

## (8) Tópicos Avançados em Deep Learning

### Redes Neurais e Arquiteturas Profundas

Moacir A. Ponti

*CeMEAI/ICMC, Universidade de São Paulo*

MBA em Ciência de Dados

`www.icmc.usp.br/~moacir — moacir@icmc.usp.br`

São Carlos-SP/Brasil – 2020

# Agenda

Detecção de Objetos e Integração regressão+classificação

Redes multi-fluxo e aprendizado de métricas

Aprendizado auto-supervisionado

# Agenda

Detecção de Objetos e Integração regressão+classificação

Redes multi-fluxo e aprendizado de métricas

Aprendizado auto-supervisionado

# Classificação + regressão

Objetivo: classificar e localizar



Saída da rede

- ▶ Classes
- ▶ Valores de uma caixa (*bounding box*)

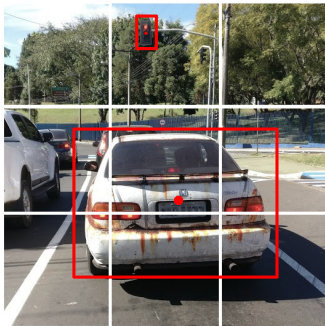
# Classificação + regressão

Formato da predição (saída da rede): presença do objeto, bounding box e classes.

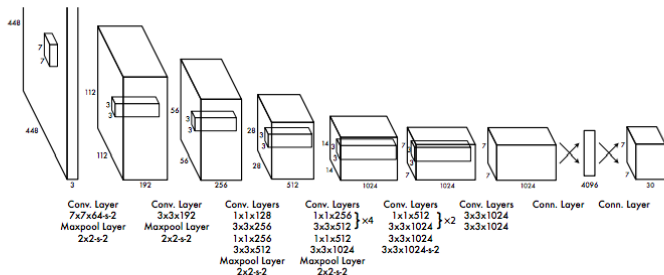


# Classificação + regressão: em um grid

Treinamento considera grid  $S \times S$  (comumente  $19 \times 19$ ) e  $B$  caixas em formatos pré-definidos, chamados de âncoras.



# YOLO: You Only Look Once



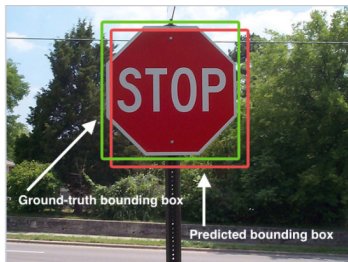
**Figure 3: The Architecture.** Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating  $1 \times 1$  convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution ( $224 \times 224$  input image) and then double the resolution for detection.


Confiança é calculada com:  $P(\text{classe}) \cdot IoU$

Saída é de tamanho  $S \times S \times (5B + C)$

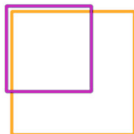
# YOLO: You Only Look Once + IoU

## Intersecção sobre União



$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


IoU: 0.4034



IoU: 0.7330



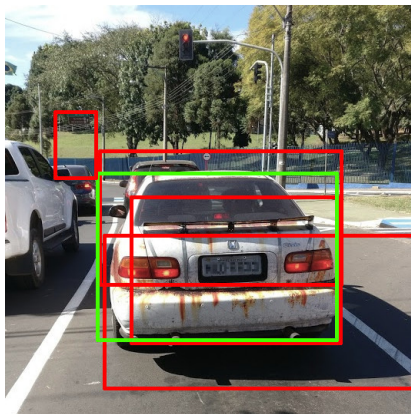
IoU: 0.9264





# YOLO: You Only Look Once + Non-Max Supression

## Supressão de não máximos



- ▶ descartar  $p_c \leq 0.6$
- ▶ selecionar maior  $p_c$
- ▶ descartar caixas com  $IoU \geq 0.5$  da anterior

# YOLO: You Only Look Once + Non-Max Supression

## Supressão de não máximos



- ▶ descartar  $p_c \leq 0.6$
- ▶ selecionar maior  $p_c$
- ▶ descartar caixas com  $IoU \geq 0.5$  da anterior

# Detecção de pontos de referência (landmark)

Exemplo: encontrar pontos de uma face

Formato da predição (saída da rede): presença do objeto de interesse, coordenadas para cada landmark



# Agenda

Detecção de Objetos e Integração regressão+classificação

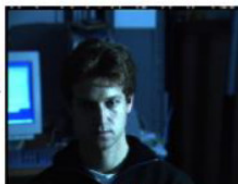
Redes multi-fluxo e aprendizado de métricas

Aprendizado auto-supervisionado

# Lidando com variações intra-classe



1.50



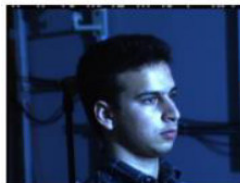
2.90



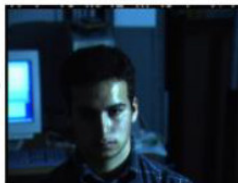
2.13



2.26



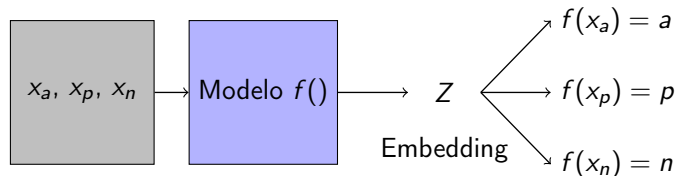
4.10



## Lidando com variações intra-classe

- ▶ Aprender a partir de instâncias diretamente para a saída pode tornar o modelo dependente de características que não representam o que gostaríamos
- ▶ A saída: aprender a partir de grupos de exemplos, em particular pares ou triplas

# FaceNet / Triplet loss



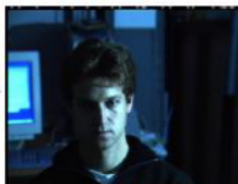
$$||a - p||^2 - ||a - n||^2$$

- O objetivo é aprender representação que obedeça distâncias

# Lidando com variações intra-classe



1.22



1.33

1.04

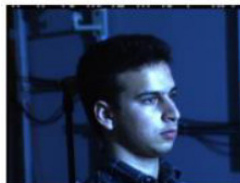


1.33

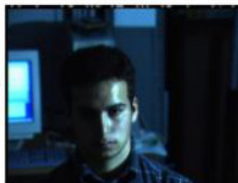


1.26

0.78

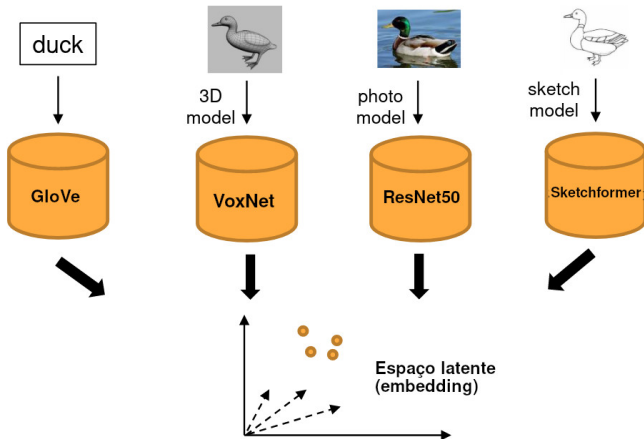


0.99

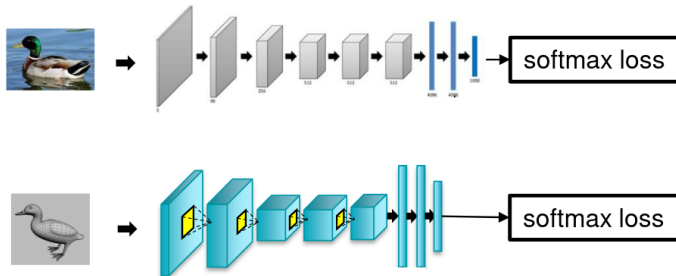




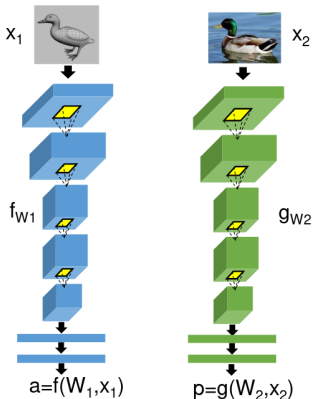
# Redes multi-fluxo e aprendizado multimodal



# Redes multi-fluxo e aprendizado multimodal



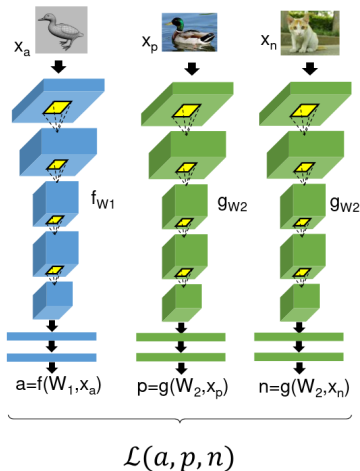
# Redes com função contrastiva



- ▶ *Entrada:* par de exemplos  $x_1, x_2$
- ▶ Modelos podem ser os mesmos ou diferentes (depende dos domínios)
- ▶ Função de custo considera as representações  $a, p$  obtidas da saída de uma das camadas
- ▶ Se  $p$  é positivo, então  $y = 0$ , senão  $y = 1$ , cancelando sempre um dos termos

$$L(a, p) = \frac{1}{2}(1 - y)|a - p|^2 + \frac{1}{2}y[\max(0, m - |a - p|^2)]$$

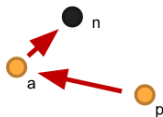
# Redes triplet



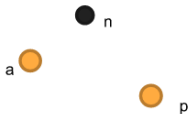
- ▶ *Entrada:* tripla  $x_a, x_p, x_n$
- ▶ Modelos podem ser os mesmos ou diferentes (depende dos domínios)
- ▶ Função de custo considera as representações obtidas da saída de uma das camadas:  $a, p, n$

$$L(a, p, n) = \frac{1}{2} [\max(0, m + |a - p|^2 - |a - n|^2)]$$

# Intuição das funções de custo

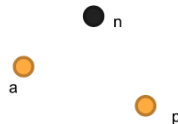


Before training



Contrastive loss

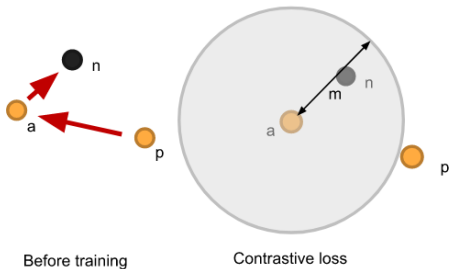
$$L(a, p) = \frac{1}{2} (1 - y) |a - p|_2^2 + \frac{1}{2} y \{ \max(0, m - |a - p|_2^2) \}$$



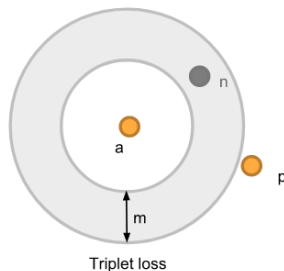
Triplet loss

$$L(a, p, n) = \frac{1}{2} \{ \max(0, m + |a - p|_2^2 - |a - n|_2^2) \}$$

# Intuição das funções de custo

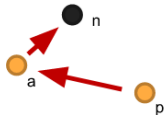


$$L(a, p) = \frac{1}{2} (1 - y) |a - p|_2^2 + \frac{1}{2} y \{ \max(0, m - |a - p|_2^2) \}$$

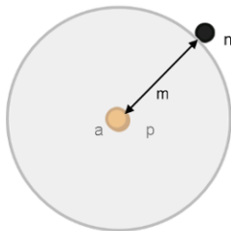


$$L(a, p, n) = \frac{1}{2} \{ \max(0, m + |a - p|_2^2 - |a - n|_2^2) \}$$

# Intuição das funções de custo

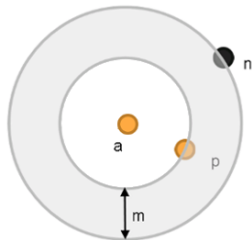


Before training



Contrastive loss

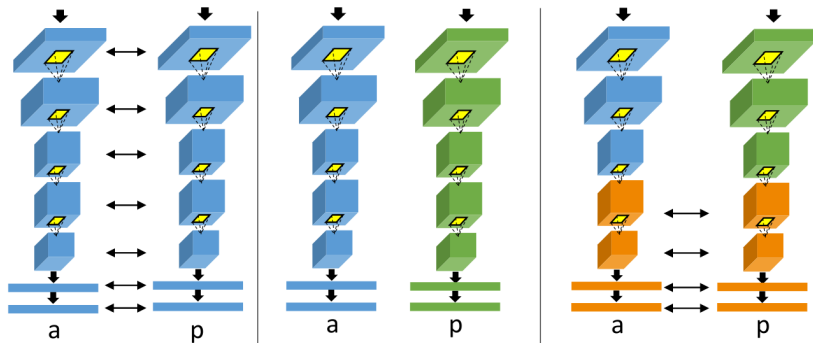
$$L(a, p) = \frac{1}{2} (1 - y) |a - p|_2^2 + \frac{1}{2} y \{ \max(0, m - |a - p|_2^2) \}$$



Triplet loss

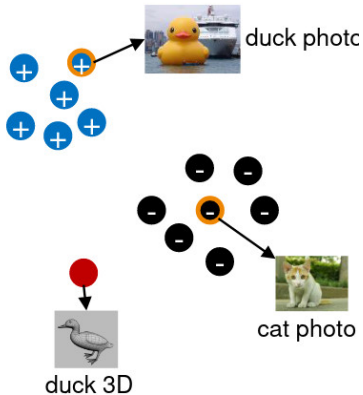
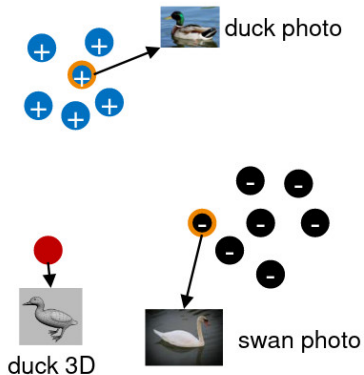
$$L(a, p, n) = \frac{1}{2} \{ \max(0, m + |a - p|_2^2 - |a - n|_2^2) \}$$

# Compartilhamento de pesos





# Estratégia de treinamento: hard positive/negative



# Agenda

Detecção de Objetos e Integração regressão+classificação

Redes multi-fluxo e aprendizado de métricas

Aprendizado auto-supervisionado

# Revisitando categorias de aprendizado

## Aprendizado por reforço

- ▶ retorno fraco a cada etapa
- ▶ funciona bem quando episódios são fáceis de computar/simular

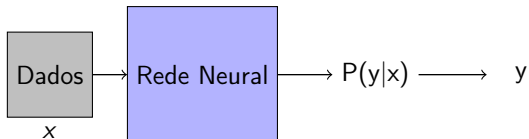
## Aprendizado supervisionado

- ▶ retorno a cada etapa depende da variabilidade e quantidade de dados
- ▶ mas raramente há dados abundantes

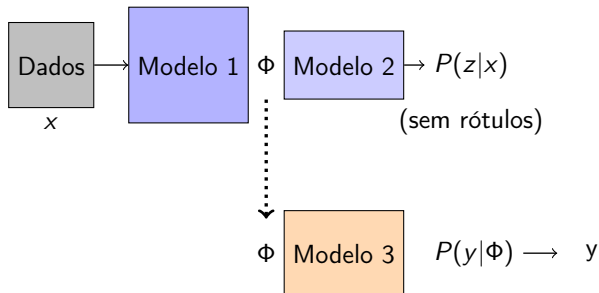
## Aprendizado auto-supervisionado

- ▶ retorno a cada etapa é similar ao supervisionado, mas computado a partir dos dados de entrada
- ▶ podemos gerar número enorme de dados para treinamento

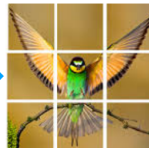
# Aprendizado supervisionado para auto-supervisionado



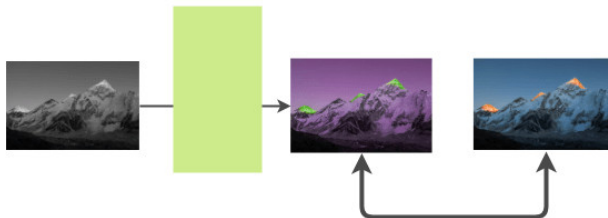
# Aprendizado supervisionado para auto-supervisionado



# Rótulos computáveis: rotação e quebra-cabeça



## Tarefas computáveis: colorização



## Tarefas auxiliares: preencher lacunas

Deixa o menino \_\_\_\_\_ bola e \_\_pren\_\_\_\_\_



## Tarefas auxiliares: preencher lacunas

Deixa o menino pegar bola e aprender

## Outras tarefas possíveis

- ▶ Redes geradoras
- ▶ Denoising Autoencoders
- ▶ Pseudo-labels com agrupamento
- ▶ Aprendizado contrastivo multidomínio: áudio + imagem, áudio + texto

# Considerações finais

- ▶ Redes profundas podem ser adaptadas e usadas em arquiteturas com mais componentes
- ▶ Funções de custo e outras tarefas tem potencial para resolver problemas aplicados
- ▶ Auto-supervisão é "a bola da vez" por diminuir dependência de dados
- ▶ O desafio é adaptar as arquiteturas e métodos aos casos em particular: estruturados, não estruturados (texto, áudio, imagens, vídeo) de acordo com a natureza dos dados.

Obrigado!