

Bootcamp: Analista de Machine Learning

Trabalho Prático

Módulo 1: Fundamentos de Deep Learning

Objetivos de Ensino

Exercitar os seguintes conceitos trabalhados no Módulo:

1. Interpretar arquiteturas de rede a partir de desenho.
2. Criar redes utilizando o Keras.

Enunciado

Neste trabalho, você utilizará duas redes para resolver o problema de classificação Fashion MNIST, que pode ser importado através do módulo 'datasets' do Keras. As redes a serem utilizadas são:

- 1) VGG-16, que pode ser importada através do módulo 'keras.applications.vgg16'. A VGG-16 é uma rede já pronta (ou seja, o número de camadas e neurônios em cada camada, por exemplo, já estão pré-definidos) que resolve o problema de classificação de imagens. Ou seja, nós não iremos criar uma rede do zero: ao invés disso, usaremos uma rede que já foi experimentada antes em outros problemas, e inclusive já podem vir com um conjunto de pesos, obtidos dos seus treinamentos anteriores. Mais informações sobre a VGG-16 podem ser obtidas em <https://keras.io/api/applications/vgg/> e <https://arxiv.org/abs/1409.1556>.
- 2) A segunda deve seguir a seguinte arquitetura, conforme já vimos nas aulas gravadas:
 - a. Camada de entrada: um neurônio por pixel;
 - b. Camada oculta 1: número de neurônios igual ao número de pixels;
 - c. Camada oculta 2: 1024 neurônios;

- d. Camada oculta 3: 2048 neurônios;
- e. Camada oculta 4: 2048 neurônios;
- f. Camada de saída.

Observações:

- 1) A VGG-16 inicialmente foi treinada para imagens coloridas. Portanto, é necessário que ela receba imagens com canais de cor. Ou seja, cada pixel da imagem será representado por mais de um valor. Tipicamente, utilizamos o sistema de cor RGB, ou seja, cada pixel da imagem é composto por três valores. Podemos dizer, portanto, que uma imagem de 32 pixels de largura e 32 pixels de altura no sistema RGB tem uma dimensionalidade de 32x32x3.
- 2) Como as imagens do Fashion MNIST são em tons de cinza (ou seja, só possuem um canal de cor e, portanto, cada imagem tem o tamanho 28x28x1), é necessário redimensionar as imagens do dataset para 32x32x3, adicionando outros dois canais para usar o VGG-Net. Uma técnica possível é a seguinte:

```
from tensorflow.image import resize
# Adiciona eixo no último índice da lista
X_train_1 = np.expand_dims(X_train, axis=-1)
# Repete 3 vezes o último índice
X_train_1 = np.repeat(X_train_1, 3, axis=-1)
# Redimensiona as imagens para 32x32
X_train_resize = resize(X_train_1, [32, 32])
```

- 3) Para instanciar a VGG-Net, informe a dimensionalidade das amostras através do argumento `input_shape=(32, 32, 3)`. Além disso, preste atenção ao número de classes do problema.
- 4) A rede VGG-Net deverá receber amostras com o tamanho (32,32,3). A segunda rede deverá receber amostras (28,28,1). Para fazer isso, você pode importar o dataset duas vezes.
- 5) Ao instanciar a VGG-Net, informe o parâmetro `weights=None`. Esse parâmetro informa que os pesos da VGG-Net serão iniciados de forma aleatória e, portanto, não utilizaremos o treinamento já realizado na VGG-16.
- 6) Perceba que é muito mais fácil importar uma rede já existente do que criar a sua própria rede.



Respostas Finais

Os alunos deverão desenvolver a prática e, depois, responder às seguintes questões objetivas: