


<div><div><div><div><div><div></div><div>Keras</div></div></div><div><div><div>Star</div><div>56,009</div></div></div></div></div></div>
About Keras
Getting started
Developer guides
Keras API reference
Models API
Layers API
Callbacks API
Optimizers
Metrics
Losses
Data loading
Built-in small datasets
Keras Applications
Mixed precision
Utilities
KerasTuner
KerasCV
KerasNLP
Code examples
Why choose Keras?
Community & governance
Contributing to Keras
KerasTuner
KerasCV
KerasNLP

Search Keras documentation...

» [Keras API reference](#) / [Callbacks API](#) / EarlyStopping

EarlyStopping

EarlyStopping class

[\[source\]](#)

```
tf.keras.callbacks.EarlyStopping(  
    monitor="val_loss",  
    min_delta=0,  
    patience=0,  
    verbose=0,  
    mode="auto",  
    baseline=None,  
    restore_best_weights=False,  
)
```

Stop training when a monitored metric has stopped improving.

Assuming the goal of a training is to minimize the loss. With this, the metric to be monitored would be 'loss', and mode would be 'min'. A `model.fit()` training loop will check at end of every epoch whether the loss is no longer decreasing, considering the `min_delta` and `patience` if applicable. Once it's found no longer decreasing, `model.stop_training` is marked True and the training terminates.

The quantity to be monitored needs to be available in `logs` dict. To make it so, pass the loss or metrics at `model.compile()`.

Arguments

- **monitor**: Quantity to be monitored.
- **min_delta**: Minimum change in the monitored quantity to qualify as an improvement, i.e. an absolute change of less than min_delta, will count as no improvement.
- **patience**: Number of epochs with no improvement after which training will be stopped.
- **verbose**: Verbosity mode, 0 or 1. Mode 0 is silent, and mode 1 displays messages when the callback takes an action.
- **mode**: One of {"auto", "min", "max"}. In `min` mode, training will stop when the quantity monitored has stopped decreasing; in `"max"` mode it will stop when the quantity monitored has stopped increasing; in `"auto"` mode, the direction is automatically inferred from the name of the monitored quantity.
- **baseline**: Baseline value for the monitored quantity. Training will stop if the model doesn't show improvement over the baseline.
- **restore_best_weights**: Whether to restore model weights from the epoch with the best value of the monitored quantity. If False, the model weights obtained at the last step of training are used. An epoch will be restored regardless of the performance relative to the `baseline`. If no epoch improves on `baseline`, training will run for `patience` epochs and restore weights from the best epoch in that set.

Example

```
>>> callback = tf.keras.callbacks.EarlyStopping(monitor='loss', patience=3)  
>>> # This callback will stop the training when there is no improvement in  
>>> # the loss for three consecutive epochs.  
>>> model = tf.keras.models.Sequential([tf.keras.layers.Dense(10)])  
>>> model.compile(tf.keras.optimizers.SGD(), loss='mse')  
>>> history = model.fit(np.arange(100).reshape(5, 20), np.zeros(5),  
...                     epochs=10, batch_size=1, callbacks=[callback],  
...                     verbose=0)  
>>> len(history.history['loss']) # Only 4 epochs are run.  
4
```