

MBA em Ciência de Dados

Redes Neurais e Arquiteturas Profundas

Módulo IV - Estratégias de Treinamento e Transferência de Aprendizado

Avaliação

Moacir Antonelli Ponti

CeMEAI - ICMC/USP São Carlos

As respostas devem ser dadas no Moodle, use esse notebook apenas para gerar o código necessário para obter as respostas

Questão 1)

Qual a relação entre o modelo chamado de "memorizador" e as redes neurais profundas?

- (a) Redes neurais com alta capacidade podem memorizar todos os exemplos de treinamento, tornando-as hábeis para generalizar para dados futuros.
 - (b) Redes neurais com alta capacidade podem memorizar todos os exemplos de treinamento, falhando em prever corretamente exemplos não vistos.
 - (c) Redes neurais com alta capacidade são imunes a convergir para modelos memorizadores, pois obtiveram resultados do estado-da-arte em muitas aplicações.
 - (d) Redes neurais com alta capacidade podem memorizar todos os exemplos de treinamento, e portanto possuem viés forte.
-

Questão 2)

O papel do uso conjunto dos métodos BatchNormalization e Regularização é o de:

- (a) Pré-processamento dos dados antes da realização do treinamento
 - (b) Gerar espaço de parâmetros esparsos, com alguns poucos parâmetros com valor alto e muitos com valores próximo a zero, melhorando a generalização
 - (c) Minimizar o problema do desaparecimento do gradiente, e ao mesmo tempo evitar que poucas unidades/neurônios se especializem demais
 - (d) Obter robustez com relação à possíveis ataques e propiciar modelos menores com acurácia similar a modelos maiores
-

Questão 3)

São práticas viáveis para o uso de aprendizado profundo com pequenas bases de dados:

- (a) Carregar uma rede neural profunda popular de um pacote de software e treiná-la a partir de pesos aleatórios
- (b) Carregar uma rede neural profunda pré-treinada em grande base de dados, e utilizar a saída da última camada da rede, ou seja as previsões das classes, como característica para modelos de aprendizado externos que permitem uso com menores bases de dados
- (c) Carregar uma rede neural profunda popular de um pacote de software e treiná-la a partir de pesos

aleatórios utilizando Batch Normalization

(d) Carregar uma rede neural profunda pré-treinada em grande base de dados, inserindo uma nova camada de saída treinando apenas essa camada com a pequena base de dados

Questão 4)

Carregue a base de dados Fashion MNIST conforme código abaixo e crie um modelo de CNN com a seguinte arquitetura, capaz de obter classificação nessa base de dados de imagens. Considere que todas as camadas convolucionais tem zeropadding, e ativação relu, exceto quando mencionado contrário.

1. Pré-processamento para aumentação contendo:
 - RandomZoom(0.1),
 - RandomContrast(0.2)
2. Convolucional 2D com 64 filtros 3×3 .
3. Batch Normalization
4. SeparableConv2D com 64 filtros 3×3 .
5. MaxPooling2D 3×3 e strides 2
6. Batch Normalization
7. SeparableConv2D com 256 filtros 3×3 .
8. MaxPooling2D 3×3 e strides 2
9. GlobalAveragePooling
10. Dropout de 0.2
11. Densa com ativação softmax

Inicialize as sementes do numpy com 1 e tensorflow com 2 e treine o modelo por 7 épocas com batch size 16, otimizador Adam e taxa de aprendizado 0.002.

Após o treinamento utilize a função predict para classificar imagens da posicao 10 a 14 no conjunto de testes ([10:15]). Quais as classes resultantes e quantas dessas estavam erradas?

- (a) 2, 5, 5, 3, 3, sendo 2 erradas
(b) 4, 5, 5, 3, 4, sendo 2 erradas
(c) 4, 5, 5, 3, 4 sendo 1 errada
(d) 4, 5, 5, 3, 4, nenhuma errada

```
In [1]: import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from numpy.random import seed
from tensorflow.random import set_seed

fashion_mnist = keras.datasets.fashion_mnist
(train_images, train_labels), (test_images, test_labels) = fashion_mnist.load_data()

train_images = train_images / 255.0
test_images = test_images / 255.0

train_labels = keras.utils.to_categorical(train_labels, 10)
test_labels = keras.utils.to_categorical(test_labels, 10)
```

In []:

In []:

Questão 5)

Carregue a base de dados MNIST do pacote Keras, e pre-processe conforme código abaixo.

Vamos utilizar o modelo treinado na questão anterior como forma de transferência de aprendizado. Se preciso reinicialize o modelo e treine-o novamente para garantir que apenas 7 épocas foram executadas. O modelo final deve ter acurácia de treinamento próxima a 0.89 (computada na base Fashion).

Agora, assuma que esse modelo já treinado está armazenado numa variável `model`. Então proceda da seguinte forma:

1. Obtendo a saída da penúltima camada (referente ao Dropout):

```
base_saida = model.layers[-2].output
```

1. Criando uma nova camada de saída que recebe como entrada a anterior

```
saida_nova = keras.layers.Dense(10, activation='softmax')(base_saida)
```

1. Criando um novo modelo tendo essa nova camada como saída

```
model2 = keras.models.Model(model.inputs, saida_nova)
```

Você pode usar o `summary` para conferir o modelo montado.

Agora inicialize as sementes do numpy para 1 e tensorflow para 2, compile e treine o novo modelo com função de custo entropia cruzada categórica, otimizador Adam com taxa de aprendizado 0.002, 16 exemplos no mini-batch e 3 épocas.

Avalie a acurácia no conjunto de testes. Em qual intervalo está a acurácia resultante, considerando arredondamento para 2 casas decimais?

- (a) [0.94,0.96]
- (b) [0.98,1.00]
- (c) [0.87,0.90]
- (d) [0.92,0.93]

```
In [5]: mnist = keras.datasets.mnist
        (train_images2, train_labels2), (test_images2, test_labels2) = mnist.load_data()
        train_images2 = train_images2 / 255.0
        test_images2 = test_images2 / 255.0
        train_labels2 = keras.utils.to_categorical(train_labels2, 10)
        test_labels2 = keras.utils.to_categorical(test_labels2, 10)
```

```
In [ ]:
```