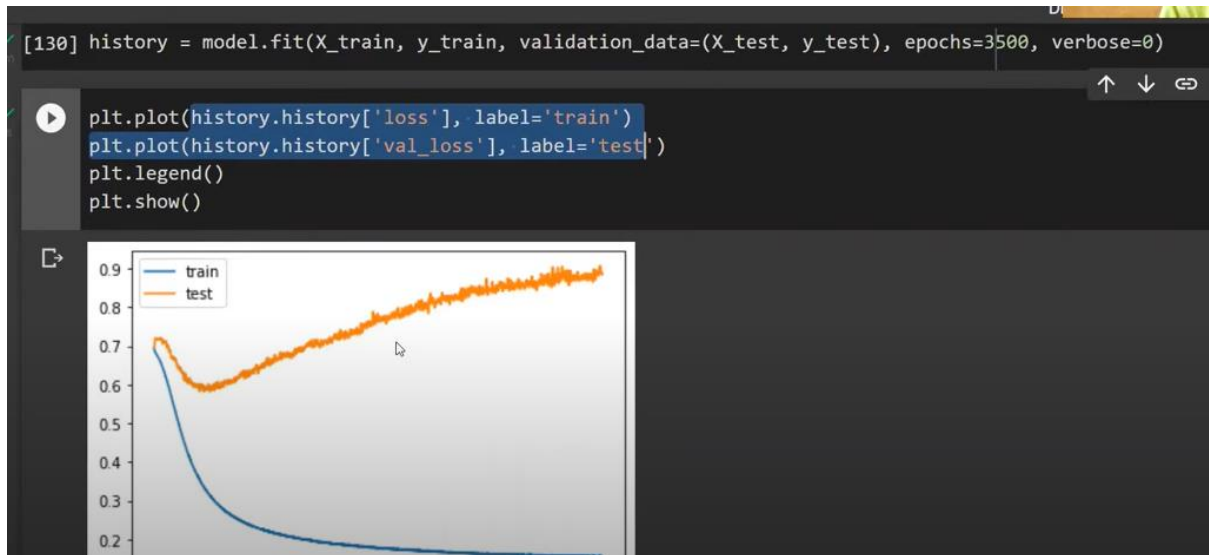


Early Stopping in Neural Network

While training Neural Network, Kitna Epochs dena hai vo param set krna hota hai,
Ese Randomly High Epochs times direct run karane se Overfitting hota hai.



As we can see Gap badhte ja raha.

Overfitting horaha hai

Early Stopping concept jo hai vo stop kr deta hai jaha se ushe lagta ki Overfit hoskta karke,

Keras mai Callback hota uskeliye

```
callback = EarlyStopping(  
    monitor="val_loss",  
    min_delta=0.00001,  
    patience=20,  
    verbose=1,  
    mode="auto",  
    baseline=None,  
    restore_best_weights=False  
)
```

```
min_delta=0.00001,
patience=20,
verbose=1,
mode="auto",
baseline=None,
restore_best_weights=False
)

history = model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=3500, callbacks=callback)
Epoch 293/3500
```

Ese fir 3k epochs ke jageh bss 300 Epochs mai kaam hogya above image.

Parameters of Early Stopping

EarlyStopping

EarlyStopping class

[source]

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

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. Defaults to `"val_loss"`.
- **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. Defaults to `0`.
- **patience:** Number of epochs with no improvement after which training will be stopped. Defaults to `0`.

If Patience = 5 set hai

Tou 5 epochs ka set dekhega kindoff ki if 5 Epochs lagatar mai koi Improvement nahi dikha tou fir vahipr Stop hojayega

- **verbose:** Verbosity mode, 0 or 1. Mode 0 is silent, and mode 1 displays messages when the callback takes an action. Defaults to `0`.

verbose = 1

- **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. Defaults to `"auto"`.

Auto Best hota hai

- **baseline:** Baseline value for the monitored quantity. If not `None`, training will stop if the model doesn't show improvement over the baseline. Defaults to `None`.

Generally nahi deta koi

- **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. Defaults to `False`.
- **start_from_epoch:** Number of epochs to wait before starting to monitor improvement. This allows for a warm-up period in which no improvement is expected and thus training will not be stopped. Defaults to `0`.