**Tóm tắt về thuật toán K-Nearest Neighbors**

* Thuật toán K-Nearest Neighbors (KNN) là thuật toán học máy có giám sát đơn giản trong Machine Learning, được ứng dụng rộng rãi vào các bài toán phân loại và hồi quy. Ý tưởng chính của thuật toán này chính là việc tập hợp các điểm dữ liệu gần nhau trong không gian thì sẽ tổng hợp thành 1 nhóm
* Thuật toán này nằm trong nhóm các thuật toán dựa trên bộ nhớ ( học lười/lazy learning ) do thuật toán này không học bất kì các đặc điểm, tính chất gì của bộ dữ liệu training. Thay vào đó, nó so sánh mối quan hệ đầu vào với lần lượt các bộ dữ liệu lưu trữ trong bộ nhớ và tính toán để đưa ra kết quả.
* Các thông số quan trọng của thuật toán bao gồm:
* K: số điểm dữ liệu lân cận (hàng xóm) gần nhất. Chọn K phù hợp để tránh bị nhiễu (K quá bé) hoặc tránh bỏ qua các chi tiết quan trọng ( K quá lớn)

## K quá nhỏ 🡪 (1) chỉ có 1 hàng xóm quyết định lớp🡪 Dễ ảnh hưởng bởi nhiễu/ ngoại lai. K quá lớn 🡪 chọn quá nhiều hàng xóm, kể cả hàng xóm xa 🡪Mất tính lân cận.

* Khoảng cách giữa các điểm dữ liệu tùy thuộc vào các thuộc tính của chúng (thông thường là KG Euclid)
* Cách thức thuật toán hoạt động:
* Lưu trữ các bộ dữ liệu huấn luyện
* Khi có đầu vào là 1 điểm mới cần dự đoán nhãn, ta tính khoảng cách từ điểm dữ liệu mới đến tất cả các điểm dữ liệu trong tập huấn luyện
* Lựa chọn K hàng xóm lân cận có khoảng cách gần nhất với điểm dữ liệu mới
* Tùy vào yêu cầu bài toán, ta đưa đầu ra:  
  Phân loại 🡪 gán điểm dữ liệu mới nhãn lớp xuất hiện nhiều nhất.

Hồi quy🡪 Tính giá trị trung bình của các điểm dữ liệu trong K hàng xóm gần nhất và gán giá trị đó cho điểm dữ liệu mới.

* **Ưu điểm**
* **Dễ hiểu và triển khai:** Thuật toán rất dễ nắm bắt và không yêu cầu nhiều kiến thức toán học phức tạp.
* **Linh hoạt:** Có thể áp dụng cho cả bài toán phân loại và hồi quy.
* **Không có giả định về phân phối dữ liệu:** thuật toán phi tham số, không yêu cầu dữ liệu phải tuân theo một phân phối xác suất cụ thể.
* Nhược điểm:
* Hiệu năng kém với tập dữ liệu lớn: việc tính khoảng cách giữa một điểm dữ liệu mới và tất cả các điểm dữ liệu trong tập huấn luyện có thể rất tốn kém về tài nguyên.
* Kết quả dễ bị dao động bởi nhiễu: Nếu dữ liệu có nhiều điểm nhiễu, kết quả có thể bị sai lệch so với thực tế chính xác

**Giới thiệu về vấn đề xử lý ảnh**

* Hệ màu RGB là hệ màu sắc tổng hợp, trong đó mọi màu sắc nào cũng có thể được tạo ra bằng cách kết hợp 3 màu cơ bản là Red, Green, Blue với các tổ hợp cường độ sáng khác nhau. Chỉ số chạy từ 0 🡪 255 với 0 là đen, 255 là trắng. Chỉ số càng thấp, màu càng tối và ngược lại
* Ảnh màu RGB được tạo nên từ số hữu hạn các phân tử pixel, mỗi pixel mang 1 màu sắc riêng trong hệ màu RGB với giá trị xác định
* Ta có thể biểu diễn ảnh dưới dạng ma trận có height hàng, row cột với height x row là độ phân giải, mỗi phần tử trong ma trận là thông số (r,g,b) của pixel đó.

A white sheet with black text

Description automatically generated

* Tensor: dữ liệu có nhiều hơn 2 chiều. Với ảnh, ta thường biểu diễn dưới dạng tensor 3 chiều height x weight x 3 (r,g,b)
* Phép tính chập trong xử lý hình ảnh:
* Tích chập: trượt ma trận nhỏ bộ lọc (kernel) qua toàn bộ tensor ảnh. Tại mỗi vị trí, kernel nhân tương ứng từng phần tử của ảnh với phần tử của kernel 🡪 cộng tất cả và đặt kết quả vào 1 pixel đầu ra mới.
* Một số kĩ thuật thay đổi đầu ra tích chập:
* Stride: thay đổi số hàng/ cột di chuyển sau mỗi lần tích chập, giúp giảm kích thước của ma trận đầu ra so với khi không sử dụng stride 🡪 giúp downsample ảnh, bỏ qua các đặc tính không quan trọng của ảnh
* Padding: Thêm các pixels mang giá trị 0 xung quanh tensor ảnh đầu vào. 🡪 Việc thêm các pixel ngoài biên giúp cho ma trận đầu ra có cùng kích thước so với ảnh ban đầu, ngoài ra giúp xử lý các đặc tính ngoài biên của ảnh mà việc thực hiện tích chập thông thường bỏ qua.
* Nếu ma trận ảnh có kích thước nxn , sau khi thực hiện convolution với padding p và stride s thì ma trận đầu ra sẽ có kích thước:

A black and white math equation

Description automatically generated with medium confidence

* Một số ứng dụng chính của phép tích chập trong việc xử lý ảnh:
* Làm mờ ảnh: sử dụng kernel có giá trị dương để khiến cho các pixel lân cận ảnh hưởng đến pixel trung tâm, làm giảm độ tương phản của ảnh đầu ra.
* Phát hiện cạnh: Sử dụng kernel có giá trị đặc biệt ( mang cả giá trị âm và dương) để tìm ra các sự thay đổi đặc biệt về độ tương phản ánh sáng.
* Phân loại / nhận diện vật thể: thường sử dụng CNN với nhiều lớp tích chập để trích xuất các đặc trưng từ ảnh, sau đó tùy thuộc vào yêu cầu bài toán để tiếp tục phân loại.