Image Classification with Convolutional Neural Networks (CNN)

1 Introduction

The goal of this project is to understand and implement Convolutional Neural Networks (CNN) for image classification using different architectures, particularly the VGG model. You will work with the CIFAR-10 dataset and optimize results using regularization and normalization techniques. You will also compare the performance of various optimizers.

2 Project Methodology

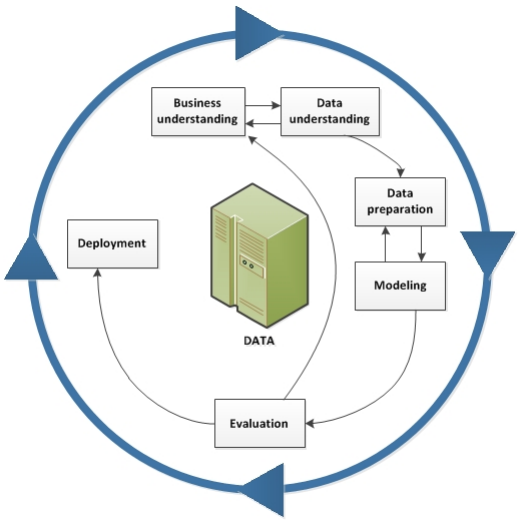

FIGURE 1 – CRISP-DM Phases

The CRISP [1] method (originally known as CRISP-DM) was originally developed by IBM in the 1960s for data-mining projects. In Data Science, it remains the most widely used methodology. It is composed of 6 steps from the understanding of the business problem to the deployment and production. This method is agile and iterative, i.e. each iteration brings additional business knowledge that allows to better tackle the next iteration. In what follows, we adopt this methodology. Thus, do not forget, at the end of each iteration, to refine and update previous steps if necessary (based on new obtained additional informations).

1. CRISP-DM : Cross-Industry Standard Process for Data Mining. Cette annexe est issue de https ://www.mygreatlearning.com/blog/why-using-crisp-dm-will-make-you-a-better-data-scientist

2.1 Data understanding and preparation

The goal of this phase is to gain a deep understanding of the CIFAR-10 dataset and its characteristics. You will analyze the dataset's structure, visualize the images, and perform basic data preprocessing steps, including one-hot encoding of labels.

Task 1 : Load the CIFAR-10 Dataset

1. Load the CIFAR-10 dataset using the tensorflow.keras.datasets library or from https://www.cs.toronto.edu/~kriz/cifar.html.
2. Display the shape of the training and test sets to understand the dimensions of the data.
3. How many training and test samples are there in the CIFAR-10 dataset ?
4. What is the shape of each image ? How many color channels are there ?

Task 2 : Visualize the Dataset

1. Plot a few random images from the dataset to understand the types of images it contains.
2. Display the corresponding labels for the images. What type of data do the labels represent ? Are they categorical or numerical ?
3. What do the images look like ? Are they easy or difficult to classify based on human vision ?

Task 3 : Normalize the Data

1. Normalize the pixel values of the images from the range [0, 255] to [0, 1].
2. Explain why normalization is important in Neural Networks models.
3. What are the benefits of normalizing the pixel values in an image dataset ?
4. How does normalization impact the performance of neural networks ?

Task 4 : Apply One-Hot Encoding to the Labels

1. Convert the labels into one-hot encoded vectors.
2. Why is it necessary to apply one-hot encoding to the class labels in a classification problem ?
3. What does one-hot encoding represent, and how does it help in training a neural network ?

Task 5 : Explore the Class Distribution

1. Plot a bar chart to show the distribution of classes.
2. Are the classes balanced in the CIFAR-10 dataset, or are some classes over/underrepresented ?
3. Why is it important to analyze the class distribution before training a model ?

2.2 Data Modeling

VGG Architectures

Many Convolutionnal Neural Network architectures could be used to classify images. In this project we propose to study Deep CNN Architectures such as VGG-alike architectures https://arxiv.org/pdf/1409.1556. A VGG-like Architecture is composed of P VGG blocks where :

— a VGG block is a sequence of : one Convolution layer, one Convolution layer, then one MaxPooling Layer.
— Convolution layer parameters : filters=32, kernel_size=(3, 3)
— MaxPooling layer uses a kernel of : (2, 2)
— We fix the ouptput layer to "nHidden = 128" neurons

We will consider the following architectures :

— LeNet5 (see Lecture)
— VGG1 : Uses one VGG block to learn features, then one hidden layer and one output layer to train model
— VGG2 : Uses two VGG blocks to learn features, then one hidden layer and one output layer to train model.
— VGG3 : Uses three VGG blocks to learn features, then one hidden layer and one output layer to train model.

Optimizers

Answer the following questions based on https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on

1. What is an optimizer in the context of deep learning ?
2. Explain the role of an optimizer during the training of a neural network. How does it affect the learning process ?
3. What are the most common types of optimizers ?
4. Compare how different optimizers (SGD, Adam, etc) update the model weights during training and how this affects learning efficiency and speed.
5. How do the different optimizers perform with the VGG architecture ?
6. Train your VGG model with different optimizers, analyze and compare their performance. Which optimizer performs best for the VGG architecture ?

Bonus - Regulization : Dropout and Batch Normalization Layers [2]

— Dropout : Explain the concept of dropout in neural networks. Why is it used, and how does it help prevent overfitting ? Describe how dropout is implemented during training and inference. What differences exist in the way dropout is applied in these two phases ?

— Batch Normalization : Define batch normalization. What are its main objectives, and how does it work ? Discuss the benefits of batch normalization. How does it impact the training speed and stability of a neural network ?

2.3 Implementaion and Performance Evaluation

Visualize the performance of different models and observe the evolution of training and validation metrics.

1. Plotting the curves : Implement a plot_history(history) function that generates two plots : The evolution of the loss function and the evolution of accuracy for both training and testing.
2. Analysis : Compare the performance of models CNN1, VGG1, VGG2, and VGG3 with regularization and different optimizers. Analyze how the different architectures affect the accuracy and loss on the training and test datasets.

3 Programming Language and librairies

1. Programming Language : Python
2. Librairies :
   — Data Manipulation : Pandas
   — Data Visualisation : Matplotlib & Seaborn
   — Data Modeling : Scikit-learn
   — TensorFlow and Keras https://www.tensorflow.org/guide/keras?hl=fr

4 Delivrables

— **A report** answering all the questions, along with relevant plots and code. It must contain an explanation of the ML methods not seen in class, the experimental results and their interpretations. You also present an overall conclusion (i.e. pre-processing, CNN architectures, Optimizers, Regularization ...) based on your results.

— **The source code** : Jupyter notebooks (you can use Google Colab Notebook - Google Colaboratory Notebooks) or A Zip file of python scripts.

5 Project groups and Deadlines

1. The project can be performed individually or in groups. The maximum number of students in each group will be three students.
2. Project submission deadline : **October 31, 2024 (11 :59 PM)** on **Moodle**.

2. https ://towardsdatascience.com/batch-normalization-and-dropout-in-neural-networks-explained-with-pytorch-47d7a8459bcd

6 Rating/Score

Your project will be graded by your teacher based on your provided work.

1. Understanding the CIFAR-10 Dataset : 15%
2. Data Preparation and Normalization : 20%
3. Implementation of CNN Models : 30%
4. Graphical Analysis and Results Discussion : 35%
5. Bonus - Model Regularization and Optimization : 20%