# Thuật toán Lan Truyền Ngược trong Mạng Nơ-ron Nhân Tạo

## Giới thiệu

Thuật toán lan truyền ngược (backpropagation) là một phương pháp tối ưu hóa cho mạng nơ-ron nhân tạo, được sử dụng để tính toán gradient của hàm mất mát (loss function) so với các trọng số của mạng. Nó dựa trên quy tắc chuỗi của đạo hàm, cho phép cập nhật các trọng số một cách hiệu quả nhằm giảm thiểu hàm mất mát.

## Các Khái Niệm Cơ Bản

* + **Mạng Nơ-ron Nhân Tạo (Artificial Neural Network - ANN)**: Hệ thống các nút (neurons) được kết nối với nhau qua các trọng số (weights). Mỗi nút thực hiện một phép biến đổi phi tuyến lên dữ liệu vào.
  + **Hàm Kích Hoạt (Activation Function)**: Hàm được áp dụng lên đầu ra của mỗi nơ-ron để giới hạn hoặc biến đổi nó. Ví dụ: sigmoid, tanh, ReLU.
  + **Hàm Mất Mát (Loss Function)**: Hàm đánh giá sự khác biệt giữa giá trị dự đoán và giá trị thực tế. Ví dụ: mean squared error, cross-entropy.
  + **Lan Truyền Tiến (Forward Propagation)**: Quá trình tính toán đầu ra của mạng từ đầu vào qua các lớp.
  + **Lan Truyền Ngược (Backpropagation)**: Quá trình tính toán gradient của hàm mất mát với các trọng số và cập nhật chúng để giảm thiểu hàm mất mát.

## Các Bước Chính của Thuật Toán Lan Truyền Ngược

1. **Lan Truyền Tiến**:
   * Đầu vào được đưa qua mạng để tính toán đầu ra .
   * Mỗi nơ-ron thực hiện tính toán , sau đó áp dụng hàm kích hoạt .
2. **Tính Toán Hàm Mất Mát**:
   * So sánh đầu ra với giá trị thực tế để tính toán hàm mất mát .
3. **Lan Truyền Ngược:**
   * Tính toán gradient của hàm mất mát với các trọng số từ đầu ra đến đầu vào.
   * Sử dụng quy tắc chuỗi để cập nhật trọng số từng lớp.
4. **Cập Nhật Trọng Số**:
   * Sử dụng thuật toán tối ưu hóa như gradient descent để cập nhật trọng số và bias .

## Các Công Thức Toán Học Cần Thiết

### Lan Truyền Tiến

Giả sử có một lớp với trọng số và bias , đầu vào :

### Tính Toán Hàm Mất Mát

Với hàm mất mát như mean squared error:

### Lan Truyền Ngược

* + 1. **Gradient của hàm mất mát với đầu ra**
    2. **Gradient của hàm kích hoạt** (với hàm sigmoid):
    3. **Gradient của hàm mất mát với :**
    4. **Gradient của hàm mất mát với trọng số** và bias :

### Cập Nhật Trọng Số

Sử dụng gradient descent:

Trong đó là tốc độ học (learning rate).

## Ví Dụ Đơn Giản

Giả sử một mạng nơ-ron đơn giản với một lớp đầu vào, một lớp ẩn, và một lớp đầu ra. Sử dụng hàm kích hoạt sigmoid và hàm mất mát mean squared error.

### Cài Đặt

* + Đầu vào
  + Trọng số
  + Bias
  + Giá trị thực tế
  + Tốc độ học

### Lan Truyền Tiến

1. Tính toán đầu ra của lớp ẩn:

2. Tính toán đầu ra cuối cùng:

### Tính Toán Hàm Mất Mát

### Lan Truyền Ngược

1. Gradient của hàm mất mát với
2. Gradient của hàm kích hoạt sigmoid tại :
3. Gradient của hàm mất mát với :
4. Gradient của hàm mất mát với trọng số và bias :

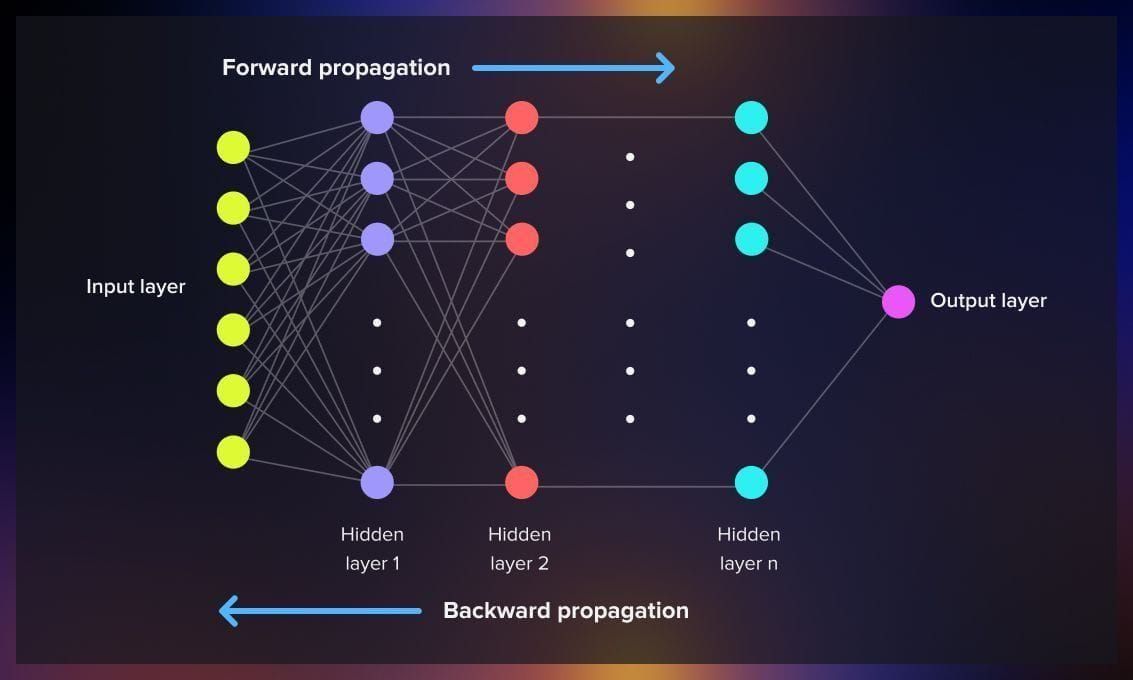
### Cập Nhật Trọng Số

Tiếp tục quá trình này cho các lớp tiếp theo và cập nhật tất cả trọng số và bias tương ứng.

## Kết Luận

Thuật toán lan truyền ngược là một phương pháp hiệu quả để tối ưu hóa trọng số trong mạng nơ-ron nhân tạo. Bằng cách tính toán gradient của hàm mất mát và cập nhật trọng số theo gradient descent, chúng ta có thể đào tạo mạng để giảm thiểu sai số dự đoán và cải thiện hiệu suất.

## Tham Khảo

 <https://serokell.io/files/a0/a05ov1m.Backpropagation_in_NN_pic1.jpg>

<https://www.youtube.com/watch?v=q555kfIFUCM&ab_channel=SirajRaval>

<https://vinodsblog.com/2019/02/17/deep-learning-backpropagation-algorithm-basics/>

<https://www.youtube.com/watch?v=q555kfIFUCM&ab_channel=SirajRaval>

<https://brilliant.org/wiki/backpropagation/#:~:text=Backpropagation%2C%20short%20for%20%22backward%20propagation,to%20the%20neural%20network's%20weights>.