ARTIFICIAL INTELLIGENCE (E016330A)
ALEKSANDRA PIZURICA
GHENT UNIVERSITY
2021/2022 AY

Assistants: Shaoguang Huang, Srđan Lazendić,
Nicolas Vercheval, Nina Žižakić
Professor: Aleksandra Pižurica

Practical Session 3: Deep Neural Networks

1 Introduction

In this practical session, we will use *Keras* [1] and *Google Colab* [2] to get familiar with different theoretical and practical aspects of deep neural networks. Convolution neural network (CNN) will be utilized for image classification in this practicum.

Google Colab is a free cloud service to run Python notebooks with GPU support, ideal for taking our first steps in deep learning.

In preparation for this practical session, take a look at *Google Colab* and *Keras*:

- https://colab.research.google.com/notebooks/welcome.ipynb
- https://keras.io/

As training a network can easily take minutes to complete, we advise you to take a look at other questions while training a model in the background.

2 Database

The assignment will be on the classification of fruits and vegetables, which is defined in collaboration with Robovision. The data is obtained from Horea Muresan's "Fruit recognition from images using deep learning" [3]. The "Fruits-360" contains 75,937 images of 120 fruits. For this practical session, we will work on a subset, already dividing the training, validation and test set. It will not be necessary to download the dataset locally.

- 175MB dataset

3 Practical tasks and questions

We will start from a provided notebook *ai_practicals_nn_2021.ipynb*.

- https://drive.google.com/file/d/1w8RaUeKdqx-j8teZU9hCJTJjCAnqVNKg/view

Copy the notebook to your own Google drive to freely make modifications. The notebook will act as a guidance to answer the different questions. You are free to make any adjustments to the script.

3.1 Loading data and preprocessing

Run the notebook up to and including section **Database**.

1. Take a look at the input data x and output data y.
   (a) Describe compactly the dataset (data type, class name, input image size, sizes of training, validation and test data, the number of samples per class, etc.)
   (b) Is there a class imbalance? Has the dataset been split evenly by class?
   (c) Plot one input image per class. According to your answer to (a), if we would add new images to the dataset, what preprocessing would be necessary (refer to the notebook)?
   (d) What should an ideal fruit classifier be invariant to? (translation? vertical flipping? horizontal flipping?). What geometric augmentation strategy can be considered?

3.2 Training

3.2.1 Apple classification

We'll start with a simple binary classification: Is the fruit in the input image an apple or not.

2. Train the provided network and study the generated logs using Tensorboard.
   (a) Keep training the model untill the training loss starts to converge.
   (b) Show the training and validation loss for each epoch (you may ignore the test set for now).
   (c) By looking at both loss curves, do we speak of overfitting, best-fit or underfitting? Explain.

Besides accuracy, other metrics like precision and recall, are often used to measure the classification performance.

3. Generate the prediction of the test set.
   (a) What does this prediction output represent?
   (b) How do you convert the output of the model to a binary prediction of apple or not-apple?
   (c) In a table, show the confusion matrix of the test set using the ground truth and prediction.
   (d) The F1-score is another score often used instead of accuracy. Provide brief definition and explain why in many cases the F1-score is a better measure of performance than the accuracy. What score (F1-score or accuracy) would you advise to use to assess our Apple classifier?

The previous assignments were done on a prefixed model, however to find the optimal network architecture for an application, numerous hyperparameters have to be correctly chosen:

- Number of layers
- Number of filters per layer

- Loss function
- Learning rate
- Batch size
- ...

To optimize the learning rate, we will train several models with different learning rates based on the given model *get_model_apple_simple*.

4. Run the cell that separately trains the network with different learning rates.
   (a) Explain what the learning rate is.
   (b) Based on the results, which learning rate would you propose to use and why?
   (c) What happens when you set the learning rate too low?
   (d) What happens when you set the learning rate too high?

To see the effect of batch size, we will train a model with varying batch sizes from very low to very high.

5. Modify the cell that trains the network with different batch sizes. For each batch size, train a new model until it converges.
   (a) Explain what the batch size is.
   (b) Based on the results, which batch size would you propose to use and why?
   (c) What happens to the running time and accuracy when you set the batch size too small (for example 1)?
   (d) What happens when you set the batch size too large (for example the whole data)?

Next, we will test the performance of different models with a varying amount of feature-maps. For this, take a look at the method **get_model_apple_simple**.

6. Based on the provided model, vary the amount of feature maps (filters) for the last convolutional layer from 1 to 64 and train separately the corresponding models. You can leave other layers of the models the same.
   (a) Which of your models has the best performance? Explain how you assessed the performance of the models.
   (b) For your best model, run <**model name**>**.summary()**. Explain how the number of parameters per layer are calculated.
   (c) In general, what are the disadvantages of a layer that is too narrow (less feature-maps) or too wide (more feature-maps). Use terms like underfitting and overfitting.

We will also analyse the influence of training data on the classification performance.

7. With the processed training set *x_train* (see Training cell), generate new training sets with different amounts of samples (e.g., 1%, 10%, 20%, 40%, 60%, 80%) by random selection (keep validation and test sets fixed), and train separately the network. Remember to update the corresponding labels as well. For each training set, make sure to train the network to be converged.

   (a) Plot the accuracies on test set with varying training data in a figure.

   (b) How does the accuracy evolve with varying size of the training set. Explain the reason.

   (c) What happens to the accuracy when using limited training data. How will you cope with the problem in this case (consider the solution such as improving network architecture, data augmentation, transfer learning or others).

   (d) According to your solution, implement your idea and report the results. Verify whether your solution works and explain the reason.

3.2.2 Fruit classification

Before now the network was trained to distinguish between apples and not-apples. We will now alter the model to distinguish between the different kinds of fruits and vegetables.

8. Take a look at the code where the previous model is build. Make now your own model that can classify multiple classes:

   (a) The amount of filters of the last layer of the model should be changed from 1 to the amount of classes there are.

   (b) Replace the activation function of this last layer from sigmoid to softmax. Explain the difference and explain in the case of multiple classes why we need to use the softmax function instead of the sigmoid function.

   (c) Replace the loss-function from mean-squared-error (mse) to categorical cross-entropy (not to be confused with binary cross-entropy). Just like accuracy vs F1-score, explain why crossentropy is preferred over mse.

9. Train your network untill it convergences or untill the model starts to overtrain.

   (a) Show the loss and accuracy with respect to number of epochs.

   (b) Using the test set, predict the classes of the test set samples and show the confusion matrix with all the classes. Indicate which fruits are difficult to distinguish by the model according to the confusion matrix.

3.3 Testing

By now you should have models to classify between apples and not-apples and to distinguish between different kind of fruits.

10. Test data preparation.

    (a) Collect your own fruit pictures online, containing at least 5 classes and no less than 4 images per class. Some form of preprocessing might be needed (see question 1). Describe the collected data set and show some representative examples for each class.

11. Test your apple classifier and fruit classifier on your own set of pictures.

    (a) Provide examples of a True Positive, True Negative, False Positive and False Negative.

    (b) Are our trained networks ready for real-life applications? If not, what modification would you suggest are needed to make the biggest improvements.

    (c) How do you deal with classifying inputs that do not belong to any of the predefined output classes?

4 Submission

The project is done in groups of maximum two students. Submit your report of the practice together with your notebook.

Important remarks:

- Write concise answers to the questions. Indicate clearly the number of each task/question when answering it. Subquestions may be combined if it improves readability.
- The overall score may be influenced by the report presentation too (pay attention to the clarity of answers and avoid sloppiness).
- The submitted file should be named after both authors and specify the project subject (e.g. Shaoguang Huang & Nina Žižakić: dnn_shuang-nzizakic.zip).
- It is the student's responsibility to check that the submitted documents are complete and not corrupted. Avoid any situation that prevents the correct evaluation of your report.
- Deadline: **December 22, 2021, 23:59 CET**

References

[1] F. Chollet *et al.*, "Keras." https://keras.io, 2021.

[2] Google, "Welcome to colaboratory!," https://colab.research.google.com/notebooks/intro.ipynb, 2021.

[3] H. Mureşan and M. Oltean, "Fruit recognition from images using deep learning," *Acta Universitatis Sapientiae, Informatica*, vol. 10, no. 1, pp. 26-42, 2018.