Aula 11
Reconhecimento de Faces

Eduardo L. L. Cabral

Objetivos

- O que é reconhecimento de faces?
- RNAs siamesas.
- Métodos:
  - Função de custo tripla;
  - Classificação binária.

O que é reconhecimento de face

- Demonstração de reconhecimento de face na entrada do Baidu.

https://www.youtube.com/watch?reload=9&v=wr4rx0Spihs

O que é reconhecimento de face

- Existem duas formas de identificar pessoas usando imagem da face:
  - Verificação de face;
  - Reconhecimento de face.
- Reconhecimento de face deve ser realizado juntamente com a verificação de que a pessoa está viva e que não é uma foto para enganar o sistema.
- Não será visto detecção de ser humano real.
- Reconhecimento e verificação de face consistem em problemas onde se tem um único dado (uma imagem, uma tentativa) $\Rightarrow$ difícil de se obter alta taxa de acerto.

O que é reconhecimento de face

- Verificação de face:
  - Dados de entrada ⇒ imagem da face de uma pessoa e sua identificação;
  - Resultado ⇒ verifica se imagem é da pessoa em questão;
  - Problema de 1 para 1 ⇒ verifica a imagem e identidade de somente uma pessoa;
  - Uma taxa de acerto de 99% é satisfatória.

O que é reconhecimento de face

- Reconhecimento de face:
  - Usa um banco de dados de muitas pessoas (*N* pessoas);
  - Dado de entrada $\Rightarrow$ imagem da face de uma pessoa;
  - Resultado $\Rightarrow$ identidade da pessoa se ela for uma das *N* pessoas cadastradas, ou “pessoas não reconhecida” se ela não estiver cadastrada
  - Se *N* = 100 e taxa de acerto é de 99% $\Rightarrow$ existe a chance de errar 1 vez em cada 100 verificações, ou seja, tem uma taxa de acerto de 1%, o que não é muito bom;
  - Em geral *N* é muito maior do que 100.

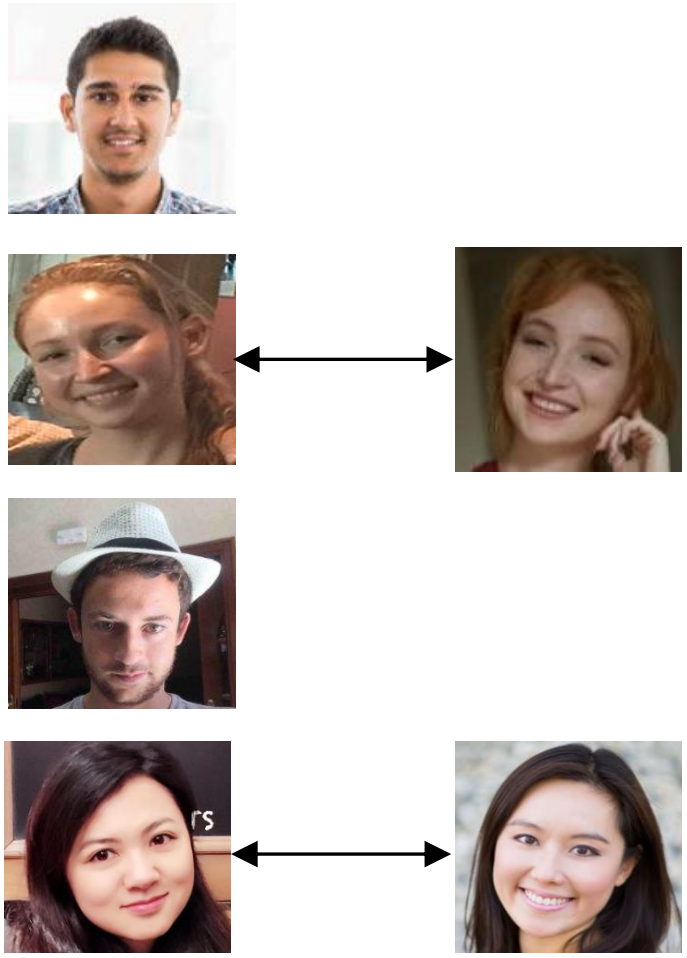
O que é reconhecimento de face

- Reconhecimento de face:
  - Erro de 1% representa uma chance grande de reconhecer uma pessoa de forma errada;
  - Precisa de uma taxa de acerto de no mínimo 99,9%, ou seja, erro de 0,1% (um erro em cada 1000 verificações);
  - Problema de 1 para *N* $\Rightarrow$ compara a imagem da face de uma pessoa com *N* outras imagens.
- Para desenvolver um algoritmo de reconhecimento de face primeiro de constrói um algoritmo para verificação de face e tendo um método com erro pequeno pode transformá-lo para reconhecer faces.

Verificação de face

- Verificação de face é um problema de uma única tentativa, ou seja, tem-se somente um único exemplo e partir desse exemplo tem-se que realizar a verificação.
- Além disso, tem-se poucas imagens para treinar o sistema $\Rightarrow$ número típico é de 10 fotos para cada pessoa.
- Aprende-se com um (ou poucos) exemplos a reconhecer uma pessoa.
- Aparece uma pessoa e uma foto é tirada $\Rightarrow$ a cada verificação tem-se uma foto diferente das que estão no banco de dados e é preciso verificar se essa pessoa é a pessoa cadastrada com a identificação fornecida.

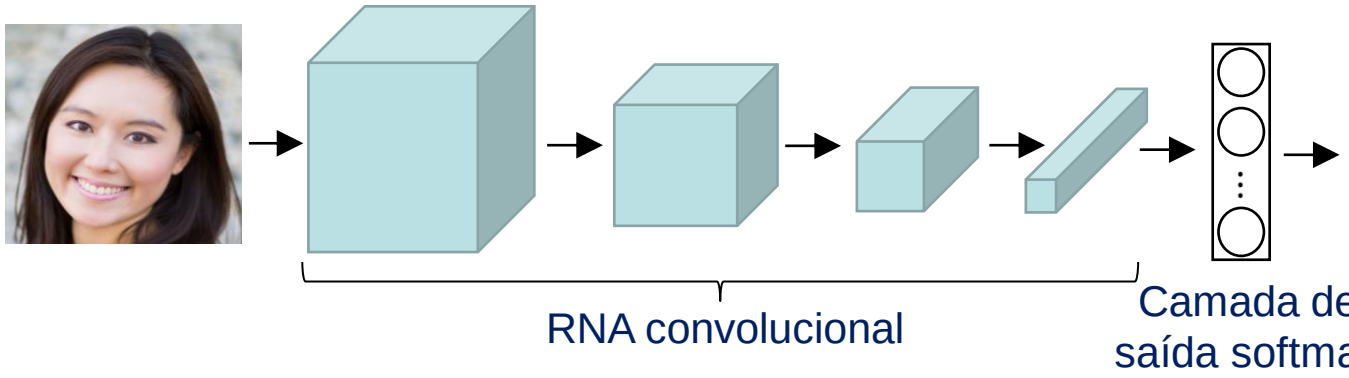
Verificação de face



- Se aparecer uma pessoa que não está cadastrada o sistema precisa identificar esse fato.
- Sistema deve aprender a partir de poucas fotos a reconhecer a pessoa novamente.

Verificação de face

- Uma solução ingênua:



- Precisa de uma camada softmax com 5 neurônios para identificar 4 pessoas (1 neurônio a mais é necessário para identificar pessoas não cadastradas).
- Esse esquema não funciona direito porque o conjunto de exemplos de treinamento é muito pequeno (apenas algumas imagens por pessoa).
- Outro problema é que para adicionar novas pessoas deve-se adicionar mais neurônios na camada de saída e retreinar a rede novamente.

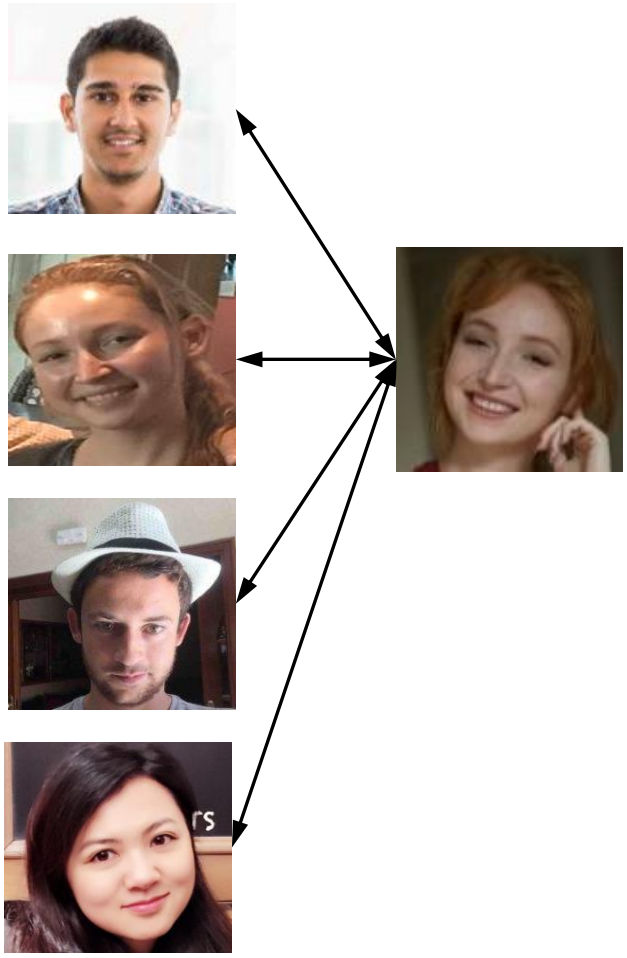
Verificação de face

- Solução $\Rightarrow$ usar uma Função de Similaridade
- Função de similaridade fornece o grau de similaridade entre duas imagens:

$$d(Im1, Im2) = \begin{cases} grau\ de \\ similaridade \\ entre\ duas \\ imagens \end{cases}$$

$$d(Im1, Im2) = \begin{cases} \leq \varepsilon \rightarrow pessoas\ iguais \\ > \varepsilon \rightarrow pessoas\ diferentes \end{cases}$$

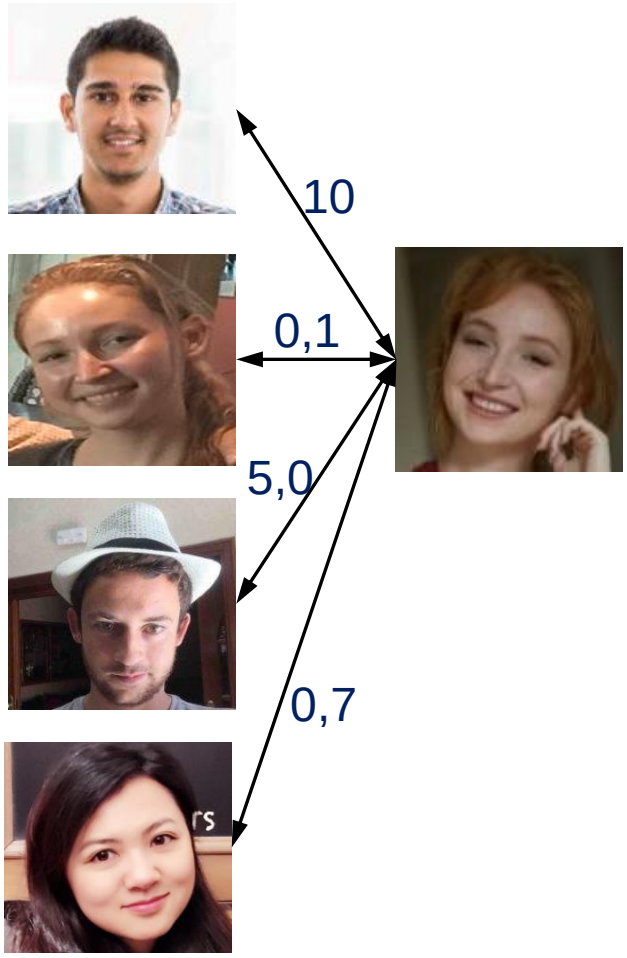
ε = número pequeno



Andrew Ng, deepleanring.ai

Verificação de face

- Calcula-se a função de similaridade para todas as imagens cadastradas no banco de dados:
  - $d \leq \varepsilon$ para alguma imagem do banco de dados $\Rightarrow$ reconhece a pessoa;
  - $d > \varepsilon$ para todas as imagens cadastradas no banco de dados $\Rightarrow$ não reconhece a pessoa.
- Com esse método é fácil incluir novas pessoas no banco de dados $\Rightarrow$ basta ter uma função de similaridade eficiente.
- Função de similaridade é implementada com uma RNA convolucional.

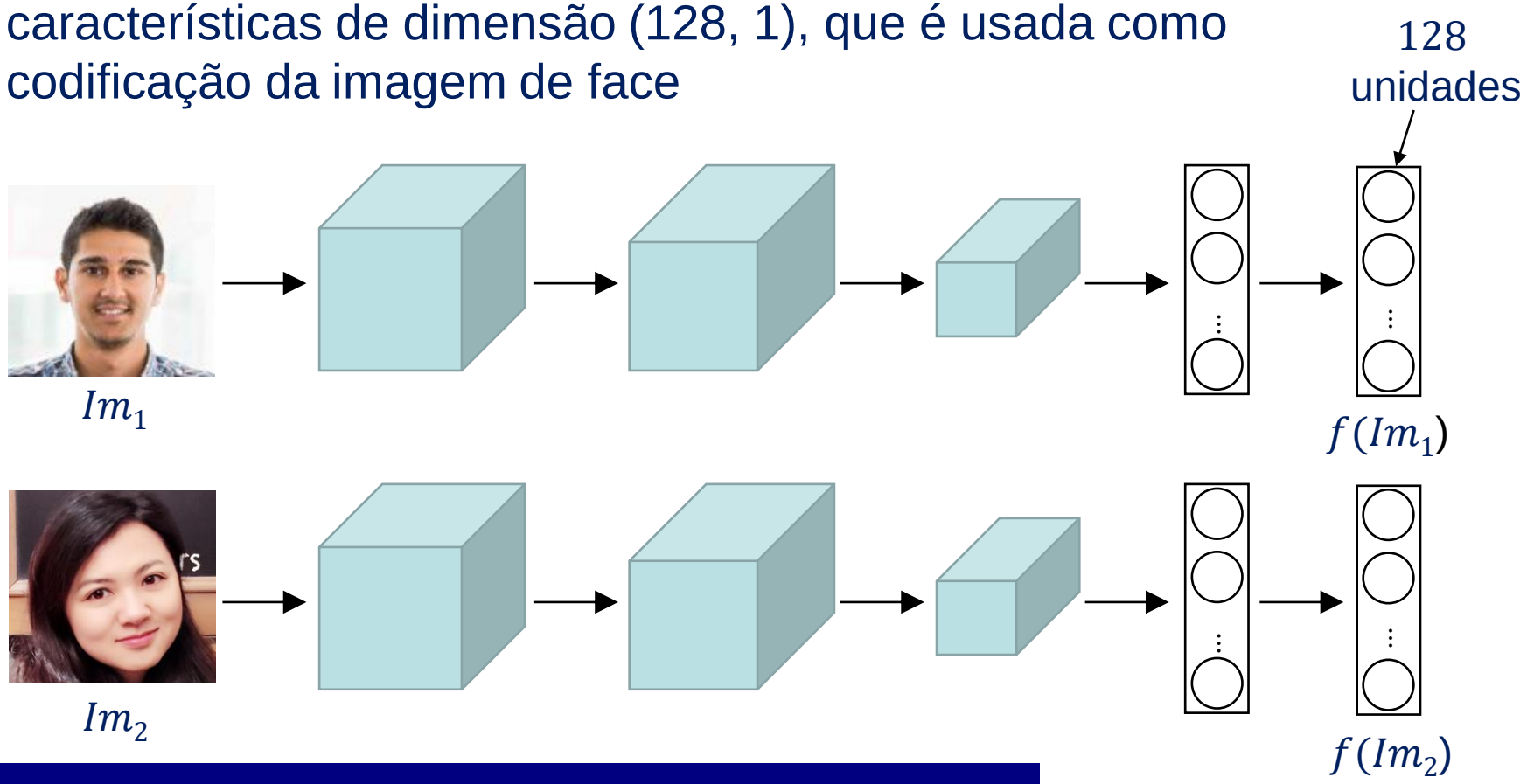


Rede siamesa

- Objetivo da função de similaridade é indicar quanto similares, ou quão diferentes são duas imagens de faces $\Rightarrow$ uso de RNA siamesa
- RNA siamesa é formada por duas RNA convolucionais iguais executadas em paralelo $\Rightarrow$ cada RNA recebe uma imagem de face
- Usa-se uma RNA pré-treinada para tarefa de classificação, que possui uma parte convolucional e outra parte densa
- Retira-se a última (ou as duas últimas) camadas densas da RNA pré-treinada e usa a saída da penúltima (ou antipenúltima) camada densa como saída da nova RNA
- Referência: Taigman et. al., 2014. DeepFace closing the gap to human level performance

Rede siamesa

- Por exemplo, se a nova camada de saída possuir 128 neurônios $\Rightarrow$ a saída de cada ramo da RNA siamesa é um vetor de características de dimensão (128, 1), que é usada como codificação da imagem de face



Rede siamesa

- Codificação das imagens pela RNA siamesa:
  - $Im_1 \Rightarrow f(Im_1)$
  - $Im_2 \Rightarrow f(Im_2)$

  } Vetores de características com dimensão (128, 1)
- Para comparar duas imagens, usando a RNA pré-treinada, calcula-se os vetores de características que codificam as duas imagens $\Rightarrow f(Im_1)$ e $f(Im_2)$
- Assume-se que $f(Im)$ é uma boa codificação das imagens
- Função de similaridade:

$$d(Im_1, Im_2) = \underbrace{\|f(Im_1) - f(Im_2)\|_2^2}_{\text{Norma 2 da diferença das codificações da duas imagens}}$$

Rede siamesa

- Treinamento da RNA deve ser feito de forma que:
  - Se Im_1 e Im_2 são da mesma pessoa, então:

    $\|f(Im_1) - f(Im_2)\|_2^2$ = pequeno

  - Se Im_1 e Im_2 são de pessoas diferentes, então:

    $\|f(Im_1) - f(Im_2)\|_2^2$ = grande

- Pode-se usar uma RNA pré-treinada, mas a tem que retreinar a rede completamente de novo.

Rede siamesa

- Duas formas de usar uma RNA siamesa para verificação de face:
  - Função de custo Tripla
  - Classificação binária

Função de custo Tripla

- Referência: Schroff et al.,2015, FaceNet: A unified embedding for face recognition and clustering
- Na Função de Custo Tripla (“Triplet Loss Function”) são necessárias 3 imagens para cada exemplo de treinamento:
  - Imagem referência $\Rightarrow$ imagem da face de uma pessoa
  - Imagem positiva $\Rightarrow$ outra imagem da face da mesma pessoa de referência
  - Imagem negativa $\Rightarrow$ imagem da face de outra pessoa diferente da referência

Referência (R)

Positiva (P)

Negativa (N)

Função de custo Tripla

- O que se espera ao analisar essas 3 imagens com uma RNA siamesa é obter os seguintes resultados:

Referência
(R)

Positiva
(P)

$$d(R,P) = pequeno$$

Referência
(R)

Negativa
(N)

$$d(R,N) = grande$$

onde $d(R,P)$ e $d(R,N)$ são as diferenças entre os vetores de características das imagens (função de similaridade)

Função de custo Tripla

- Matematicamente isso pode ser expresso por:

$$\underbrace{\|f(R) - f(P)\|^2}_{d(R,P)} < \underbrace{\|f(R) - f(N)\|^2}_{d(R,N)}$$

- Rearranjando tem-se:

$$\underbrace{\|f(R) - f(P)\|^2}_{\text{deve ser } 0} - \underbrace{\|f(R) - f(N)\|^2}_{\text{deve ser o maior possível}} < 0$$

- De fato deseja-se que essa diferença seja a maior possível assim, para garantir isso introduz-se um termo no lado direito:

$$\|f(R) - f(P)\|^2 - \|f(R) - f(N)\|^2 < -\alpha$$

Função de custo Tripla

- Rearranjando tem-se:

$$\|f(R) - f(P)\|^2 - \|f(R) - f(N)\|^2 + \alpha < 0$$

α = constante positiva que representa a menor distância entre $d(R,N)$ e $d(R,P)$

- Esse critério garante que $\Rightarrow d(R,N) \gg\gg d(R,P)$
- Exemplo para $\alpha = 0{,}2$:
  - Se $d(R,P) = 0{,}5$ e $d(R,N) = 0{,}51 \Rightarrow$ não satisfaz o critério
  - No caso de $d(R,P) = 0{,}5$, para satisfazer o critério $d(R,N)$ deve ser no mínimo igual a $0{,}7$
- Esse critério garante que o treinamento a distância entre imagens de faces de pessoas diferentes fiquem “distantes”

Função de custo Tripla

- **Função de erro:**

  Dadas 3 imagens de faces $\Rightarrow R, P, N$:

$$Erro(R,N,P) = \max(\|f(R) - f(P)\|^2 - \|f(R) - f(N)\|^2 + \alpha, 0)$$

  – Essa função de erro garante que a diferença $d(R,P) - d(R,N)$ seja maximizada, pois o valor mínimo que a função retorna é zero

  – Tentando minimizar a função $Erro(R,N,P)$ no final tenta-se de fato manter $d(R,P) - d(R,N) + \alpha$ menor do que zero.

Função de custo Tripla

- **Função de custo tripla:**

$$J(\mathbf{W}, \mathbf{B}) = \sum_{i}^{m} Erro(R^{(i)}, P^{(i)}, N^{(i)})$$

onde:

m = número de triplas

W e **B** = parâmetros da RNA siamesa

- Para o treinamento são necessárias várias fotos da mesma pessoa, pelo menos 10 fotos de cada pessoa
- Não é possível fazer o treinamento com uma única foto por pessoa

Função de custo Tripla

- Como formar as triplas:
  - Selecionar aleatoriamente as triplas torna difícil o treinamento porque o critério é facilmente satisfeito para imagens muito diferentes

$$\|f(R) - f(P)\|^2 - \|f(R) - f(N)\|^2 + \alpha < 0$$

  - Deve-se selecionar triplas que são difíceis de serem treinadas, $d(R, N) \approx d(R, P)$, que ocorre quando as imagens são parecidas $\Rightarrow$ isso faz com que a RNA tenha que aprender a separá-las
  - Antes de iniciar o treinamento deve-se calcular a distância $d\left(Im^{(i)}, Im^{(j)}\right)$ entre todas as imagens e escolher as triplas mais convenientes

Função de custo Tripla

- Exemplo de como formar as triplas:

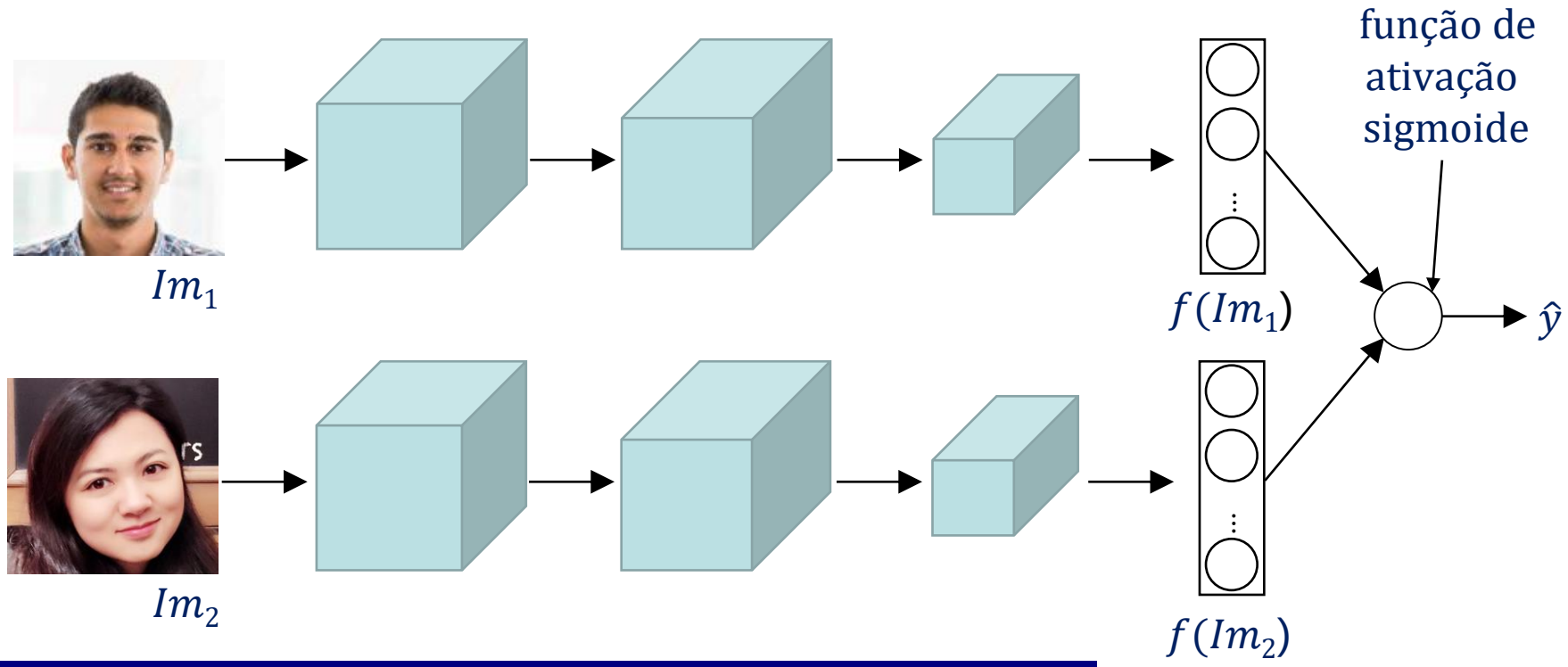
Referência (R)	Positiva (P)	Negativa (N)
		
		
		

Função de custo Tripla

- Treinamento da RNA é feito normalmente para minimizar a função de custo
- Sistemas comerciais de reconhecimento de face são treinados com milhões de imagens (FaceNet e Deepface)
- Algumas RNAs treinadas para reconhecer faces estão disponíveis e podem ser usadas livremente

Verificação de face com classificação binária

- Existe outra forma de reconhecimento de face baseada em um problema de classificação binária
- Usa-se também uma RNA siamesa



Verificação de face com classificação binária

- As duas RNAs são iguais e calculam o vetor de características
- Unidade de saída da RNA:

ex. 128 unidades

$$\hat{y} = \sigma\left(\sum_{k=1}^{128} w_k \left| f(Im_i)_k - f\left(Im_j\right)_k \right| + b\right)$$

onde:

$\sigma()$ = função de ativação sigmoide

w_k = pesos das ligações

b = viés

$\left| f(Im_i)_k - f\left(Im_j\right)_k \right|$ = soma das diferenças absolutas dos elementos dos vetores de características das duas imagens

Verificação de face com classificação binária

- Existem outras possibilidades para a função da unidade de saída da RNA (ver referência)
- Uma possibilidade é a função χ^2 (Qui-Quadrado) no lugar da soma absoluta das diferenças:

$$\hat{y} = \sigma\left(\sum_{k=1}^{128} w_k \underbrace{\frac{\left[f(Im_i)_k - f\left(Im_j\right)_k\right]^2}{f(Im_i)_k + f\left(Im_j\right)_k}}_{\chi^2} + b\right)$$

- Referência: Taigman et. al., 2014. DeepFace closing the gap to human level performance

Verificação de face com classificação binária

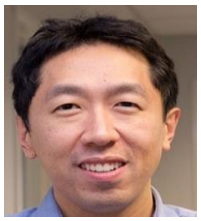
- Nesse caso:
  - Entradas da RNA siamesa $\Rightarrow$ duas imagens de face
  - Saída $\Rightarrow$ $\begin{cases} y = 1 \text{ se as imagens correspondem à mesma pessoa} \\ y = 0 \text{ se as duas imagens são de pessoas diferentes} \end{cases}$
- Os dois ramos da RNA siamesa são iguais
- Esse método funciona tão bem quanto ao método que usa a função de custo Tripla
- Não precisa calcular os vetores de características toda vez que vai realizar uma verificação $\Rightarrow$ basta calcular os vetores de características da cada pessoa uma única vez.

Verificação de face com classificação binária

- Exemplos:


$y = 1$


$y = 0$

$y = 0$


$y = 1$