

- 이 문제를 해결하기 위해 사용하는 기법이 **조기 종료(early stopping)**로 **'이전 epoch 때와 비교해서 오차가 증가했다면 학습을 중단한다'**라는 방법이다.
- 조기 종료를 구현할 때, 주의할 사항은 바로 직전의 epoch 오차를 비교하기만해서는 안된다. 그림 14.1의 그래프가 보여주는 것처럼 오차값은 각 epoch마다 올라가기도 하고 내려가기도 하며, 드롭아웃을 적용한 경우에는 아직 학습이 되지 않은 뉴런이 존재할 가능성이 있기 때문에 오차값이 상하로 움직이게 된다.



따라서 **'어떤 일정한 epoch 수를 거듭하면서 계속해서 오차가 증가하면 학습을 중단한다'**는 방식으로 구현해야 한다.

- 위 코드에서 **patience**는 **'오차를 보기 위해 과거 몇 epoch까지 거슬러 올라갈 것인가'**를 설정하는 값이다.

```
early_stopping = EarlyStopping(patience=10, verbose=1)
```

```
EarlyStopping(monitor='val_loss', min_delta=0.0001, patience=2, verbose=0, mode='auto')
```

and here are the results:

```
val_loss: 0.6811
val_loss: 0.6941
val_loss: 0.6532
val_loss: 0.6546
val_loss: 0.6534
val_loss: 0.6489
val_loss: 0.6240
val_loss: 0.6285
val_loss: 0.6144
val_loss: 0.5921
val_loss: 0.5731
val_loss: 0.5956
val_loss: 0.5753
val_loss: 0.5977
```

There are three consecutively worse runs by loss, let's look at the numbers:

```
val_loss: 0.5921 < current best
val_loss: 0.5731 < current best
val_loss: 0.5956 < patience 1
val_loss: 0.5753 < patience 2
val_loss: 0.5977 < patience >2, stopping the training
```

People typically define a patience, i.e. the number of epochs to wait before early stop if no progress on the validation set. The patience is often set somewhere between 10 and 100 (10 or 20 is more common), but it really depends on your dataset and network.

Example with patience = 10: