**Dropout Rate in Neural Networks**

The **dropout rate** is a regularization technique used in neural networks to **prevent overfitting** by randomly "dropping" (setting to zero) a proportion of neurons during training.

**How It Works**

* During each training iteration, a fraction of neurons is **randomly deactivated** (along with their connections).
* This forces the network to learn **redundant representations**, improving generalization.
* Dropout is **only applied during training**; during inference (prediction), all neurons are active, and their outputs are scaled accordingly.

**Dropout Rate Meaning**

The **dropout rate** is the percentage of neurons that are dropped in each training step.

* **Example:** A dropout rate of **0.2 (20%)** means that **20% of the neurons** in a given layer are turned off during each training iteration.

**Example in Code (Keras)**

from tensorflow.keras.layers import Dense, Dropout

from tensorflow.keras.models import Sequential

model = Sequential([

Dense(64, activation='relu', input\_shape=(100,)),

Dropout(0.2), # 20% of neurons are randomly dropped

Dense(32, activation='relu'),

Dropout(0.3), # 30% dropout in this layer

Dense(10, activation='softmax')

])

Here, dropout is applied after the first and second dense layers to **reduce overfitting**.

**Choosing a Dropout Rate**

* **Common values:** 0.1 to 0.5
* **Too low (e.g., 0.1):** May not prevent overfitting effectively.
* **Too high (e.g., 0.5+):** Can under-train the model by removing too many neurons.

**Key Takeaways**

✅ Helps prevent overfitting  
✅ Encourages robustness in the network  
✅ Typically used in hidden layers  
✅ Applied **only during training**, not during inference

Let me know if you need further clarification! 🚀