More layers

CONVOLUTIONAL LAYERS

layer_conv_1d() 1D, e.g. temporal convolution

layer_conv_2d_transpose() Transposed 2D (deconvolution)

layer_conv_2d() 2D, e.g. spatial convolution over images

layer_conv_3d_transpose() Transposed 3D (deconvolution)
layer_conv_3d() 3D, e.g. spatial convolution over volumes

layer_conv_lstm_2d() Convolutional LSTM

layer_separable_conv_2d() Depthwise separable 2D

layer_upsampling_1d()
layer_upsampling_2d()
layer_upsampling_3d() Upsampling layer

layer_zero_padding_1d()
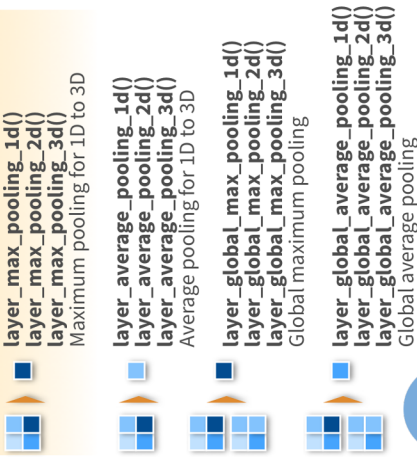**layer_zero_padding_2d()**
layer_zero_padding_3d() Zero-padding layer

layer_cropping_1d()
layer_cropping_2d()
layer_cropping_3d() Cropping layer

POOLING LAYERS

layer_max_pooling_1d()
layer_max_pooling_2d()
layer_max_pooling_3d() Maximum pooling for 1D to 3D

layer_average_pooling_1d()
layer_average_pooling_2d()
layer_average_pooling_3d() Average pooling for 1D to 3D

layer_global_max_pooling_1d()
layer_global_max_pooling_2d()
layer_global_max_pooling_3d() Global maximum pooling

layer_global_average_pooling_1d()
layer_global_average_pooling_2d()
layer_global_average_pooling_3d() Global average pooling

ACTIVATION LAYERS

layer_activation(object, activation) Apply an activation function to an output

layer_activation_leaky_relu() Leaky version of a rectified linear unit

layer_activation_parametric_relu() Parametric rectified linear unit

layer_activation_thresholded_relu() Thresholded rectified linear unit

layer_activation_elu() Exponential linear unit

DROPOUT LAYERS

layer_dropout() Applies dropout to the input

layer_spatial_dropout_1d()
layer_spatial_dropout_2d()
layer_spatial_dropout_3d() Spatial 1D to 3D version of dropout

RECURRENT LAYERS

layer_simple_rnn() Fully-connected RNN where the output is to be fed back to input

layer_gru() Gated recurrent unit - Cho et al

layer_cudnn_gru() Fast GRU implementation backed by CuDNN

layer_lstm() Long-Short Term Memory unit - Hochreiter 1997

layer_cudnn_lstm() Fast LSTM implementation backed by CuDNN

LOCALLY CONNECTED LAYERS

layer_locally_connected_1d()
layer_locally_connected_2d() Similar to convolution, but weights are not shared, i.e. different filters for each patch

Preprocessing

SEQUENCE PREPROCESSING

pad_sequences() Pads each sequence to the same length (length of the longest sequence)

skipgrams() Generates skipgram word pairs

make_sampling_table() Generates word rank-based probabilistic sampling table

TEXT PREPROCESSING

text_tokenizer() Text tokenization utility

fit_text_tokenizer() Update tokenizer internal vocabulary

save_text_tokenizer(); load_text_tokenizer() Save a text tokenizer to an external file

texts_to_sequences();
texts_to_sequences_generator() Transforms each text in texts to sequence of integers

texts_to_matrix(); sequences_to_matrix() Convert a list of sequences into a matrix

text_one_hot() One-hot encode text to word indices

text_hashing_trick() Converts a text to a sequence of indexes in a fixed-size hashing space

text_to_word_sequence() Convert text to a sequence of words (or tokens)

IMAGE PREPROCESSING

image_load() Loads an image into PIL format.

flow_images_from_data()
flow_images_from_directory() Generates batches of augmented/normalized data from images and labels, or a directory

image_data_generator() Generate minibatches of image data with real-time data augmentation.

fit_image_data_generator() Fit image data generator internal statistics to some sample data

generator_next() Retrieve the next item

image_to_array(); image_array_resize()
image_array_save() 3D array representation

Pre-trained models

Keras applications are deep learning models that are made available alongside pre-trained weights. These models can be used for prediction, feature extraction, and fine-tuning.

application_xception()
xception_preprocess_input() Xception v1 model

application_inception_v3()
inception_v3_preprocess_input() Inception v3 model, with weights pre-trained on ImageNet

application_inception_resnet_v2()
inception_resnet_v2_preprocess_input() Inception-ResNet v2 model, with weights trained on ImageNet

application_vgg16(); application_vgg19() VGG16 and VGG19 models

application_resnet50() ResNet50 model

application_mobilenet()
mobilenet_preprocess_input()
mobilenet_decode_predictions()
mobilenet_load_model_hdf5() MobileNet model architecture

IM·GENET
ImageNet is a large database of images with labels, extensively used for deep learning

imagenet_preprocess_input()
imagenet_decode_predictions() Preprocesses a tensor encoding a batch of images for ImageNet, and decodes predictions

Callbacks

A callback is a set of functions to be applied at given stages of the training procedure. You can use callbacks to get a view on internal states and statistics of the model during training.

callback_early_stopping() Stop training when a monitored quantity has stopped improving
callback_learning_rate_scheduler() Learning rate scheduler
callback_tensorboard() TensorBoard basic visualizations