[[fit]](https://keras.io/models/model/#fit)

* the training function for keras model for numpy training data
* It is used for training a model without data augmentation

<model>.fit(x=None,

y=None,

batch\_size=None,

epochs=1,

verbose=1,

callbacks=None,

validation\_split=0.0,

validation\_data=None,

shuffle=True,

class\_weight=None,

sample\_weight=None,

initial\_epoch=0,

steps\_per\_epoch=None,

validation\_steps=None,

validation\_freq=1)

* **x**: Numpy array of training data (if the model has a single input), or list of Numpy arrays (if the model has multiple inputs). If input layers in the model are named, you can also pass a dictionary mapping input names to Numpy arrays. x can be None (default) if feeding from framework-native tensors (e.g. TensorFlow data tensors).
* **y**: Numpy array of target (label) data (if the model has a single output), or list of Numpy arrays (if the model has multiple outputs). If output layers in the model are named, you can also pass a dictionary mapping output names to Numpy arrays. y can be None (default) if feeding from framework-native tensors (e.g. TensorFlow data tensors).
* **batch\_size**: Integer or None. Number of samples per gradient update. If unspecified, batch\_size will default to 32.
* **epochs**: Integer. Number of epochs to train the model. An epoch is an iteration over the entire x and y data provided. Note that in conjunction with initial\_epoch, epochs is to be understood as "final epoch". The model is not trained for a number of iterations given by epochs, but merely until the epoch of index epochs is reached.
* **verbose**: Integer. 0, 1, or 2. Verbosity mode. 0 = silent, 1 = progress bar, 2 = one line per epoch.
* **callbacks**: List of keras.callbacks.Callback instances. List of callbacks to apply during training and validation (if ). See callbacks.
* **validation\_split**: Float between 0 and 1. Fraction of the training data to be used as validation data. The model will set apart this fraction of the training data, will not train on it, and will evaluate the loss and any model metrics on this data at the end of each epoch. The validation data is selected from the last samples in the x and y data provided, before shuffling.
* **validation\_data**: tuple (x\_val, y\_val) or tuple (x\_val, y\_val, val\_sample\_weights) on which to evaluate the loss and any model metrics at the end of each epoch. The model will not be trained on this data. validation\_data will override validation\_split.
* **shuffle**: Boolean (whether to shuffle the training data before each epoch) or str (for 'batch'). 'batch' is a special option for dealing with the limitations of HDF5 data; it shuffles in batch-sized chunks. Has no effect when steps\_per\_epoch is not None.
* **class\_weight**: Optional dictionary mapping class indices (integers) to a weight (float) value, used for weighting the loss function (during training only). This can be useful to tell the model to "pay more attention" to samples from an under-represented class.
* **sample\_weight**: Optional Numpy array of weights for the training samples, used for weighting the loss function (during training only). You can either pass a flat (1D) Numpy array with the same length as the input samples (1:1 mapping between weights and samples), or in the case of temporal data, you can pass a 2D array with shape (samples, sequence\_length), to apply a different weight to every timestep of every sample. In this case you should make sure to specify sample\_weight\_mode="temporal" in compile().
* **initial\_epoch**: Integer. Epoch at which to start training (useful for resuming a previous training run).
* **steps\_per\_epoch**: Integer or None. Total number of steps (batches of samples) before declaring one epoch finished and starting the next epoch. When training with input tensors such as TensorFlow data tensors, the default None is equal to the number of samples in your dataset divided by the batch size, or 1 if that cannot be determined.
* **validation\_steps**: Only relevant if steps\_per\_epoch is specified. Total number of steps (batches of samples) to validate before stopping.
* **validation\_freq**: Only relevant if validation data is provided. Integer or list/tuple/set. If an integer, specifies how many training epochs to run before a new validation run is performed, e.g. validation\_freq=2 runs validation every 2 epochs. If a list, tuple, or set, specifies the epochs on which to run validation, e.g. validation\_freq=[1, 2, 10] runs validation at the end of the 1st, 2nd, and 10th epochs.

[fit\_generator]

* train a model with data generator
* it is usually used in the case of data augmentation.

<model>.fit\_generator(generator,

steps\_per\_epoch=None,

epochs=1,

verbose=1,

callbacks=None,

validation\_data=None,

validation\_steps=None,

validation\_freq=1,

class\_weight=None,

max\_queue\_size=10,

workers=1,

use\_multiprocessing=False,

shuffle=True,

initial\_epoch=0)

* **generator**: A generator or an instance of Sequence (keras.utils.Sequence) object in order to avoid duplicate data when using multiprocessing. The output of the generator must be either
  + a tuple (inputs, targets)
  + a tuple (inputs, targets, sample\_weights).
  + This tuple (a single output of the generator) makes a single batch. Therefore, all arrays in this tuple must have the same length (equal to the size of this batch). Different batches may have different sizes. For example, the last batch of the epoch is commonly smaller than the others, if the size of the dataset is not divisible by the batch size. The generator is expected to loop over its data indefinitely. An epoch finishes when steps\_per\_epoch batches have been seen by the model.
* **steps\_per\_epoch**: Integer. Total number of steps (batches of samples) to yield from generator before declaring one epoch finished and starting the next epoch. It should typically be equal to ceil(num\_samples / batch\_size) Optional for Sequence: if unspecified, will use the len(generator) as a number of steps.
* **epochs**: Integer. Number of epochs to train the model. An epoch is an iteration over the entire data provided, as defined by steps\_per\_epoch. Note that in conjunction with initial\_epoch, epochs is to be understood as "final epoch". The model is not trained for a number of iterations given by epochs, but merely until the epoch of index epochs is reached.
* **verbose**: Integer. 0, 1, or 2. Verbosity mode. 0 = silent, 1 = progress bar, 2 = one line per epoch.
* **callbacks**: List of keras.callbacks.Callback instances. List of callbacks to apply during training. See callbacks.
* **validation\_data**: This can be either
  + a generator or a Sequence object for the validation data
  + tuple (x\_val, y\_val)
  + tuple (x\_val, y\_val, val\_sample\_weights)
  + on which to evaluate the loss and any model metrics at the end of each epoch. The model will not be trained on this data.
* **validation\_steps**: Only relevant if validation\_data is a generator. Total number of steps (batches of samples) to yield from validation\_data generator before stopping at the end of every epoch. It should typically be equal to the number of samples of your validation dataset divided by the batch size. Optional for Sequence: if unspecified, will use the len(validation\_data) as a number of steps.
* **validation\_freq**: Only relevant if validation data is provided. Integer or collections.Container instance (e.g. list, tuple, etc.). If an integer, specifies how many training epochs to run before a new validation run is performed, e.g. validation\_freq=2 runs validation every 2 epochs. If a Container, specifies the epochs on which to run validation, e.g. validation\_freq=[1, 2, 10] runs validation at the end of the 1st, 2nd, and 10th epochs.
* **class\_weight**: Optional dictionary mapping class indices (integers) to a weight (float) value, used for weighting the loss function (during training only). This can be useful to tell the model to "pay more attention" to samples from an under-represented class.
* **max\_queue\_size**: Integer. Maximum size for the generator queue. If unspecified, max\_queue\_size will default to 10.
* **workers**: Integer. Maximum number of processes to spin up when using process-based threading. If unspecified, workers will default to 1. If 0, will execute the generator on the main thread.
* **use\_multiprocessing**: Boolean. If True, use process-based threading. If unspecified, use\_multiprocessing will default to False. Note that because this implementation relies on multiprocessing, you should not pass non-picklable arguments to the generator as they can't be passed easily to children processes.
* **shuffle**: Boolean. Whether to shuffle the order of the batches at the beginning of each epoch. Only used with instances of Sequence (keras.utils.Sequence). Has no effect when steps\_per\_epoch is not None.
* **initial\_epoch**: Integer. Epoch at which to start training (useful for resuming a previous training run).