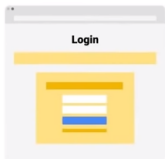
Inteligência Artificial



5.0%
ACERTO NO
reCAPTCHA



Invisible reCAPTCHA



DIREÇÃO AUTÔNOMA

ALVINN



TECNICAS AVANÇADAS

Técnicas Avançadas

Avaliação do problema

- Treino e validação cruzada
- Viés e variância
- Análise de erro
- Precisão e revocação

Máquina de vetores de suporte

- Núcleo Gaussiano
- Núcleos "esotéricos": String, χ^2 , histograma, etc...
- SVM multi-classe
- SVM transdutivo e estruturado

CLASSIFICADOR DE SPAM



Exemplo de e-mail SPAM???

```
> Anyone knows how much it costs to host a web portal ?
```

```
>
```

```
Well, it depends on how many visitors youre expecting. This can be  
anywhere from less than 10 bucks a month to a couple of $100. You  
should checkout http://www.rackspace.com/ or perhaps Amazon EC2 if  
youre running something big..
```

```
To unsubscribe yourself from this mailing list, send an email to:  
groupname-unsubscribe@egroups.com
```

Pré-processamento do e-mail



- Letras minúsculas
MachineLEARNING ⇒

Pré-processamento do e-mail



- Letras minúsculas
MachineLEARNING \Rightarrow machinelearning



Pré-processamento do e-mail

- Letras minúsculas
MachineLEARNING \Rightarrow machinelearning
- Remoção do HTML
<title>MachineLearning</title> \Rightarrow



Pré-processamento do e-mail

- Letras minúsculas
MachineLEARNING \Rightarrow machinelearning
- Remoção do HTML
<title>MachineLearning</title> \Rightarrow MachineLearning



Pré-processamento do e-mail

- Letras minúsculas
MachineLEARNING \Rightarrow machinelearning
- Remoção do HTML
<title>MachineLearning</title> \Rightarrow MachineLearning
- Normalização de URL
professorpalhares.com.br \Rightarrow



Pré-processamento do e-mail

- Letras minúsculas
MachineLEARNING \Rightarrow machinelearning
- Remoção do HTML
<title>MachineLearning</title> \Rightarrow MachineLearning
- Normalização de URL
professorpalhares.com.br \Rightarrow "endereçohttp"



Pré-processamento do e-mail

- Letras minúsculas
MachineLEARNING \Rightarrow machinelearning
- Remoção do HTML
<title>MachineLearning</title> \Rightarrow MachineLearning
- Normalização de URL
professorpalhares.com.br \Rightarrow "endereçohttp"
- Normalização de email
eu@professorpalhares.com.br \Rightarrow



Pré-processamento do e-mail

- Letras minúsculas
MachineLEARNING \Rightarrow machinelearning
- Remoção do HTML
<title>MachineLearning</title> \Rightarrow MachineLearning
- Normalização de URL
professorpalhares.com.br \Rightarrow "endereçohttp"
- Normalização de email
eu@professorpalhares.com.br \Rightarrow "endereçoemail"



Pré-processamento do e-mail

- Letras minúsculas
MachineLEARNING \Rightarrow machinelearning
- Remoção do HTML
<title>MachineLearning</title> \Rightarrow MachineLearning
- Normalização de URL
professorpalhares.com.br \Rightarrow "endereçohttp"
- Normalização de email
eu@professorpalhares.com.br \Rightarrow "endereçoemail"
- Normalização de números
1987 \Rightarrow



Pré-processamento do e-mail

- Letras minúsculas
MachineLEARNING \Rightarrow machinelearning
- Remoção do HTML
<title>MachineLearning</title> \Rightarrow MachineLearning
- Normalização de URL
professorpalhares.com.br \Rightarrow "endereçohttp"
- Normalização de email
eu@professorpalhares.com.br \Rightarrow "endereçoemail"
- Normalização de números
1987 \Rightarrow "número"



Pré-processamento do e-mail

- Letras minúsculas
MachineLEARNING \Rightarrow machinelearning
- Remoção do HTML
<title>MachineLearning</title> \Rightarrow MachineLearning
- Normalização de URL
professorpalhares.com.br \Rightarrow "endereçohttp"
- Normalização de email
eu@professorpalhares.com.br \Rightarrow "endereçoemail"
- Normalização de números
1987 \Rightarrow "número"
- Normalização de símbolos
\$ \Rightarrow



Pré-processamento do e-mail

- Letras minúsculas
MachineLEARNING \Rightarrow machinelearning
- Remoção do HTML
<title>MachineLearning</title> \Rightarrow MachineLearning
- Normalização de URL
professorpalhares.com.br \Rightarrow "endereçohttp"
- Normalização de email
eu@professorpalhares.com.br \Rightarrow "endereçoemail"
- Normalização de números
1987 \Rightarrow "número"
- Normalização de símbolos
\$ \Rightarrow "dollar"



Pré-processamento do e-mail

- Letras minúsculas
MachineLEARNING \Rightarrow machinelearning
- Remoção do HTML
<title>MachineLearning</title> \Rightarrow MachineLearning
- Normalização de URL
professorpalhares.com.br \Rightarrow "endereçohttp"
- Normalização de email
eu@professorpalhares.com.br \Rightarrow "endereçoemail"
- Normalização de números
1987 \Rightarrow "número"
- Normalização de símbolos
\$ \Rightarrow "dollar"
- Stemização de palavras
custou, custavam, custaria \Rightarrow



Pré-processamento do e-mail

- Letras minúsculas
MachineLEARNING \Rightarrow machinelearning
- Remoção do HTML
<title>MachineLearning</title> \Rightarrow MachineLearning
- Normalização de URL
professorpalhares.com.br \Rightarrow "endereçohttp"
- Normalização de email
eu@professorpalhares.com.br \Rightarrow "endereçoemail"
- Normalização de números
1987 \Rightarrow "número"
- Normalização de símbolos
\$ \Rightarrow "dollar"
- Stemização de palavras
custou, custavam, custaria \Rightarrow "custo"



Pré-processamento do e-mail

- Letras minúsculas
MachineLEARNING \Rightarrow machinelearning
- Remoção do HTML
<title>MachineLearning</title> \Rightarrow MachineLearning
- Normalização de URL
professorpalhares.com.br \Rightarrow "endereçohttp"
- Normalização de email
eu@professorpalhares.com.br \Rightarrow "endereçoemail"
- Normalização de números
1987 \Rightarrow "número"
- Normalização de símbolos
\$ \Rightarrow "dollar"
- Stemização de palavras
custou, custavam, custaria \Rightarrow "custo"
- Remoção de não palavras
pontuação é removida e espaços são concatenados



Resultado do pré-processamento

```
anyon know how much it cost to host a web portal well it depend on  
how mani visitor your expect thi can be anywher from less than  
number buck a month to a coupl of dollarnumb you should checkout  
httpaddr or perhap amazon ecnumb if your run someth big to  
unsubscribe yourself from thi mail list send an email to emailaddr
```



Resultado do pré-processamento

```
anyon know how much it cost to host a web portal well it depend on
how mani visitor your expect thi can be anywher from less than
number buck a month to a coupl of dollarnumb you should checkout
httpaddr or perhap amazon ecnumb if your run someth big to
unsubscribe yourself from thi mail list send an email to emailaddr
```

Podemos utilizar uma lista para transformar as palavras em números

1. aa	86. anyon	1898. zero
2. ab	...	1899. zip
3. abil	916. know	
...	...	



Resultado do pré-processamento

```
anyon know how much it cost to host a web portal well it depend on
how mani visitor your expect thi can be anywher from less than
number buck a month to a coupl of dollarnumb you should checkout
httpaddr or perhap amazon ecnumb if your run someth big to
unsubscribe yourself from thi mail list send an email to emailaddr
```

Podemos utilizar uma lista para transformar as palavras em números

1. aa	86. anyon	1898. zero
2. ab	...	1899. zip
3. abil	916. know	
...	...	

Nosso e-mail agora pode ser representados apenas por números

```
86 916 794 1077 883 370 1699 790 1822 1831 883 431 1171 794 1002
1893 1364 592 1676 238 162 89 688 945 1663 1120 1062 1699 375 1162
479 1893 1510 799 1182 1237 810 1895 1440 1547 181 1699 1758 1896
688 1676 992 961 1477 71 530 1699 531
```

Aplicação de SVM

Comparando as palavras do e-mail com uma BlackList de SPAM, podemos classifica-las.

our click remov garante visit basenumb dollar will price pleas
nbsp most lo ga dollarnumb

$$w = \begin{bmatrix} \textit{anyon} \\ \textit{know} \\ \vdots \\ \textit{visitor} \\ \vdots \\ \textit{dollarnumb} \\ \vdots \\ \textit{to} \\ \textit{emailaddr} \end{bmatrix} \Rightarrow n = \begin{bmatrix} 86 \\ 916 \\ \vdots \\ 1750 \\ \vdots \\ 450 \\ \vdots \\ 1699 \\ 531 \end{bmatrix} \Rightarrow x = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 1 \\ \vdots \\ 0 \\ 0 \end{bmatrix}$$

APLICAÇÕES EM ECÔNOMIA

Machine learning na bolsa

Table 17

Classification of the reviewed articles about financial market prediction using machine learning techniques. Note: AUC: Area Under the receiver operating characteristic Curve; GMM: Generalized Method of Moments; MAD: Mean Absolute Deviation.

Reference	Market/s	Asset/s	Predictive variable/s	Prediction/s	Main method/s	Performance Measure/s
Al Nasser et al. (2015)	USA	Index	Text	Direction	Analysis of sentiment	Return
Ang and Quek (2006)	Singapore	Stocks	TA	Prices	Neural networks	Return
Armano et al. (2005)	USA, Italy	Indices	TA	Prices	Neural networks, GA	Sharpe rate
Ballings et al. (2015)	Europe	Stocks	Fundamentalist	Direction	Neural networks, SVM, kNN, and RF	AUC
Barak et al. (2017)	Iran	Stocks	Fundamentalist	Return and risk	Neural networks, SVM, decision trees	Accuracy
Bezerra and Albuquerque (2017)	Brazil, Japan	Indices	Prices	Volatility	SVR, GARCH	MAE, RMSE
Cao et al. (2005)	China	Stocks	Fundamentalist	Return	Neural networks, CAPM	MAD, MAPE, MSE
Chang and Fan (2008)	Taiwan	Index	TA	Prices	kNN, DWT, fuzzy logic	MAPE
Chang et al. (2009)	Taiwan	Stocks	TA	Direction	Neural networks, CBR	Return
Chen et al. (2003)	Taiwan	Index	Fundamentalist	Return	Neural networks, GMM	Return
Chen et al. (2014)	Taiwan, Hong Kong	Indices	TA	Prices	Fuzzy logic	RMSE
Chen and Chen (2016)	USA, Taiwan	Indices	TA	Return	Pattern recognition	Return
Chen et al. (2017)	China	Indices	TA	Direction	SVM, kNN	MAPE, RMSE, AUC
Chiang et al. (2016)	Multiple	Indices	TA	Direction	Neural networks	Accuracy, return
Enke and Thawornwong (2005)	USA	Index	Fundamentalist	Direction	Neural networks	RMSE
Fernandez-Rodríguez et al. (2000)	Spain	Index	Prices	Direction	Neural networks	Accuracy, Sharpe rate
Gorenc Novak and Velušček (2016)	USA	Stocks	TA	Direction	SVM	Return, Sharpe rate
Hájek et al. (2013)	USA	Stocks	Fundamentalist	Return	Neural networks, SVR, analysis of sentiment	MSE
Hassan et al. (2007)	USA	Stocks	Prices	Prices	Neural networks, GA	MAPE
Huang and Tsai (2009)	Taiwan	Index	TA	Prices	SVR	MSE, MAE, MAPE
Kara et al. (2011)	Turkey	Index	TA	Direction	Neural networks, SVM	Accuracy
Kamstra and Donaldson (1996)	Multiple	Indices	Prices	Volatility	Neural networks	MSE, MAE
Kim and Han (2000)	Korea	Index	TA	Direction	Neural networks, GA	Accuracy
Kimoto et al. (1990)	Japan	Index	Fundamentalist	Direction	Neural networks	MAPE
Krauss et al. (2017)	USA	Stocks	Prices	Returns	Neural networks, RF, decision trees	Return, Sharpe rate
Labossiere et al. (2015)	Brazil	Stocks	Indices	Maximums, minimums.	Neural networks	MAE, MAPE, RMSE
Leigh et al. (2002)	USA	Index	Prices, volume	Prices	Neural networks, GA	Return
Leung et al. (2000)	USA, United Kingdom, Japan	Indices	Fundamentalist.	Return	Neural networks, LDA, regressions	Return
Li and Kuo (2008)	Taiwan	Indices	Prices	Prices	DWT, SOM	MSE, MAE
Mo and Wang (2017)	China, USA	Indices	Prices	Correlation	Neural networks	MAE, RMSE, MAPE
Oliveira et al. (2017)	Multiple	Indices	Text	Return, volume, volatility	Neural networks, SVM, RF	MAE
Pai and Lin (2005)	USA	Stocks	Prices	Prices	SVM	MAE, MAPE, MSE, RMSE
Pan et al. (2017)	USA	Index	Fundamentalist, prices	Prices	SVM	RMSE, MAE
Patel et al. (2015)	India	Indices, stocks	TA	Direction	Neural networks, SVM, RF, NB	Accuracy

Machine learning na bolsa

Table 18

Main forecasting techniques applied by each reviewed reference.

Main Method	Number of References	References
Neural Networks	42	Ang and Quek (2006), Armano et al. (2005), Ballings et al. (2015), Barak et al. (2017), Cao et al. (2005), Chang et al. (2009), Chen et al. (2003), Chiang et al. (2016), Enke and Thawornwong (2005), Fernandez-Rodriguez et al. (2000), Hájek et al. (2013), Hassan et al. (2007), Kara et al. (2011), Kamstra and Donaldson (1996), Kim and Han (2000), Kimoto et al. (1990), Krauss et al. (2017), Laboissiere et al. (2015), Leigh et al. (2002), Leung et al. (2000), Mo and Wang (2017), Oliveira et al. (2017), Patel et al. (2015), Pei et al. (2017), Rodríguez-González et al. (2011), Thawornwong et al. (2003), Tsai and Hsiao (2010), Tsai et al. (1998), Wang et al. (2012), Weng et al. (2017), Yan et al. (2017), Yoon et al. (1993), Zhong and Enke (2017), Abu-Mostafa and Ariya (1996), Donaldson and Kamstra (1999), Enke and Thawornwong (2005), Huang et al. (2005), Kim (2003), Kumar and Thenmozhi (2014), Tay and Cao (2001), Thawornwong and Enke (2004), Lahmiri (2014a), Lahmiri and Boukadoum (2015)
SVM/SVR	20	Ballings et al. (2015), Barak et al. (2017), Bezerra and Albuquerque (2017), Chen et al. (2017), Gorenc Nowak and Velušček (2016), Hájek et al. (2013), Huang and Tsai (2009), Kara et al. (2011), Oliveira et al. (2017), Pai and Lin (2005), Pan et al. (2017), Patel et al. (2015), Schumaker and Chen (2009), Weng et al. (2017), Yu et al. (2009), Huang et al. (2005), Kim (2003), Kumar and Thenmozhi (2014), Tay and Cao (2001), Lahmiri (2014b)
RF/Decision Trees	7	Ballings et al. (2015), Barak et al. (2017), Krauss et al. (2017), Oliveira et al. (2017), Patel et al. (2015), Weng et al. (2017), Kumar and Thenmozhi (2014)
Sentiment/Text Analysis	5	Al Nasser et al. (2015), Hájek et al. (2013), Schumaker and Chen (2009), Weng et al. (2017), Oliveira et al. (2017)
kNN	4	Ballings et al. (2015), Chang and Fan (2008), Chen et al. (2017), Zhang et al. (2017)
ARIMA/GARCH	4	Bezerra and Albuquerque (2017), Donaldson and Kamstra (1999), Zhang et al. (2017), Kumar and Thenmozhi (2014)
Fuzzy Logic	4	Chang and Fan (2008), Chen et al. (2014), Wang (2002), Wang (2003)
LDA	3	Leung et al. (2000), Yoon et al. (1993), Huang et al. (2005)
NB	1	Patel et al. (2015)

Table 19

Main optimization techniques applied by each reviewed reference.

Main Method	Number of References	References
GA	11	Armano et al. (2005), Bezerra and Albuquerque (2017), Hassan et al. (2007), Kim and Han (2000), Leigh et al. (2002), Wang et al. (2012), Yu et al. (2009), Donaldson and Kamstra (1999), Göçken et al. (2016), Chen and Chen (2016), Tsai and Hsiao (2010)
DWT	6	Li and Kuo (2008), Chang and Fan (2008), Chiang et al. (2016), Ortega and Khashanah (2014), Xiao et al. (2013), Lahmiri (2014a)
PCA	3	Zhong and Enke (2017), Son, Noh, and Lee (2012), Tsai and Hsiao (2010)
PSO	3	Yan et al. (2017), Chiang et al. (2016), Xiao et al. (2013)
SOM	2	Li and Kuo (2008), Cao (2003)

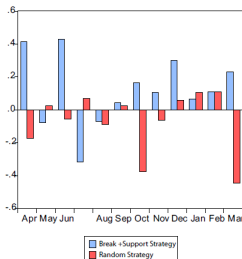
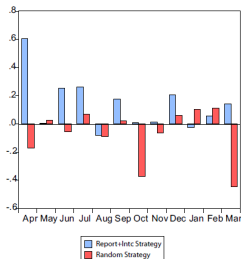
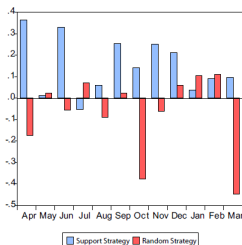
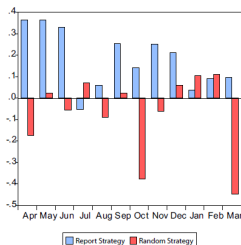
Análise de notícias - Tweet Term Trading

Table 3

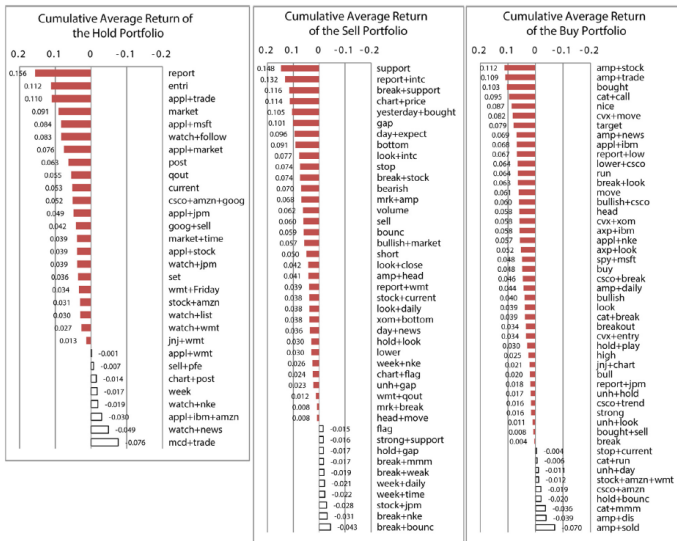
The decision rules for the individual occurrence of the term in the StockTwits postings.

Decision rule: If the term	{ bearish botoom bounc flag lower sell short stop su pport volume }	appears in a tweet message then the decision would be Sell
Decision rule: If the term	{ current entri market post report set week }	appears in a tweet message then the decision would be Hold
Decision rule: If the term	{ bought break breakout bull bullish buy head high look move nice run }	appears in a tweet message then the decision would be Buy

Análise de notícias - Tweet Term Trading



Análise de notícias - Tweet Term Trading



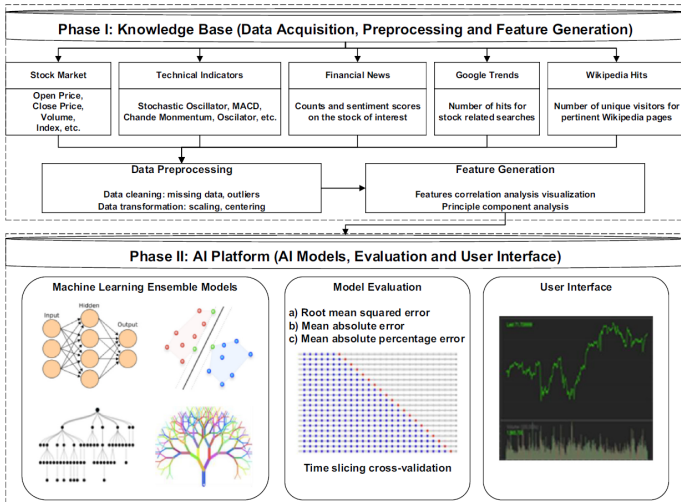


Comparação de várias técnicas

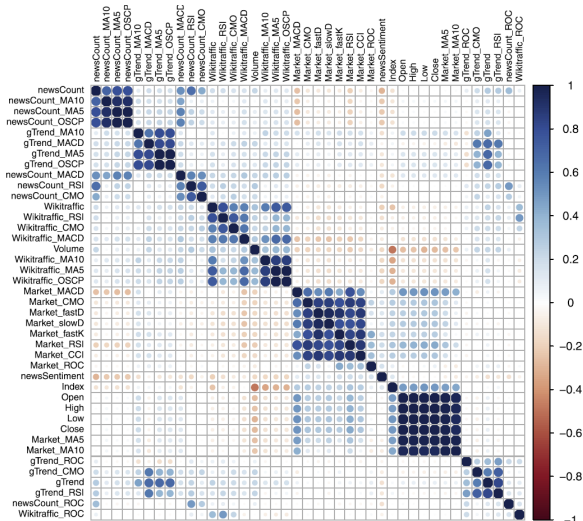
- Fontes de dados
 - Preditores tradicionais
 - Notícias financeiras
 - Google Trends
 - Wikipédia
 - Indicadores técnicos
- Técnicas utilizadas
 - Neural Network
 - Support Vector Machine
 - Boosted Regression Tree
 - Random Forest
 - Séries Temporais
 - RSE/MAE/MAPE

<https://github.com/martinwg/stockprediction>

Comparação de várias técnicas



Comparação de várias técnicas



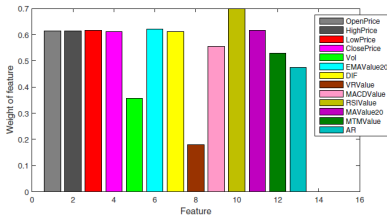
Comparação de várias técnicas

Table 5

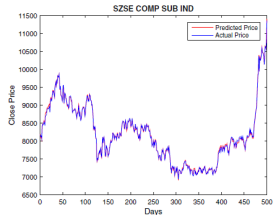
The performance of the ensemble models for the 1-day ahead price prediction on different stocks.

		No PCA			w/PCA		
		MAE	MAPE	RMSE	MAE	MAPE	RMSE
Amazon.com, Inc. (\$AMZN)	NNRE	5.9176	0.8308	7.6973	6.6202	<u>0.9296</u>	8.6108
	SVRE	12.2509	1.7032	15.5321	17.9721	2.5373	22.6149
	BRT	4.2573	0.6009	5.6486	7.2265	1.0279	9.6373
	RFR	3.2064	<u>0.4538</u>	4.5678	9.5250	1.3088	12.7378
	HYBRID	5.0736	0.7115	6.3352	7.4839	1.0472	9.0729
Apple Inc. (\$AAPL)	NNRE	0.8276	0.8104	1.0929	0.8969	0.8768	1.1781
	SVRE	1.9908	1.9623	2.5911	2.8127	2.7419	3.7080
	BRT	0.5194	0.5064	0.7030	0.5318	<u>0.5165</u>	0.6927
	RFR	0.3905	<u>0.3831</u>	0.5481	0.7609	0.7507	0.9985
	HYBRID	0.7322	0.7204	0.9918	0.9896	0.9636	1.2188
Chevron Corporation (\$CVX)	NNRE	0.6260	0.7951	0.8192	0.6645	0.8458	0.8795
	SVRE	0.5521	0.6952	0.6535	0.3812	0.4799	0.3821
	BRT	0.4755	0.6037	0.6532	0.2626	<u>0.3312</u>	0.3449
	RFR	0.3118	<u>0.3962</u>	0.4335	0.6537	0.8187	0.8888
	HYBRID	0.4530	0.5755	0.6088	0.4451	0.5625	0.5855
The Coca-Cola Company (\$KO)	NNRE	0.2372	0.5663	0.3277	0.2477	0.5904	0.3405
	SVRE	0.5100	1.2150	0.6748	0.6927	1.6560	0.9037
	BRT	0.1650	0.3941	0.2184	0.1161	<u>0.2768</u>	0.1488
	RFR	0.1105	<u>0.2643</u>	0.1595	0.2098	0.4964	0.2921
	HYBRID	0.1651	0.3946	0.2267	0.1653	0.3929	0.2226
The Walt Disney Company (\$DIS)	NNRE	0.5341	0.5325	0.7111	0.6318	0.6298	0.8699
	SVRE	0.3974	0.4064	0.3985	1.1131	1.1384	1.1161
	BRT	0.2845	<u>0.2866</u>	0.3661	0.4624	<u>0.4611</u>	0.6346
	RFR	0.3410	0.3366	0.4664	0.8705	0.8694	1.1494
	HYBRID	0.3408	0.3398	0.4689	0.5379	0.5384	0.7096

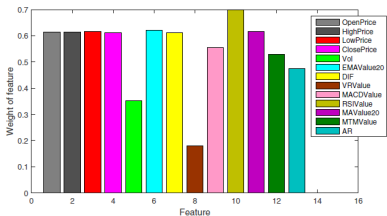
SVM (aprendizagem) + KNN (otimização)



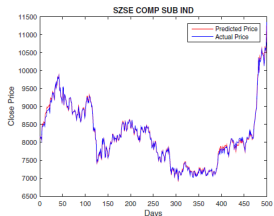
(a) The distribution of the feature weight for 20 days ahead



(b) 20 days ahead prediction of closing price

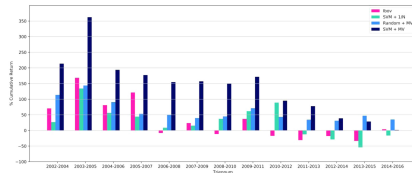
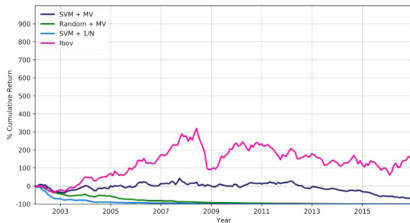
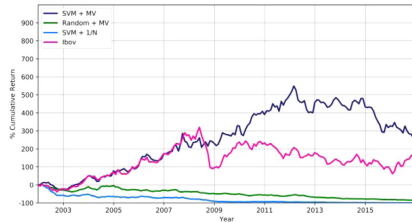
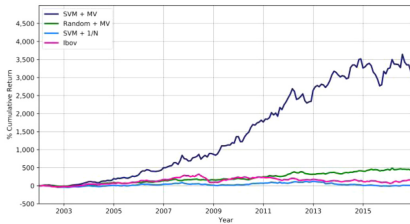


(c) The distribution of the feature weight for 30 days ahead



(d) 30 days ahead prediction of closing price

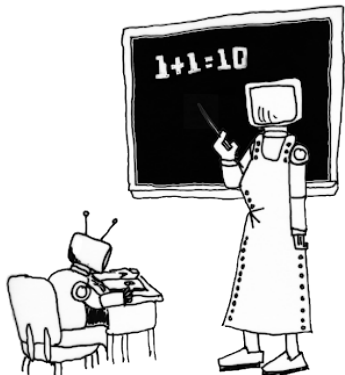
SVM (aprendizagem) + MV (otimização)



Aprendizagem não-supervisionada



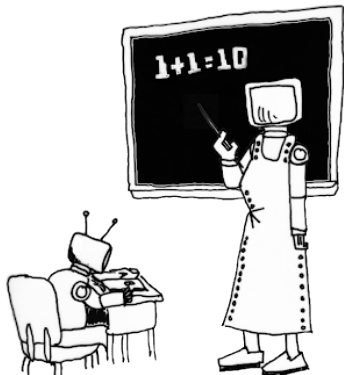
Aprendizagem supervisionada



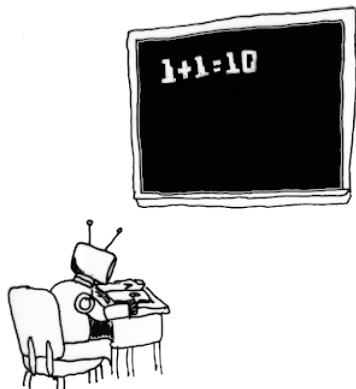
Aprendizagem não-supervisionada



Aprendizagem supervisionada



Aprendizagem não-supervisionada



Referências Bibliográficas



Palhares Junior, E. *Como fazer uma apresentação decente usando Latex*, São Paulo, setembro de 2017.