Mục Lục

[Chương 1: Giới thiệu tổng quan 2](#_Toc8929671)

[1.1: Đặt vấn đề 2](#_Toc8929672)

[1.2: Phương pháp giải quyết: 2](#_Toc8929673)

[Chương 2: Cơ sở lý thuyết 4](#_Toc8929674)

[2.1: Đặc trưng HOG 4](#_Toc8929675)

[2.1.1: Tổng quan về HOG 4](#_Toc8929676)

[2.1.2: Các bước trích xuất HOG 4](#_Toc8929677)

[2.2: NN (Neural Network ) 11](#_Toc8929678)

[2.2.1: Cấu trúc mạng Neural Network 11](#_Toc8929679)

[2.2.2: Sigmoid Neurons 14](#_Toc8929680)

[2.2.3: Lan truyền thẳng 16](#_Toc8929681)

[2.2.4: Softmax function 17](#_Toc8929682)

[2.2.5: Lan truyền ngược và đạo hàm 18](#_Toc8929683)

# Giới thiệu tổng quan

## Đặt vấn đề

Với tình hình giaothông hiện nay*,* tình trạng ùn tắc giao thông tại một số tuyến đường trọng điểm trên địa bàn Thành Phố đã diễn ra thường xuyên gây bức xúc và gây khó khăn cho việc đi lại của người dân và du khách. Mặt khác, thời tiết nắng nóng, oi bức khiến các con đường trở nên vắng vẻ hơn.Tại nhiều nút giao thông, việc phải dừng chờ đèn đỏ từ 70 - 90 giây thực sự là ám ảnh đối với người đi xe máy, xe đạp.

Vì vậy, việc quản lý giao thông trên các ngã ba, ngã tư hay những nơi ùn tắc giao thông thì một yêu cầu quan trọng là phát hiện và đếm số lượng xe ô tô hay xe máy lưu thông từ dữ liệu video thu được thông qua hệ thống camera quan sát để giải quyết bài toán quản lý, điều hành, phân luồng và điều khiển giao thông là một điều cấp bách và cần thiết đối với tình hình giao thông hiện nay.

## Phương pháp giải quyết:

Với tốc độ phát triển khoa học kỹ thuật ngày càng phát trển mạnh mẽ và phổ biến toàn cầu. Nhiều công nghệ giải thuật học máy ra đời nhằm phục vụ cho nhu cầu công nghệ của con người.

Một số thuật toán mà chúng ta có thể áp dụng:

* Giải thuật SIFT (Scale-invariant Feature Transform): là một thuật toán tiêu biểu và có hiệu quả khá cao. Dựa theo các đặc trưng cục bộ bất biến trong ảnh với việc thay đổi tỷ lệ ảnh, quay ảnh, đôi khi thay đổi điểm nhìn hoặc thay đổi cường độ chiếu sáng của ảnh.
* Giải thuật SURF (Speed Up Robust Feature): là một phương pháp phát hiện và mô tả hình ảnh mạnh mẽ, do Herbert Bay đưa ra lần đầu vào năm 2006. Nó được lấy cảm hứng từ SIFT. Tuy nhiên SURF nhanh hơn nhiều so với SIFT và nó chống lại sử biến đổi hình ảnh khác nhau tốt hơn SIFT. Bộ phát hiện đặc trưng của SURF sử dụng phép xấp xỉ trên ma trận Hessian và ảnh tích hợp để làm giảm thời gian tính toán một cách đáng kể. SURF được mô tả bởi vector có số chiều ít hơn SIFT nên tốc độ so khớp nhanh hơn, tuy nhiên độ bền vững vẫn được bảo đảm.
* HOG (histogram of oriented gradients): là một feature descriptor được sử dụng trong computer vision và xử lý hình ảnh, dùng để detec một đối tượng.HOG được tính toán trên một lưới dày đặc các cell và chuẩn hóa sự tương phản giữa các block để nâng cao độ chính xác. Hog được sử dụng chủ yếu để mô tả hình dạng và sự xuất hiện của một object trong ảnh

Học máy, có tài liệu gọi là Máy học, (machine learning) là một lĩnh vực của [trí tuệ nhân tạo](http://vi.wikipedia.org/wiki/Tr%C3%AD_tu%E1%BB%87_nh%C3%A2n_t%E1%BA%A1o) liên quan đến việc nghiên cứu và xây dựng các kĩ thuật cho phép các hệ thống "học" tự động từ dữ liệu để giải quyết những vấn đề cụ thể. Học máy có hiện nay được áp dụng rộng rãi bao gồm [máy truy tìm dữ liệu](http://vi.wikipedia.org/wiki/M%C3%A1y_truy_t%C3%ACm_d%E1%BB%AF_li%E1%BB%87u), [chẩn đoán y khoa](http://vi.wikipedia.org/wiki/Ch%E1%BA%A9n_%C4%91o%C3%A1n), phát hiện [thẻ tín dụng giả](http://vi.wikipedia.org/w/index.php?title=Th%E1%BA%BB_t%C3%ADn_d%E1%BB%A5ng_gi%E1%BA%A3&action=edit&redlink=1), phân tích [thị trường chứng khoán](http://vi.wikipedia.org/wiki/Th%E1%BB%8B_tr%C6%B0%E1%BB%9Dng_ch%E1%BB%A9ng_kho%C3%A1n), phân loại các [chuỗi DNA](http://vi.wikipedia.org/w/index.php?title=Chu%E1%BB%97i_DNA&action=edit&redlink=1), [nhận dạng tiếng nói](http://vi.wikipedia.org/wiki/Nh%E1%BA%ADn_d%E1%BA%A1ng_ti%E1%BA%BFng_n%C3%B3i) và [chữ viết](http://vi.wikipedia.org/w/index.php?title=Nh%E1%BA%ADn_d%E1%BA%A1ng_ch%E1%BB%AF_vi%E1%BA%BFt_tay&action=edit&redlink=1), [dịch tự động](http://vi.wikipedia.org/wiki/D%E1%BB%8Bch_t%E1%BB%B1_%C4%91%E1%BB%99ng), [chơi trò chơi](http://vi.wikipedia.org/w/index.php?title=Tr%C3%B2_ch%C6%A1i_chi%E1%BA%BFn_l%C6%B0%E1%BB%A3c&action=edit&redlink=1) và [cử động rô-bốt](http://vi.wikipedia.org/w/index.php?title=C%E1%BB%AD_%C4%91%E1%BB%99ng_r%C3%B4-b%E1%BB%91t&action=edit&redlink=1) (robot locomotion). Trong lĩnh vực máy học có nhiều phương pháp học khác nhau.sau đây là một số phương pháp học máy.

* Thuật toán **cực đại hóa kỳ vọng** (tiếng Anh hay được gọi là **EM** viết tắt của **Expectation-Maximization**) là một kỹ thuật được dùng rộng rãi trong thống kê và học máy để giải bài toán tìm hợp lý cực đại hoặc hậu nghiệm cực đại (MAP) của một mô hình xác suất có các biến ẩn.
* SVM(Support Vector Machine): là một thuật toán thuộc nhóm Supervised Learning (Học có giám sát) dùng để phân chia dữ liệu thành các nhóm riêng biệt.
* Mạng nơ-ron nhân tạo (Neural Network - **NN**) là một mô hình lập trình rất đẹp lấy cảm hứng từ mạng nơ-ron thần kinh. Kết hợp với các kĩ thuật học sâu (Deep Learning - **DL**), NN đang trở thành một công cụ rất mạnh mẽ mang lại hiệu quả tốt nhất cho nhiều bài toán khó như nhận dạng ảnh, giọng nói hay xử lý ngôn ngữ tự nhiên.

Hệ thống của chúng tôi sử dụng camera giám sát đặt cố định trên trên ngã ba hay ngã tư để thu nhận thông tin môi trường và xử lý. Đối với bài toán nhận dạng xe ô tô hay xe máy, chúng tôi tiến hành chia bài toán thành các bài toán nhỏ: (1) Tiền xử lý ảnh đầu vào, (2) tính các vecto đặc trưng HOG, (3) huấn luyện mạng noron,(4)Phân biệt và đếm số lượng xe.

* (1). Tiền xử lý: Ảnh đầu vào sẽ được thu thập tất cả các trường hợp nằm ở các vị trí khác nhau trong cửa sổ. Được phân chia cụ thể vào xe máy, xe ô tô và hình ảnh nền. Tất cả các ảnh được quy về kích cỡ size 64:64
* (2). Tính vecto đặc trưng HOG: Đây là bước quang trọng quyết định tới toàn bộ kết quả của giai đoạn sau. Để phát hiện và phân biệt loại xe có nhiều cách tiếp cận khác nhau, như khớp mẫu, mạng nơ – ron, HOG... dựa trên tỷ lệ kích thước chiều dài trên rộng, tỉ lệ pixel trên hình ảnh, chúng tôi tính toán ra các vecto đặc trưng cho từng ảnh khác nhau
* (3) huấn luyện mạng noron: Đầu vào là bảng excel tổng hợp tất cả các vecto đặc trưng HOG cho tất cả các ảnh. Sau đó huấn luyện tạo ra mạng noron mong muốn.
* (4)Phân biệt và đếm số lượng xe:  Chúng tôi tiến hành huấn luyện bộ nhận dạng xe bằng đặc trưng HOG. Từ kết quả huấn luyện này, đối với mỗi frame ảnh đầu vào thu được về tình trạng xe lưu thông ở đường giao nhau, chúng tôi sử dụng các cửa sổ con quét qua tất cả các vùng trên frame ảnh để xác định vùng chứa ôtô hay xe máy. Từ đó thống kê số lượng xe trên điểm dừng đèn đỏ ở đoạn đường giao nhau là một điều cần thiết cho việc phân luồng giao thông và quy định thời gian được lưu thông.

# Cơ sở lý thuyết

## Đặc trưng HOG

### Tổng quan về HOG

HOG (Histogram Of Oriented Gradients) là một feature descriptor được sử dụng để phát hiện một đối tượng. Các khái niệm về HOG được nêu ra từ năm 1986 tuy nhiên cho đến năm 2005 HOG mới được sử dụng rộng rãi sau khi Navneet Dalal và Bill Triggs công bố những bổ sung về HOG. HOG được tính toán trên một lưới dày đặc các cell và chuẩn hóa sự tương phản giữa các block để nâng cao độ chính xác. HOG được sử dụng chủ yếu để mô tả hình dạng và sự xuất hiện của một đối tượng trong ảnh. Đặc trưng được dùng nhiều trong lĩnh vực phát hiện đối tượng. Ý tưởng ra đời của HOG là do hình dạng và trạng thái xuất hiện của vật có thể được đặc trưng bằng sự phân bố về cường độ và hướng của cạnh.

### Các bước trích xuất HOG



#### Ảnh Input

Scale ảnh chứa đối tượng cần nhận dạng



Ảnh input

#### Chẩn hóa Gamma & Colour

Mục đích: Làm giảm bớt ảnh hưởng của sự thay đổi ánh sang và ảnh hưởng của bóng mờ



Ảnh đã đưa về ảnh xám, kí hiệu là: I

#### Tính Gradients

Dx = [-1 0 1], D­y = [-1 0 1]T

Tính đạo hàm ảnh theo phương X: IX = I \* Dx



Ảnh Ix sau khi đã đạo hàm theo phương x của ảnh I

Tính đạo hàm ảnh theo phương Y: IY = I \* DY



Ảnh Iy sau khi đã đạo hàm theo phương y của ảnh I

Cường độ hướng biến thiên Gradient tại mỗi pixel:

#### Chia hướng & Gom nhóm các đặc trưng tại mỗi cell

Việc lưu trữ chính xác giá trị hướng biến thiên của từng điểm ảnh (x, y) tốn nhiều chi phí. Nên phải chia không gian hướng biến thiên thành các bin để tối ưu hóa

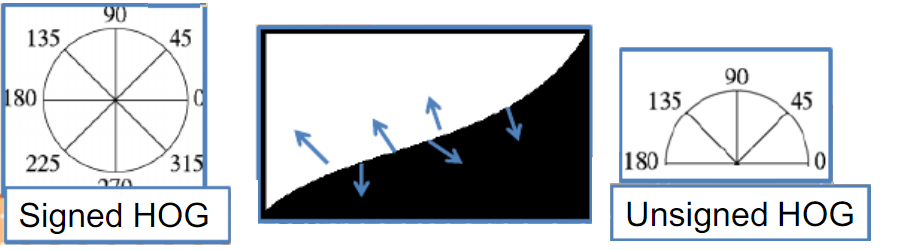
Góc hướng nghiên r tại pixel(x, y) có độ lớn(x, y) được rời rạc hóa giá trị vào 1 trong p bins

* Trường hợp rời rạc hóa unsigned-HOG:

B(x, y) = round()mod p

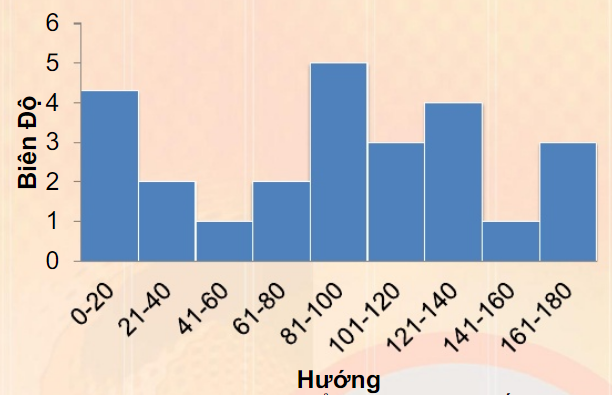
* Trường hợp rời rạc hóa signed-HOG

B(x, y) = round()mod p

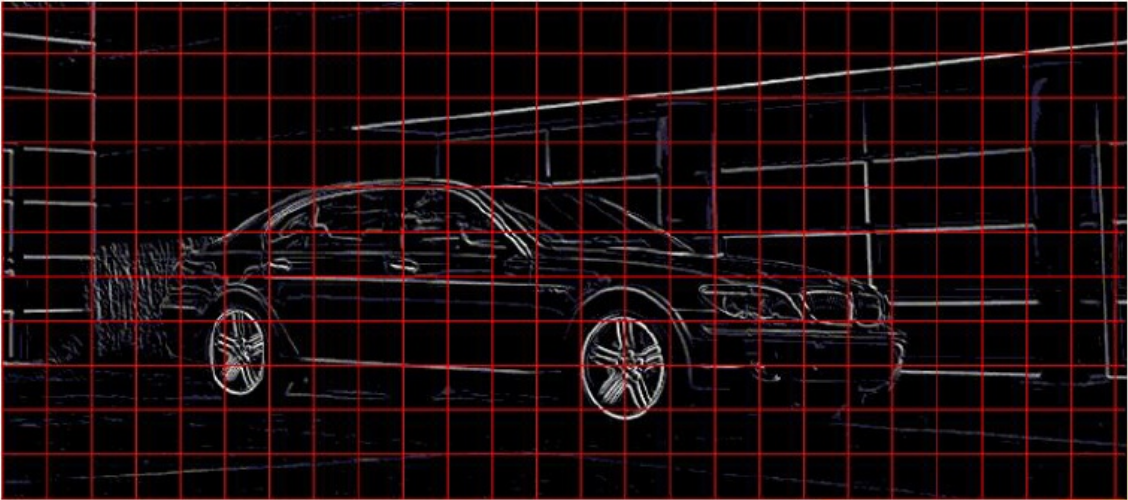


Gradient Histogram cho từng Cell. Giá trị bin được định lượng bởi tổng cường độ biến thiên của các pixels thuộc về bin đó. Độ lớn của Cell phụ thuộc vào các thông số là số lượng Bin, Kích thước của Cell. Cell càng lớn thì độ chính giá càng giảm vì những điểm xa nhau thì xác xuất cùng một đối tượng là thấp. Cell càng nhỏ thì các đặc trưng của HOG thu được là nhiều, dẫn đến dữ liệu của HOG lớn, làm cho độ phức tạp tăng lên, dẫn đến mất nhiều thời gian và bộ nhớ. Các tham số thường được lựa chọn

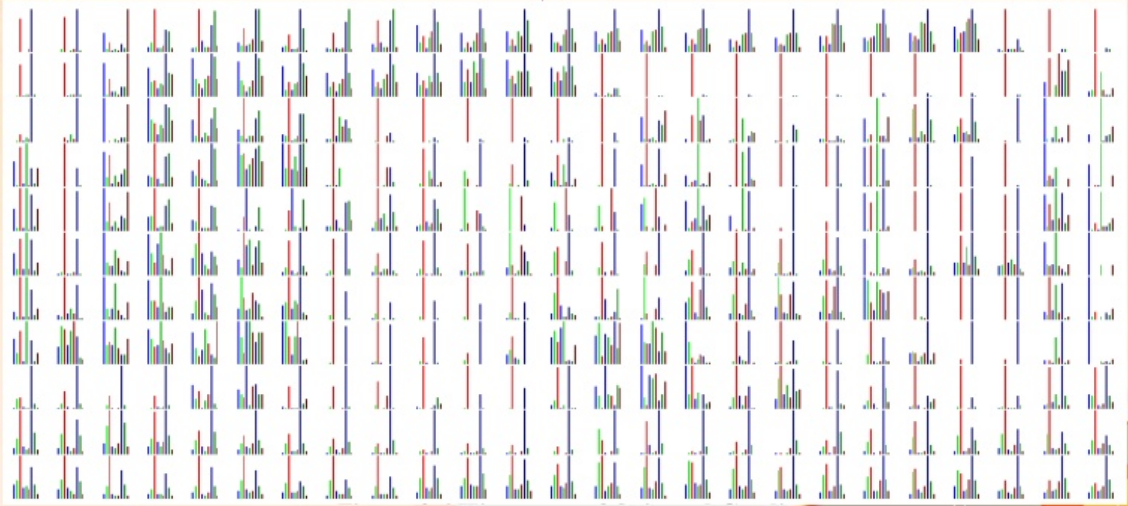
* + Số bin hướng có thể lựa chọn: 9 bins / 18bins / 36 bins
  + Kích thước Cell: 8x8 pixels,
  + Kích thước Block: 2x2 cells



Gradient Histogram của Cell với 9 Bins, 8x8 pixels



Chia cell cho ảnh

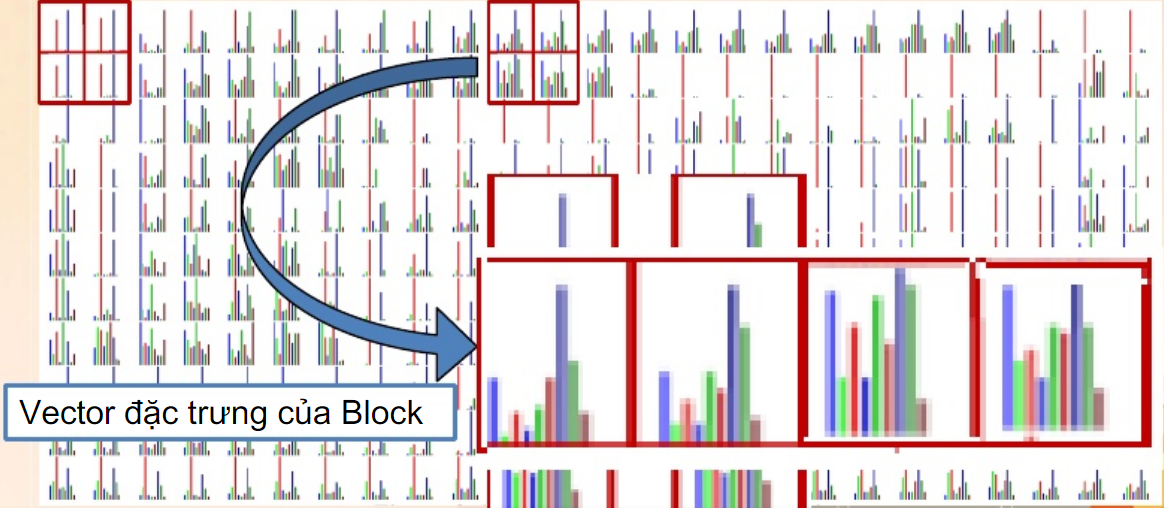


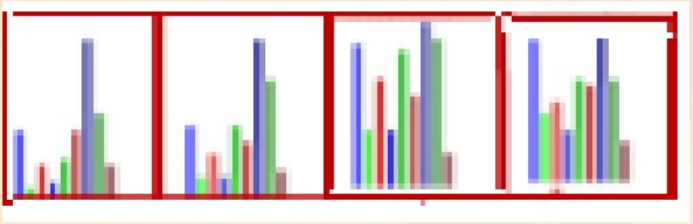
Histogram cho từng cell

#### Trích đặc trưng cho khối

Vector đặc trưng của mỗi khối được tính bằng cách ghép nối các vector đặc trưng của từng cell trong khối với nhau

Vector đặc trưng của 1 block có thể được minh họa như sau:



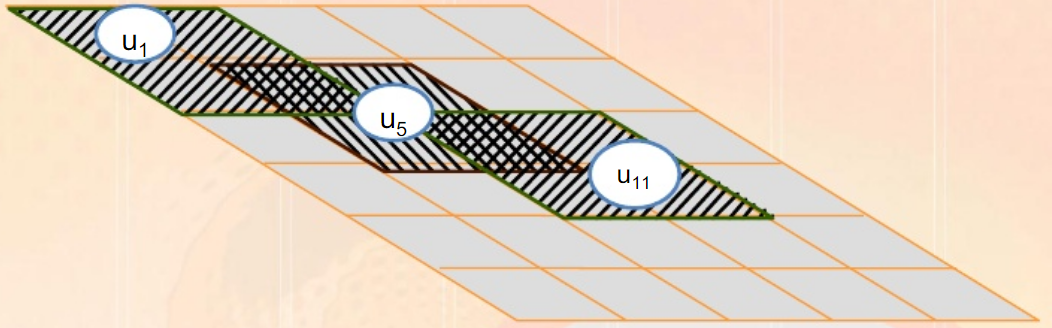


Xét giả thiết:

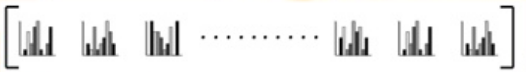
* + Kích thước block: 2x2(cells)
  + Số thành phần của vector đặc trưng tại mỗi cell: 18
  + Số thành phần của vector đặc trưng tại mỗi Block: 18 \* (2 \* 2) = 72 (thành phần)

#### Tính vector đặc trưng cho cửa sổ

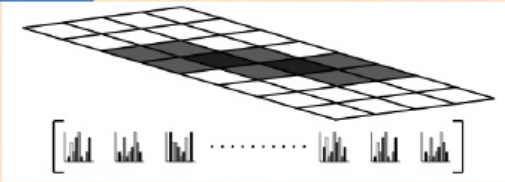
Ghép nối vector đặc trưng của các block, ta thu được vector đặc trưng của cửa sổ. Lưu ý: Các Block trong cửa sổ có thể nằm chồng lên nhau



Vector đặc trưng cho cửa sổ: [u1, ..., u5, ..., u11, ...]



* Kích thước vector đặc trưng của cửa sổ:
  + Xét giả thiết:
    - Kích thước cửa sổ: 64x64(pixels)
    - Số overlap cell giữa 2 Block: 1(cell)
    - Số thành phần của vector đặc trưng tại mỗi Block: 72



* + - * Số block có trong 1 cửa sổ:
      * Số thành phần của vector đặc trưng cho cửa sổ:

72 \* n­Blocks = 72 \* 49 = 3528(thành phần)

## ****NN**** (Neural Network )

### Cấu trúc mạng Neural Network

Một mạng neural gồm 3 lớp gồm: Input, hidden layer và output

  
*hình 2.2.1 : Cấu trúc mạng NN cơ bản*

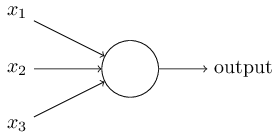
Mô tả :

* **Tầng vào** (*input layer*): Là tầng bên trái cùng của mạng thể hiện cho các đầu vào của mạng.
* **Tầng ra** (*output layer*): Là tầng bên phải cùng của mạng thể hiện cho các đầu ra của mạng.
* **Tầng ẩn** (*hidden layer*): Là tầng nằm giữa tầng vào và tầng ra thể hiện cho việc suy luận logic của mạng.

Tầng vào và Tầng ra gồm 1 lớp với số lượng các nút khác nhau tùy thuộc vào từng bài toán thực tế. Riêng tầng ẩn (hidden layer) là tầng có thể chứa nhiều lớp ẩn, số lượng lớp ẩn tùy thuộc vào từng mạng neural riêng biệt , để tối ưu hóa một mô hình và tìm ra một mô hình tốt nhất có thể thì số lớp ẩn của một mạng neural là một trong những tham số vô cùng quan trọng.

  
hình 2.2.2 Cấu trúc NN gồm 3 đầu vào 2 lớp ẩn

Các nút ở lớp này liên kết với tất cả các nút ở lớp kia tạo ra một mạng các nút kết nối với nhau. Như vậy một nút sẽ nhận một hoặc nhiều đầu X vào dạng nhị phân và cho ra một kết quả output dạng nhị phân duy nhất. Các đầu vào được điều phối tầm ảnh hưởng bởi các tham số trọng lượng tương ứng w của nó, còn kết quả đầu ra được quyết định dựa vào một ngưỡng quyết định b nào đó:



Hinh: ảnh 1 nút cơ bản với 3 đầu vào

Đặt *b*=−threshold , ta có thể viết lại thành:

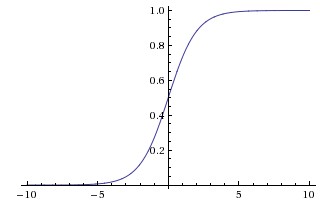
Để dễ hình dung, ta lấy ví dụ 1 quyết định dựa vào 4 yếu tố sau:

Thì ta coi 4 yếu tố đầu vào là x1, x2, x3, x4 ​ và nếu *o*=0 thì quyết định không thực hiện ,*o*=1 thì quyết định thực hiện. Giả sử mức độ quan trọng của 4 yếu tố trên lần lượt là  ​w1=0.05, *w2*​=0.5, *w3*​=0.2, *w4*​=0.25 và chọn ngưỡng *b*= -0.5 thì ta có thể thấy rằng yếu tố w1có ảnh hưởng chỉ 5% tới quyết định và w2 ảnh hưởng tới 50% quyết định

Nếu gắn *x*0​=1 và *w*0​=*b*, ta còn có thể viết gọn lại thành:

### Sigmoid Neurons

Với đầu vào và đầu ra dạng nhị phân, ta rất khó có thể điều chỉnh một lượng nhỏ đầu vào để đầu ra thay đổi chút ít, nên để linh động, ta có thể mở rộng chúng ra cả khoảng [0, 1]. Lúc này đầu ra được quyết định bởi một hàm sigmoid *σ*(X). Hàm sigmoid có công thức:



Sigmoid Function

Đặt z = X thì công thức của nút lúc này sẽ có dạng:

Tới đây thì ta có thể thấy rằng mỗi sigmoid neuron cũng tương tự như một bộ phân loại tuyến tính (logistic regression) bởi xác suất P(yi = 1|xi ;W) = *σ*(X).

Thực ra thì ngoài hàm sigmoid ra, ta còn có thể một số hàm khác như tanh, ReLU để thay thế hàm sigmoid bởi dạng đồ thị của nó cũng tương tự như sigmoid. Một cách tổng quát, hàm perceptron được biểu diễn qua một hàm kích hoạt (activation function) f(z)f(z) như sau:



* Output layer nhiều khi không có activation function mà sử dụng trực tiếp giá trị đầu vào z(l)izi(l) của mỗi unit. Hoặc nói một cách khác, activation function chính là hàm identity, tức đầu ra bằng đầu vào. Với các bài toán classification, output layer thường là một Softmax regression layer giúp tính xác suất để một điểm dữ liệu rơi vào mỗi class.
* Mặc dù activation function cho mỗi unit có thể khác nhau, trong cùng một network, activation như nhau thường được sử dụng. Điều này giúp cho việc tính toán được đơn giản hơn

### Lan truyền thẳng

Tất cả các nốt mạng (nơ-ron) được kết hợp đôi một với nhau theo một chiều duy nhất từ tầng vào tới tầng ra. Tức là mỗi nốt ở một tầng nào đó sẽ nhận đầu vào là tất cả các nốt ở tầng trước đó mà không suy luận ngược lại. Hay nói cách khác, việc suy luận trong mạng NN là suy luận tiến (feedforward):

Trong đó, *n*(*l*) số lượng nút ở tầng *l* tương ứng và *aj*(*l*)​ là nút mạng thứ *j* của tầng *l*. Còn  *wij*(*l*+1)​ là tham số trọng lượng của đầu vào *aj*(*l*)​ đối với nút mạng thứ *i* của tầng *l*+1 và *bi*(*l*+1)​ là độ lệch (bias) của nút mạng thứ *i* của tầng *l*+1. Đầu ra của nút mạng này được biểu diễn bằng  *ai*(*l*+1)​ ứng với hàm kích hoạt *f*(*zi*​) tương ứng.

Riêng với tầng vào, thông thường  a(1) cũng chính là các đầu vào x tương ứng của mạng.

Để tiện tính toán, ta coi  *a*0(*l*)​ là một đầu vào và *wi*0(*l*+1) =​*bi*(*l*+1)​ là tham số trọng lượng của đầu vào này. Lúc đó ta có thể viết lại công thức trên dưới dạng véc-tơ:

Nếu nhóm các tham số của mỗi tầng thành một ma trận có các cột tương ứng với tham số mỗi nút mạng thì ta có thể tính toán cho toàn bộ các nút trong một tầng bằng véc-tơ:

.

### Softmax function

Chúng ta cần một mô hình xác suất sao cho với mỗi input x, ai thể hiện xác suất để input đó rơi vào class i Vậy điều kiện cần là các ai phải dương và tổng của chúng bằng 1. Để có thể thỏa mãn điều kiện này, chúng ta cần nhìn vào mọi giá trị zi và dựa trên quan hệ giữa các zi này để tính toán giá trị của ai. Ngoài các điều kiện ai lớn hơn 0 và có tổng bằng 1, chúng ta sẽ thêm một điều kiện cũng rất tự nhiên nữa, đó là: giá trị zi=X càng lớn thì xác suất dữ liệu rơi vào class I càng cao. Điều kiện cuối này chỉ ra rằng chúng ta cần một hàm đồng biến ở đây.

Chú ý rằng zi có thể nhận giá trị cả âm và dương. Một hàm số mượt đơn giản có thể chắc chắn biến zi thành một giá trị dương, và hơn nữa, đồng biến, là hàm **exp**(zi)= Điều kiện mượt để thuận lợi hơn trong việc tính đạo hàm sau này. Điều kiện cuối cùng, tổng các aiai bằng 1 có thể được đảm bảo nếu:

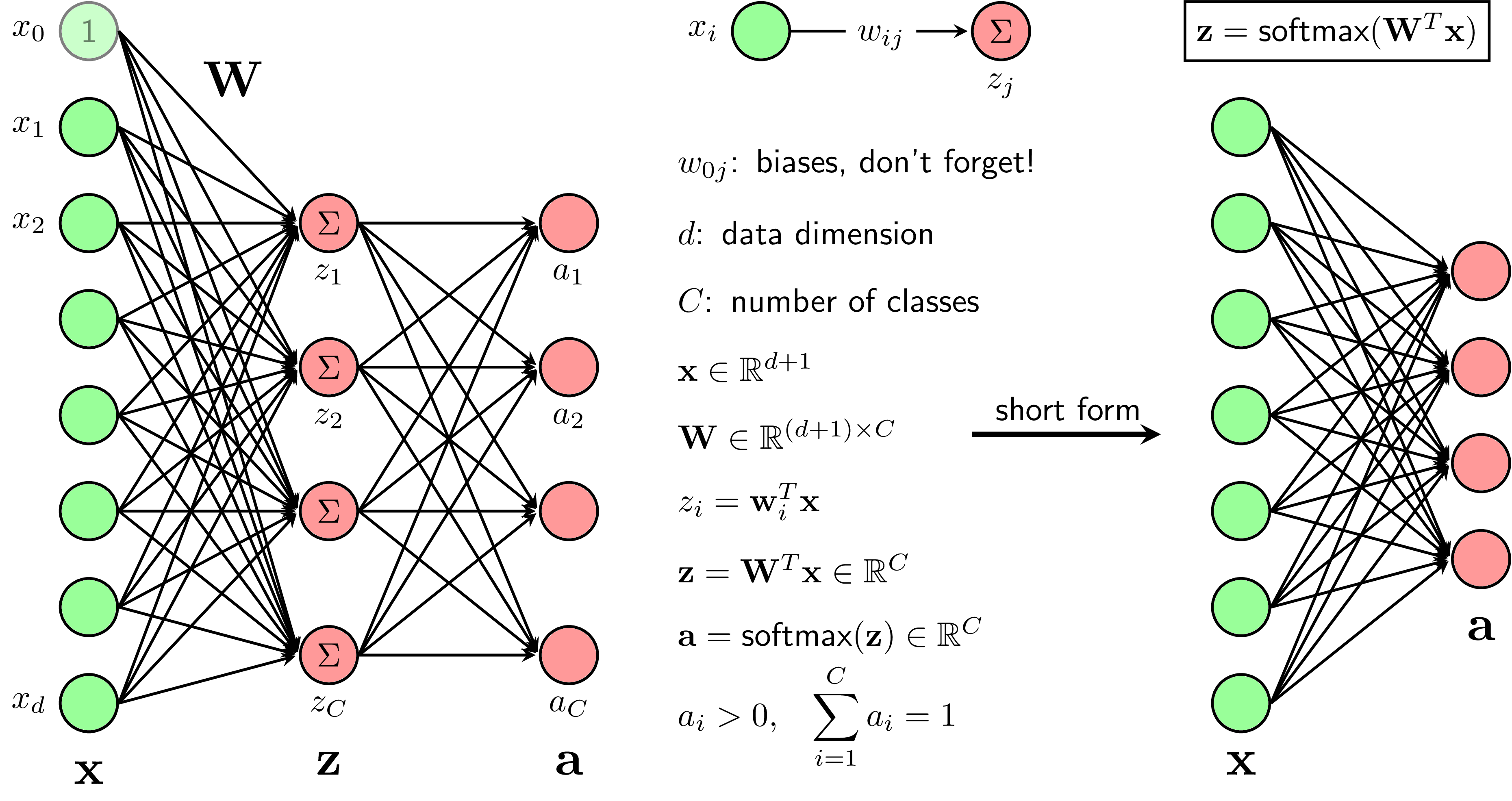
∀i=1,2,…,C

Hàm số này, tính tất cả các aiai dựa vào tất cả các zi, thõa mãn tất cả các điều kiện đã xét: dương, tổng bằng 1, giữ được thứ tự của zi. Hàm số này được gọi là softmax function. Chú ý rằng với cách định nghĩa này, không có xác suất ai nào tuyệt đối bằng 0 hoặc tuyệt đối bằng 1, mặc dù chúng có thể rất gần 0 hoặc 1 khi zi rất nhỏ hoặc rất lớn khi so sánh với các zj,j≠i

Lúc này, ta có thể giả sử rằng:

Trong đó, P(y=i|x;W) được hiểu là xác suất để một điểm dữ liệu x rơi vào class thứ ii nếu biết tham số mô hình (ma trận trọng số) là W.

Hình vẽ dưới đây thể hiện mạng Softmax Regression dưới dạng neural network:



Hình 2: Mô hình Softmax Regression dưới dạng Neural network.

Ở phần bên phải, hàm tuyến tính Σ và hàm softmax (activation function) được tách riêng ra để phục vụ cho mục đích minh họa. Dạng short form ở bên phải là dạng hay được sử dụng trong các Neural Networks, lớp a được ngầm hiểu là bao gồm cả lớp z.

### Lan truyền ngược và đạo hàm

Để tính đạo hàm của hàm lỗi ∇*J*(W) trong mạng NN, ta sử dụng một giải thuật đặc biệt là giải thuật **lan truyền ngược** (backpropagation). Nhờ có giải thuật được sáng tạo vào năm 1986 này mà mạng NN thực thi hiệu quả được và ứng dụng ngày một nhiều cho tới tận ngày này. Về cơ bản phương pháp này được dựa theo quy tắc chuỗi đạo hàm của hàm hợp và phép tính ngược đạo hàm để thu được đạo hàm theo tất cả các tham số cùng lúc chỉ với 2 lần duyệt mạng

Tính đạo hàm theo *z* ngược lại từ *l*=(*L*−1)→2 theo công thức:

= ((**W(**l+1))T )

với z(*l*) tính được ở bước 1 và  tính được ở vòng lặp ngay trước

Tính đạo hàm  
Tính đạo hàm theo tham số wbằng công thức:

với a(*l-1*) tính được ở bước 1 và  tính được ở vòng lặp ngay trước.

3. Thuật toán đếm số xe

3.1 Mô tả sơ đồ khối

3.1.1 Sơ đồ khối training

Bảng training

Cơ sở dữ liệu

Tách đặc trưng

Huấn luyện mạng nơ ron

Mô hình tối ưu

3.3.2 Sơ đồ khối testing

Mô tả sơ đồ khối:

ảnh test

Output

Mô hình đã tối ưu

Tách đặc trưng

* Khối dữ liệu: gồm hình ảnh (input data ), số lượng ảnh tùy thuộc vào thành phần của các nhóm target(classes). Số lượng hình ảnh tùy vào từng đề tài và mục đích riêng. Đối với bài toán nhận dạng thì số classes thưởng trên 2
* Khối tách đặc trung : sử dụng HOG để tách đặc trưng. Hog là thuật toán dựa vào biến thiên gradient theo 2 phương đứng và ngang. Giữ lại những đặc điểm biến thiên theo hai phương này
* Mô hình huấn luyện: gồm 3 lớp: 1 lớp đầu vào , 1 lớp đàu ra và 1 lớp ẩn (có thể thay đổi ). Lớp đầu vào gồm sau khi qua HOG được vector đặc trưng theo HOG. Lớp đầu ra với 3 đầu ra là xác xuất của các class
* Mô hình tối ưu; là mô hình sau khi train test và quá trình điều chỉnh từu dữ liệu đến chính mô hình ta được một mô hình tối ưu cho kết quả tốt nhất có thể dựa vào
* Bảng training : giúp ta ghi lại kết quả của từng lần test với sự thay đổi và điều chỉnh thông số cũng như dữ liệu. Việc này giúp ta tìm ra mô hình tối ưu
* Ảnh testing: là ảnh chứa nhiều classes cần phân biệt, nói cách khác chính là đầu vào của hệ thống khi hoàn thiện.
* Đầu ra : là xác xuất hay các classes mà chúng ta muốn xác định

Đặc trưng các function trong mạng neural :

1. Hàm kích hoạt Sigmod : Sigmod là hàm sử dụng
2. Early stopping : để tránh việc mô hình gặp vấn đề overfitting, hàm sử dụng thông số validation để khống chế việc mô hình chúng ta học một các overfitting
3. Solfmax