Estimativa de similaridade em imagens usando Python

Wesin Ribeiro Alves

Agenda

- Introdução
- Few shot learning
- Redes convolucionais
- Redes siamesas
- Constrative loss
- Triplet loss
- Implementação (tensorflow/keras)
- Considerações finais

Parte I (teoria – 50 min)

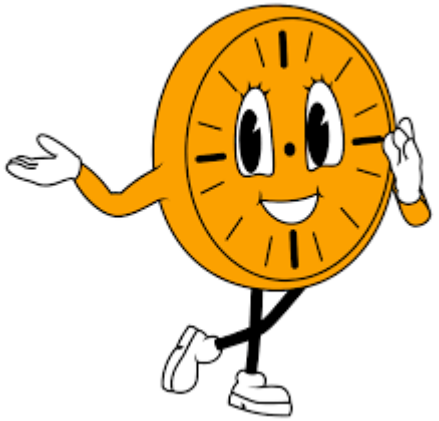
Few shot learning, redes siamesas

Parte II (prática – 50 min)

Contrastive loss

Parte III (prática – 50 min)

Triplet loss



Few shot learning

Support Set

Armadillo

Pangolin

Query

Armadillo or Pangolin?

Conjunto de treinamento

São animais da mesma espécie?

São animais da mesma espécie?

Few shot learning

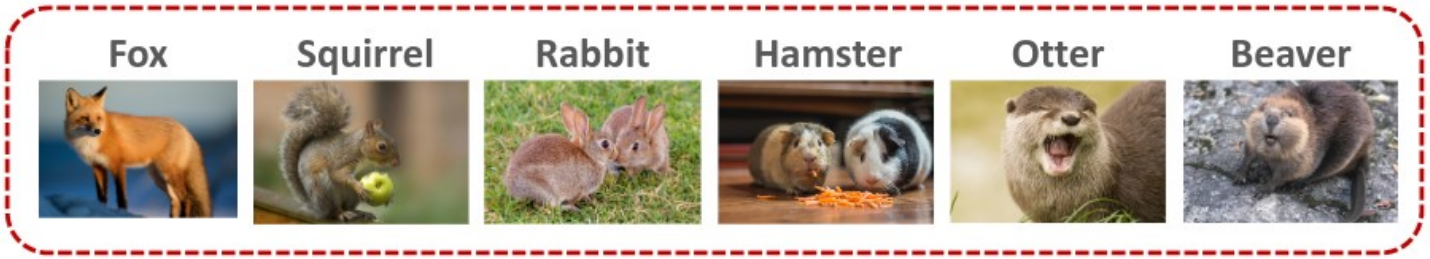
Query:

Support Set:

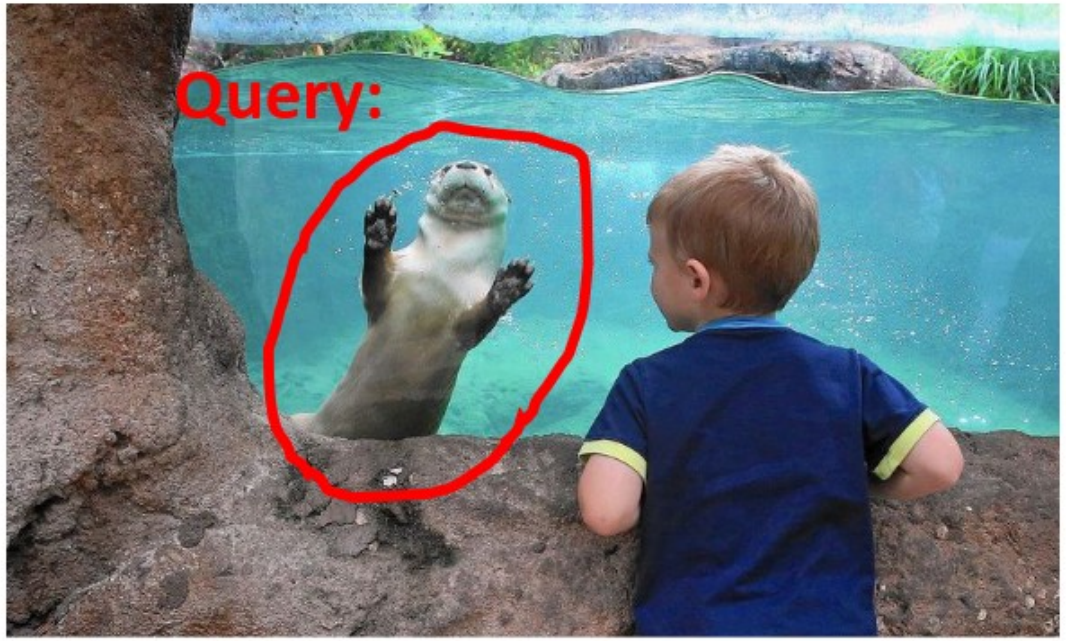
Meta learning (aprender a aprender)

Qual espécie de animal é mais semelhante?

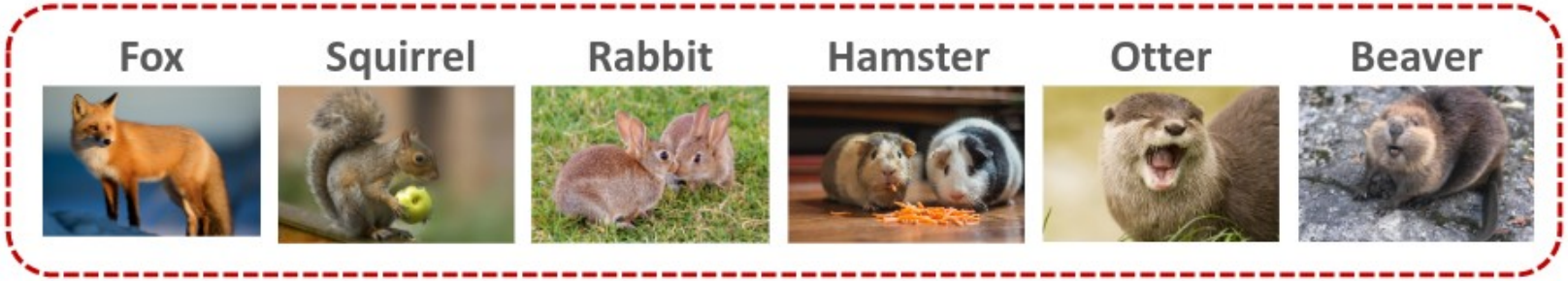
Give him the cards:



Em termos técnicos

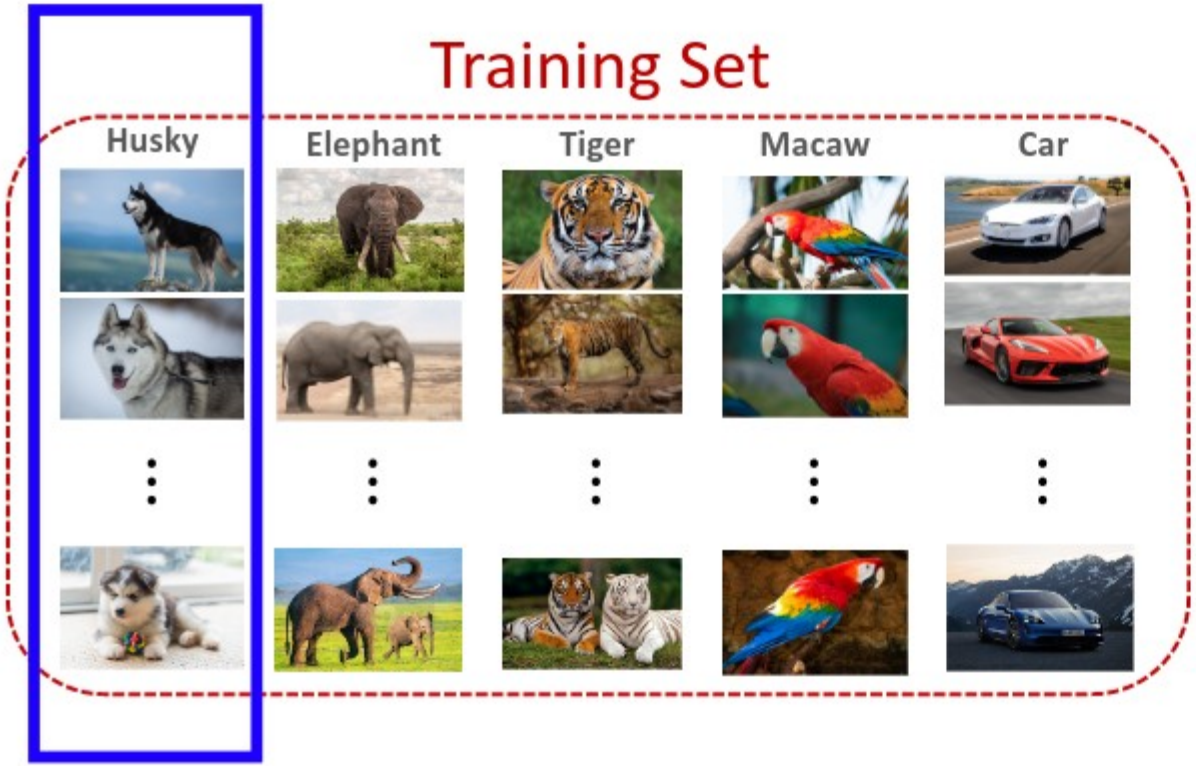


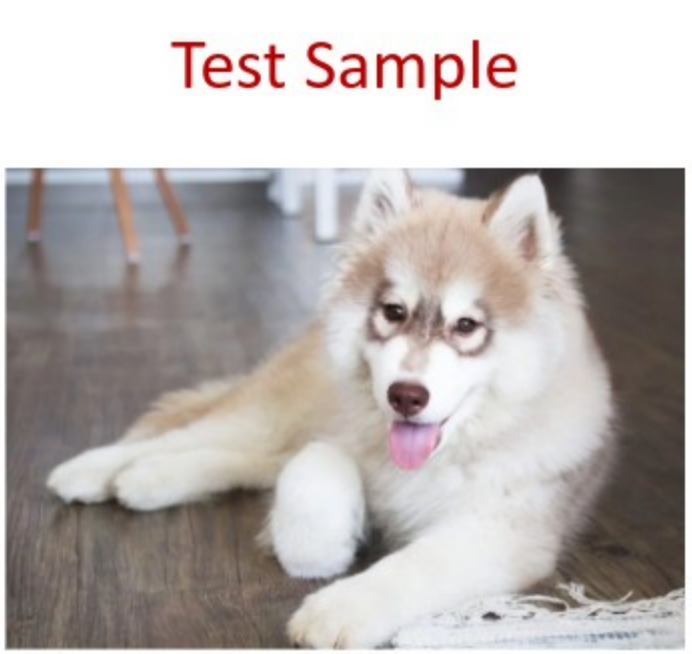
Support set:



Aprendizado supervisionado tradicional

- Amostra de teste nunca vista anteriormente
- Amostra de teste pertence a uma classe



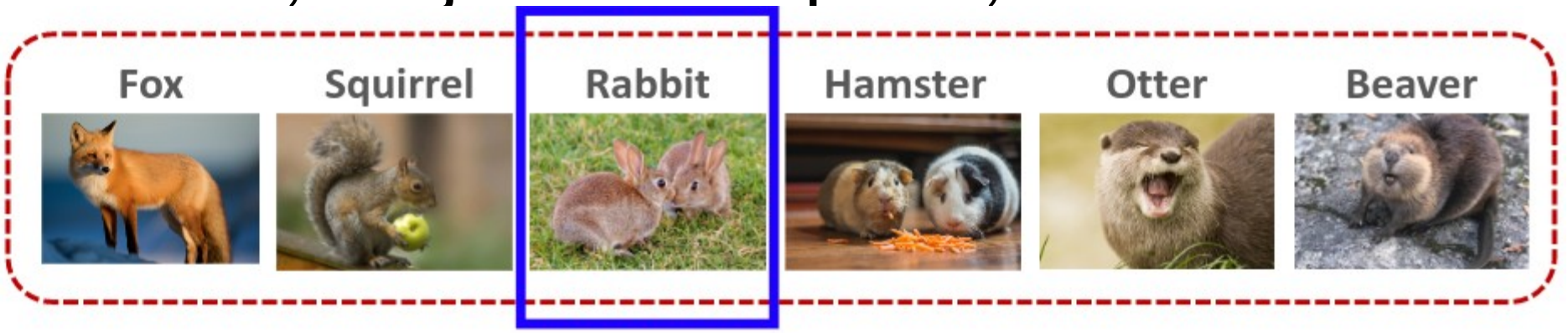


Few shot learning

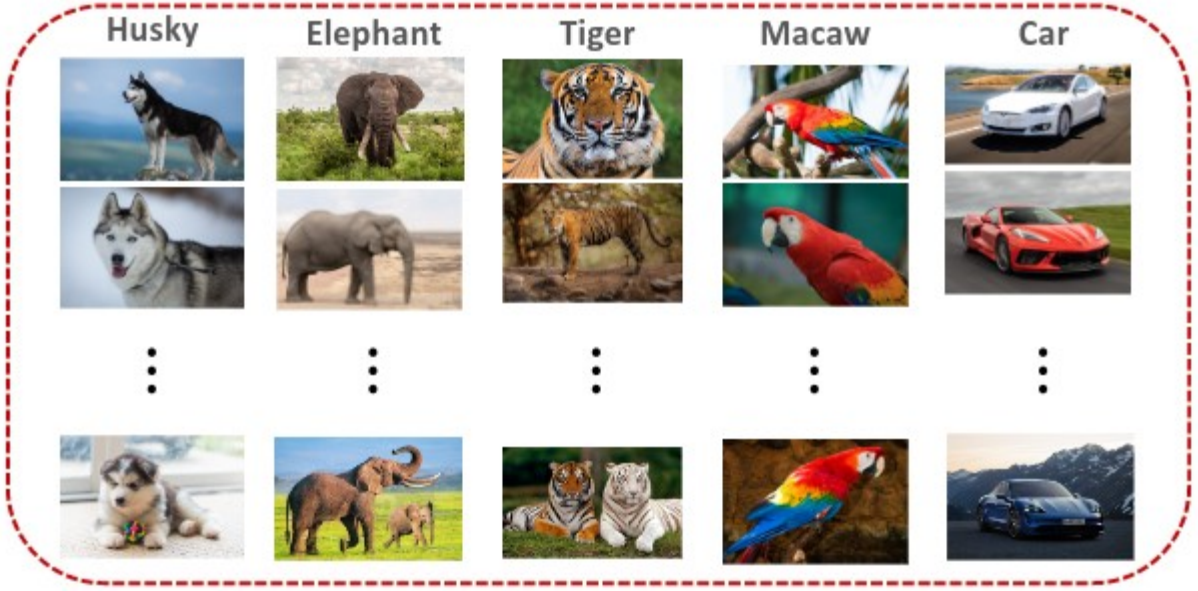
- Amostra de consulta nunca vista anteriormente
- Amostra de consulta pertence a classe desconhecida

Conjunto de treino, conjunto de suporte, e consulta

Support Set:

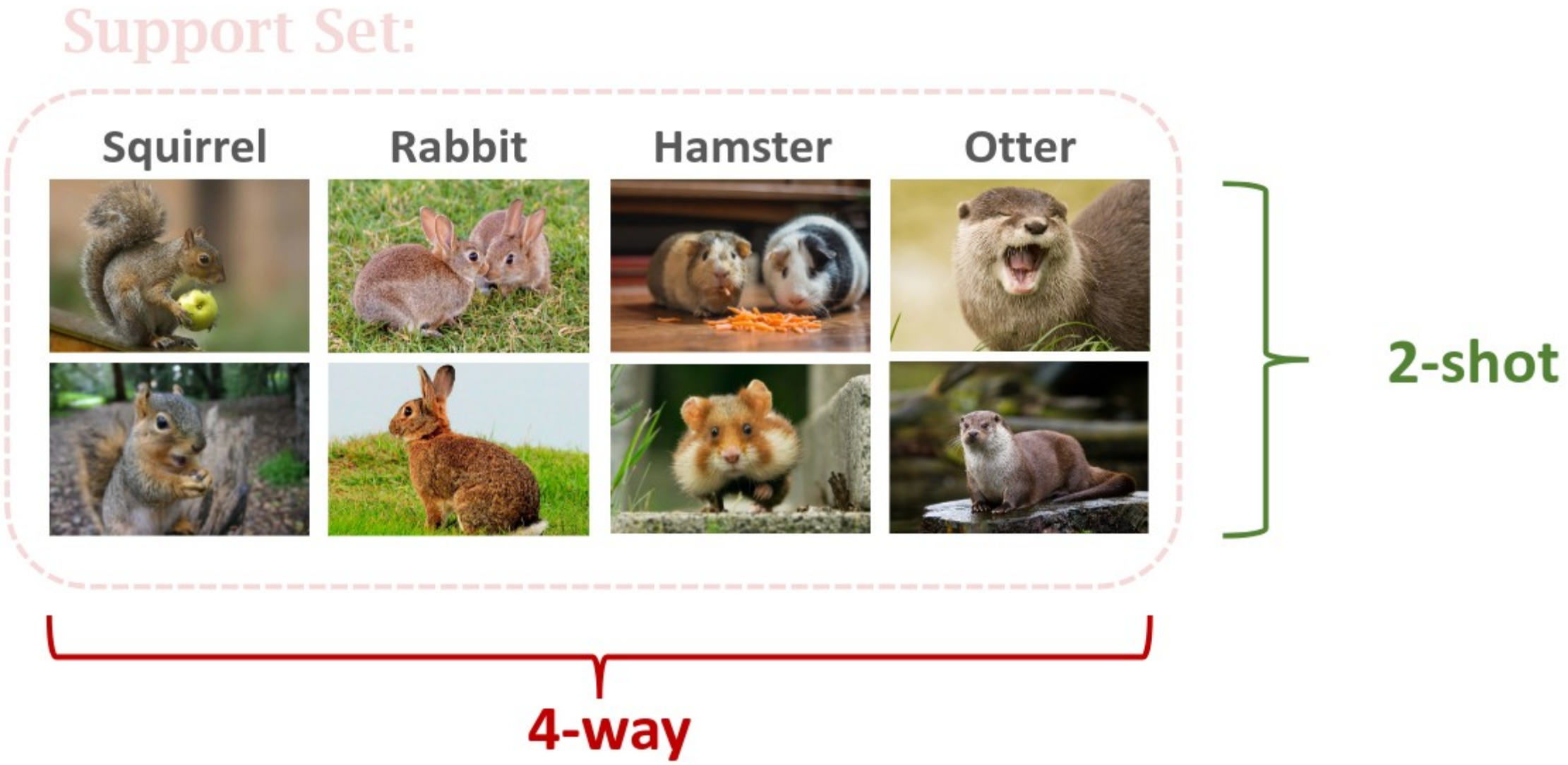


Training Set



Query Sample

K-way n-shot support set



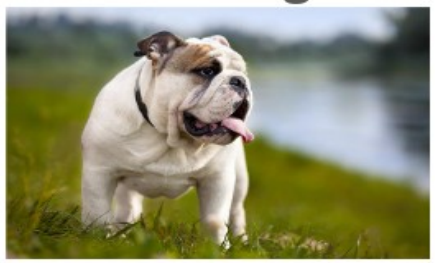
Ideia básica do few-shot learning

- Primeiro, aprende uma função de similaridade em um conjunto de treinamento
- Então, aplica a função aprendida para fazer a predição
  - Compara a consulta com cada amostra no conjunto suporte
  - Encontra a amostra com o grau de similaridade mais alta.

Bulldog

$\mathbf{x}_1$

Bulldog



$\mathbf{x}_2$

Fox

$\mathbf{x}_3$

$$\text{sim}(\mathbf{x}_1, \mathbf{x}_2) = 0 \quad \text{sim}(\mathbf{x}_1, \mathbf{x}_3) = 1, \quad \text{and} \quad \text{sim}(\mathbf{x}_2, \mathbf{x}_3) = 1$$

Que animal é esse?

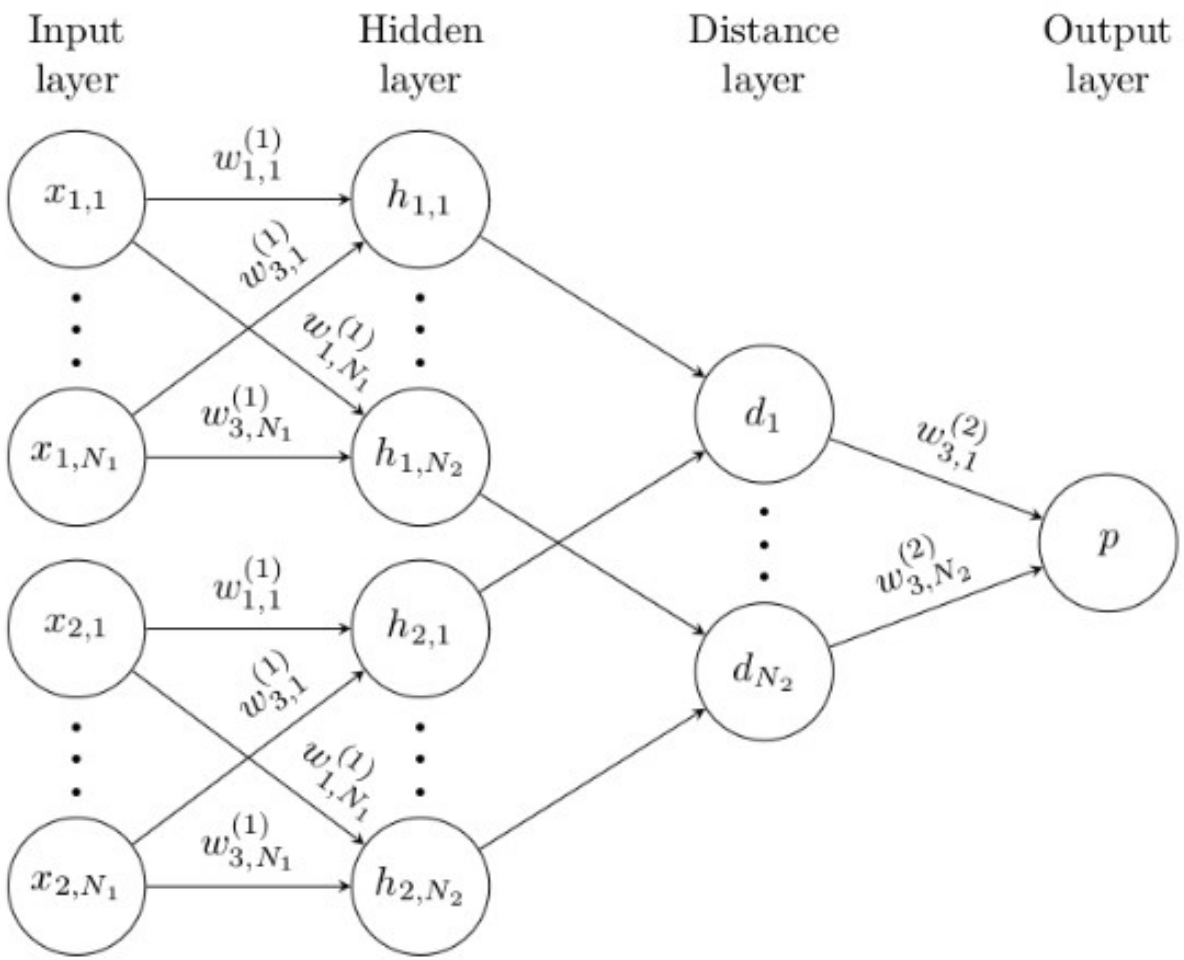
What is in the image?

Query:

sim = 0.2	sim = 0.1	sim = 0.03	sim = 0.05	sim = 0.7	sim = 0.5
Greyhound	Bulldog	Armadillo	Pangolin	Otter	Beaver
					

Redes neurais siamesas

- Duas ou mais redes neurais
- Compartilham o mesmo peso
- Extrai o vetor de características
- Calcula a similaridade entre esses vetores
- Propabilidade de pertencer a mesma classe



Conjunto de treinamento

Preparação dos dados de treinamento

Positive Samples

( ,  , 0)

( ,  , 0)

( ,  , 0)

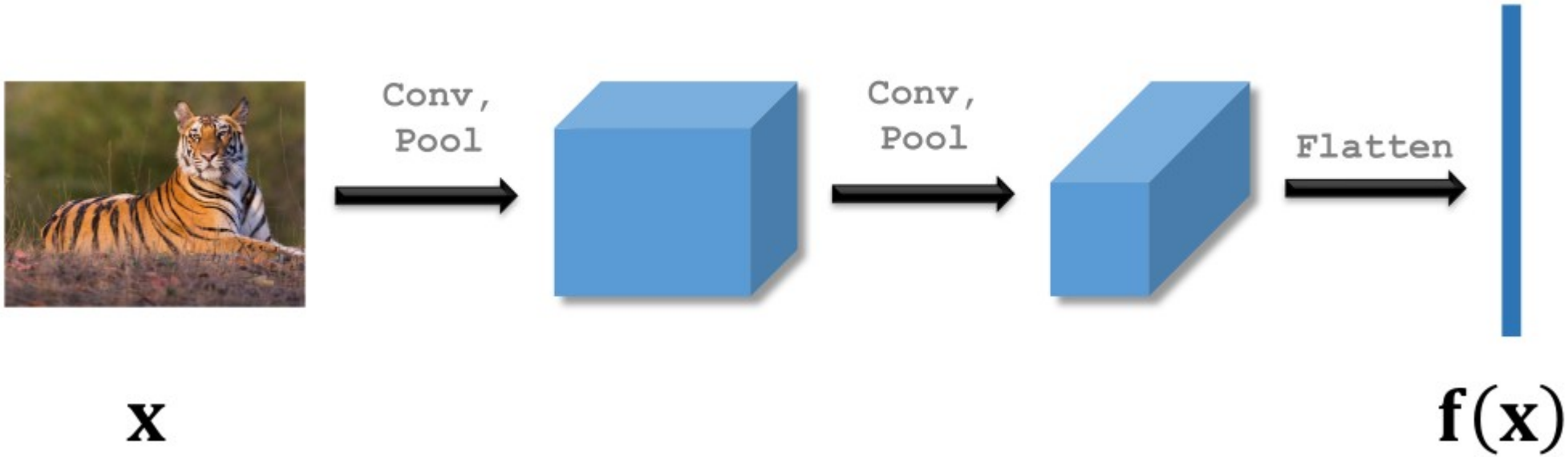
Negative Samples

( ,  , 1)

( ,  , 1)

( ,  , 1)

CNN para extrair o vetor de características



Contrastive loss

$\mathbf{x}_1$

$\mathbf{x}_2$

f

f

$\mathbf{h}_1$

$\mathbf{h}_2$

$\mathbf{z} = |\mathbf{h}_1 - \mathbf{h}_2|$

Dense Layers

Sigmoid

$\text{sim}(\mathbf{x}_1, \mathbf{x}_2)$

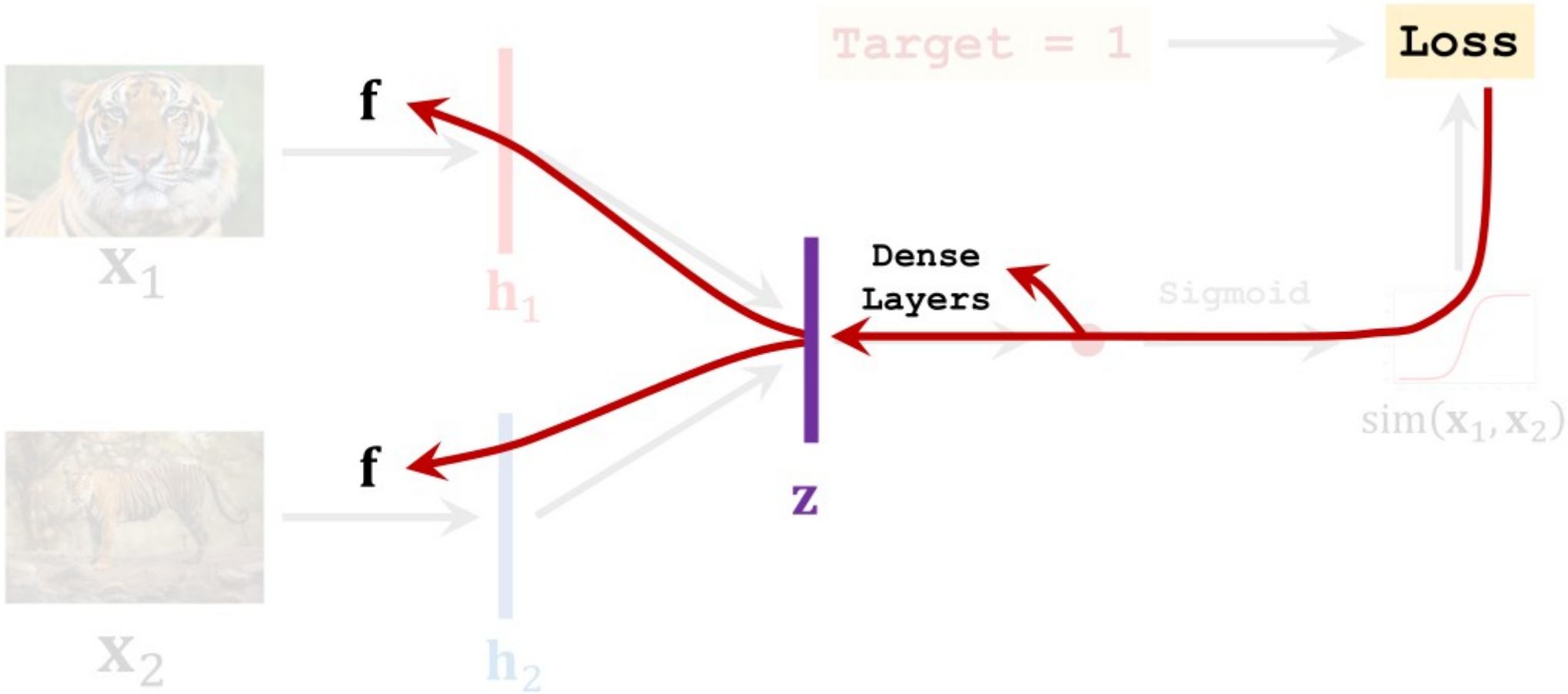
Target = 1

Loss

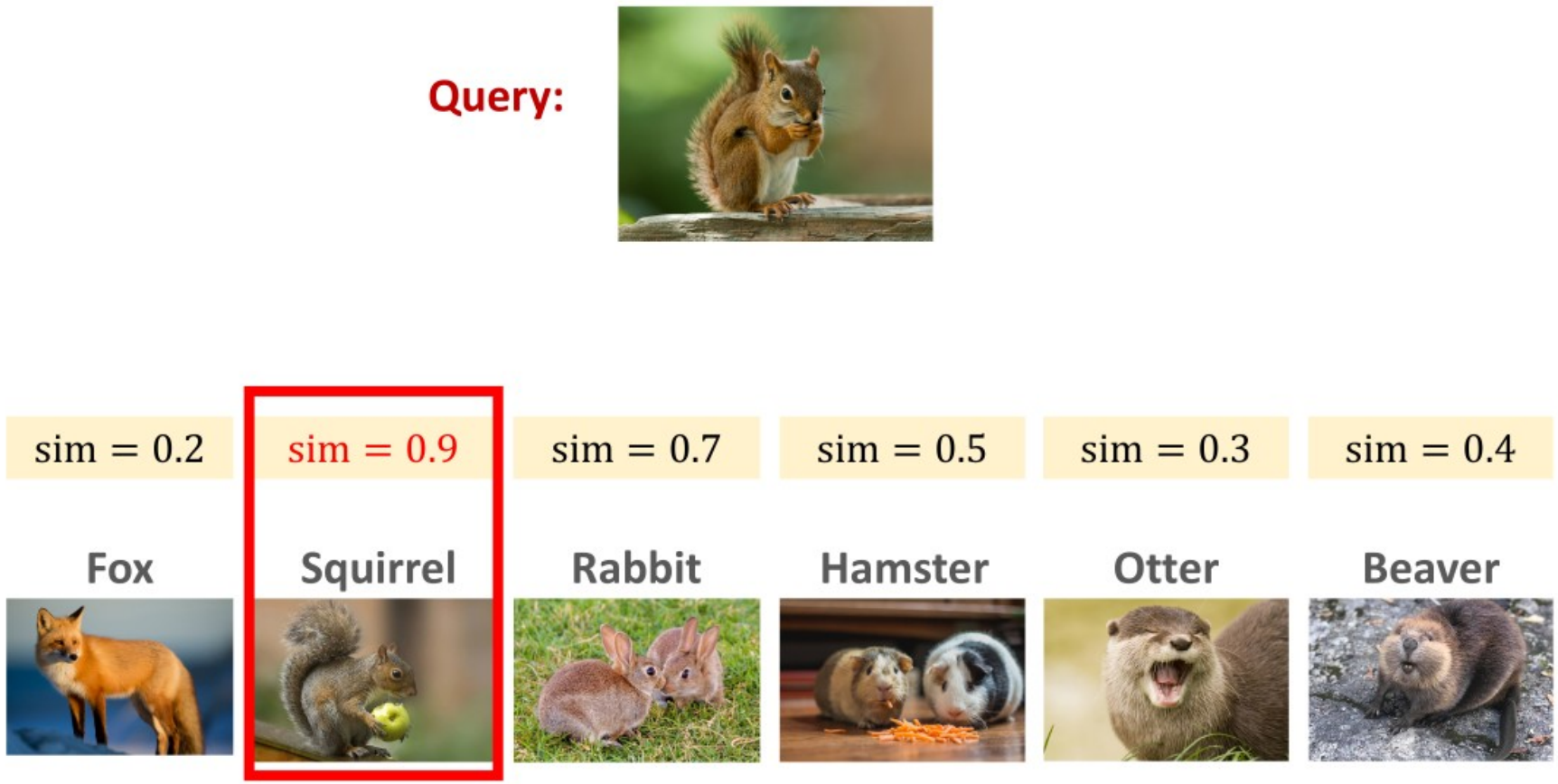
$$L(W,(Y,X_1,X_2)^i) = (1-Y)L_G\left(E_W(X_1,X_2)^i\right) + YL_I\left(E_W(X_1,X_2)^i\right)$$

$$E_W(X_1,X_2) = ||G_W(X_1) - G_W(X_2)||$$

Retropropagação do gradiente de erros (descida do gradiente)

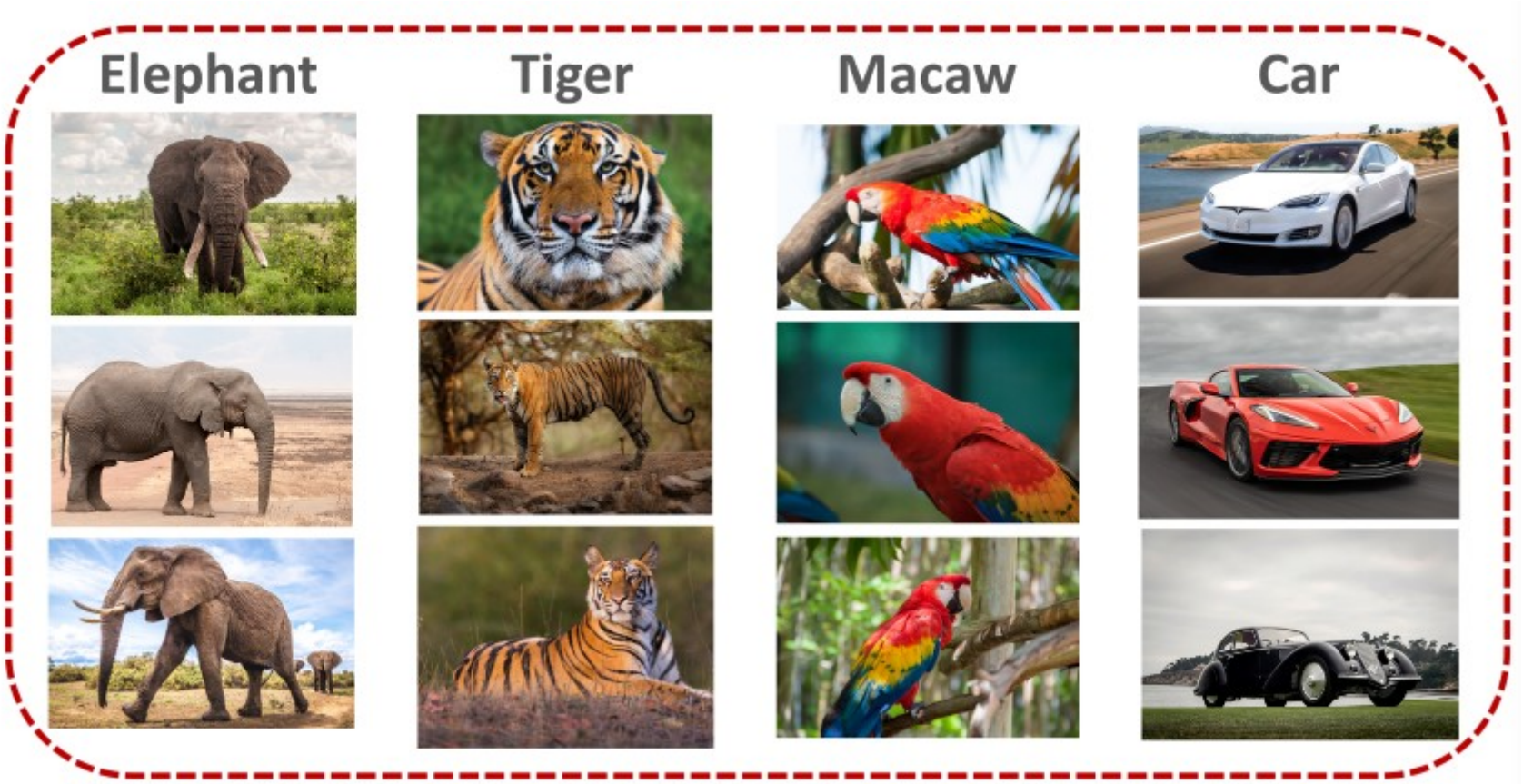


One-shot prediction



Triplet loss

- Outra maneira de treinar a rede siamesa



Amostra âncora

- Selecionada aleatóriamente

Amostra positiva

- Mesma classe da âncora

Amostra negativa

- classe diferente

Função similaridade

$\mathbf{x}^+$
(positive)

$\mathbf{f}$

$\mathbf{f}(\mathbf{x}^+)$

$\mathbf{x}^a$
(anchor)

$\mathbf{f}$

$\mathbf{f}(\mathbf{x}^a)$

$\mathbf{x}^-$
(negative)

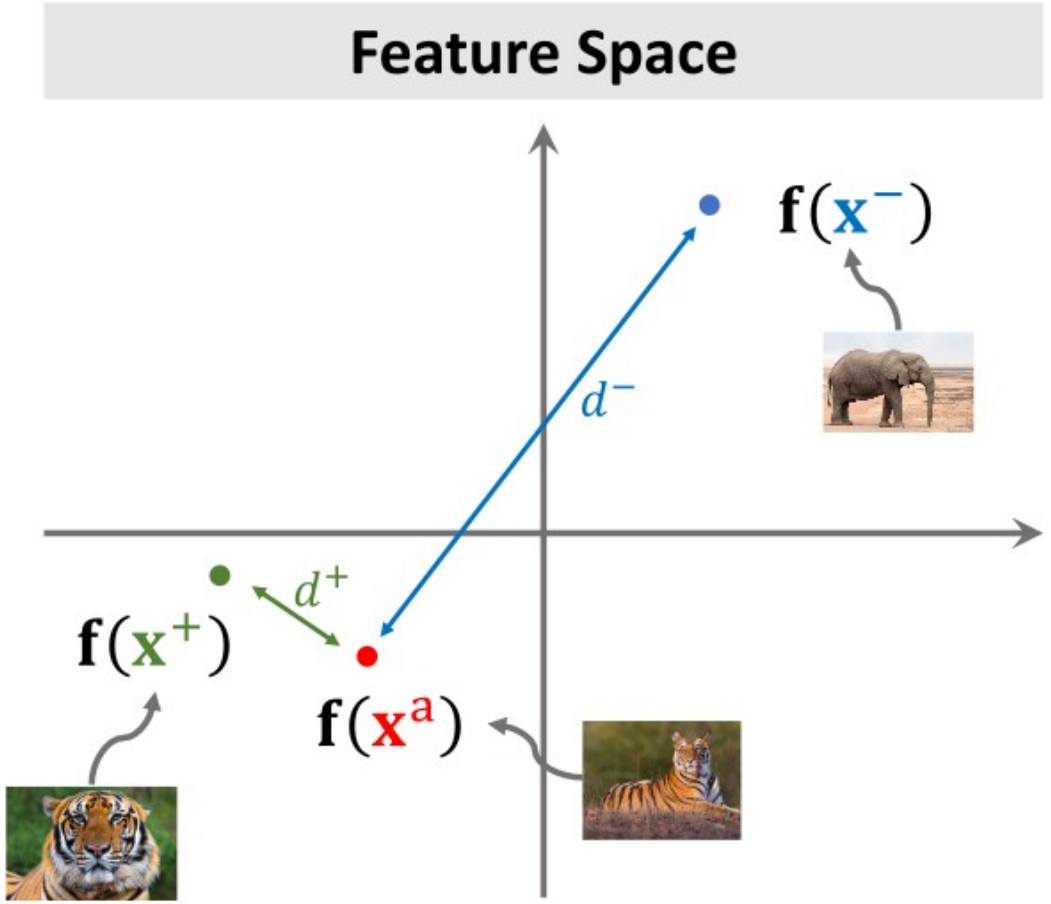
$\mathbf{f}$

$\mathbf{f}(\mathbf{x}^-)$

$$d^+ = \left|\left|\mathbf{f}(\mathbf{x}^+) - \mathbf{f}(\mathbf{x}^a)\right|\right|_2^2$$

$$d^- = \left|\left|\mathbf{f}(\mathbf{x}^a) - \mathbf{f}(\mathbf{x}^-)\right|\right|_2^2$$

d+ deve ser menor que d-



Por baixo dos panos

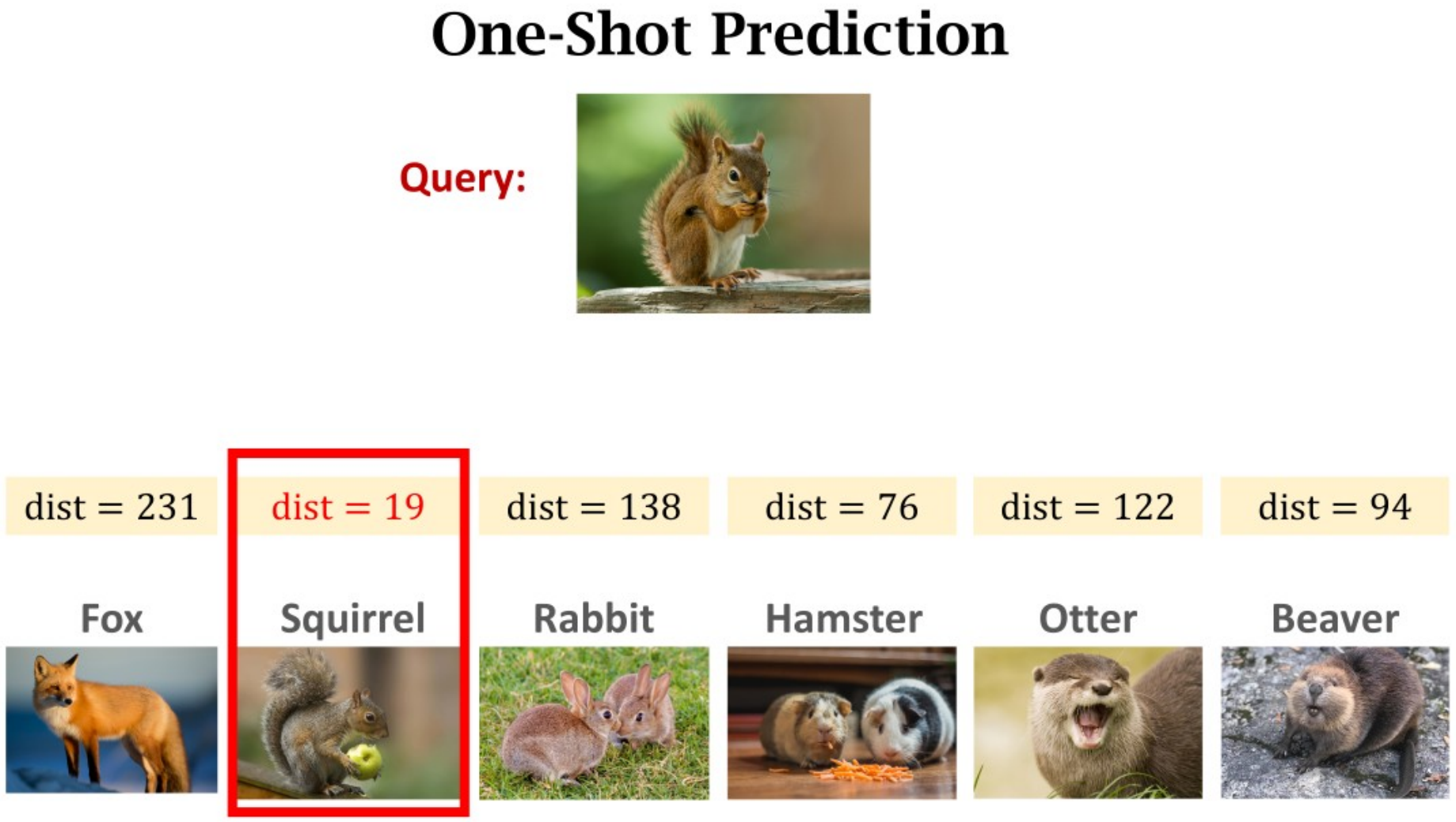
$\mathbf{x}^+$
(positive)

$\mathbf{x}^a$
(anchor)

$\mathbf{x}^-$
(negative)

- Encourage $d^+ = \left|\left|\mathbf{f}(\mathbf{x}^+) - \mathbf{f}(\mathbf{x}^a)\right|\right|_2^2$ to be small.
- Encourage $d^- = \left|\left|\mathbf{f}(\mathbf{x}^a) - \mathbf{f}(\mathbf{x}^-)\right|\right|_2^2$ to be big.
- If $d^- \geq d^+ + \alpha$, then no loss. ($\alpha > 0$ is margin.)
- Otherwise, the loss is $d^+ + \alpha - d^-$.
- $\text{Loss}(\mathbf{x}^a, \mathbf{x}^+, \mathbf{x}^-) = \max\{0,\ d^+ + \alpha - d^-\}$.
- Update the CNN (function $\mathbf{f}$) to decrease the loss.

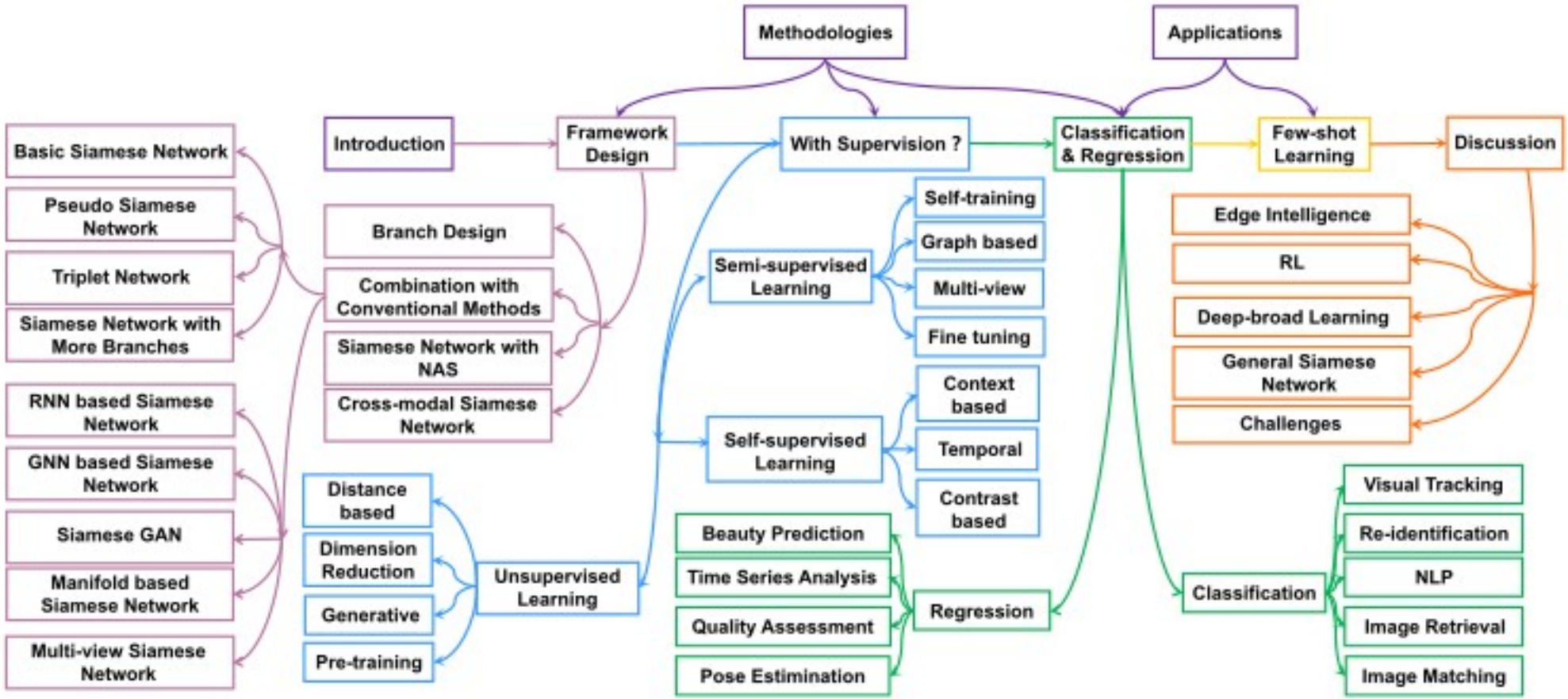
One-shot prediction



Aplicações com redes siamesas

Field	Application	Method
Classification	Visual tracking	SiamFC [129], Cen and Jung [130], Dong and Shen [131], SiamRPN++ [132], SiamBAN-ACM [133], HP-siam [134], DP-Siam [135], HASiam [136], Qi et al. [137], CLNet [138]
	Reidentification	Wu et al. [139], CASN [140], Lin et al. [141], Chung et al. [142], Mai et al. [69]
	NLP	Bhati et al. [143], SDCM [144], HM-LGSN [145], ISA-SNN [146], SS-CNN [147]
	Image retrieval	Qi et al. [148], Khokhlova et al. [149], Shimoda and Yanai [150], DLRTN [151]
	Image matching	sHybridNet [152], Melekhov et al. [153], Kertesz et al. [154], Joshi et al. [155]
Regression	Beauty prediction	Gattupalli et al. [156], R^2-ResNeXt [157], R^3CNN [30]
	Time series analysis	SmLSTM [158]
	Quality assessment	Annaland et al. [159], Prabhudesai and Duong [160], RGR [161]
	Pose estimation	PSGMN [57], Gao and Wang [162], Basaru et al. [163], Yu et al. [164]

Quadro geral de redes siamesas



Referências

- Koch et al – Siamese neural networks for one-shot image recognition (2015)
- Schroff et al – FaceNet: A unified embedding for face recognition and clustering (2015)
- Rosenfeld et al – Totally looks like – how humans compare, compared to machines (2018)
- Li et al – A survey on siamese network: methodologies, applications, and opportunities (2022)
- https://youtu.be/dG8le1YWUI8?si=KXrt3SXgwpS4w5mw
- https://youtu.be/4S-XDefSjTM?si=gDXJZJGdtPZtOPau

Obrigado por sua atenção!

@engmoderno

engmoderno.com.br