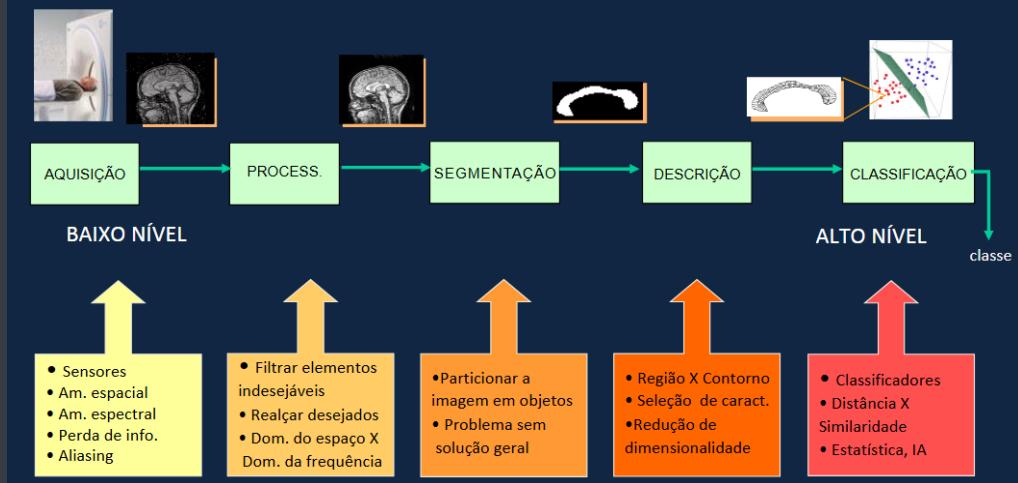


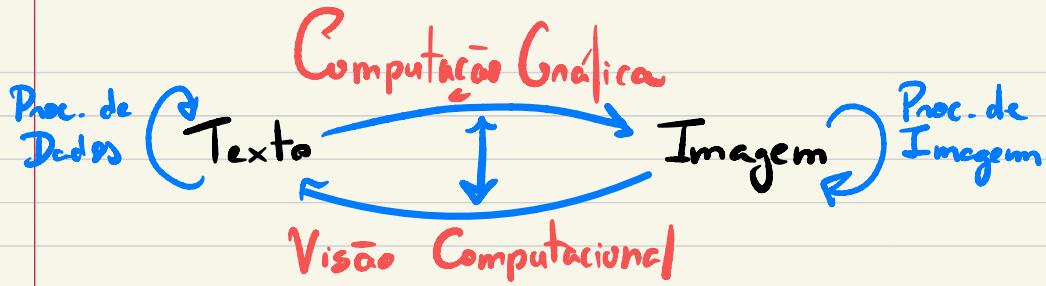
Resumo 1a Prova PAI

O Processo de Visão Computacional Clássico



Sophia
Carnezza





> Imagem digital = uma matriz onde cada pixel é um nível de cinza ou cor.

tam. de armazenamento em bytes

- n.º de linhas e colunas da imagem = resolução
- pixel de mais dimensão = voxcel
- tamanho de uma imagem = $L \times C \times \lg N$ bytes

linhas \leftrightarrow colunas 8² → 4 bits distintos

ex: imagem de 256×512 pixels
c, 256 tonalidades de cinza = tam = $\frac{2^8 \cdot 2^9 \cdot 8}{8} = 128$ KBytes

Amostragem Espacial
define sua resolução

vs Amostragem Espectral
define sua quantização
(n.º de tons / níveis de cinza
ou n.º de bits por pixel)

> Transformações Geométricas:

↳ **Translação**: move um ponto para outro local adicionando valores de translação às coordenadas

$$\begin{array}{l} \xrightarrow{\quad T \quad} P'(x', y') \\ P(x, y) \end{array}$$

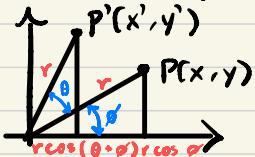
$$x' = x + dx \quad \text{ou} \quad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} dx \\ dy \end{bmatrix} \quad \text{ou} \quad P' = P + T$$

↳ **Escala**: altera o tamanho do objeto multiplicando as coordenadas dos pontos por uma escala

$$x' = x \cdot S_x \quad y' = y \cdot S_y \quad \text{ou} \quad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad \text{ou} \quad P' = SP$$

- se $S_x = S_y$, a escala é uniforme. Se $S_x \neq S_y$, não é.
- se $S_x, S_y < 1$, o tam. é reduzido e o objeto se move p/ mais perto da origem, e vice-versa.

↳ **rotacão**: gira pontos por um ângulo θ em relação à origem ($\theta > 0$ = rotacão no sentido anti-horário)



$$\cos(\theta) = x/r \quad \text{ou} \quad x = r \cos(\theta)$$

$$\sin(\theta) = y/r \quad \text{ou} \quad y = r \sin(\theta)$$

$$\cos(\phi + \theta) = x'/r \quad \text{ou} \quad x' = r \cos(\phi + \theta) = r \cos(\phi) \cos(\theta) - r \sin(\phi) \sin(\theta)$$

$$\sin(\phi + \theta) = y'/r \quad \text{ou} \quad y' = r \sin(\phi + \theta) = r \cos(\phi) \sin(\theta) + r \sin(\phi) \cos(\theta)$$



$$x' = x \cos(\theta) - y \sin(\theta) \quad \text{ou} \quad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad \text{ou} \quad P' = R P$$

↳ **espelhamento ou reflexao**

$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x_d \\ y_d \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix}$$

> Morfologia

- Subconjunto $A \subseteq B$

- União $A \cup B$

- Interseção $A \cap B$

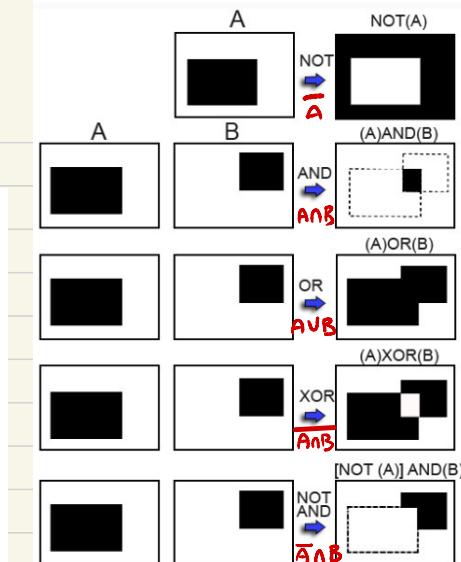
Conjuntos disjuntos/mutualmente exclusivos: $A \cap B = \emptyset$

- Complemento $A^c \equiv \{w \mid w \notin A\}$

- Diferença $A - B \equiv \{w \mid w \in A, w \notin B\} = A \cap B^c$

- Reflexão $B \equiv \{w \mid w = -b, \forall b \in B\}$

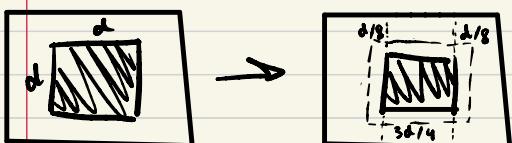
- Translação $(A)z \equiv \{c \mid c = a + z, \forall a \in A\}$



> Elemento estruturante = conjunto de pixels com valor de frente (foreground = 1). As demais posicões são fundo (0)

→ erosão = usada p/ encolher o elemento A usando o elemento estruturante B:

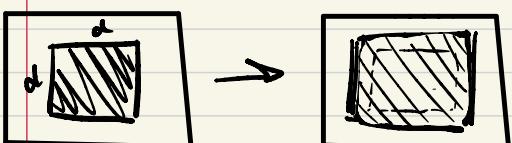
$$A \ominus B = \{z \mid (B)z \subseteq A\}$$



erosão → um pixel central só permanece se TODOS os pixels da elemento estruturante (B) forem cobertos por pixels do objeto A quando B é centrado nele.

→ dilatação = expande um elemento A usando a reflexão de um elemento estruturante B:

$$A \oplus B = \{z \mid (\hat{B})z \cap A \neq \emptyset\}$$



dilatação → adicionamos pixels do elemento estruturante (B) a todos os pixels "1".

$$(A \ominus B)^c = A^c \oplus \hat{B} \rightarrow (\text{dilatação e erosão são duals uma da outra!})$$

remove elementos finos,
áreas isoladas.

→ **abertura** = suaviza contornos, elimina saliências

- primeiro erodir A por B, depois dilatar o resultado por B

$$A \circ B = (A \ominus B) \oplus B$$

→ **fechamento** = suaviza reções de contornos, funde quebras estreitas, concavidades longas e finas, elimina pequenos buracos e preenche lacunas em contornos.

- primeiro dilatar A por B e, em seguida, erodir o resultado por B

$$A \bullet B = (A \oplus B) \ominus B$$

→ **extração de contorno** = eroda A por B, então faça a diferença entre A e a erosão

$$\beta(A) = A - (A \ominus B)$$

→ **deteção de padrões** = transformada de "Hit or Miss", usa uma versão de elemento estruturante pr/ localizar padrões

$$I \oplus B = \{z | (B)_z \subseteq I\}$$

> Segmentação

→ partitiona imagens em regiões que representam os objetos da cena ou que possuam determinadas propriedades (textura, cor, etc)

pixels parecidos (por regiões) → homogeneidade tipos → descontinuidade
(cor, luz, intensidade)

conectividade = 2 pixels pertencem a uma região conectada se existir um caminho entre elas formado por pixels vizinhos

cores = RGB, CMY, HSI ou HSV

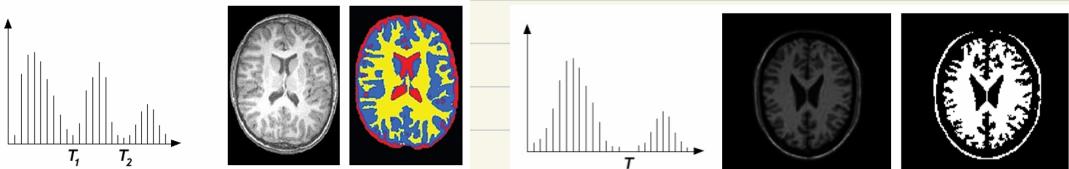
→ demonstra a frequência de ocorrência dos valores de uma variável em sua distribuição de propriedades

Segmentação por regiões

- limiarização: usar o histograma p/ definir valores de corte (limiar) p/ partitionar imagens em regiões.

→ T

- binarização: quando apenas um valor de limiar é definido.



→ une limiarização quando tiver um contraste claro e distinto entre o objeto e o fundo.

(o objeto x tem
regiões c/ valor
t > ?)

- algoritmo de rotulação: dado um limiar $t > ?$

- varrer a imagem → e ↓ cte achar um ponto do objeto X
- verificar a vizinhança de X ecriar um rótulo p/ X se nenhum dos pontos da vizinhança tiver rótulo. Se algum dos pontos tiver rótulo, ^{atracção} novo rótulo.
- Se tiver múltiplos rótulos dos pontos de vizinhança, intervir-lhos em uma lista de equivalência
- repetir essa etapa p/ os outros pontos e uniforizar rótulos equivalentes
- geralmente rótulo de X = 1 e o q não for X = 0.

(se adaptar
à imagem)

a) - algoritmo de otimização: iterar pelos valores do limiar t , do 1º ao penúltimo tom de cinza, buscando o que maximiza a variação entre classes.

* $p(i) = \text{probabilidade de ocorrer o tom de cinza } i$.

$$\sigma_b^2(t) = q_1(t)q_2(t)[m_1(t) - m_2(t)]^2$$

$$m_1(t) = \frac{1}{q_1(t)} \sum_{i=0}^t ip(i) \quad m_2(t) = \frac{1}{q_2(t)} \sum_{i=t+1}^{NC-1} ip(i)$$

$$q_1(t) = \sum_{i=0}^t p(i) \quad q_2(t) = \sum_{i=t+1}^{NC-1} p(i)$$

* O fundo será formado pelos tons de 0 a t e o objeto pelos tons $t+1$ a $NC-1$

enemies by taking stocks
of arms and a party of gunners
and a dozen or so of the best
artillery men of the town are
now steadily posted on the
island. Demolition work is being
done and the fortifications
are being strengthened. The
old gun emplacements are
being held there and certain
parts of the fortifications are
being repaired.

and wanted his letters to be kept by
the Archbishop of Geraldine
Dublin. Jackson would have
left Connaught at the other part
of Stony Bonwick, for A of
the Sum of two thousand
pounds, he left Connaught with
such but by this present
letter every one and Connaught
such or he there any day
between the dates or period of so
long as one year to reward where

Instrument Six Lettered Stockley
I know and part of Gurney
Andrew Jackson of the fourth
late Governor of all other parts
and late Governor of Monroe for a
part of the sum of two thousand
four hundred and fifty dollars
each and by these presents
he also excepts and conveys
Jackson in the town of
certain tracts or parcels of land
southern one thousand acre

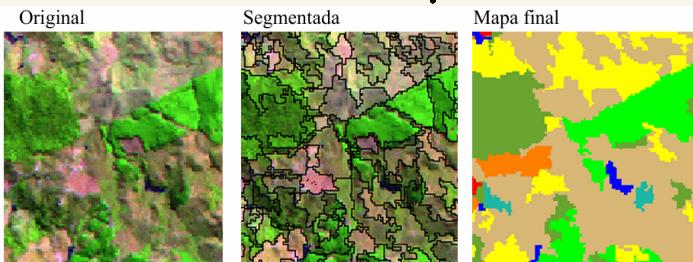
Limiarização Global

Limiarização Local

- crescimento de rugiāo!

Agrupa-nas regiões com base na similaridade. É útil quando se sabe a localização aproximada dos objetos (pág anterior e menu).

- ✓ escolhe-se uma **semente** x e um critério de similaridade c_1 relativamente à intensidade da semente.
 - ✓ se a condição de similaridade for satisfeita, atribui-se o mesmo rótulo de x aos vizinhos.
 - ✓ repete-se o processo até não ter pontos novos visitados.
 - ✓ escolhe-se outra semente não rotulada e repete tudo.

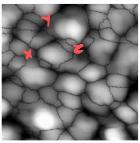
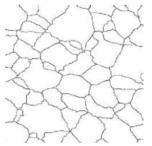
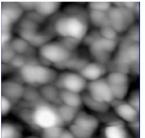
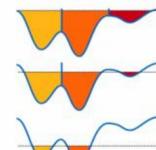


pode regredir
demais é o VZLW

- watershed: Beucher and Meyer

- rotule cada mínimo com um rotulo distinto
 - inicialize um conjunto S vazio.
 - extraia de S um nó x de altitude mínima F , ou seja $F(x) = \min \{ F(y) \mid y \in S\}$
 - atribua o rotulo de x a cada nó não rotulado y adjacente a x e insira y em S .
 - repita o passo 2 até que S esteja vazio.

Imagen vista como um mapa topológico a ser inundado



→ a intensidade dos pixels representa a "altitude"

Segmentação por Contorno

→ determinamos o subconjunto de pixels conectados que separam um objeto do resto da imagem. → detectam a base do objeto

- algoritmo de Rosenfeld:

→ percorrer a imagem até encontrar o 1º pixel contendo sinal de objeto.

→ atribuir o pixel a P e o pixel a esquerda a Q.

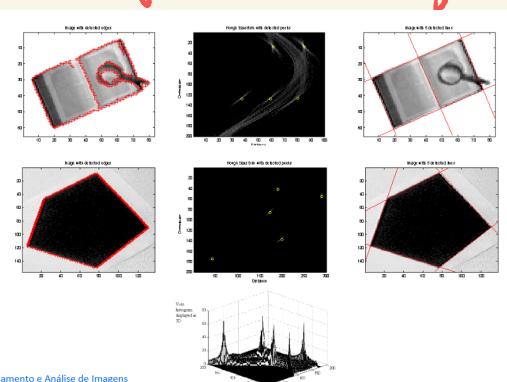
→ a partir de Q (no sentido horário) verifico se o próx. vizinho de P pertence ao fundo. Se sim, atribui a Q. Se não, a P e atribuímos P à vizinhança de contorno

→ repetir isso até que Q retorne ao valor inicial.



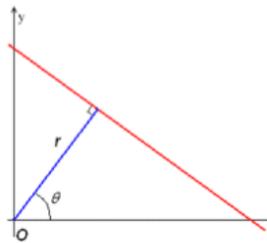
- transformada de Hough:

muito usado
pr detectar
retas ou
círculos



- Detecta linhas pela colinearidade de pontos

- Cada linha é da forma $y = ax + b$. Crie espaço de parâmetros (a, b) . A nova equação será: $b = -ax + y$
- Porém, o espaço (a, b) está no intervalo $[-\infty, \infty]$
- Crie espaço de parâmetros (ρ, θ) : representação polar da reta, onde ρ indica a distância entre a origem e a reta e θ sua orientação. Eles são intervalos finitos!



essa técnica transforma o espaço (x, y) em um espaço (ρ, θ) , que descreve retas

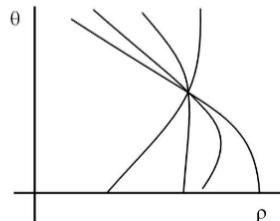
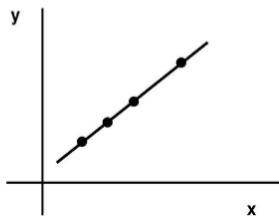
\downarrow distância \downarrow ângulo

Passo a passo:

- o espaço funcionará como uma variável **acumuladora**; inicie-o todo com 0. Para cada ponto da imagem original, computar (ρ, θ) para as retas que passam pelo ponto. Neste espaço, a equação é da forma:

$$\rho(\theta) = x \cdot \cos(\theta) + y \cdot \sin(\theta)$$

- Após computar as senóides para todos os pontos, detectar os contadores máximos. Estes são os (ρ, θ) de cada reta na imagem original.



* cada ponto da borda da imagem vota nos possíveis retas que poderiam passar por ele nesse espaço de parâmetros

* onde os votos se acumularem, um pico, corresponde a uma reta na imagem original.

Usar quando precisa de segmentações muito precisas e refinadas.

(SNARES) → quando há curvas muito complexas

2 - **contorno ativo**: uma curva é traçada dentro (ou em torno) do objeto a ser segmentado. A curva se expande ou contrai até encontrar a fronteira do objeto, moldando seu formato.

→ Representação e Descricão:

Representação

esqueletos/
eixos médios
p/ forma da
imagem

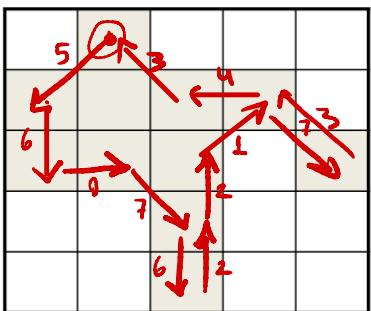
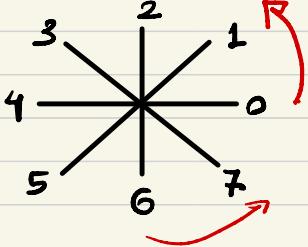
como a imagem é expressa ou codificada
p/ análise
características externas (contorno)
cor e.
internas (pixels)
p/ caracterizar objetos
por cor ou textura.

p/ forma da
imagem

→ representação por contorno

→ código de codia = representa a derivada do contorno de um objeto.

- "higher-left boundary pixel" é de onde geralmente começam.
- Negue em sentido horário!



CC: 5 6 0 7 6 2 2 1 7 3 4 3

* o "código de codia anterior" é o código de codia da parte de resto imediatamente anterior

→ e o anterior (C_{i-1}) ao primeiro é o último valor de C_i (pq forma um "loop").

- Objetivo da Esqueletização: Encontrar uma representação do objeto com apenas 1 pixel de espessura que mantenha sua estrutura e conectividade original. Pense em "descascar" o objeto até sobrar apenas seu "esqueleto".
- Algoritmo de Zhang & Wang: É um processo iterativo que remove pixels das bordas do objeto em passadas sucessivas. Ele usa um conjunto de condições para garantir que a remoção de um pixel:

 - Não quebre o objeto em partes desconectadas.
 - Não remova a ponta de uma linha.
 - Não apague um objeto por completo.

O processo continua até que nenhuma remoção de pixel seja mais possível sem violar essas regras.

representação por região

P ₉	P ₂	P ₃
P ₈	P ₁	P ₄
P ₇	P ₆	P ₅

→ eixos médios e esqueleto = extração do Esqueleto de Zhang & Wang:

trânsito $S(P_i)$

↓

quuntas vezes

há uma transiçao

de 0 pr. 1 ao

redor de P_i

(só depois de P_2)

→ Passo 1 → p/ cada ponto P_i do objeto, marque-o se:
 • $2 \leq V(P_i) \leq 6$ • $P_2 \cdot P_4 \cdot P_6 = 0$
 • $S(P_i) = 1$ * • $P_4 \cdot P_6 \cdot P_8 = 0$

→ remove os pontos marcados

→ Passo 2 → p/ cada ponto P_i do objeto, marque-o se:
 • $2 \leq V(P_i) \leq 6$ • $P_6 \cdot P_8 \cdot P_2 = 0$
 • $S(P_i) = 1$ • $P_8 \cdot P_2 \cdot P_4 = 0$

→ remove os pontos marcados

→ repita até não haver + pontos a serem removidos

→ introdução dos elementos virtuais de imagem
 p/ um formato + compreensível p/ nós

Descrição = extraer características que resultem em alguma info. quantitativa de interesse

> Características:

- área (nº de pixels do objeto)
- perímetro (nº de pixels do contorno do objeto)
- circularidade ou compactade
- diâmetro (dist. em linha reta de 2 pontos do contorno)
- raio máx. e min.
- retângulo báscio = menor retângulo circunscreto paralelo ao eixo de inclinação
- eixos máximos e mínimos = lados do retângulo báscio
- excentricidade = eixo máximo / eixo mínimo
- retangularidade = área / área retângulo báscio

$$\frac{1 \cdot (\text{qtd. Pares Codificados})}{\sqrt{2} \cdot (\text{qtd. Impares Cod. Codificados})}$$

$$C = \frac{P^2}{4\pi A}$$

* circularidade = área / área círculo perfeito.

* a circularidade resulta em 1 p/ um círculo perfeito.

representação
por contorno

$$\Phi - S_i = (C_i - C_{i-1} + 8) \bmod 8$$

(resto da divisão)

- curva phi-S = derivada do código de codificação:

representa a curvatura
do contorno:

- $P_{\Phi-S} = 0 \rightarrow$ contorno plano

- $1 \leq (P_{\Phi-S} + 8) \bmod 8 \leq 4 \rightarrow$ contorno convexo

- senão, contorno é côncavo

- nº de forma = normalização da curva Phi-S:

- calcular $(P_S + 8) \bmod 8$

- rotacionar o vetor resultante até que se obtenha o menor número octal.

- atrinuturas = função unidimensional de uma fronteira
↳ distância da fronteira ao centroíde (centro de distribuição de pixels do objeto)

*Centroíde:

$T = \text{nº de pixels do objeto}$

$S_x = \text{Soma dos coordenados}$

X de cada pixel do objeto

$S_y = \text{Soma dos coordenados}$

Y de cada pixel do objeto

$C_x = \text{coordenada } X \text{ do centroíde}$

$C_y = \text{coordenada } Y \text{ do centroíde}$

Temos:

$$C_x = S_x / T$$

$$C_y = S_y / T$$

- fecho-convexo = menor polígono convexo, H , que engloba todo o objeto S .

$$\text{deficiência convexa} = D = H - S$$

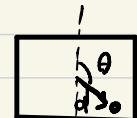


* Texturas

- padrões virtuais com certas propriedades de homogeneidade
- relacionado com coeficientes de:
 - uniformidade
 - regularidade
 - densidade
 - intensidade
 - rugosidade
 - apreza
 - variações de intensidade luminosa
- definidas por medidas de:
 - suavidade
 - rugosidade
 - regularidade

* Matrizes de Co-ocorrência

- $I(N, M)$ = imagem quantizada em G níveis de cinza
- P = matriz $G \times G$. Cada elemento dela designa a prob. de ocorrência simultânea de 2 níveis de cinza i, j



Operador $p(d, \theta)$



Operador $p(dx, dy)$

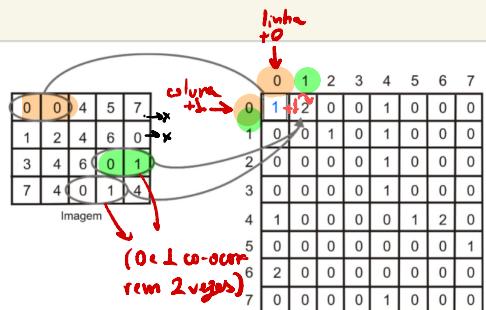
> na matriz de co-ocorrência circular, apenas d é usado!

Matrizes de Co-ocorrência

Assim:

1. Percorre-se a imagem na forma descrita pelo operador $p(d, \theta)$ ou $p(dx, dy)$ ou $p(d)$, contando-se quantas vezes um tom de cinza co-ocorre com outro na posição dada pelo operador

2. A matriz de probabilidades é calculada dividindo-se os valores de co-ocorrências pelo somatório da matriz.



Matriz de co-ocorrência de tons de cinza $C_{1,0}$. Para calcular $P_{1,0}$ dividir $C_{1,0}$ por 16

$C_{1,0} \rightarrow p(x, y) \rightarrow p(x+1, y+3)$

operadores de deslocamento

a partir de um pixel procurar um pixel vizinho

després de formar a matriz de co-ocorrência c/ 16x16 valores, montarmos a matriz de probab.

X Descritores de Haralick

Característica	Descrição	Fórmula Matemática
Homogeneidade	Distribuição de pixels.	$\sum_i \sum_j \frac{p(i,j)}{(1+ i-j)}$
Probabilidade Máxima	Indica a direção mais importante da textura a ser examinada.	$\max_{i,j} p(i,j)$
Entropia	Mede a informação contida em p, muitos valores nulos representam pouca informação.	$-\sum_i \sum_j p(i,j) \log_2 p(i,j)$

Entropia Variações / detalhes

Momento de diferenças ordem k	Distorção da imagem. Este descritor apresenta valores pequenos se p tiver maiores valores na diagonal principal.	$\sum_i \sum_j (i-j)^k p(i,j)$
Momento inverso de diferenças de ordem k	Inverso de contraste. Este descritor apresenta valores maiores pequenos se p tiver pequenos valores na diagonal principal.	$\sum_i \sum_j \frac{p(i,j)}{(i-j)^k}$
Energia ou Uniformidade	Retorna a soma dos elementos elevados ao quadrado dentro da matriz de co-ocorrência de tons de cinza. Faixa de valores possíveis: 0 a 1. A energia possui valor 1 para uma imagem constante (mesmo tom de cinza em toda a sua extensão).	$\sum_i \sum_j p^2(i,j)$
Variância ou Contraste	Retorna uma medida do contraste entre as intensidades de um pixel analisado e do pixel vizinho. A comparação é realizada em todos os pixels da imagem. Para uma imagem constante (mesmo tom de cinza em toda a extensão), o contraste é 0 (zero). Contraste da imagem corresponde ao Momento de ordem 2.	$\sum_i \sum_j (i-j)^2 p(i,j)$
Variância Inversa	Inverso de contraste.	$\sum_i \sum_j \frac{p(i,j)}{(i-j)^2}, i \neq j$

★ Momentos Invariantes de Hu

$f(x, y)$: intensidade de uma imagem digital

momento central
normalizado da imagem
por os índices p e q

$$M_{pq} = \sum_1^{nx} \sum_1^{ny} (x - \bar{x})^p (y - \bar{y})^q f(x, y)$$

valor do
pixel na posição
(x, y) da imagem

momento normalizado

$$n_{pq} = \frac{M_{pq}}{M_{00}^{\left(\frac{p+q}{2} + 1\right)}}$$

Momentos invariantes de Hu:

$$I_1 = \eta_{20} + \eta_{02}$$

$$I_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

$$I_3 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$I_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2$$

$$I_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

$$I_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

$$I_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

→ São um conjunto de 7 momentos calculados a partir de uma imagem que permanecem invariantes (seus valores não mudam).

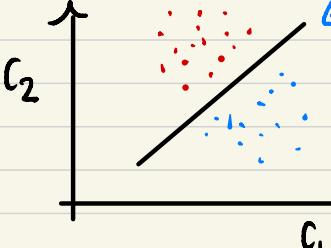
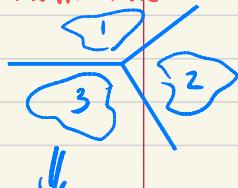
→ mesmo que a imagem sofra translação, referâncias ou mudanças de escala (redimensionamento), os objetos não reconhecidos nas imagens.

Classificação

última etapa do processo de vazio computacional

↳ como rotular objetos na imagem → FUNÇÃO DE DECISÃO

problemas multi-classe:



uma fronteira no espaço de características

⇒ cenário ideal: classes bem separadas (máximas distâncias com desvio padrão pequeno).

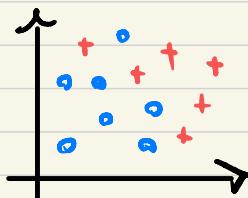
matriz de confusão

Vizinho + próximo (KNN)

um conjunto p/ cada classe

* passo a passo:

- 1- escolher um conjunto de treinamento
- 2- escolher características discriminantes
- 3- escolher um método / função de decisão
- 4- determinar os parâmetros a partir do conjunto de treino



"alarme falso"
erro tipo 1

- Falso Positivo (FP)
- Verdadeiro Positivo (TP)
- Falso Negativo (FN)
- Verdadeiro Negativo (TN)

$$\text{Especificidade} = \frac{TN}{(TN+FP)}$$

$$\text{Precisão} = \frac{TP}{(TP+FP)}$$

$$\text{Score F1 ou Dice} = \frac{2TP}{(2TP+FP+FN)}$$

$$\text{Acurácia} = \frac{TP+TN}{(TP+TN+FP+FN)}$$

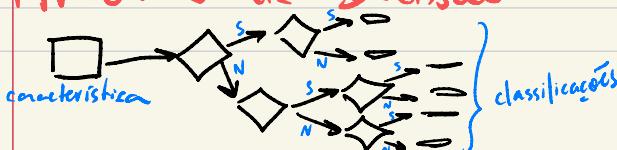
↳ porcentagem geral de acertos
(quanto ele acertou no total.
pode enganar em classes desbalanceadas)

$$\text{Sensibilidade} = \frac{TP}{(TP+FN)}$$

ou Revocação (TP+FN)
RECALL
(quanto dos positivos REAIS o modelo conseguiu encontrar)

↳ Confiança (de todos que o modelo falou que eram vdd, quais realmente eram).

Árvores de Decisão



* não liga p/ escala ou p/
como as características variam
(vai sempre escolher a de maior efeito)

Distância Euclidiana:

vetores {
 $P = (p_1, p_2 \dots p_n)$

$Q = (q_1, q_2 \dots q_n)$

$$d_e = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2}$$

cordenadas dos pontos

$$= \sqrt{\sum_{i=1}^n (p_i - q_i)^2}$$

* Função de decisão:

↳ o padrão x pertence a

classe C_i com média m_i se:

$$\sqrt{(x - m_i)^T (x - m_i)} < \sqrt{(x - m_j)^T (x - m_j)}, \forall i \neq j$$

* Variância

covariância

↓

normaliza o
espaço considerando a
estatística dos
dedos!

Distância de Mahalanobis:

$$d_i(x) = (x - m_i)^T C_i^{-1} (x - m_i)$$

+ solitária

leva em conta a
distância e a forma
como os dedos de cada
classe se distribuem

$$C_i[j, k] = \sum_{s=1}^N \frac{(x_s[j] - m_i[j])(x_s[k] - m_i[k])}{N-1}$$

* Função de decisão:

↳ o padrão x pertence a

classe c_i com média m_i e matriz de covariância C_i se:

$$D_i(x) < D_j(x), \forall i \neq j$$

Classificador de Bayes:

↳ inclui a probabilidade a priori de cada classe

$$d_i(x) = \ln P(c_i) - \frac{1}{2} \ln |C_i| - \frac{1}{2} (x - m_i)^T C_i^{-1} (x - m_i)$$

matriz de confusão:

(as classes que o modelo previu)

Obj \ Classe	Círculo	Quadrado	Triângulo
Círculo	10	2	1
Quadrado	5	20	2
Triângulo	0	0	10

classes verdadeiras

ERROS

(o que o objeto realmente é)

ACERTOS

SVM - support vector machines

- escolhem o melhor hiperplano de acordo com algum critério (ex: margem máxima)
- otimização → processo de treinamento
- o set de treino é reduzido a um número pequeno de support vectors.

* Função Discriminante:

ele associa um vetor de feature x a uma classe w_j se:

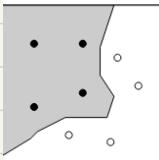
• p/ vários classes $\Rightarrow g_i(x) > g_j(x), \forall j \neq i$

• p/ 2 classes $\Rightarrow g(x) = g_1(x) - g_2(x)$

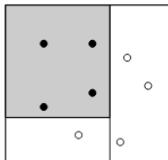
• Decidimos w_1 se $g(x) > 0$.

• Decidimos w_2 se $g(x) < 0$

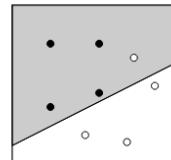
It can be arbitrary functions of x , such as:



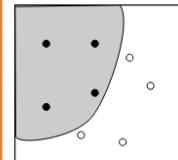
Nearest Neighbor



Decision Tree



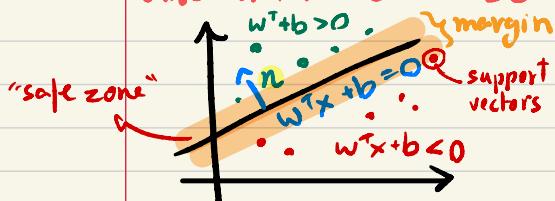
Linear Functions
 $g(x) = w^T x + b$



Nonlinear Functions

discriminante linear:

$$g(x) = w^T x + b$$



a largura da margem é: $\frac{2}{\|w\|}$

$$\text{vetor normal} \Rightarrow n = \frac{w}{\|w\|}$$

* margem = o tamanho que o discriminante poderia aumentar sem atingir um ponto de class.

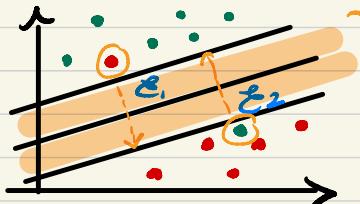
depois os pontos $\{(x_i, y_i)\}_{i=1,2,\dots,n}$

- $y_i = +1$, $w^T x_i + b > 0$ (ou ≥ 1)
- $y_i = -1$, $w^T x_i + b < 0$ (ou ≤ -1)

$$\text{maximize } \frac{2}{\|w\|} = \text{minimize } \frac{1}{2} \|w\|^2$$

* logo, ele desça escolher o discriminante que entrega a maior safe zone possível, para robustez.

* E se tiver ruído? \rightarrow adicionamos os "slack variables"



soft margin

traz flexibilidade à margem

ξ_i
variáveis de folga

medem o quanto cada ponto viola uma margem ideal

controle o overfitting

$$\star \text{ minimizamos } \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i$$

tal que $y_i(x^T x_i + b) \geq 1 - \xi_i$

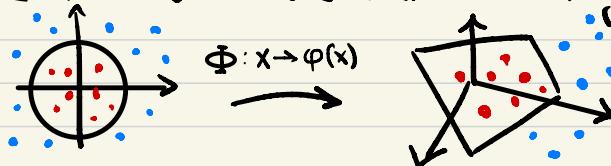
$\star \cup C$ controla o overfitting

↳ se ele for muito grande, a penalidade é grande \Rightarrow overfitting

↳ se ele for muito pequeno \Rightarrow underfitting (sacrifica separabilidade linear p/ ganhar estabilidade)

SVM's não lineares:

↳ o espaço de input pode ser mapeado para um espaço de mais dimensões onde o training set é separável.



- Examples of commonly-used kernel functions:

- Linear kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$
- Polynomial kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^p$
- Gaussian (Radial-Basis Function (RBF)) kernel:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$$
- Sigmoid:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\beta_0 \mathbf{x}_i^T \mathbf{x}_j + \beta_1)$$

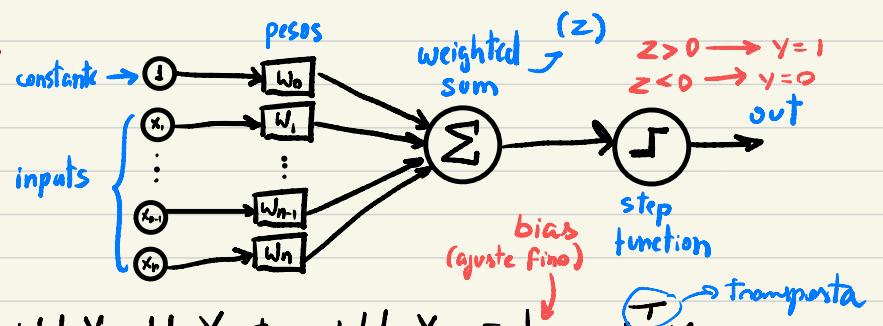
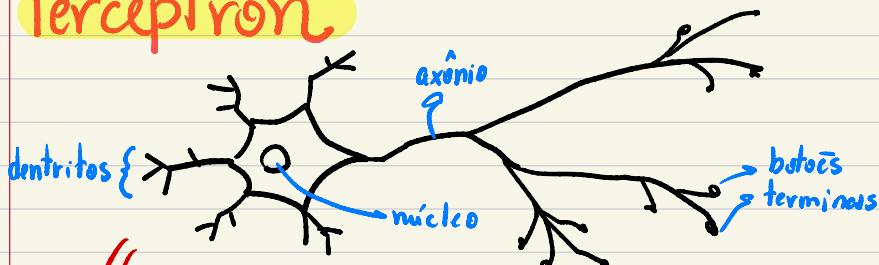
\star Algoritmo:

- 1- escolha uma função do Kernel
- 2- escolha um valor p/ C
- 3- resolva o quadratic programming problem
- 4- construa a função discriminante a partir dos vetores supports.

Multiclass classification

- Specify $n(n-1)/2$ classifiers of the form "one against one" and choose the "most voted" class.
- Specify n classifiers of the form "one against all" and choose the class with larger score.
- Specify a tree of classifiers of the form "one against the remaining" until a single class is selected.

Perceptron

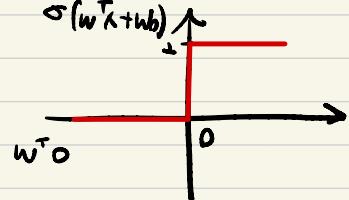


$$Z = w_0 + w_1 x_1 + w_2 x_2 + \dots + w_n x_n = b + w^T x$$

\downarrow Ativação total
(step function)
degrau

$$a = \hat{y} = \sigma(z)$$

$\sigma(w^T x + b)$



* Um perceptron é um classificador binário: $y=1$ se $z>0$.
nenão, $y=0$.

* Cada neurônio dos caminhos
encadrados e da saída possui um
peso de biais associado.

1- initialize w .

2-repete o processo até que w seja estável

3-p/ cada sample (x, y) no dataset:

- compute $\bar{y} = \sigma(b + w^T x)$

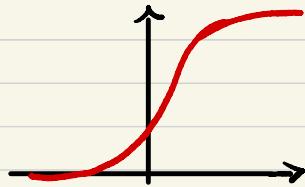
- se $\bar{y} \neq y$, ajuste cada peso w_i p/ que \bar{y} seja + perto de y .

se o valor calculado não for
o real, ajustamos os pesos

* XOR → não dá p/
separar tudo em um
único limiar

Regressão Logística:

$$\ln \frac{P(y=1)}{1-P(y=1)} = b + w^T x$$



Sigmoid activation function

$$P(y=1) = \frac{1}{1 + e^{-z}}$$

probabilidade de dado pertencer à classe 1

- atualizaremos os pesos
- computaremos uma função de perda

$$J(w, b) = \frac{1}{m} \sum_{i=1}^m L(y^{(i)}, \bar{y}^{(i)})$$

↳ cross entropy loss function
(precisamos minimizar essa!!)

Gradiente Descendente:

algoritmo de otimização p/ ajustar os parâmetros de um modelo p/ minimizar uma função de custo ou erro

$$z = w^T x + b$$

$$\bar{y} = a = \sigma(z)$$

$$L(a, y) = -(y \log(a) + (1-y) \log(1-a))$$

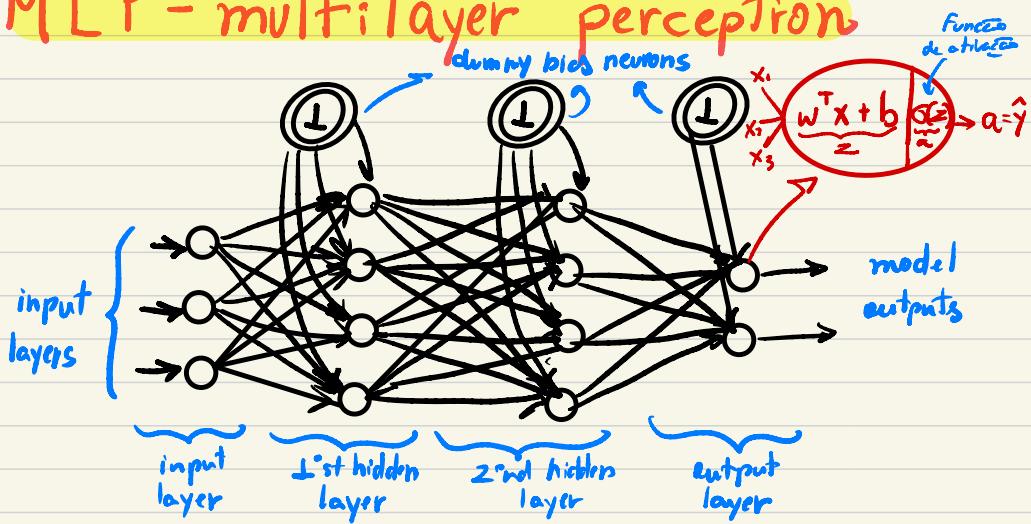
$$\partial L(a, y) / \partial a = -\frac{y}{a} + \frac{1-y}{1-a}$$

$$\partial L(a, y) / \partial z = a - y$$

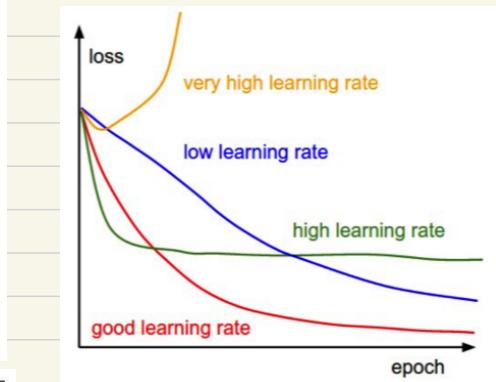
$$\partial L(a, y) / \partial w_i = x_i \partial L(a, y) / \partial z$$

$$\partial L(a, y) / \partial b = \partial L(a, y) / \partial z$$

MLP - multi layer perceptron



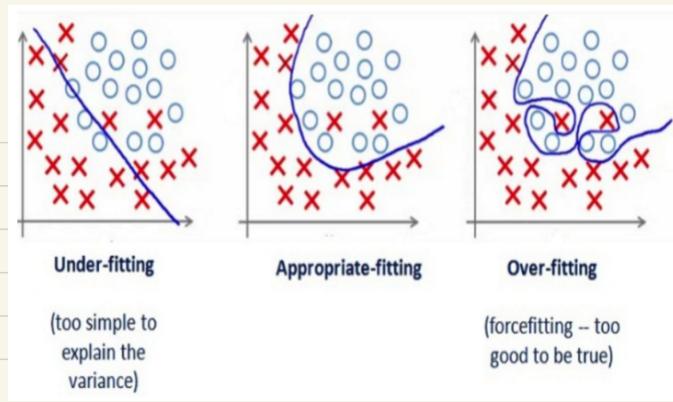
- X is the $n^{[0]}, m$ data matrix of m samples
- Y is the $n^{[L]}, m$ label matrix of m samples
- $[l]$ is the layer index, $l=0..L$
- L is the number of layers
- $x^{(i)}$ denotes the i -th sample
- $x^{(i)}_j$ denotes the j -th variable of the i -th sample
- Each layer l has $n^{[l]}$ units
- $W^{[l]}$ is the $n^{[l]}, n^{[l-1]}$ weight matrix of layer l
- $b^{[l]}$ is the $n^{[l]}$ bias array in layer l
- $Z^{[l]}$ is the $n^{[l]}, m$ matrix with $W^{[l]}A^{[l-1]} + b^{[l]}$
- $A^{[l]}$ is the $n^{[l]}, m$ matrix with $g(Z^{[l]})$
- $X=A^{[0]}; Y^{\wedge}=A^{[L]}$



Name	Plot	Equation	Derivative
Sigmoid	Sigmoid	$f(x) = \sigma(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1-f(x))$
Tanh	Tanh	$f(x) = \tanh(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$	$f'(x) = 1 - f(x)^2$
Rectified Linear Unit (relu)	Relu	$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Leaky Rectified Linear Unit (Leaky relu)	Leaky Relu	$f(x) = \begin{cases} 0.01x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0.01 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$

passo a passo:

- 1- definimos hiper/parâmetros
- 2- loop p/ n. de iterações:
 - a. forward propagation
 - b. computa a função de custo
 - c. backward propagation
 - d. atualiza os parâmetros p/ predição
- 3- use os parâmetros treinados p/ predição



BIAS & VARIÂNCIA

Symptoms	Underfitting	Just right	Overfitting
Regression illustration	• High training error • Training error close to test error • High bias	• Training error slightly lower than test error	• Very low training error • Training error much lower than test error • High variance
Classification illustration			
Deep learning illustration			
Possible remedies	• Complexity model • Add more features • Train longer		• Perform regularization • Get more data

bias alto → - tentar uma rede maior (T hidden layers, T units)
 - teste por + tempo
 - tente uma arquitetura que encaixe melhor
 - tente algoritmos de otimização diferentes

performance dos dados de treino

variância alta → - pegue mais dados
 - tente regularizar
 - tente uma arquitetura que encaixe melhor

performance dos dados de teste

- * Técnicos p/ diminuir overfitting
 - regulregularização
 - data augmentation
 - early stopping (para de iterar)

* L2 Regularization

- inclus inclus um termo na função da custo p/ diminuir os pesos da rede.

$$J(W, b) = -\frac{1}{m} \sum L(y, \hat{y}) + \frac{\lambda}{2m} \sum_j \sum_i (w_{ji}^{[l]})^2$$

$$dw^{[l]} = (from_bp) + \frac{\lambda}{m} w^{[l]}$$

$$w^{[l]} = w^{[l]} - \alpha dw^{[l]}$$

* Dropout Regularization

- Inverted dropout: Randomly block a fraction p of units from each hidden layer (make $a=0$), at each iteration of the training process (no dropout to test!)
- Normalize the remaining by dividing $a/(1-p)$
- As any unit can be blocked, the network tends to get more balanced weights
- Larger layers may have larger dropout fractions
- Input layer usually has no dropout

- * Técnicos p/ acelerar convergência
 - normalização
 - momentum
 - mini-batch descent
 - inicialização

* Normalizações

- traz os variáveis p/ mesma escala
- pode aumentar a velocidade de treinamento

$$\frac{x - \text{mean}}{\text{st dev}}$$

* Momentum

"elisa" os gradientes e acelera convergência

$$v_{dW} = \beta v_{dW} + (1 - \beta)dW$$

$$v_{db} = \beta v_{db} + (1 - \beta)db$$

$$W = W - \alpha v_{dW}, \quad b = b - \alpha v_{db}$$

Multiple classes

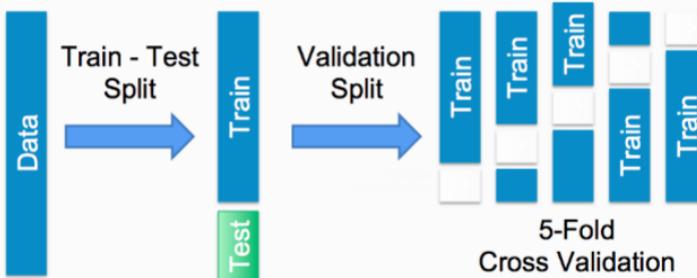
- In order to classify into n classe, use a n-unit output layer
- Each unit's output should reflect the probability of the input being each class
- Use the softmax activation function:

$$\alpha_i = \hat{y}_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$$

and the Loss Function (Maximum Likelihood Estimator):

$$L(y, \hat{y}) = -\sum_{j=1}^n y_j \log \hat{y}_j; dz = \hat{y} - y$$

Data Split Strategies



Mapeamento de Técnicas de PAI (Processamento e Análise de Imagens)

Fase Principal	Objetivo Prático	Técnica(s) Chave dos Slides	Conceito / Aplicação (Exemplos Reais)
I. PROCESSAMENTO	Corrigir distorções e ruídos antes da análise.	Transformações Geométricas (Translação, Rotação, Escala, Afins)	Registro de Imagens: Alinhar duas tomadas diferentes de um mesmo objeto (ex: alinhamento de imagens médicas ou satélites) para comparação ou fusão.
<i>Filtrar / corrigir</i>	Simplificar a forma e remover ruídos finos.	Morfologia Matemática (Erosão, Dilatação, Abertura, Fechamento)	Pré-processamento e Limpeza: Eliminar "ilhas" de ruído (Abertura) ou preencher pequenos buracos e quebras em contornos (Fechamento) de um objeto segmentado, como em análise de vasos sanguíneos.
	Identificar o contorno de um objeto binário.	Morfologia Matemática (Extração de Contorno: $\beta(A)=A-(A \ominus B)$)	Medição de Perímetro: Calcular a fronteira exata de um objeto após a segmentação, como a medição do contorno de uma célula ou tumor.
	Localizar um padrão específico em uma imagem.	Morfologia Matemática (Transformada Hit-or-Miss)	Contagem ou Localização: Encontrar ocorrências de pequenos marcadores ou formas específicas (como cantos) em um sistema de inspeção.
II. SEGMENTAÇÃO	Separar o objeto de interesse do fundo (Foreground/Background).	Limiarização Global (Ex: Algoritmo de Otsu)	Binarização de Documentos: Converter texto manuscrito ou impresso em uma imagem binária para facilitar o OCR (reconhecimento de caracteres), maximizando a variância entre as classes "texto" e "fundo".
<i>Encontrar o objeto</i>	Segmentar múltiplas regiões homogêneas.	Crescimento de Regiões (Region Growing)	Segmentação de Imagens de Satélite: Agrupar pixels com cores ou texturas semelhantes (ex: floresta, água, área urbana) para criar um mapa temático da paisagem.
	Encontrar contornos em imagens complexas (mapa topográfico).	Watershed (Divisor de Águas)	Separação de Células Aglomeradas: Em bioimagem, separar objetos que estão tocando uns nos outros, tratando a imagem como um relevo onde as bordas são os divisores de água.
	Determinar o contorno preciso de uma estrutura conhecida.	Contorno Ativo (Snakes)	Medição de Tumores ou Órgãos: Segmentar estruturas com contornos irregulares, como o ventrículo em uma imagem cardíaca, onde uma curva inicial se molda à fronteira.
	Detectar retas em uma imagem.	Transformada de Hough (Espaço)	Visão Robótica/Autônoma: Identificar linhas retas em estradas, bordas de edifícios ou trilhos para navegação e reconhecimento de cena.

III. DESCRIÇÃO	Calcular a área e o formato de um objeto após segmentação.	Características Geométricas (Área, Perímetro, Circularidade/Compaquidade)	Análise Biomédica: No exemplo dado: Calcular a área de um tumor a partir de uma tomografia segmentada, usando a contagem de pixels internos (área) e comparando com o perímetro (circularidade).
<i>Medir o objeto / extraír características</i>		Caracterizar a complexidade da forma.	Descritores de Forma (Código de Cadeia, Curva Phi-S, Assinaturas, Fecho-Convexo)
	Analizar o "padrão visual" de uma superfície.	Descritores de Textura (Matrizes de Co-ocorrência / Descritores de Haralick)	Reconhecimento de Formas: Distinguir letras, algarismos ou peças industriais, fornecendo uma descrição da forma que é invariante à translacão e rotação.
	Descrever objetos com invariância à rotação, escala e translacão.	Momentos Invariantes de Hu	Inspeção de Materiais: Identificar defeitos em tecidos, madeira ou metais, ou classificar o tipo de solo em imagens geológicas com base na sua rugosidade e regularidade tonal.
IV. CLASSIFICAÇÃO	Atribuir um padrão a uma classe em problemas linearmente separáveis.	Perceptron / Regressão Logística	Classificação Simples: Usado para problemas binários onde as características são facilmente separáveis por uma linha reta (ex: classificar um objeto em duas categorias muito distintas).
<i>Atribuir um rótulo ao objeto</i>		Support Vector Machine (SVM) com Kernel RBF	Reconhecimento de Padrões: Classificar faces, dígitos manuscritos ou texturas complexas, onde o truque do kernel mapeia os dados para um espaço de alta dimensão para separação linear.
	Classificar problemas não lineares, como o XOR, e criar classificadores complexos.	Multi-Layer Perceptron (MLP)	Deep Learning Inicial: Classificação de imagens com múltiplas classes, onde camadas ocultas aprendem a representar características abstratas.
	Classificar padrões baseado na similaridade de amostras conhecidas.	Classificador do Vizinho Mais Próximo (KNN)	Diagnóstico Rápido: Classificar um novo paciente/objeto com base nas características mais próximas de exemplos já rotulados no conjunto de treinamento.
	Classificar padrões em que a dispersão dos dados é importante.	Classificador de Distância de Mahalanobis ou Classificador de Bayes	Classificação Estatística em Medicina/Biometria: Classificar amostras levando em conta a correlação entre as características (ex: peso e altura) e a probabilidade a priori de cada classe.