

ACH2016 Inteligência Artificial

Exercício Programa (EP)

Classificação

(Parte 1)

1. Conjunto de dados Fashion MNIST

Considere o conjunto de dados Fashion MNIST (semelhante ao MNIST) com 70.000 imagens de moda rotuladas de tamanho 28x28 disponível em:

<https://www.kaggle.com/datasets/zalando-research/fashionmnist>

O conjunto de treinamento consiste de 60.000 exemplos e um conjunto de teste de 10.000 exemplos.

2. Classificador utilizado:

- Convolutional neural network (CNN)

3. Calibração

Testar, comparar e analisar a influência de diferentes valores de parâmetros. Essa comparação deve ser feita usando, por exemplo, 5-fold cross-validation **nos dados de treinamento** e utilizando a métrica acurácia.

Responder as seguintes perguntas:

- A. Fixe o valor de filters, kernel_size, strides e padding para cada camada de convolução-pooling e uma quantidade fixa de neurônios na camada densa. Compare a acurácia obtida considerando diferentes quantidades de camadas de convolução-pooling, por exemplo 1, 2 ou 3. Qual é a melhor quantidade de camadas de convolução-pooling?
- B. Considerando a melhor quantidade de camadas de convolução-pooling encontrado no item anterior, qual é a melhor quantidade de feature maps (filters) em cada camada?
- C. Considerando a melhor quantidade de camadas de convolução-pooling e a melhor quantidade de feature maps em cada camada encontrados, qual é o melhor tamanho da camada densa?
- D. O que é dropout? Avalie diferentes porcentagens de dropout. Quanto dropout é melhor?
- E. O que é batch normalization? Aplique batch normalization e avalie como ela ajuda?
- F. O que é data augmentation?. Aplique data augmentation e avalie como ele ajuda?

Links interessantes sobre o assunto:

<https://www.kaggle.com/code/cdeotte/how-to-choose-cnn-architecture-mnist>

<https://machinelearningmastery.com/how-to-develop-a-convolutional-neural-network-from-scratch-for-mnist-handwritten-digit-classification/>

4. Avaliação no conjunto de dados de teste

Considerando a melhor arquitetura da CNN encontrada no item 3, calcule o valor da Acurácia no conjunto de dados de teste.

6. Algumas regras gerais

- Os alunos devem se organizar em grupos de 2 até 4 integrantes.
- Todas as entregas deverão ser feitas via Sistema e-Disciplinas, dentro dos deadlines estabelecidos.
- Podem ser usadas bibliotecas das linguagens de programação Python e R tais como: scikit-learn, Keras, H2O e mlr, entre outras.
- O código deve ser bem documentado (em DETALHES) de forma que seja simples identificar passagens do código que são importantes para a verificação do entendimento do grupo sobre a lógica dos procedimentos implementados.
- Os vídeos poderão conter entre 10 e 12 minutos de gravação e todos os integrantes do grupo devem participar da apresentação no vídeo. Uma janela com a imagem do aluno realizando a explicação deverá fazer parte dos vídeos.

7. Entregas:

- a) Código desenvolvido pelo grupo: o grupo deverá fazer o upload de todos os arquivos fonte.
- b) Vídeo de apresentação: o grupo deverá gravar um vídeo no qual apresenta sua codificação, em detalhes, e apresenta a execução de sua codificação sobre o conjunto de dados. Detalhes esperados no vídeo (listagem não exaustiva):
 - a. Apresentação dos integrantes do grupo (nomes completos).
 - b. Descrição do conjunto de dados.
 - c. Explicações sobre o código, ressaltando os detalhes e a lógica das implementações.
 - d. Demonstração da execução do código. Elaborar gráficos/tabelas.
 - e. Respostas às perguntas sobre calibração (item 3)
 - e. Incluir uma descrição curta das dificuldades encontradas.

IMPORTANTE: o tempo de vídeo é um critério de estabelecimento de avaliação igualitária entre os alunos, ou seja, vídeos “acelerados” implicarão em descontos na nota, podendo inclusive ser um fator determinante para a atribuição de nota 0 (zero) para esse quesito do trabalho. Uma possibilidade pode ser acelerar o vídeo enquanto mostra um treinamento, mas a gravação da voz deve estar em velocidade normal.

IMPORTANTE: você pode disponibilizar o seu vídeo em qualquer repositório público ou no sistema e-disciplinas. Se você disponibilizar em um repositório público você deverá depositar no sistema e-disciplinas, até a data limite de entrega do trabalho, um arquivo **.pdf** com o link e a duração exata do vídeo.

Outros links:

#02 Convolutional neural network: MNIST Dataset

<https://www.youtube.com/watch?v=9cPMFTwBdM4>

Deep Learning with Keras + TensorFlow - (Pt.1) Prepare the Fashion MNIST Dataset

<https://www.youtube.com/watch?v=N3oMKS1AfVI>

Deep Learning with Keras + TensorFlow - (Pt.2) Build the CNN

<https://www.youtube.com/watch?v=IOZGYzTn9Z8&t=11s>

<https://towardsdatascience.com/mnist-cnn-python-c61a5bce7a19>

<https://www.kaggle.com/code/pavansanagapati/a-simple-cnn-model-beginner-guide>

<https://towardsdatascience.com/a-simple-2d-cnn-for-mnist-digit-recognition-a998dbc1e79a>

<https://www.codingninjas.com/codestudio/library/applying-cnn-on-mnist-dataset>

<https://www.geeksforgeeks.org/applying-convolutional-neural-network-on-mnist-dataset/>

Este é um documento preliminar e pode sofrer alterações ao longo do tempo no sentido de torná-lo mais didático e mais detalhado.