**Tóm Tắt bài giảng**

(không gian tấm hình màu) Rm\*n\*kX----‑f---->Y (đầu ra) Rp Ví dụ: { gấu,đại bàng, trái cây,mặt người } 🡪R4 Ví dụ : x là mèo y =[0,0,1,0]

Chia hàm f thành 2 hàm nhỏ

g1 Z Rd

g3  **h1 Y = [h1(z),….,hP(z)] so sánh**

**g2 h2 y = [y1,……,yp] Rp**

**g4 hp distance( khoảng cách)**

**d(Y,y) = nếu Y giống y thì d nhỏ**

**nếu Y khác y thì d lớn**

-Ví dụ: g1 là tóc màu gì---->1 con số

g2 là tóc dài hay ngắn ---->1 con số

g3 là tóc kiểu gì -----> 1 con số

g4 là khôn mặt ra sao---->1 con số

g = [g1,….,gd**] :basic(hàm cơ sở)**

g(x) = Z =[z1=g1(x)….,zd=gd(x)] Rd : vector tọa độ(coordinates) ­

**vậy machine learning là đi tìm hàm f gồm g và h .Và đi tìm hàm f đầu ra mà so với đầu ra mà mình biết là phải nhỏ nhất**

để tìm khoảng cách nhỏ nhất ta có

D(f,f\*) Yt,yt)

Với bài toán linear (hàm tuyến tính )

H = wx +b

Nhưng nếu đầu ra y là 1 số R thì để tính khoảng cách thì ta phải:

Regression: d(y,Y) = trị tuyệt đối thì máy tính khó tính toán

(y–Y)^2---->t-Yt)/N

- Vậy bài toán đơn giản nhất của linear regression là:

Ví dụ đầu vào là các bức hình người mình chuyển qua 1 con số có kích cỡ của người đã có tọa độ từ đó mình chuyển qua hàm tuyến tính để đưa ra chiều cao của người đó. Thì mình so sánh chiều cao của hàm tính và chiều cao thực tế và tính sai sô bình phương để tìm hàm tính chiều cao tốt nhất