**1. Recap: Training and Validation**

* Previously, we introduced **training** and **validation datasets**.
* Training → teaches the model.
* Validation → checks model performance during training to detect overfitting.

**2. Purpose of the Test Dataset**

* **Goal:** Measure the **predictive power** of the final model on **new, unseen data**.
* Think of it as applying the model in **real life**.
* Test dataset **cannot be used for training or validation**. It’s only for final evaluation.

**3. Workflow Summary**

1. **Get a dataset**
2. **Split the dataset into three parts:**
   * **Training:** largest portion (e.g., 70–80%)
   * **Validation:** smaller portion (e.g., 10–20%)
   * **Test:** remaining portion (e.g., 10%)

No strict rule; these ratios are commonly used.

1. **Train the model** using only the training dataset.
2. **Validate the model** every epoch (or periodically) using the validation dataset:
   * Check **training loss** and **validation loss**
   * If validation loss increases → overfitting → stop training
3. **Test the model** using the test dataset:
   * The accuracy here is the **realistic performance** you would expect in deployment.

**4. Key Notes**

* **Training dataset:** Teaches the model (weights updated here).
* **Validation dataset:** Checks overfitting and guides early stopping (weights not updated).
* **Test dataset:** Measures final performance on unseen data.

**5. Important Concepts**

* Validate **every epoch** → after each pass through the training data.
* Training loss and validation loss should **decrease together**.
* If validation loss increases → stop → model is overfitting.
* Test dataset gives the **true measure of model accuracy** in real life.

✅ **Takeaway:**

* Splitting data into **training / validation / test** is essential.
* **Training → learning**, **Validation → monitoring**, **Test → final evaluation**.
* The test dataset is the **last step**, representing the model’s real-world performance.