Lumina

Getting Started with PrismRCL

PrismRCL is a Windows-based AI classification algorithm. Below are basic prepping, training, and inference examples to get you started with PrismRCL. Before you start training PrismRCL on your data, please make sure that your image or text data is organized and shaped correctly. Our algorithm supports two types of data: images in the PNG format and text data. Each text sample must be in its own file.

Required structure of your dataset:

```
parent_folder/
            class1_folder/img1.png, img2.png, etc.
            class2_folder/
            ...

parent_folder/
            class1_folder/file1.txt, file2.txt, etc.
            class2_folder/
            ...
```

Note: file names across all class folders must be unique!

1. Installation and Activation:

PrismRCL does not require installation. Just unzip the files included in the package into a single folder on your hard drive and you're ready to go! Please note that PrismRCL is intended as a developer tool that you can build applications and scripts on top of. Although a native Windows application, it is designed to be run from the command line as shown in the examples below. It does not have an interactive user interface. All logs and outputs are directed to text files on the file system and are fully configurable by the user. This makes it easy to run PrismRCL in batch mode and allows for automation and the ability to run machine learning tasks unattended.

PrismRCL comes with a free 30-day trial period. If you've purchased a subscription from us, please use your license key anytime during the trial to activate the software. If you're activating the software before running your first PrismRCL job, it is recommended that you run the included wizard to complete the activation (turboactivate.exe). PrismRCL will not load a GUI until after you've run at least one successful job from the command line.

2. Finding the optimal training parameters for your dataset:

Before you start training the algorithm on your data, you will need to set some parameters that will optimize the training and result in the best model for your data. PrismRCL offers a special feature that can find the optimal training parameters for your dataset automatically. We call it **auto-optimize**. No more trying random parameters endlessly. Let PrismRCL do the work for you! The three examples below demonstrate how you can use this time saving and optimization tool.

The command below will find and save the parameters to the log directory in a file of this format: **_optimize_summary_mm_dd_yy_hh_mm_ss.txt**.

C:\PrismRCL\PrismRCL.exe auto-optimize data=C:\data\train_data log=c:\log_files\

data: path to the training image dataset shaped as explained above.

log: path to where PrismRCL will save the training session log and results files.

The command below will create a model using parameters from auto-optimize (evaluation method, rclticks, boxdown, *and possibly imaginaryslice* are stored in model).

C:\PrismRCL\PrismRCL.exe auto-optimize data=C:\data\train_data savemodel=c:\models\best_model.classify log=c:\log_files\

data: path to the training image dataset shaped as explained above.

savemodel: path to where PrismRCL will save the trained model.

log: path to where PrismRCL will save the training session log and results files.

The command below will create a model using parameters from auto-optimize (evaluation method, rclticks, boxdown, *and possibly imaginaryslice* are stored in model) and evaluates model on test data, all in one pass.

C:\PrismRCL\PrismRCL.exe auto-optimize data=C:\data\train_data testdata=C:\data\test_data savemodel=c:\models\best_model.classify log=c:\log_files\

data: path to the training image dataset shaped as explained above.

testdata: path to the test dataset to be used for evaluating the model during training.

savemodel: path to where PrismRCL will save the trained model.

log: path to where PrismRCL will save the training session log and results files.

3. Create a new training session with test split:

The command below will start training session given a training dataset and a test split size.

C:\PrismRCL\PrismRCL.exe fractal rclticks=15 data=C:\PrismRCL\data\dataset\train-data testsize=0.1 savemodel=C:\PrismRCL\models\model_name.classify log=C:\PrismRCL\logfiles\job_folder stopwhendone

The evaluation method for this training session is **fractal** and **rclticks** is set (used for image data only).

data: path to the training image dataset shaped as explained above.

testsize: fraction of training dataset to be used for testing the model during training.

savemodel: path to where PrismRCL will save the trained model.

log: path to where PrismRCL will save the training session log and results files.

stopwhendone: instructs PrismRCL to shutdown automatically once the training session ends.

4. Create a new training session with test dataset:

The command below will start training session given a training dataset and a test dataset.

```
C:\PrismRCL\PrismRCL.exe chisquared rclticks=15 data=C:\PrismRCL\data\dataset\train-data testdata=
C:\PrismRCL\data\dataset\test-data savemodel=C:\PrismRCL\models\model_name.classify
log=C:\PrismRCL\logfiles\job_folder stopwhendone
```

This is very similar to the training command above. The only difference is that it uses a separate test dataset for evaluating the model, as opposed to using a split out of the training data. The evaluation method for this training session is **chisquared** and **rclticks** is set (used for image data only).

data: path to the training image dataset shaped as explained above.

testdata: path to the test dataset to be used for evaluating the model during training.

savemodel: path to where PrismRCL will save the trained model.

log: path to where PrismRCL will save the training session log and results files.

stopwhendone: instructs PrismRCL to shutdown automatically once the training session ends.

5. Create a new transfer learning session with test split:

The command below will start transfer learning training session given a training dataset and a test split size.

C:\PrismRCL\PrismRCL.exe naivebayes rclticks=15 transferlearn=c:\PrismRCL\models\existing_model_name.classify data=C:\PrismRCL\data\dataset\train-data testsize=0.1 savemodel=C:\PrismRCL\models\new_model_name.classify log=C:\PrismRCL\logfiles\job_folder stopwhendone

In this example, PrismRCL will use a pretrained model to train further on more training data. The evaluation method for this training session is **naivebayes** and **rclticks** is set (used for image data only).

transferlearn: path to the pretrained model to be used as the starting point for transfer learning session.

data: path to the training image dataset shaped as explained above.

testsize: fraction of training dataset to be used for testing the model during training.

savemodel: path to where PrismRCL will save the trained model.

log: path to where PrismRCL will save the training session log and results files.

stopwhendone: instructs PrismRCL to shutdown automatically once the training session ends.

6. Combine two or more models into one:

The command below will combine three models together and save them into one final model. Please note that when adding models together, make sure that the individual models were created using chunks from the same dataset and using the same training parameters. That's what makes it possible for the models to be combined.

```
C:\PrismRCL\PrismRCL.exe loadmodel=c:\PrismRCL\models\bettermodel.classify
addmodel=c:\PrismRCL\models\model998.classify;c:\PrismRCL\models\model_999.classify savemodel=c:
\PrismRCL\models\bestmodel.classify stopwhendone
```

PrismRCL can combine or "add up" models that were created in separate training sessions. To add models, all you need is to specify the locations on disk where your base model and additional models are. No parameters are needed.

loadmodel: path to the pre-trained model to be used as the starting point for transfer learning session.

addmodel: paths to the additional models to be combined with the based training image dataset shaped as explained above. Model paths must be separated by semicolons.

savemodel: path to where PrismRCL will save the trained model.

stopwhendone: instructs PrismRCL to shutdown automatically once the training session ends.

7. Create a new inference session:

The command below will start an inference session using a pre-trained model and an inference dataset (not used in training or testing the model).

C:\PrismRCL\PrismRCL.exe fractal rclticks=15 loadmodel= C:\PrismRCL\models\best_model.classify testdata=D:\deploy\data\dataset\inference-data inftotext=C:\PrismRCL\output\inference_output.txt log=D:\PrismRCL\logfiles\job_folder stopwhendone

Note: When running inference, please make sure that the evaluation method and rclticks value are the same as in the training session that generated your pre-trained model!

The evaluation method for this inference session is **fractal** and **rclticks** is set (used for image data only).

loadmodel: path to the pre-trained model to be used as the starting point for transfer learning session.

data: path to the training image dataset shaped as explained above.

testdata: path to the inference dataset that you'd like your pre-trained model to classify.

inftotext: path to the output text file where predicted classes for all inference images will be saved. The output file will contain two columns: name of image and predicted class.

log: path to where PrismRCL will save the inference session log and results files.

stopwhendone: instructs PrismRCL to shutdown automatically once the training session ends.