

https://www.tensorflow.org/guide/keras/custom_callback

Early stopping at minimum loss

First example showcases the creation of a Callback that stops the Keras training when the minimum of loss has been reached by mutating the attribute `model.stop_training` (boolean). Optionally, the user can provide an argument `patience` to specify how many epochs the training should wait before it eventually stops.

`tf.keras.callbacks.EarlyStopping` provides a more complete and general implementation.

```
import numpy as np

class EarlyStoppingAtMinLoss(tf.keras.callbacks.Callback):
    """Stop training when the loss is at its min, i.e. the loss stops
    decreasing.

    Arguments:
        patience: Number of epochs to wait after min has been hit. After this
        number of no improvement, training stops.
    """

    def __init__(self, patience=0):
        super(EarlyStoppingAtMinLoss, self).__init__()

        self.patience = patience

        # best_weights to store the weights at which the minimum loss occurs.
        self.best_weights = None

    def on_train_begin(self, logs=None):
        # The number of epoch it has waited when loss is no longer minimum.
        self.wait = 0
        # The epoch the training stops at.
        self.stopped_epoch = 0
        # Initialize the best as infinity.
        self.best = np.Inf

    def on_epoch_end(self, epoch, logs=None):
        current = logs.get('loss')
        if np.less(current, self.best):
            self.best = current
            self.wait = 0
            # Record the best weights if current results is better (less).
            self.best_weights = self.model.get_weights()
        else:
            self.wait += 1
            if self.wait >= self.patience:
                self.stopped_epoch = epoch
                self.model.stop_training = True
                print('Restoring model weights from the end of the best epoch.')
                self.model.set_weights(self.best_weights)

    def on_train_end(self, logs=None):
        if self.stopped_epoch > 0:
            print('Epoch %05d: early stopping' % (self.stopped_epoch + 1))
```