ICyTE - LPI - AI Toolbox

Documentation

1. Introduction

This is a toolbox for working on Deep Learning Neural Networks and general Artifitial Intelligence models (AI). It has been developed by Diego S. Comas, Agustín Amalfitano, Juan I. Iturriaga, Luciana Simón González, Gustavo J. Meschino, Franco R. Ércoli, and Virginia L. Ballarin at the Image Processing Lab, Institute of Scientific and Technological Research in Electronics (ICyTE), National University of Mar del Plata-CONICET, Argentina.

- ICyTE Website: https://icyte.conicet.gov.ar/.
- Image Processing Lab social media: https://www.instagram.com/lpi_icyte/.

LICENSE INFORMATION: If you use any part of this code for your experiments, please include the following references in your work:

- "Interpreting Deep Convolutional Features in Medical Images". Diego S. Comas, Agustín Amalfitano, Gustavo J. Meschino, Juan I. Iturriaga, Franco R. Ércoli, and Virginia L. Ballarin. International Workshop on Artificial Intelligence and Analytics, Eureka 2023. November 2023. Jaén, España. pp. 102.

2. Modules and functions

The current version of the toolbox includes the next modules with specific functions.

Module “dataframes”

This module contains functions for using Pandas Data Frames, which were adopted as a general approach for working with data and images.

Function name	Description
`balance_df`	This function performs dataset balancing from a DATAFRAME.

`hold_out_df`	This function makes a partition in a DATAFRAME to form TRAIN, VALIDATION, and TEST data considering HOLD-OUT.
`findin_df`	This function finds distinct values in a specific field of a DATAFRAME.
`filter_df`	This function filters a DATAFRAME.
`sampling_df`	This function samples a DATAFRAME.
`create_iter_df`	This function creates an iterator from a DATAFRAME.
`read_csv`	This function reads a "csv" file and returns a PANDAS DATAFRAME.
`walk_into_subdirs`	This function walks through the subdirectories looking for files to create a DATAFRAME.
`save_df`	This function saves a PANDAS DATAFRAME in a "csv" file.

Module “deep_nets”

This module contains functions for creating, freezing, training, and validation of Deep Neural Networks.

Function name	Description
`create_cnn`	This function generates a CNN using Keras.
`train_net`	This function trains a Neural Network using the method fit de Keras.
`eval_net`	This function evaluates a neural network against the test set and returns the validation measures.
`create_early_stopping`	This function generates a specific EarlyStopping function.
`get_conv_indexs`	This function goes through the layers and saves the index of the last conv layer of each block.
`compute_loss`	This function computes the "loss" as the mean of the activation of a specific filter in our target layer. To avoid border effects, we exclude border pixels.
`load_saved_model`	This function load a Neural Network from a H5

	file.
`evaluate_image`	This function evaluates a Neural Network for a specific image.
`set_net_outputs`	This function sets specific indexes to as outputs.
`implementing_hold_out`	This function implements hold-out on a CNN defined by means "dict_params".
`evaluate_class_models`	This function implements the evaluation of classification models from a DATAFRAME.
`reshape_input_model`	This function allows to change the input shape of a model.
`rgb_to_gray`	This function creates a layer for converting RGB to GRAYSCALE using [0.333, 0.333, 0.333] as weights.

Module “files”

This module contains functions developed for managing files and folders.

Function name	Description
`dict_conversion`	This function performs conversion of format in specified keys of a DICTIONARY.
`read_params_file`	This function read a TXT file containing parameters and return a DICTIONARY of parameters
`file_to_list`	This function reads a TXT file and returns the content in a LIST (an element per line).
`input_files`	This function returns a list of files with particular conditions.
`check_key`	This check if a KEY is defined in a DICTIONARY. It is useful for checking parameters read from a TXT file.
`images_paths_and_names`	This function returns the full path of all the files in "full_path" and their names.
`create_folder`	This function creates a folder.

Module “graphs”

This module contains functions for doing graphs.

Function name	Description
`show_history`	This function shows the history graphs from a training history.

Module “images”

This module contains functions for image processing.

Function name	Description
`augmentation_generator`	This function shows the history graphs from a training history.
`apply_aug_offline`	This function applies offline augmentation to all the images in a folder from an "augmentor" object generated from “augmentation_generator" or "imgaug" toolbox.

Module “net_visualization”

This module contains functions for net visualizations.

Function name	Description
`normalize_fmaps`	This function normalizes a feature map with values in [0, 255].
`save_fmaps_as_png`	This function saves single images for all the feature maps.
`save_fmaps_as_mosaicos`	This function saves a feature map as a mosaic.
`info_dic`	This function saves an info.mat file inside of each image folder.
`layers_dic`	This function saves a feature map as MATLAB files.
`gradient_ascent_step`	This function computes gradient ascent. It simply computes the gradients of the loss above with regard to the input image, and updates the update image so as to move it towards a state that will activate the target filter more strongly.
`initialize_image`	This function initialize an IMAGE with random

	values from a uniform distribution in [0, 1].
`maximize_filter`	This function obtains the image that maximizes a specific filter in a Neural model using gradient maximization.
`deprocess_image`	This function converts the resulting input image of "gradient_ascent_step" back to a displayable form, by normalizing it, center-cropping it, and restricting it to the [0, 255] range.
`feature_visualization_layer`	This function performs feature-visualization on a specific filter. The results are saved accordingly.
`grad_cam_explainer`	This function performs grad-cam based explainers from a model and a set of images.

Module “settings”

This module contains functions for hardware settings.

Function name	Description
`toolbox_version`	This function returns the version of the ICyTE - LPI - AI Toolbox informing the HASH of the repository.
`memory_limit`	This function limits the memory to be used by GPUs.

Module “texts”

This module contains functions for managing text files and text data.

Function name	Description
`save_list_to_text`	This function saves a LIST in a text file.
`str_into_list`	This function converts a STRING to a LIST.
`list_into_str`	This function convert a LIST to a STRING.