2AMM10 Assignment 1:
Omniglot character recognition

Deadline: 23 May 2022 23:59

1 Introduction

The goal of this assignment is to build a model that can recognize handwritten characters. We are given a set of images that we can to use to develop the model. The task is the following: Given a query image x_q and a support set of images $\{x_i\}, i \in [0..4]$. Find which images in the support set correspond to the character given in the query image. The train data is organized in sets of query and support set images. The characters that the model is trained on are different from the characters in the testing dataset and any future data that the model will be used on. In other words, the model needs to learn to recognize and match images that it has not seen before.

1.1 Data

The data consists of characters from alphabets from around the world. These images in the dataset are organized in tuples of a query image x_q, support set images $\{x_i\}, i \in [0..4]$ and an indexes of the support set image $\{j\}$ that correspond to the query image Figure 1. Such that $\{j\}, j \in \{0..4\}$ and $1 \leq |\{j\}| \geq 3$.

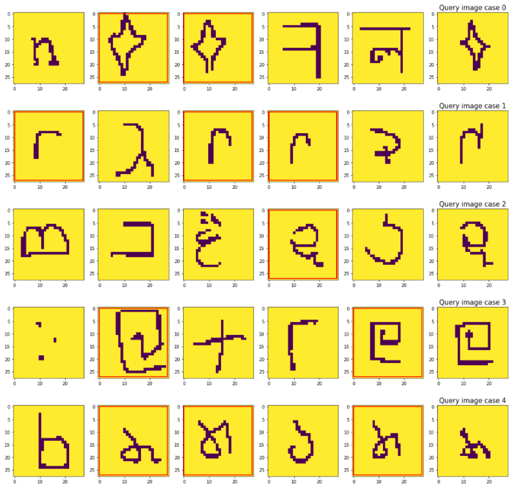

Figure 1: Example tuples of images from the Omniglot dataset. In each tuple, the first five columns are support images and the last one is the query image.

You are given 10000 tuples for training, and 1000 tuples for testing. None of the characters that appear in the training data images appear in the testing data.

1.2 Assignment Tasks

Build a solution that can match the character in the query image x_q with one more characters from the given support set $\{x_i\}$. The tuples in the train and test both contain between 1 and 3 images matching the query. The characters seen during training are different from the characters seen during testing.

2 Deliverables

You should hand in a report in PDF and a Jupyter Notebook solution.

1. **Jupyter Notebook.** Implement your model in PyTorch, and provide the solution in a Jupyter Notebook. Skeleton file that load the provided data are provided. You can extend these file to develop the solution.

2. **Report.** The report should contain the following sections:

   - **Problem formulation.** Develop a machine learning problem formulation for the task. What kind of machine learning task fits the problem description? What is a good measure of performance for your solution? Motivate your choices!

   - **Model formulation.** Develop a model that fits the problem formulation. Here, your model should be formulated on a high level using mathematical notation and accompanying explanations. Motivate your model by explaining how relevant characteristics and symmetries of the data are accommodated for by your design choices.

   - **Implementation.** Describe the procedure(s) you used to train the model. Your report should concisely describe all necessary information for someone to reproduce your implementation(s) and training procedure(s) from scratch. This includes, but is not limited to, the number and type of layers in your neural network(s), activation functions, loss function, optimization algorithm, etc. Describe how your model is used in the solution for the task and justify your decisions.

   - **Experiments and results.** Conduct suitable experiments and obtain results to evaluate the performance of your solution. Carefully describe the experiment setup and report and interpret your results (on the train, and test sets).

   - **Results and analysis.** Conduct suitable experiments and obtain results to evaluate the performance of your solution. Analyse the performance of your model and present your conclusions based on this analysis.

   - **Conclusion.** Summarize your approach and results, and concisely discuss limitations and some directions for future research.

   A skeleton file for the report is provided, although you are not obligated to use this format. *Do* make sure that all the (sub)sections in this skeleton are present in your submitted report.