

Redes Neurais Artificiais

Transfer Learning



INFORMAÇÃO,
TECNOLOGIA
& INOVAÇÃO

Introdução

2

- ❑ O treinamento de uma rede neural profunda em grandes conjuntos de dados pode levar semanas
- ❑ Podemos reduzir esse tempo de treinamento utilizando pesos pré-treinados de outra rede neural artificial:
 - ▣ Transfer Learning
- ❑ Trata-se de aproveitar as representações de atributos de um modelo pré-treinado, para evitar o treinamento de novo modelo do 0

Introdução

3

- ❑ Os modelos pré-treinados são geralmente induzidos em conjuntos de dados grandes. Seus pesos podem então serem reutilizados em outras tarefas.
- ❑ Esses modelos podem ser utilizados diretamente para fazer previsões em tarefas novas, ou então integrados no processo de treinamento de novos modelos.
 - ❑ Ideia: menos tempo de treinamento e melhor generalização
- ❑ É muito comum na prática que se use modelos pré-treinados da literatura, como o VGG e o Inception.

Introdução

4

- A maioria dos modelos pré-treinados usados em transfer learning são redes neurais convolucionais (CNNs).
- Relembrando, uma CNN é composta basicamente por duas partes principais:
 - ▣ Base convolucional: camadas de convolução e pooling empilhadas, que geram atributos das imagens.
 - ▣ Classificador: camadas totalmente conectadas, que classificam as imagens baseado nos atributos gerados.

Introdução

5



Fonte: <https://towardsdatascience.com/transfer-learning-from-pre-trained-models-f2393f124751>

- A rede aprende representações hierárquicas dos atributos.
 - ▣ Atributos aprendidos pela primeira camada são mais gerais e podem ser reutilizados em diferentes domínios.
 - ▣ Atributos aprendidos pela camadas mais profundas são mais específicos e dependem dos dados e domínio.

Reutilizando um Modelo Pré-Treinado

6

- Iniciamos removendo o classificador original (camadas totalmente conectadas), adicionamos um novo classificador, fazemos um ajuste fino (fine-tuning) do modelo.
- Há três estratégias para o ajuste fino:
 - ▣ 1 - Treinamento de todo o modelo.
 - ▣ 2 - Treinamento de algumas camadas, deixando outras “congeladas”.
 - ▣ 3 - Congelamento da base convolucional.

Reutilizando um Modelo Pré-Treinado

7

- Treinamento de todo o modelo.
 - Utiliza a arquitetura do modelo pré-treinado e treina o modelo de acordo com os dados.
 - Nesse caso, o modelo será treinado desde o início, portanto é necessário um conjunto de dados grande e bastante poder computacional.

Reutilizando um Modelo Pré-Treinado

8

- Treinamento de algumas camadas, deixando outras “congeladas”.
 - Camadas mais rasas são mais gerais (independentes do problema), enquanto camadas mais profundas tratam atributos mais específicos (dependentes do problema).
 - Precisamos então escolher quais camadas treinar e o quanto treinar, e congelar camadas cujos pesos não queremos treinar.
 - Para conjuntos de dados pequenos e grandes números de hiperparâmetros, congelamos mais camadas para evitar overfitting.
 - Para conjuntos de dados grandes e menos hiperparâmetros, podemos usar mais camadas para o treinamento

Reutilizando um Modelo Pré-Treinado

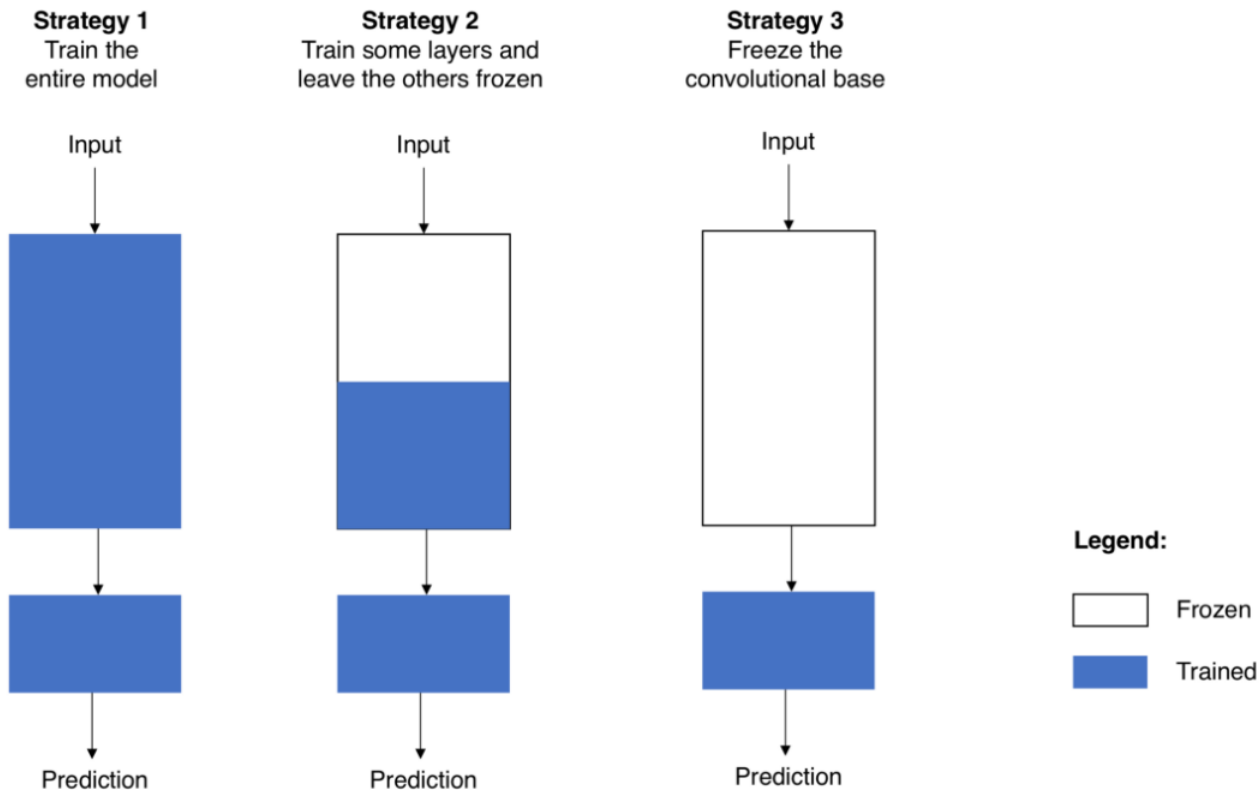
9

- ❑ Congelamento da base convolucional.
 - ▣ Usa a arquitetura e pesos das camadas convolucionais e de pooling em seus formatos originais, e usa a saída como entrada para o classificador (camadas densas).
 - ▣ Assim, o modelo pré-treinado é usado como um extrator de atributos fixo.
 - ▣ Útil quando temos pouco poder computacional, um conjunto de dados pequeno, ou o modelo pré-treinado foi construído em um conjunto de dados similar ao do problema que queremos solucionar

Reutilizando um Modelo Pré-Treinado

10

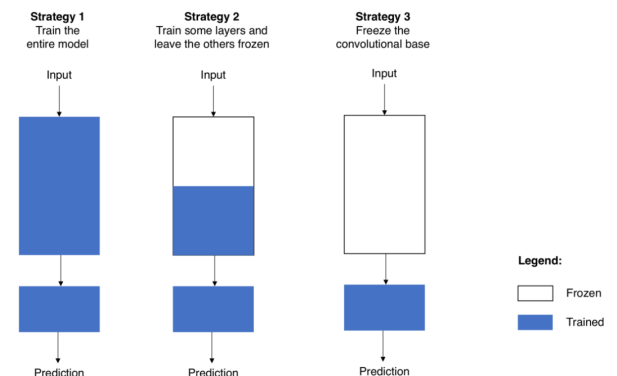
□ As três estratégias de fine-tuning:



Reutilizando um Modelo Pré-Treinado

11

- Nas estratégias 1 e 2 é preciso tomar cuidado com a taxa de aprendizado utilizada.
- Uma taxa de aprendizado muito alta pode levar à perda de conhecimento adquirido previamente.
- Manter uma pequena taxa de aprendizado garantirá que os pesos da CNN não sejam distorcidos muito cedo e em grande intensidade.



○ Processo de Transfer Learning

12

- Seleção de um modelo pré-treinado.
 - ▣ Podemos escolher um modelo dentre vários modelos disponíveis na literatura, tais como VGG e Inception
 - ▣ O Keras disponibiliza diferentes modelos
 - <https://keras.io/api/applications/>

Available models

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth	Time (ms) per inference step (CPU)	Time (ms) per inference step (GPU)
Xception	88	79.0%	94.5%	22.9M	81	109.4	8.1
VGG16	528	71.3%	90.1%	138.4M	16	69.5	4.2
VGG19	549	71.3%	90.0%	143.7M	19	84.8	4.4
ResNet50	98	74.9%	92.1%	25.6M	107	58.2	4.6
ResNet50V2	98	76.0%	93.0%	25.6M	103	45.6	4.4
ResNet101	171	76.4%	92.8%	44.7M	209	89.6	5.2
ResNet101V2	171	77.2%	93.8%	44.7M	205	72.7	5.4
ResNet152	232	76.6%	93.1%	60.4M	311	127.4	6.5
ResNet152V2	232	78.0%	94.2%	60.4M	307	107.5	6.6
InceptionV3	92	77.9%	93.7%	23.9M	189	42.2	6.9
InceptionResNetV2	215	80.3%	95.3%	55.9M	449	130.2	10.0

○ Processo de Transfer Learning

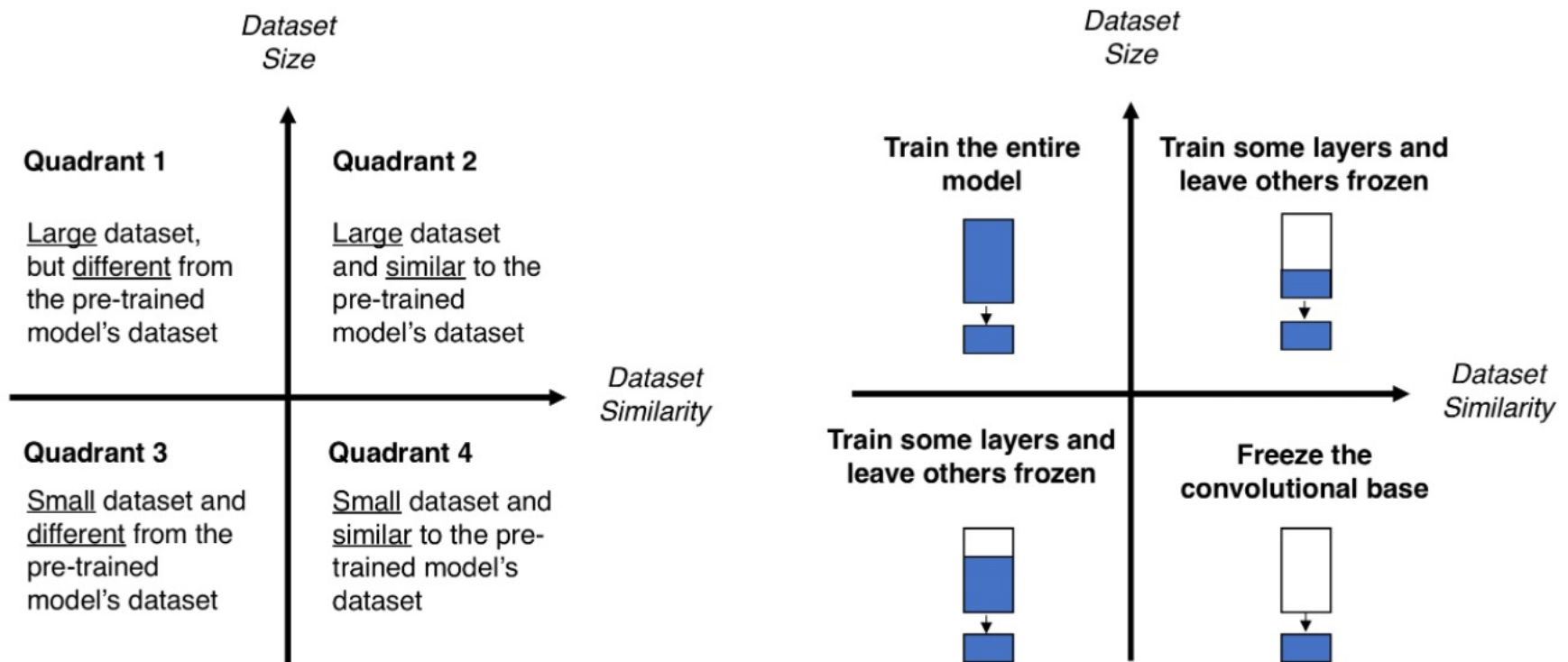
13

- Classifique o problema adequadamente.
 - Classifica o problema considerando o tamanho do conjunto de dados e sua semelhança com o conjunto de dados no qual o modelo pré-treinado foi treinado.
 - Conjunto de dados pequeno: menos de 1 000 imagens por classe.
 - Considerando semelhança entre conjuntos de dados, se a nova tarefa é identificar cães e gatos, o ImageNet é um conjunto de dados semelhante, pois possui imagens de cães e gatos.
 - No entanto, se a nova tarefa é identificar células cancerígenas, o ImageNet não pode ser considerado um conjunto de dados semelhante.

○ Processo de Transfer Learning

14

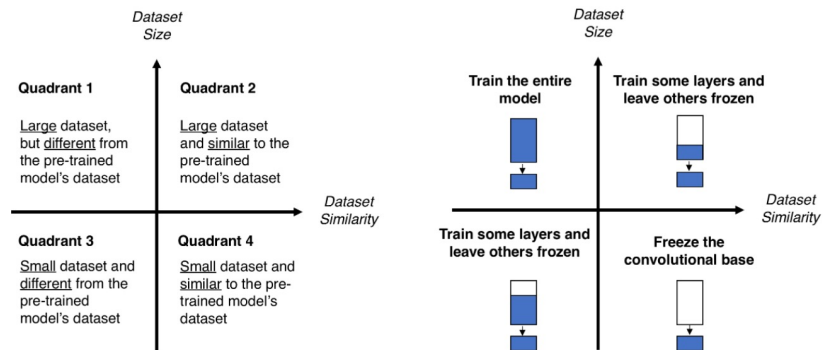
- Ajuste fino (fine tuning) do modelo.



○ Processo de Transfer Learning

15

□ Ajuste fino

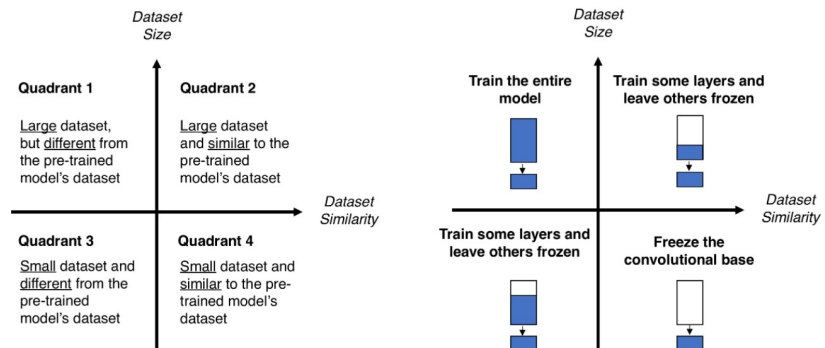


- Quadrante 1: use a estratégia 1. Como tem-se um grande conjunto de dados, pode-se treinar um modelo do zero.
- Apesar da dissimilaridade do conjunto de dados, na prática, ainda pode ser útil inicializar o modelo a partir de um modelo pré-treinado, usando sua arquitetura e pesos

○ Processo de Transfer Learning

16

□ Ajuste fino

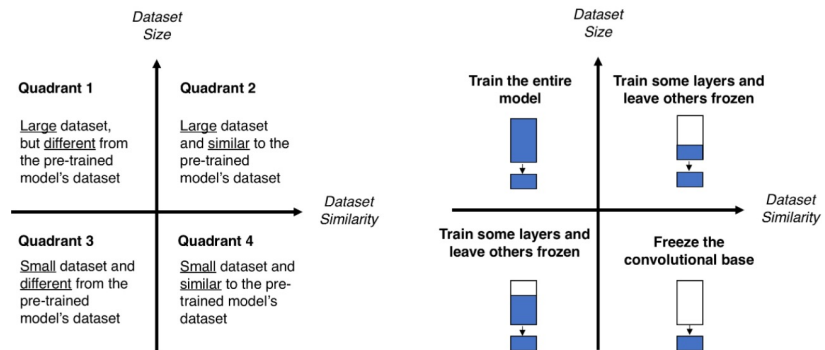


- Quadrante 2: provavelmente a melhor opção seja a estratégia 2. Como tem-se um grande conjunto de dados, o overfitting não deve ser um problema.
- No entanto, como os conjuntos de dados são semelhantes, pode-se aproveitar conhecimento anterior e poupar tempo de treinamento.
- Deve ser suficiente treinar o classificador e as camadas superiores da base convolucional.

○ Processo de Transfer Learning

17

□ Ajuste fino

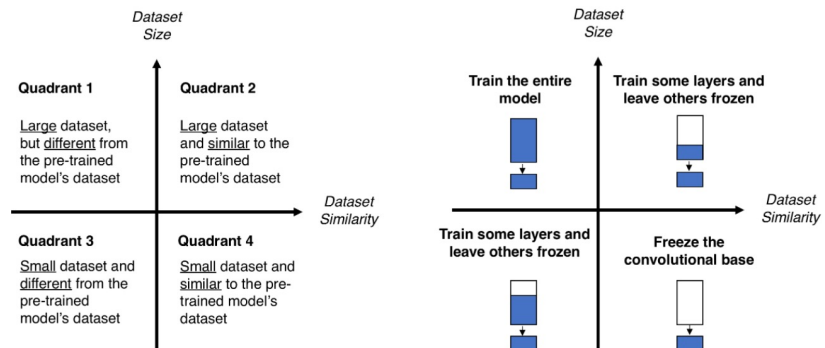


- Quadrante 3: cenário mais difícil. A estratégia 2 é recomendada. É preciso pensar no número ideal de camadas para treinar e congelar.
- Se usar muitas camadas para treino, pode ocorrer overfitting. Se usar poucas camadas, pode não aprender nada.
- Data augmentation pode ser considerado

○ Processo de Transfer Learning

18

□ Ajuste fino



- Quadrante 4: a estratégia mais recomendada é a 3. Apenas removemos o classificador (camadas densas).
- Executa-se a parte convolucional como um extrator de características, que são dadas como entrada para o treinamento de outro classificador.
- Esse classificador pode ser uma rede neural densa.

Classificadores em DNNs profundas

19

- Como já mencionado, modelos resultantes do processo de Transfer Learning, baseados no pré-treino de CNNs são formados por duas partes:
 - ▣ Base Convolutacional, que faz extração de características.
 - ▣ Classificador, que classifica os dados usando as características extraídas pela base convolutacional.

Classificadores em DNNs profundas

20

- Diferentes estratégias podem ser utilizadas como classificador no topo de CNNs profundas:
 - Camadas totalmente conectadas utilizando a função softmax na camada de saída.
 - Global average pooling: ao invés de camadas totalmente conectadas, usa-se uma camada de pooling global, cuja saída vai direto para uma softmax.
 - Classificadores como uma Máquina de Vetores de Suporte linear (SVMs).

Obrigado!

21

□ Perguntas?

