(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

Agenda

Detecção de Objetos, regressão+classificação e pixel-to-pixel

Redes multi-fluxo e aprendizado de métricas

Aprendizado auto-supervisionado

BERT: pré-treinamento de encoders de transformers bidirecionais

Agenda

Detecção de Objetos, regressão+classificação e pixel-to-pixel

Redes multi-fluxo e aprendizado de métricas

Aprendizado auto-supervisionado

BERT: pré-treinamento de encoders de transformers bidirecionais

Classificação + regressão

Objetivo: classificar e localizar

Saída da rede

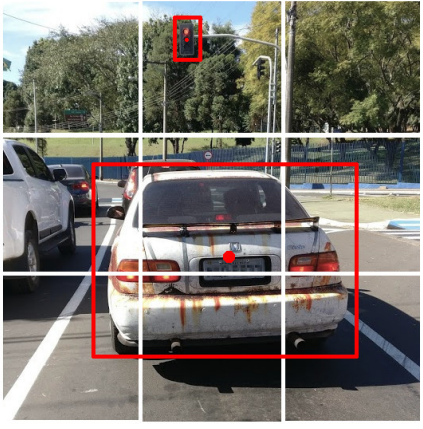
- Classes

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 19×19) e B caixas em formatos pré-definidos, chamados de âncoras.



YOLO: You Only Look Once

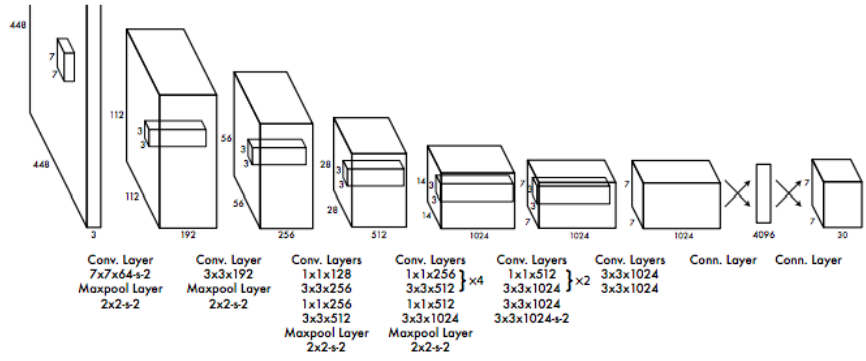
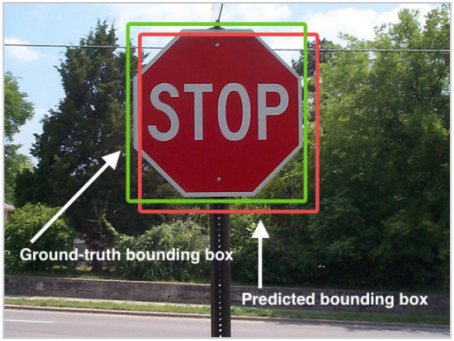


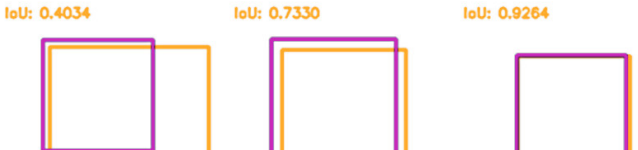
Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224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

YOLO: You Only Look Once + IoU

Intersecção sobre União

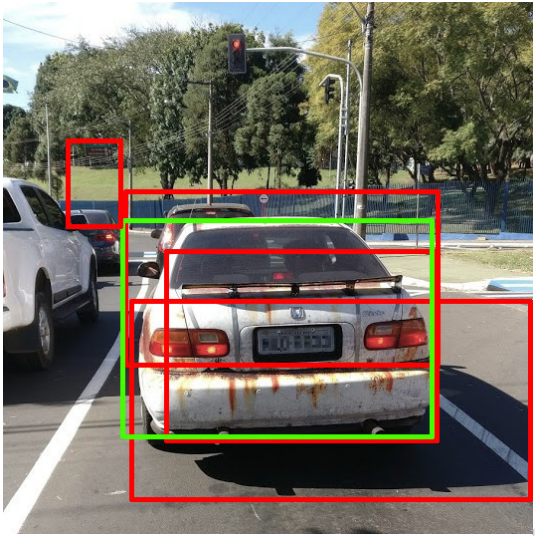


$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$



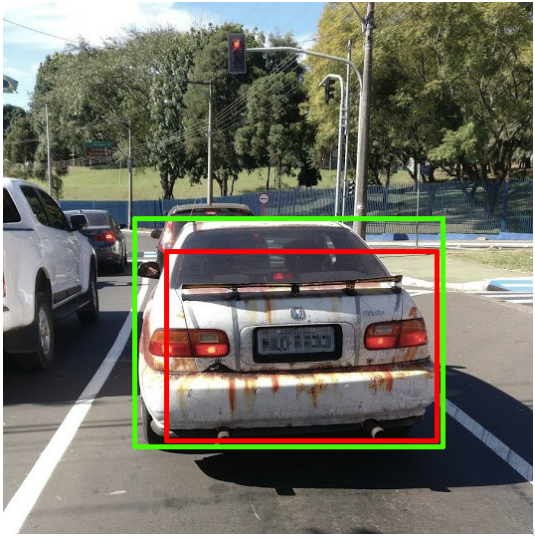
YOLO: You Only Look Once + Non-Max Supression

Supressão de não máximos



YOLO: You Only Look Once + Non-Max Supression

Supressão de não máximos

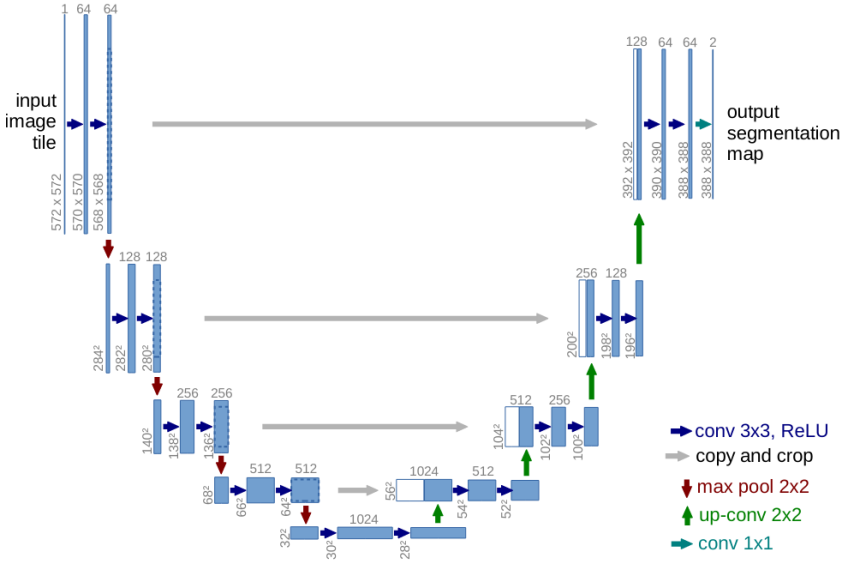


Detecção de pontos de referência (landmark)

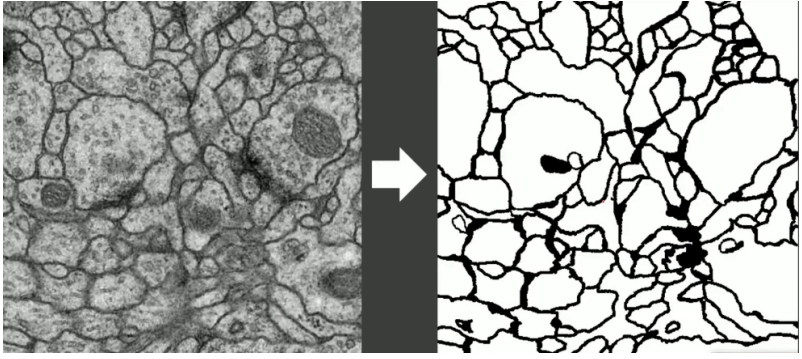
Exemplo: encontrar pontos de uma face

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

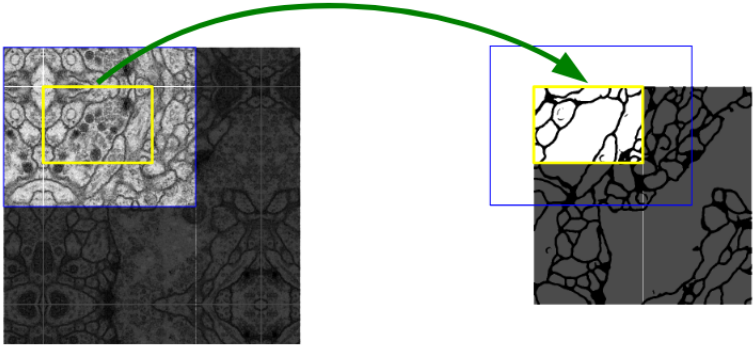
U-Net


input image tile
output segmentation map
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1

U-Net



U-Net



U-Net

Função de custo

- Softmax pixel-a-pixel (ao longo dos canais) + Entropia Cruzada
- Pesos computados para:
  1. compensar desbalanceamento (fundo é comumente mais proeminente do que os alvos)
  2. dar mais peso às bordas das regiões a serem segmentadas

Treinamento:

- Data augmentation: utilizando deformação suave das imagens
- Inicialização dos pesos por camada
  1. baseada na camada anterior
  2. distribuição Gaussiana/normal $\sigma = \sqrt{2/N}$, N sendo o número de nós de entrada
     — exemplo: camada anterior com 64 filtros 3×3, $\sigma = \sqrt{2/(9 \cdot 64)} = \sqrt{2/576}$

Agenda

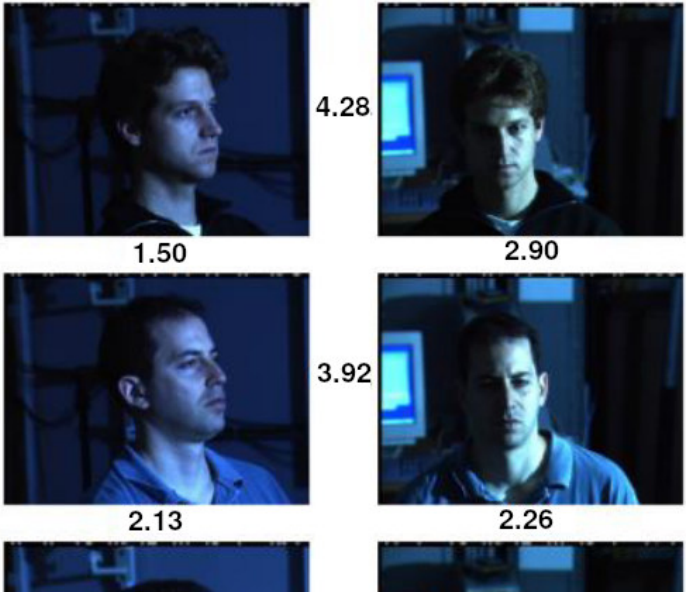
Detecção de Objetos, regressão+classificação e pixel-to-pixel

Redes multi-fluxo e aprendizado de métricas

Aprendizado auto-supervisionado

BERT: pré-treinamento de encoders de transformers bidirecionais

Lidando com variações intra-classe



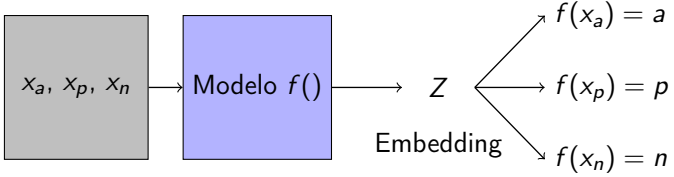
Lidando com variações intra-classe

- Aprender a partir de instâncias diretamente para a saída pode tornar o modelo dependente de características que não representam o que gostaríamos

Lidando com variações intra-classe

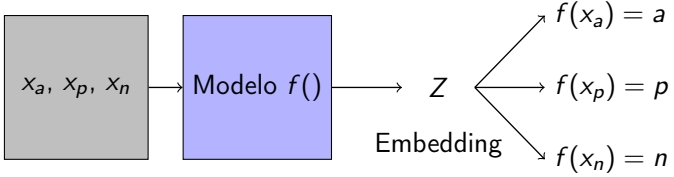
- Aprender a partir de instâncias diretamente para a saída pode tornar o modelo dependente de características que não representam o que gostaríamos
- A saída: aprender a partir de grupos de exemplos, em particular pares ou triplas

FaceNet / Triplet loss



- O objetivo é aprender representação que obedeça distâncias

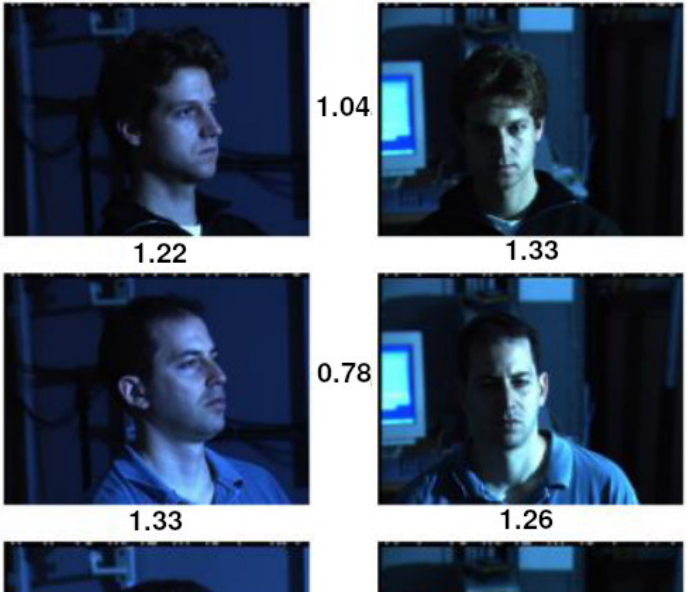
FaceNet / Triplet loss



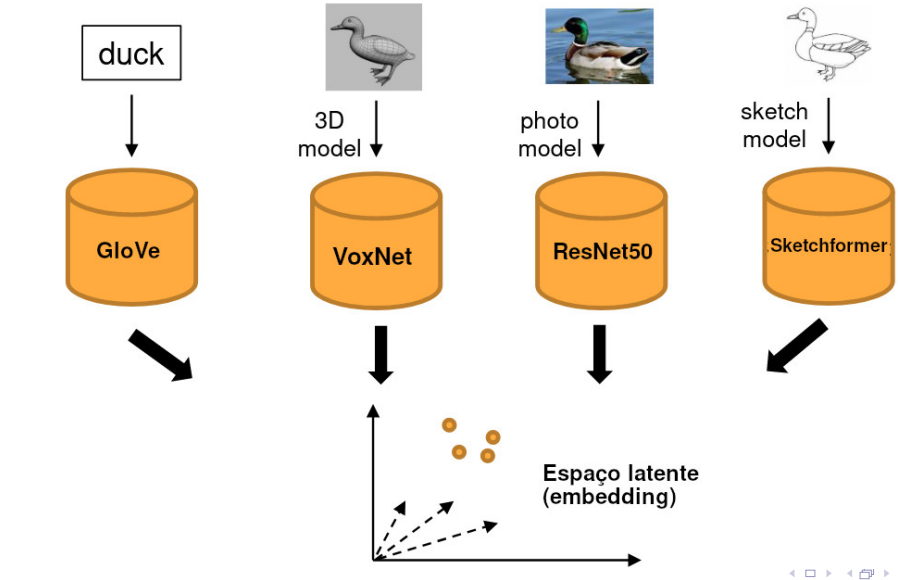
$$||a - p||^2 - ||a - n||^2$$

- O objetivo é aprender representação que obedeça distâncias

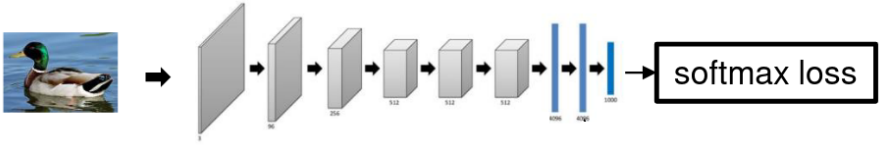
Lidando com variações intra-classe



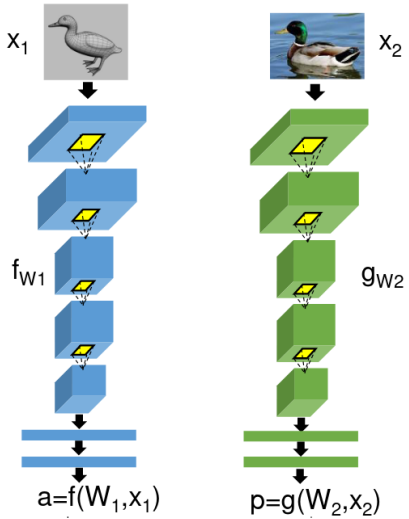
Redes multi-fluxo e aprendizado multimodal



Redes multi-fluxo e aprendizado multimodal



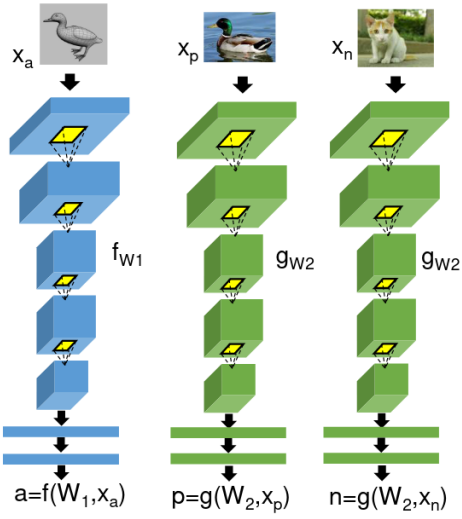
Redes com função contrastiva



- *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

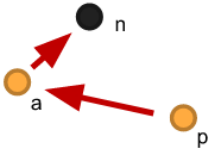
$$L(a, p) = \frac{1}{2}(1 - y)|a - p|^2 + \frac{1}{2}y[\max(0, m - |a - p|^2)]$$

Redes triplet

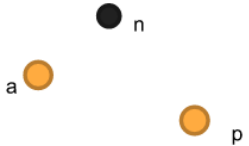


- *Entrada*: tripla x_a, x_p, x_n
- Modelos podem ser os mesmos ou diferentes (depende dos domínios)
- Função de custo considera as representações obtidas da saída de uma das camadas: a, p, n

Intuição das funções de custo

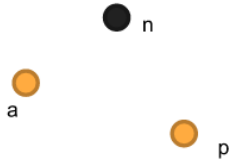


Before training



Contrastive loss

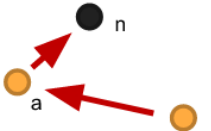
$$L(a,p) = \frac{1}{2}(1-y)|a-p|_2^2 + \\ + \frac{1}{2}\, y\, \{\max(0, m - |a-p|_2^2\}$$



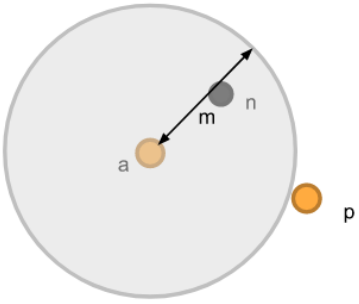
Triplet loss

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

Intuição das funções de custo

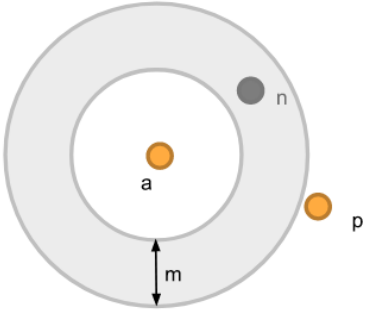


Before training



Contrastive loss

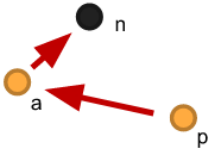
$$L(a,p) = \frac{1}{2}(1-y)|a-p|_2^2 + \\ + \frac{1}{2}\, y\, \{\max(0, m - |a-p|_2^2\}$$



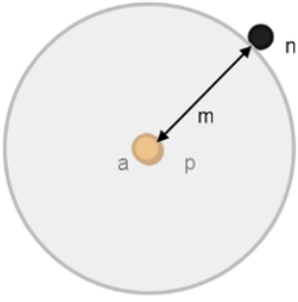
Triplet loss

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

Intuição das funções de custo

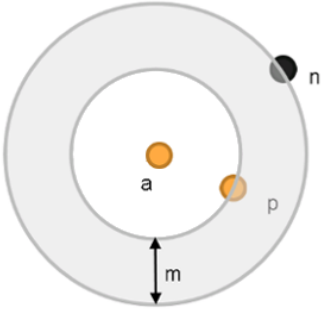


Before training



Contrastive loss

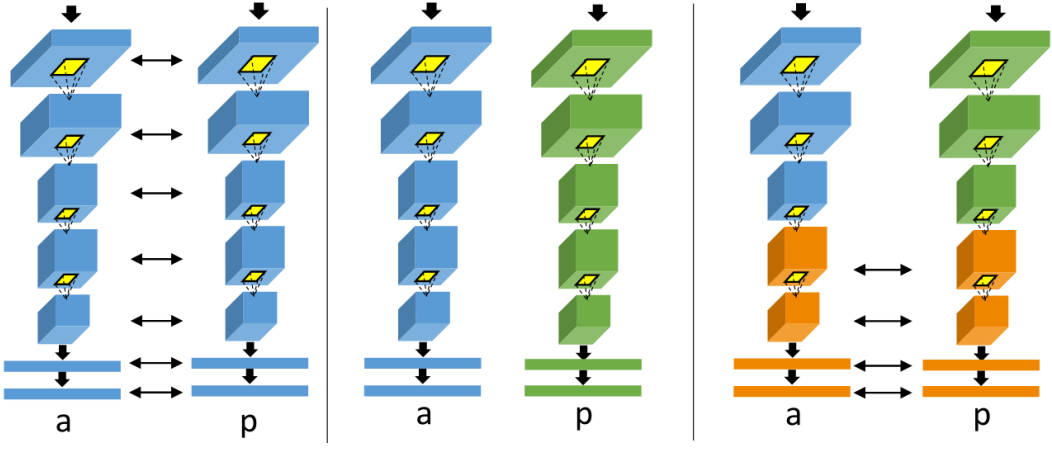
$$L(a,p) = \frac{1}{2}(1-y)|a-p|_2^2 + \\ + \frac{1}{2}\, y\, \{\max(0, m - |a-p|_2^2\}$$



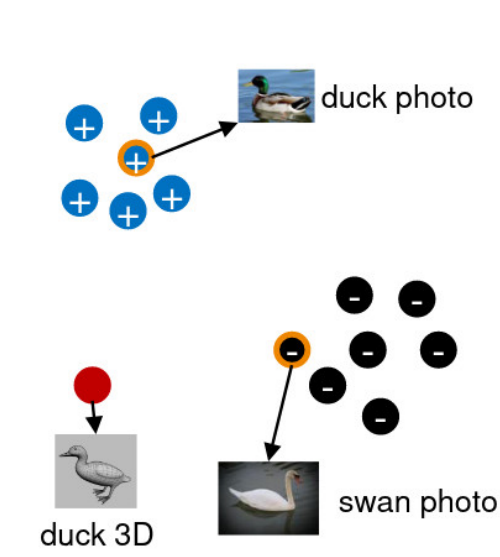
Triplet loss

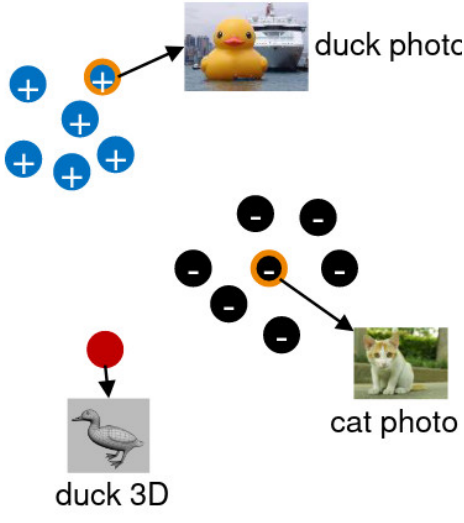
$$L(a,p,n) = \frac{1}{2}\ \{\max(0, m + \\ + |a-p|_2^2 - |a-n|_2^2\}$$

Compartilhamento de pesos



Estratégia de treinamento: hard positive/negative





Agenda

Detecção de Objetos, regressão+classificação e pixel-to-pixel

Redes multi-fluxo e aprendizado de métricas

Aprendizado auto-supervisionado

BERT: pré-treinamento de encoders de transformers bidirecionais

Revisitando categorias de aprendizado

Aprendizado por reforço

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

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

Revisitando categorias de aprendizado

Aprendizado por reforço

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

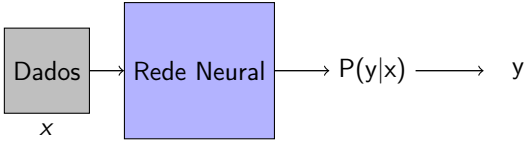
Aprendizado supervisionado

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

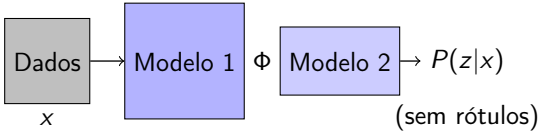
Aprendizado auto-supervisionado

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

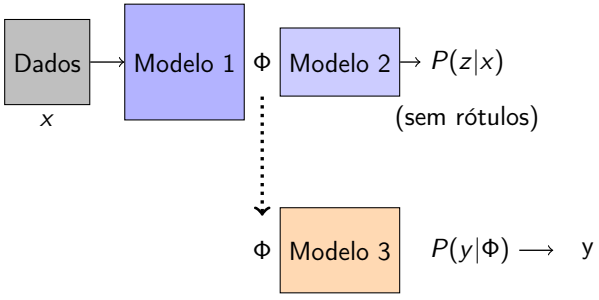
Aprendizado supervisionado para auto-supervisionado



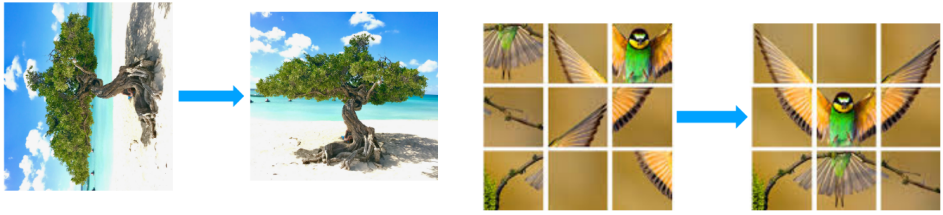
Aprendizado supervisionado para auto-supervisionado



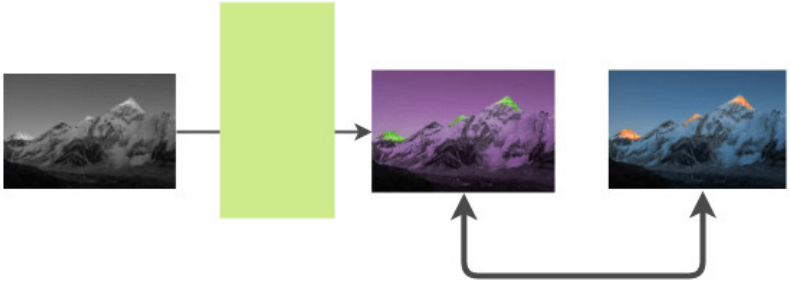
Aprendizado supervisionado para auto-supervisionado



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 constrastivo multidomínio: áudio + imagem, áudio + texto

Agenda

Detecção de Objetos, regressão+classificação e pixel-to-pixel

Redes multi-fluxo e aprendizado de métricas

Aprendizado auto-supervisionado

BERT: pré-treinamento de encoders de transformers bidirecionais

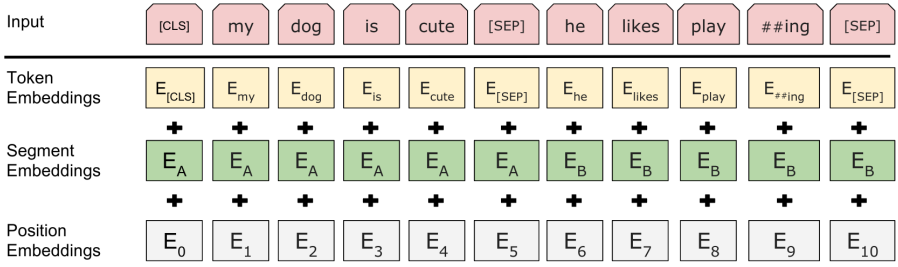
BERT e Transformer

Método para **pré-treinar** encoders to tipo Transformer

— BERT Base e BERT Large, modelos com blocos Transformer:

- Base:
  - 12 camadas (blocos Transformer)
  - Embedding com 768 dimensões
  - 110 milhões de parâmetros
- Large:
  - 24 camadas (blocos Transformer)
  - Embedding com 1024 dimensões
  - 336 milhões de parâmetros

BERT Treinamento e Embeddings

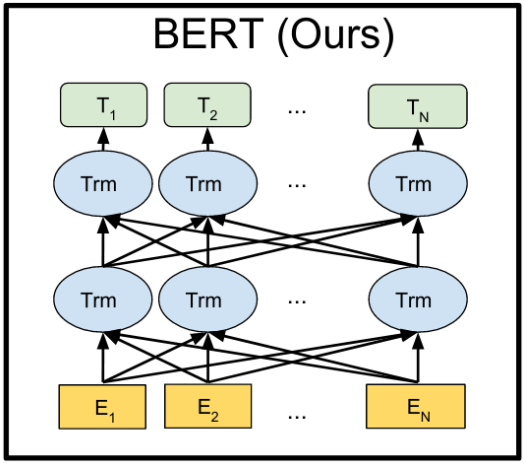


BERT: pré-treinamento inspirado em ELMo

- GloVe e similares
  - vetor fixo por palavra independente do contexto
  - ex. manga, bateria, pilha.
- ELMo olha para a sentença antes de atribuir vetor
  - usa LSTM bidirecional para criar o embedding
  - aprende (sem labels) a predizer a próxima palavra (e a anterior)
  - BERT usa essa ideia mas com transformers

vamos deixar o menino jogar bola
$\rightarrow$ vamos deixar o menino jogar [mask]
[mask] deixar o menino jogar bola $\leftarrow$
vamos deixar o $\rightarrow$ [mask] $\leftarrow$jogar bola

BERT Bidirectional Transformer



BERT: segunda tarefa de pré-treinamento

$$\text{Input} = \texttt{[CLS] the man went to [MASK] store [SEP]}$$
$$\texttt{he bought a gallon [MASK] milk [SEP]}$$
$$\text{Label} = \texttt{IsNext}$$

$$\text{Input} = \texttt{[CLS] the man [MASK] to the store [SEP]}$$
$$\texttt{penguin [MASK] are flight ##less birds [SEP]}$$
$$\text{Label} = \texttt{NotNext}$$

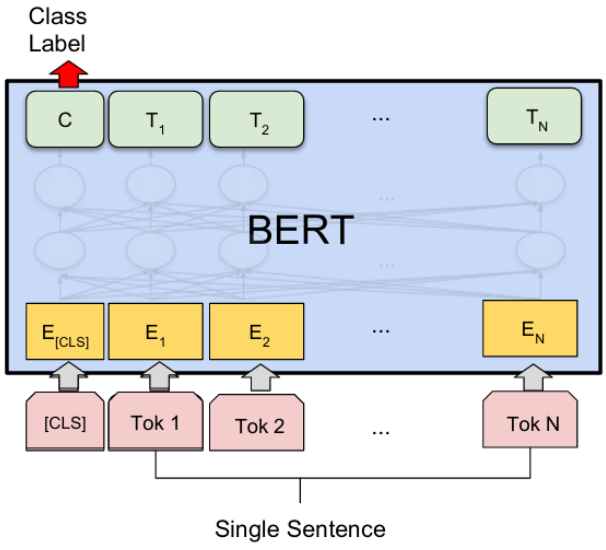
BERT: ULM-FiT

- ULM-FiT usa metodos efetivos para pré-treinamento para além:
  - de word embeddings
  - de word embeddings contextualizados
- Modelo de linguagem + estratégia para ajustar modelo para várias tarefas
- Descongelamento gradual: da última camada até a primeira

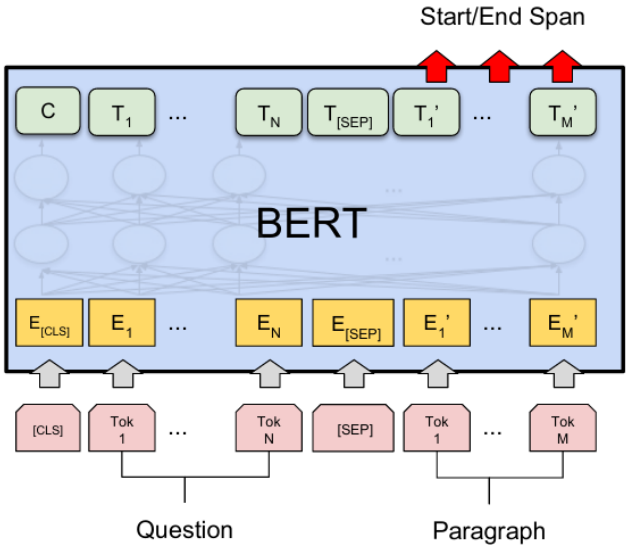
BERT: datasets

- Book Corpus: 800M palavras
- Wikipedia Inglês: 2.4B palavras

BERT: para classificação



BERT: para encontrar resposta em texto



Considerações finais

- Redes profundas podem ser adaptadas e usadas em arquiteturas com mais componentes

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 e pré-treinamento são indicadas como potenciais para diminuir dependência de dados

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 e pré-treinamento são indicadas como potenciais para 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!