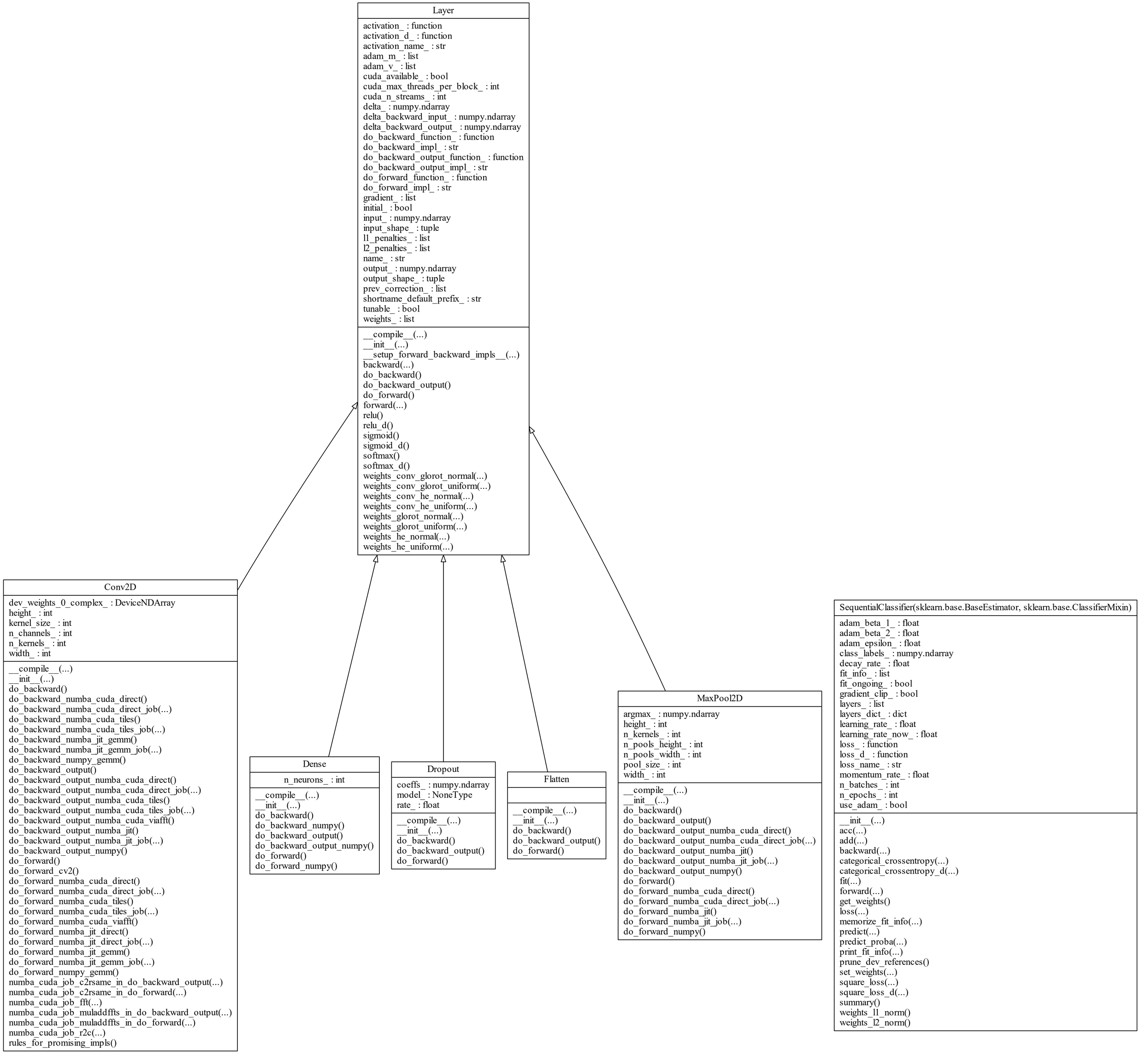
Layer
activation_ : function
activation_d_ : function
activation_name_ : str
adam_m_ : list
adam_v_ : list
cuda_available_ : bool
cuda_max_threads_per_block_ : int
cuda_n_streams_ : int
delta_ : numpy.ndarray
delta_backward_input_ : numpy.ndarray
delta_backward_output_ : numpy.ndarray
do_backward_function_ : function
do_backward_impl_ : str
do_backward_output_function_ : function
do_backward_output_impl_ : str
do_forward_function_ : function
do_forward_impl_ : str
gradient_ : list
initial_ : bool
input_ : numpy.ndarray
input_shape_ : tuple
l1_penalties_ : list
l2_penalties_ : list
name_ : str
output_ : numpy.ndarray
output_shape_ : tuple
prev_correction_ : list
shortname_default_prefix_ : str
tunable_ : bool
weights_ : list
__compile__(...)
__init__(...)
__setup_forward_backward_impls__(...)
backward(...)
do_backward()
do_backward_output()
do_forward()
forward(...)
relu()
relu_d()
sigmoid()
sigmoid_d()
softmax()
softmax_d()
weights_conv_glorot_normal(...)
weights_conv_glorot_uniform(...)
weights_conv_he_normal(...)
weights_conv_he_uniform(...)
weights_glorot_normal(...)
weights_glorot_uniform(...)
weights_he_normal(...)
weights_he_uniform(...)
Conv2D
dev_weights_0_complex_ : DeviceNDArray
height_ : int
kernel_size_ : int
n_channels_ : int
n_kernels_ : int
width_ : int
__compile__(...)
__init__(...)
do_backward()
do_backward_numba_cuda_direct()
do_backward_numba_cuda_direct_job(...)
do_backward_numba_cuda_tiles()
do_backward_numba_cuda_tiles_job(...)
do_backward_numba_jit_gemm()
do_backward_numba_jit_gemm_job(...)
do_backward_numpy_gemm()
do_backward_output()
do_backward_output_numba_cuda_direct()
do_backward_output_numba_cuda_direct_job(...)
do_backward_output_numba_cuda_tiles()
do_backward_output_numba_cuda_tiles_job(...)
do_backward_output_numba_cuda_viafft()
do_backward_output_numba_jit()
do_backward_output_numba_jit_job(...)
do_backward_output_numpy()
do_forward()
do_forward_cv2()
do_forward_numba_cuda_direct()
do_forward_numba_cuda_direct_job(...)
do_forward_numba_cuda_tiles()
do_forward_numba_cuda_tiles_job(...)
do_forward_numba_cuda_viafft()
do_forward_numba_jit_direct()
do_forward_numba_jit_direct_job(...)
do_forward_numba_jit_gemm()
do_forward_numba_jit_gemm_job(...)
do_forward_numpy_gemm()
numba_cuda_job_c2rsame_in_do_backward_output(...)
numba_cuda_job_c2rsame_in_do_forward(...)
numba_cuda_job_fft(...)
numba_cuda_job_muladdffts_in_do_backward_output(...)
numba_cuda_job_muladdffts_in_do_forward(...)
numba_cuda_job_r2c(...)
rules_for_promising_impls()
Dense
n_neurons_ : int
__compile__(...)
__init__(...)
do_backward()
do_backward_numpy()
do_backward_output()
do_backward_output_numpy()
do_forward()
do_forward_numpy()
Dropout
coeffs_ : numpy.ndarray
model_ : NoneType
rate_ : float
__compile__(...)
__init__(...)
do_backward()
do_backward_output()
do_forward()
Flatten
__compile__(...)
__init__(...)
do_backward()
do_backward_output()
do_forward()
MaxPool2D
argmax_ : numpy.ndarray
height_ : int
n_kernels_ : int
n_pools_height_ : int
n_pools_width_ : int
pool_size_ : int
width_ : int
__compile__(...)
__init__(...)
do_backward()
do_backward_output()
do_backward_output_numba_cuda_direct()
do_backward_output_numba_cuda_direct_job(...)
do_backward_output_numba_jit()
do_backward_output_numba_jit_job(...)
do_backward_output_numpy()
do_forward()
do_forward_numba_cuda_direct()
do_forward_numba_cuda_direct_job(...)
do_forward_numba_jit()
do_forward_numba_jit_job(...)
do_forward_numpy()
SequentialClassifier(sklearn.base.BaseEstimator, sklearn.base.ClassifierMixin)
adam_beta_1_ : float
adam_beta_2_ : float
adam_epsilon_ : float
class_labels_ : numpy.ndarray
decay_rate_ : float
fit_info_ : list
fit_ongoing_ : bool
gradient_clip_ : bool
layers_ : list
layers_dict_ : dict
learning_rate_ : float
learning_rate_now_ : float
loss_ : function
loss_d_ : function
loss_name_ : str
momentum_rate_ : float
n_batches_ : int
n_epochs_ : int
use_adam_ : bool
__init__(...)
acc(...)
add(...)
backward(...)
categorical_crossentropy(...)
categorical_crossentropy_d(...)
fit(...)
forward(...)
get_weights()
loss(...)
memorize_fit_info(...)
predict(...)
predict_proba(...)
print_fit_info(...)
prune_dev_references()
set_weights(...)
square_loss(...)
square_loss_d(...)
summary()
weights_l1_norm()
weights_l2_norm()