

Explanation of Training Parameters in `model.fit()`

1. Epochs

An epoch represents one complete pass through the entire training dataset.

- Why use multiple epochs?

Neural networks often require multiple passes through the data to adjust weights sufficiently and learn patterns.

- Key Points:

- Too few epochs may result in underfitting (model doesn't learn enough).

- Too many epochs can lead to overfitting (model memorizes training data).

- Typical Value: Start with 10-50 epochs and adjust based on validation metrics.

2. Batch Size

The batch size defines the number of samples the model processes before updating its weights.

- Why not process all data at once?

- Memory constraints: Large datasets may not fit into memory.

- Efficiency: Processing smaller batches is faster and allows for more frequent updates.

- Common Values: Powers of 2 (16, 32, 64, 128).

- Smaller batch sizes (e.g., 32): More stable but slower training.

- Larger batch sizes (e.g., 128): Faster but may require more tuning to generalize.

3. Validation Split

The validation split specifies the fraction of training data set aside for validation.

- Why use validation data?

Validation data monitors the model's performance on unseen data during training, helping to:

- Detect overfitting or underfitting.
- Guide decisions like stopping training early.
- Typical Value: 10-20% of the training data.

4. Combining These Parameters

- Epochs: Define how many complete passes through the dataset the model makes.
- Batch Size: Controls the granularity of weight updates (how many samples are processed at once).
- Validation Split: Monitors performance on unseen data to guide training decisions.

These parameters together allow for controlled and effective training of a neural network.

5. Code Example

```
history = model.fit(
    X_train, y_train,
    epochs=20,      # Train for 20 complete passes through the data
    batch_size=32,  # Process 32 samples at a time
    validation_split=0.2 # Use 20% of training data for validation
)
```

What happens here:

- The dataset is split into 80% training and 20% validation data.
- For each epoch:
 - The training data is split into batches of 32 samples.
 - The model processes each batch, calculates the loss, and updates weights.
- At the end of each epoch, the model's performance is evaluated on the validation data.
- This process is repeated for 20 epochs.