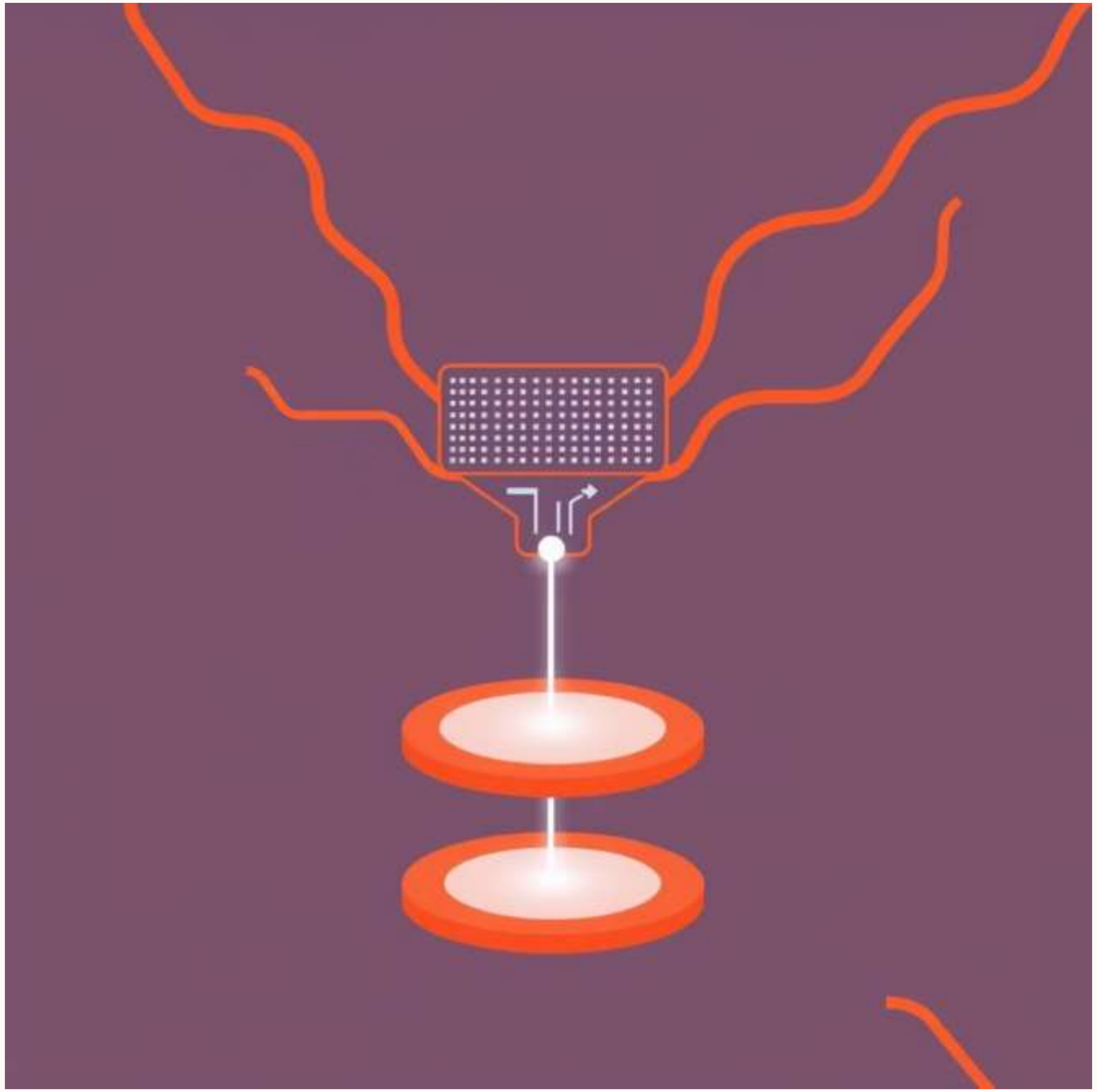
Training Neural Networks & Optimization

Explore the fundamental processes behind teaching neural networks to learn and improve performance.

FORGESPRINT
Precision Driven Proficiency

What is Backpropagation?



Learning Algorithm

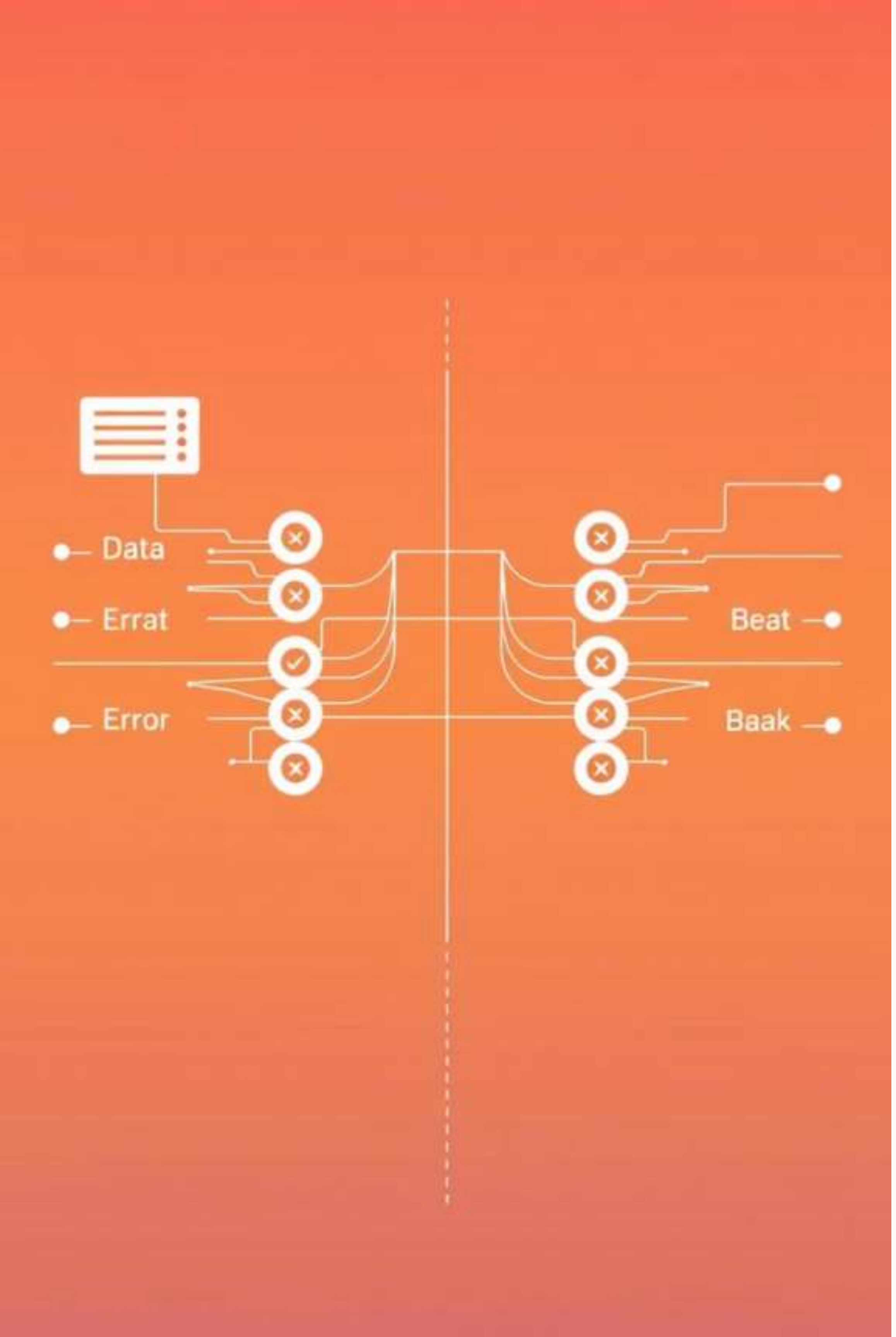
The core method for neural network training.

Error Measurement

Calculates how far off predictions are.

Weight Adjustment

Fine-tunes connections to reduce errors.



How Backpropagation Works

1. **Forward Pass**

   Input data moves through the network to generate an output.

2. **Calculate Loss**

   Compare predicted output with the actual target.

3. **Backward Pass**

   Error signals are sent back through the network.

4. **Update Weights**

   Adjust network connections based on error signals.

FORGESPRINT
Precision Driven Proficiency

Introduction to Gradient Descent

An optimization algorithm that helps minimize the "loss" or error in a neural network. It iteratively adjusts the network's weights to find the lowest point of the loss function.

$$w := w - \eta \nabla L(w)$$

- **w**: network weights
- **η**: learning rate
- **∇L**: gradient of the loss function

FORGESPRINT
Precision Driven Proficiency

Understanding the Gradient

Slope of Loss

Indicates the steepest direction of the error curve.

Direction & Magnitude

Tells how much and in which way to change weights.

Minimize Error

The ultimate goal is to reach the lowest error point.

Variants of Gradient Descent

Different approaches to updating weights, each with trade-offs.

Batch GD

Uses all data for one update.

Stochastic GD

Updates per single data point.

Mini-Batch GD

Updates per small group of data.

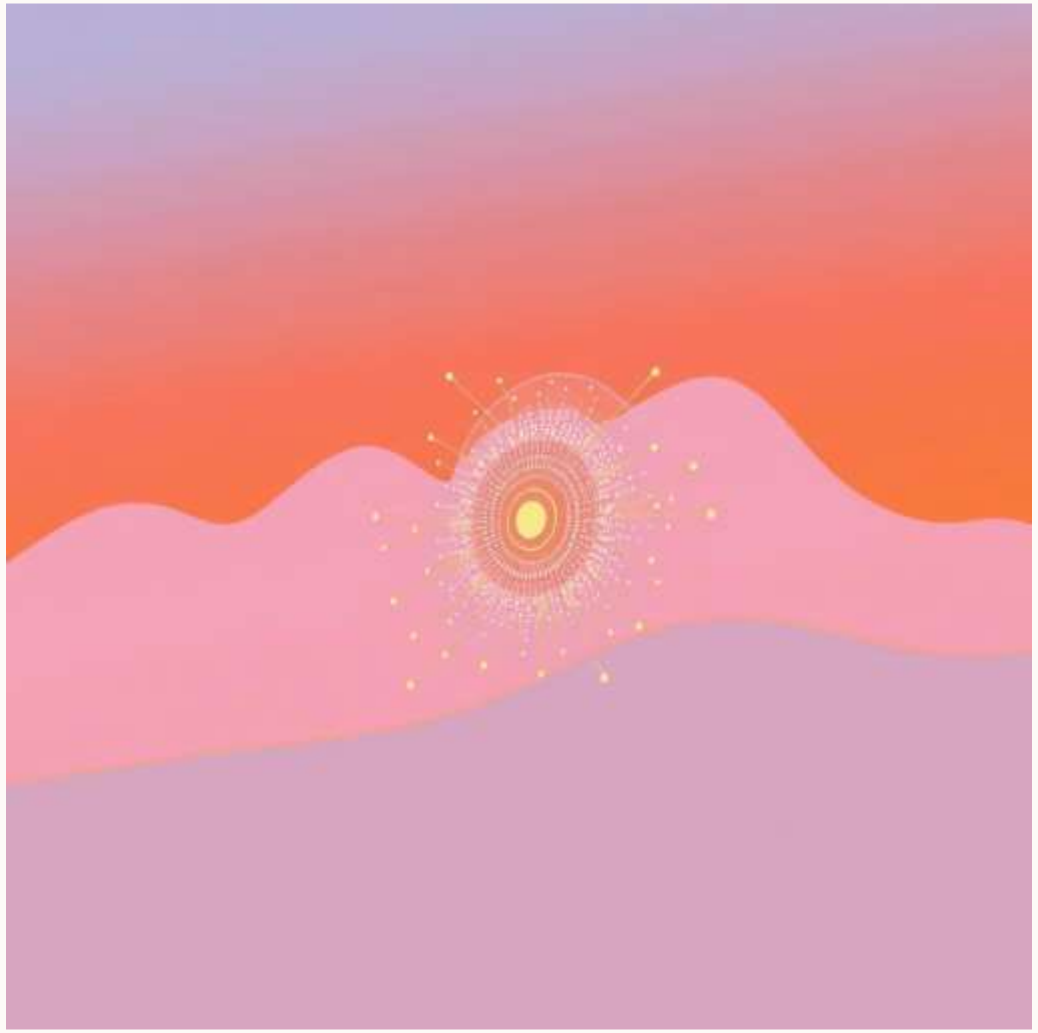
Batch Gradient Descent

Calculates the gradient using the **entire training dataset** before updating the weights. This leads to a very stable convergence.

$$w := w - \eta \frac{1}{N} \sum_{i=1}^{N} \nabla L_i(w)$$

N: Total number of training samples. Each **L_i** is the loss for one sample.

Stable, but can be very slow for large datasets.



Stochastic & Mini-Batch Gradient Descent

1

Stochastic GD

Update per sample: Weights updated after processing just one training example.

⚠ Noisy updates, but very fast.

2

Mini-Batch GD

Update per small batch: Updates weights after processing a small subset of training data (e.g., 32-256 samples).

Balances speed and stability, widely used.

Learning Rate: Key Hyperparameter

The learning rate (η) controls the size of the steps taken during weight updates.

High Learning Rate

May overshoot the optimal solution, causing instability.

Low Learning Rate

Leads to very slow convergence, extending training time.

Finding the right balance is crucial for efficient training.

FORGESPRINT
Precision Driven Proficiency

Summary & Takeaways

Backpropagation

The engine for calculating error gradients.

Gradient Descent

The optimizer for adjusting network weights.

GD Variants

Batch, Stochastic, Mini-Batch offer different speeds/stabilities.

Learning Rate

Crucial for effective and efficient training.

Mastering these basics unlocks the power of neural networks!

FORGESPRINT
Precision Driven Proficiency