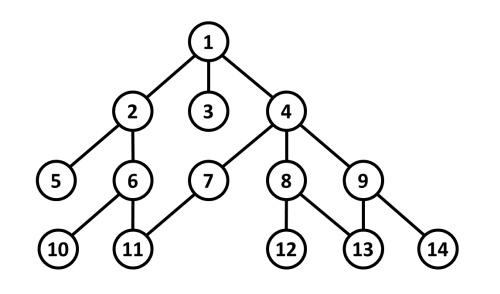
**Câu 1:** Giải thuật tìm kiếm theo chiều rộng, theo chiều sâu

* Trình bày giải thuật
* Cài đặt giải thuật cho bài toán 3 can nước

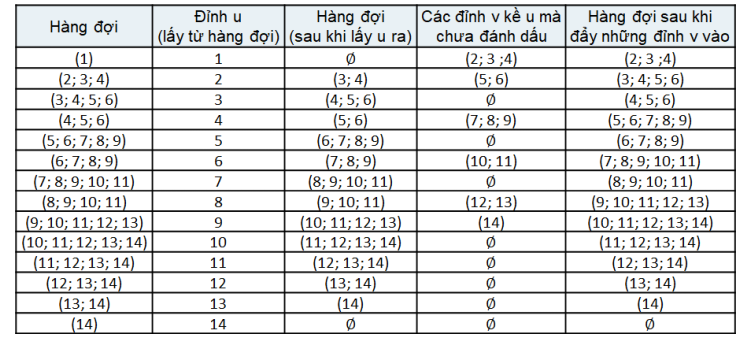
**Tìm kiếm chiều rộng:**

Giải thuật theo chiều rộng là 1 giải thuật tìm kiếm trong đồ thị hoặc cây dựa vào việc từng đỉnh từ gốc đến lá theo chiều rộng (theo tầng).

Giải thuật này dùng cấu trúc dữ liệu dạng quere:

`

ở đây ta có ví dụ sau cách duyệt sau:



ta suy ra các bc sau:

Bước 1 : khởi tạo 1 hàng đợi quere

Bước 2: Lặp qua từng phần tử trong hàng đợi

* Lấy phần tử đầu tiên trong hàng đợi , check xem nó ở đỉnh nào
* Nếu đỉnh này là đỉnh cần tìm , kết thúc tìm kiếm và trả về kết quả.
* Nếu không , thêm các đỉnh kề của đỉnh này vào hàng đợi và đánh dấu đỉnh này đã được duyệt qua

Bước 3: Nếu không tìm thấy đỉnh cần tìm , kết thúc => kq ko tìm thấy

**Tìm kiếm chiều sâu :**

Tương tự với giải thuật chiều rộng cùng sử dụng cấu trúc đồ thị hoặc cây

Về cách hoạt động thì tìm kiếm theo chiều sâu thì quét dọc , tức là quét theo từng cột

Dạng tìm kiếm này dùng cấu trúc dữ liệu dạng stack

Để tạo một giải thuật tìm kiếm theo chiều sâu (DFS), ta có thể thực hiện những bước sau:

1. Khai báo một stack để lưu trữ các nút đã được xét.
2. Thêm nút bắt đầu vào stack và đánh dấu nó là đã được xét.
3. Lặp lại các bước sau cho đến khi stack rỗng: a. Lấy nút cuối cùng trong stack ra khỏi stack và kiểm tra nếu nó là nút đích, kết thúc tìm kiếm. b. Tìm các nhánh con còn chưa được xét của nút hiện tại và thêm chúng vào stack và đánh dấu chúng là đã được xét.

**giải thuật cho bài toán 3 can nước**

Bài toán 3 can nước (Three Jugs Problem) là một bài toán tìm kiếm trên đồ thị, trong đó có 3 cái can có dung tích khác nhau và một cái tách. Mục tiêu của bài toán là tìm ra cách sử dụng tách để đổ nước từ các can vào nhau để đạt được một số dung tích cho trước.

1. Chuyển một can nước từ nguồn A sang B.
2. Chuyển một can nước từ nguồn B sang C.
3. Chuyển một can nước từ nguồn C sang A.

Để giải bài toán này, ta có thể sử dụng giải thuật tìm kiếm theo chiều sâu (DFS). Cụ thể, ta có thể thực hiện những bước sau:

1. Khai báo một stack để lưu trữ các trạng thái của 3 can nước. Mỗi trạng thái được lưu trữ dưới dạng một tuple gồm 3 số nguyên tương ứng với dung tích của 3 can.
2. Thêm trạng thái ban đầu (dung tích của 3 can ban đầu) vào stack và đánh dấu nó là đã được xét.
3. Lặp lại các bước sau cho đến khi stack rỗng: a. Lấy trạng thái cuối cùng trong stack ra khỏi stack và kiểm tra nếu nó là trạng thái đích (dung tích của 3 can đạt mục tiêu), kết thúc tìm kiếm và trả về các bước thực hiện

**Câu 2:**

Giải thuật tìm kiếm dựa trên hàm đánh giá

* Trình bày giải thuật
* Minh họa giải thuật cho bài toán tìm đường đi với các hàm đánh giá khác nhau
* Cài đặt các giải thuật A\* cho bài toán 8-puzzle

**Trình bày giải thuật**

Giải thuật tìm kiếm dựa trên hàm đánh giá (heuristic search) là một loại giải thuật tìm kiếm trên cây hoặc đồ thị, trong đó mỗi nút được đánh giá bằng một hàm đánh giá (heuristic function) xác định độ ưu tiên của nút đó. Giải thuật này sử dụng ưu tiên hàng đợi để lưu trữ các nút có độ ưu tiên cao nhất. Khi tìm kiếm, nó luôn lấy ra nút có độ ưu tiên cao nhất để xét tiếp

**Câu 3 :**

Giải thuật Minimax và các cải tiến của nó khi không gian trò chơi lớn

* Trình bày giải thuật Minimax
* Cài đặt giải thuật minimax cho trò chơi bốc bi
* Trình bày các cải tiến của Minimax cho trò chơi cờ vua

**Câu 4:**

Hồi qui tuyến tính (Linear Regression)

* Trình bày bài toán hồi qui tuyến tính
* Trình bày giải thuật Hồi qui tuyến tính
* Cài đặt giải thuật Hồi qui tuyến tính cho bài toán dự đoán cân nặng từ chiều cao của người VN
* Sử dụng thư viện scikit-learn để giải bài toán hồi qui tuyến tính ở trên (dự đoán chiều cao)
* Tìm các bài toán thực tế và sử dụng Hồi qui tuyến tính để giải quyết.

**Câu 5:**

Giải thuật tối ưu Gradient Descent

* Trình bày giải thuật gradient descent tổng quát (cho hàm nhiều biến)
* Cài đặt giải thuật gradient descent tìm Min của hàm một biến

**Câu 6:**Giải thuật lan truyền ngược Backpropagation (Mạng nơron nhân tạo)

* Mô hình Multi-layer Perceptron (MLP) là như thế nào?
* Trình bày giải thuật Backpropagation để học mô hình MLP.
* Cài đặt giải thuật Backpropagation cho bài toán phân 3 lớp (dữ liệu hai chiều) và nêu được ưu điểm của MLP với Softmax regression.
* Sử dụng thư viện Scikit-learn để học và đánh giá mô hình MLP (với số lượng node ẩn khác nhau) cho bài toán nhận diện chữ số (với 2 tập dữ liệu là: MNIST và Digits)
* Tìm các bài toán phân lớp khác để minh họa MLP.
* (Option) Tìm hiểu về Deep learning với tập dữ liệu Fashion-MINIST

**Giải thuật lan truyền ngược Backpropagation (Mạng nơron nhân tạo):**

Giải thuật lan truyền ngược (backpropagation) là một phương pháp học tự động cho mô hình mạng nơron nhân tạo (artificial neural network). Nó được sử dụng để tính toán đạo hàm của hàm mất mát (loss function) với respect đến các trọng số (weights) của mô hình, để có thể cập nhật các trọng số này tag cách sử dụng các thuật toán tối ưu hóa như stochastic gradient descent.

Giải thuật backpropagation làm việc tag cách sử dụng một quá trình lan truyền ngược (backward propagation) để tính toán đạo hàm của hàm mất mát với respect đến các trọng số của mô hình. Trong quá trình này, ta sẽ tính toán đạo hàm của hàm mất mát với respect đến các trọng số tại từng lớp của mô hình, bắt đầu từ lớp cuối cùng và lan truyền ngược lại cho đến lớp đầu tiên. Khi tính toán đạo hàm tại mỗi lớp, ta sử dụng công thức toán học của mô hình và đạo hàm của hàm mất mát tại lớp tiếp theo để tính toán đạo hàm của hàm mất mát tại lớp đó.

Giải thuật backpropagation được sử dụng trong mô hình mạng nơron nhân tạo để học từ dữ liệu huấn luyện và điều chỉnh các trọng số của mô hình để dự đoán chính xác hơn cho dữ liệu mới. Để sử dụng giải thuật backpropagation, ta cần:

1. Khởi tạo ngẫu nhiên các trọng số của mô hình.
2. Cho mô hình một bộ dữ liệu huấn luyện.
3. Tính toán đầu ra của mô hình cho mỗi mẫu dữ liệu trong bộ dữ liệu huấn luyện tag cách sử dụng công thức toán học của mô hình.
4. Tính toán hàm mất mát (loss function) cho mỗi mẫu dữ liệu tag cách so sánh đầu ra dữ liệu mới.
5. Sử dụng giải thuật backpropagation để tính toán đạo hàm của hàm mất mát

**Mô hình Multi-layer Perceptron (MLP) là như thế nào?**

Mô hình Multi-layer Perceptron (MLP) là một loại mô hình mạng nơron nhân tạo có nhiều lớp được gắn liền với nhau. Mỗi lớp gồm nhiều nơron (neurons) và các nơron trong cùng một lớp được kết nối với nhau tag các cạnh trọng số (weighted edges). Các nơron trong lớp đầu tiên (input layer) nhận dữ liệu đầu vào, các nơron trong lớp cuối cùng (output layer) trả về kết quả dự đoán của mô hình, và các lớp giữa (hidden layers) là các lớp để xử lý dữ liệu.

Để tính toán đầu ra của mô hình MLP, ta sử dụng công thức toán học sau:

Đầu ra của nơron i trong lớp j:

y\_i^j = f(x\_i^j)

Trong đó:

* y\_i^j là đầu ra của nơron i trong lớp j.
* f là hàm kích hoạt (activation function).
* x\_i^j là tổng trọng số của nơron i trong lớp j:

x\_i^j = ∑w\_ij^k \* y\_k^(j-1) + b\_i^j

Trong đó:

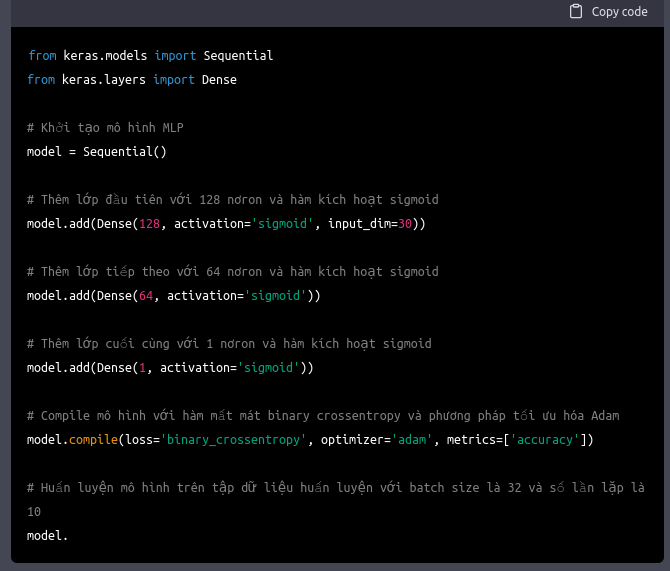
* w\_ij^k là trọng số của cạnh kết nối giữa nơron i trong lớp j và nơron k trong lớp j-1.
* y\_k^(j-1) là đầu ra của nơron k trong lớp j-1.
* b\_i^j là bias của nơron i trong lớp j

**Trình bày giải thuật Backpropagation để học mô hình MLP.**

Để học mô hình MLP tag giải thuật backpropagation, ta cần:

1. Khởi tạo ngẫu nhiên các trọng số của mô hình.
2. Cho mô hình một bộ dữ liệu huấn luyện (training data).
3. Tính toán đầu ra của mô hình cho mỗi mẫu dữ liệu trong bộ dữ liệu huấn luyện tag cách sử dụng công thức toán học của mô hình.
4. Tính toán hàm mất mát (loss function) cho mỗi mẫu dữ liệu tag cách so sánh đầu ra dự đoán của mô hình với giá trị đích thực tế của mẫu dữ liệu đó.
5. Sử dụng giải thuật backpropagation để tính toán đạo hàm của hàm mất mát với respect đến các trọng số của mô hình.
6. Lặp lại bước 3 đến bước 6 cho đến khi mô hình đạt độ chính xác cao hoặc số lần lặp đạt mức tối đa.
7. Sử dụng mô hình đã học để dự đoán cho các mẫu dữ liệu mới.

Để học mô hình MLP tag giải thuật backpropagation, ta có thể sử dụng một số thư viện phổ biến như TensorFlow, PyTorch, Keras hoặc Theano để xây dựng và huấn luyện mô hình MLP. Các thư viện này đều cung cấp các hàm và lớp cho việc xây dựng mô hình MLP và sử dụng giải thuật backpropagation để huấn luyện mô hình.

Ví dụ, ta có thể sử dụng Keras để xây dựng và huấn luyện mô hình MLP cho bài toán phân loại nhị phân (binary classification) như sau:

**Cài đặt giải thuật Backpropagation cho bài toán phân 3 lớp (dữ liệu hai chiều) và nêu được ưu điểm của MLP với Softmax regression.**

Để cài đặt giải thuật Backpropagation cho bài toán phân 3 lớp (dữ liệu hai chiều), ta có thể làm theo các bước sau:

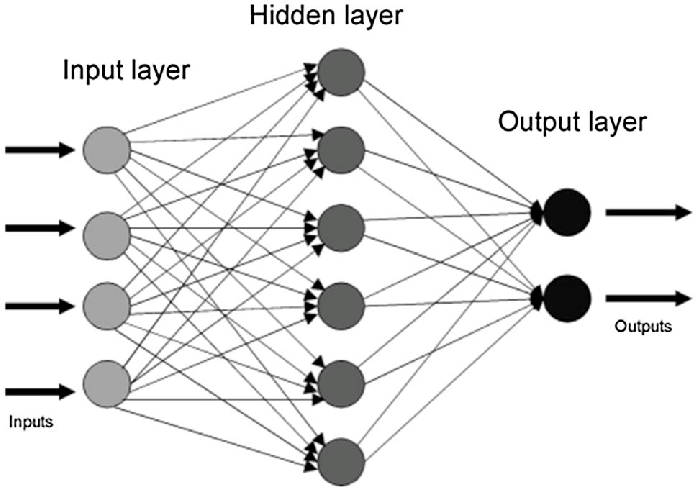
1. Khởi tạo trọng số và bias cho các lớp: Để khởi tạo trọng số và bias cho các lớp, ta có thể sử dụng các giá trị ngẫu nhiên từ khoảng (0, 1) hoặc sử dụng các phương pháp khác như khởi tạo trọng số theo phân phối chuẩn hoặc khởi tạo trọng số theo phân phối Xavier.
2. Tính toán đầu ra của mỗi lớp: Để tính toán đầu ra của mỗi lớp, ta cần tính toán hàm kích hoạt của mỗi lớp. Ví dụ, nếu ta sử dụng hàm kích hoạt sigmoid cho lớp ẩn và hàm Softmax cho lớp đầu ra, ta có thể tính toán đầu ra của mỗi lớp tag cách sử dụng công thức sau:

Đầu ra của lớp ẩn: a = sigmoid(w \* x + b)

Đầu ra của lớp đầu ra: y = Softmax(w \* a + b)

Trong đó, w và b là trọng số và bias của lớp tương ứng, x là đầu vào của mô hình, a là đầu ra của lớp ẩn và y là đầu ra của lớp đầu ra.

Các bước chi tiết như sau:



1. Khởi tạo trọng số cho các lớp đầu vào, lớp ẩn và lớp đầu ra.
2. Định nghĩa hàm kích hoạt sigmoid cho lớp ẩn và hàm Softmax cho lớp đầu ra.
3. Sử dụng các đầu vào để tính toán các giá trị đầu ra từ các lớp đầu vào và lớp ẩn bằng cách sử dụng công thức: output = sigmoid(dot(inputs, weights\_input\_to\_hidden) + biases\_hidden)
4. Tính toán giá trị đầu ra từ lớp đầu ra bằng cách sử dụng công thức: output = softmax(dot(hidden\_outputs, weights\_hidden\_to\_output) + biases\_output)
5. Tính toán độ lỗi bằng cách sử dụng hàm mất mát cross-entropy.
6. Tính toán đạo hàm của hàm mất mát theo các trọng số và bias của lớp đầu ra và lớp ẩn bằng cách sử dụng các công thức backpropagation.
7. Cập nhật các trọng số và bias của các lớp bằng cách sử dụng công thức gradient descent.

Ưu điểm của của mô hình này :

1. Mô hình MLP có thể học các quan hệ phức tạp đầu vào đầu ra
2. Mô hình MLP có thể được sử dụng cho các bài toán phân lớp và hồi quy
3. Mô hình MLP có thể được áp dụng cho những bài toán phân đa lớp và đa nhân
4. Mô hình MLP có khả năng học từ các dữ liệu không giống nhau
5. Mô hình này có thể dự đoán xác suất cho từng lớp đầu ra , do đó nó có thể được sử dụng trong các bài toán phân loại