

MBA em Ciência de Dados

Redes Neurais e Arquiteturas Profundas

Módulo II - Redes Neurais Convolucionais (CNNs)

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)

Considere duas camadas de redes neurais profundas que recebem uma entrada com um total de 2500 dimensões:

- A. Camada densa com 10 neurônios, cuja entrada é um vetor de 2500 dimensões;
- B. Camada convolucional com 100 neurônios (filtros) de tamanho 3×3 , cuja entrada é uma matriz de $50 \times 50 = 2500$ dimensões

Qual o total de parâmetros a serem aprendidos em cada camada?

- (a) A = 25000 parâmetros; B = 1010 parâmetros
 - (b) A = 2500 parâmetros; B = 900 parâmetros
 - (c) A = 25010 parâmetros; B = 1000 parâmetros
 - (d) A = 25010 parâmetros; B = 900 parâmetros
-

Questão 2)

Considere o conceito de "campo receptivo local" como uma região de certo tamanho dos dados de entrada que é processada de forma a gerar a saída. Considere ainda dois tipos de unidades de processamento de redes neurais profundas:

- A. Neurônio de camada densa (tipo Perceptron), que recebe por entrada um vetor com 3072 dimensões
- B. Neurônio de camada convolucional (filtro) de tamanho $K \times K$

x P, que recebe por entrada uma imagem com 32 x 32 x 3 dimensões.

Qual é o tamanho do campo receptivo para cada valor de saída computado por A e B?

- (a) $A = 1$; $B = 32 \times 32 \times 3$
- (b) $A = 3072$; $B = K \times K \times P$
- (c) $A = K \times K$; $B = K \times K \times P$
- (d) $A = 3072$; $B = 32 \times 32 \times 3$

Questão 3)

Utilizando a biblioteca Keras, projete uma rede neural para processar dados unidimensionais (dimensionalidade do vetor de entrada = 11000), e que contenha as seguintes camadas:

1. camada max pooling com tamanho de pool = 10
2. camada convolucional 1 com 100 filtros de tamanho 5, sem padding
3. camada global average pooling

Essa arquitetura poderia ser utilizada para receber por entrada segundos de áudio a 11kHz e aprender um espaço de características compacta com a camada Global Average Pooling.

Quais as dimensionalidades das saídas de cada camada?

- (a) 1=(1100,1); 2=(1096,100); 3=(100)
- (a) 1=(10010,1); 2=(1993,100); 3=(100,100)
- (c) 1=(10,1); 2=(1000,5); 3=(100)
- (d) 1=(1100,1); 2=(1096,5); 3=(100)

```
In [1]: import tensorflow as tf
        from tensorflow import keras
```

Questão 4)

Carregue a base de dados Fashion-MNIST conforme o código abaixo e exiba as 10 primeiras imagens dessa base de dados. Normalize os dados das imagens de forma a que os valores estejam entre 0 e 1, depois converta as classes para o tipo categórico utilizando o `tf.keras.utils.to_categorical`.

A seguir, crie uma CNN para classificar imagens dessa base de dados, contendo como camadas:

1. convolucional 1 com 32 filtros de tamanho 3×3 , com padding e stride 2 (nas

- duas direções)
2. convolucional 2 com 64 filtros de tamanho 1×3 , com padding e stride 1, 2
 3. convolucional 3 com 64 filtros de tamanho 3×1 , com padding e stride 2, 1
 4. convolucional 4 com 128 filtros de tamanho 3×3 , sem padding.
 5. global average pooling
 6. classificador softmax

Quais os tamanhos das saídas de cada camada?

- (a) 1=(14,14,32); 2=(14,7,64); 3=(7,7,64); 4=(5,5,128); 5=(128)
- (a) 1=(28,28,32); 2=(14,7,64); 3=(7,7,64); 4=(5,5,128); 5=(128); 6=(10)
- (c) 1=(28,28,32); 2=(14,7,64); 3=(7,14,64); 4=(5,5,128); 5=(128); 6=(10)
- (d) 1=(14,14,32); 2=(14,7,64); 3=(7,7,64); 4=(5,5,128); 5=(128); 6=(10)

```
In [2]: import numpy as np
import matplotlib.pyplot as plt

from tensorflow.keras.datasets import fashion_mnist
(x_train, y_train), (x_test, y_test) = fashion_mnist.load_data()
```

In []:

Questão 5)

Defina as sementes aleatórias do numpy para 1 e do tensorflow para 2. Depois, utilizando a arquitetura definida no exercício anterior, configure a rede para treinar com a configuração abaixo, salvando o histórico da perda e acurácia para as épocas.

- otimizador: SGD
- taxa de aprendizado: 0.09
- função de custo: categorical_crossentropy
- métrica: accuracy
- épocas: 15
- batchsize: 64

Após o processo de aprendizado, obtenha a acurácia calculada no conjunto de treinamento e no conjunto de testes utilizando a função `evaluate()` e escolha a opção cujo intervalo se enquadre nos valores obtidos.

- (a) Acurácia Treinamento = (0.92, 0.95), Acurácia Teste = (0.90, 0.93)
- (b) Acurácia Treinamento = (0.86, 0.90), Acurácia Teste = (0.80, 0.83)
- (c) Acurácia Treinamento = (0.96, 1.00), Acurácia Teste = (0.83, 0.89)
- (d) Acurácia Treinamento = (0.87, 0.92), Acurácia Teste = (0.84, 0.89)

```
In [3]: from numpy.random import seed  
        seed(1)  
        from tensorflow.random import set_seed  
        set_seed(2)
```

```
In [ ]:
```