**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KỸ THUẬT MÁY TÍNH**

**DƯƠNG MINH TÂM – 18521367**

**LÊ TRƯỜNG THỊNH - 18521438**

**BÁO CÁO ĐỒ ÁN MÔN HỌC**

**THIẾT KẾ VÀ HIỆN THỰC MẠNG**

**NƠ RON TÍCH CHẬP TRÊN FPGA**

**TP. HỒ CHÍ MINH, 2021**

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA KỸ THUẬT MÁY TÍNH**

**DƯƠNG MINH TÂM – 18521367**

**LÊ TRƯỜNG THỊNH - 18521438**

**BÁO CÁO ĐỒ ÁN MÔN HỌC**

**THIẾT KẾ VÀ HIỆN THỰC MẠNG**

**NƠ RON TÍCH CHẬP TRÊN FPGA**

**GIẢNG VIÊN HƯỚNG DẪN**

**ThS. Trương Văn Cương**

**TP. HỒ CHÍ MINH, 2021**

MỤC LỤC

[Chương 1. GIỚI THIỆU TỔNG QUAN ĐỀ TÀI 1](#_Toc69894840)

[1.1. Đặt vấn đề 1](#_Toc69894841)

[1.2. Hướng nghiên cứu đề tài 3](#_Toc69894842)

[1.3. Mục tiêu của đề tài 3](#_Toc69894843)

[Chương 2. CƠ SỞ LÝ THUYẾT 4](#_Toc69894844)

[2.1. Convolutional Neural Network – Mạng Nơron tích chập 4](#_Toc69894845)

[2.1.1. Convolutional Layer – Lớp tích chập 5](#_Toc69894846)

[2.1.2. Pooling Layer – Lớp tổng hợp 5](#_Toc69894847)

[2.1.3. Fully Connected Layer – Lớp kết nối đầy đủ 6](#_Toc69894848)

[2.1.4. Các hàm kích hoạt thường dùng 7](#_Toc69894849)

# GIỚI THIỆU TỔNG QUAN ĐỀ TÀI

## Đặt vấn đề

Máy học hiện đang đóng vai trò quan trọng trong đời sống con người, phương pháp đang rất phổ biến và có nhiều ứng dụng là mạng nơ-ron tích chập (CNN). CNN được xây dựng dựa trên ý tưởng mô phỏng lại các tế bào nơ ron trong não người, và được ra mắt lần đầu vào năm những năm 80 của thế kỉ XX bởi Kunihiko Fukushima, Sei Miyake, Takayuki Ito [1]. Kể từ đó, các mô hình CNN ngày càng được phát triển với số lượng tham số của mạng ngày càng dần và độ chính xác càng cao, điển hình như: Lenet – 1997 [2] với 60 ngàn trọng số, AlexNet - 2012 [3] với 60 triệu trọng số, VGGNet -2014 [4] 138 triệu trọng số, GoogleNet - 2014 [5] với 4 triệu trọng số. Về cơ bản, kiến trúc CNN bao gồm 4 thành phần chính: Convolution layer, Pooling layer, Activation ReLU và Fully Connected layer được ghép nối với nhau theo cấu trúc nhiều tầng.

Ngày nay, CNN được ứng dụng nhiều trong lĩnh vực thị giác máy tính do khả năng trích xuất tốt đặc trưng đối tượng, phù hợp với các bài toán phát hiện và nhận diện đối tượng trên ảnh. Tuy nhiên, do số lượng tham số cần tính toán trong CNN khá lớn đòi hỏi tài nguyên với tốc độ xử lý cao, người ta thường chọn GPU là phần cứng để huấn luyện và hiện thực mạng. Mặc dù vậy, GPU lại có những điểm hạn chế như: tiêu tốn nhiều năng lượng sử dụng, chi phí cao và tốc độ xử lý chậm hơn so với FPGA. FPGA có khả năng cung cấp hàng ngàn đơn vị bộ nhớ cho việc tính toán giúp tăng tốc độ xử lý, giảm chi phí tính toán. Bên cạnh đó, FPGA còn có khả năng tái lập trình, cho phép tối ưu về mặt thông lượng và sử dụng ít năng lượng hơn. Chính nhưng điểm nổi bật trên khiến cho FPGA trở nên thích hợp trong các ứng dụng nhúng. Với sự phát triển của FPGA như ngày nay và sự hỗ trợ nhiệt tình về phần mềm, mã nguồn mở từ các nhà sản xuất, chúng ta có thể dễ dàng xây dựng mô hình CNN trên FPGA cho các ứng dụng xử lý thời gian thực.

Hiện nay, các mô hình CNN được triển khai chủ yếu tập trung xử lý trên server với GPU cấu hình cao, dữ liệu hình ảnh từ các camera sẽ được truyền về server để xử lý. Cũng chính vì vậy mà thời gian để truyền dữ liệu từ camera đến server và ngược lại sẽ dài và phụ thuộc vào băng thông mạng. Ngoài ra, nếu có quá nhiều tác vụ cần xử lý cùng một lúc, server có thể sẽ bị quá tải, ảnh hướng đến các tác vụ khác.

Năm 2015, các tác giả C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao và J. Cong công

bố nghiên cứu về đề tài mang tên “Optimizing FPGA-based Accelerator Design for

Deep Convolutional Neural Networks”, nhưng trong nghiên cứu, nhóm tác giả chỉ

hiện thực lớp tích chập trên FPGA. Năm 2016, các tác giả Zhiqiang Liu, Yong Dou,

Jingfei Jiang và Jinwei Xu đến từ National Laboratory for Parallel và Distributed

Processing National University of Defense Technology đã công bố bài báo mang

tên “Automatic Code Generation of Convolutional Neural Networks in FPGA

Implementation”. Tuy nhiên, các tác giả xây dựng mạng CNN có thể tái cấu hình

trên nền tảng FPGA dựa trên dữ liệu số fixed-point – fixed-point có tầm biểu diễn

số khá hẹp nhưng các phép tính đơn giản vì giống như phép tính trên số nguyên. Để

khắc phục các nhược điểm trên và kết hợp các ưu điểm của các nghiên cứu trên,

trong đề tài này, nhóm thực hiện xây dựng mô hình CNN theo cách tiếp cận HDL ở

nền tảng FPGA, có khả năng tái cấu hình mạng và sử dụng các phép tính trên

Floating-point để có tầm biểu diễn số rộng hơn." sửa thành "Đề hướng đến việc xử lý dữ liệu ngay trên camera trước khi truyền về server và khắc phục những hạn chế của GPU, năm 2015, C. Zhang và cộng sự đã đề xuất phương pháp “Optimizing FPGA-based Accelerator Design for

Deep Convolutional Neural Networks”, tuy nhiên trong bài báo này, nhóm tác giả chỉ

hiện thực lớp tích chập trên FPGA. Năm 2016, các tác giả Zhiqiang Liu, Yong Dou,

Jingfei Jiang và Jinwei Xu đến từ National Laboratory for Parallel và Distributed

Processing National University of Defense Technology đã công bố bài báo mang

tên “Automatic Code Generation of Convolutional Neural Networks in FPGA

Implementation”. Tuy nhiên, các tác giả xây dựng mạng CNN có thể tái cấu hình

trên nền tảng FPGA dựa trên dữ liệu số fixed-point – fixed-point có tầm biểu diễn

số khá hẹp nhưng các phép tính đơn giản vì giống như phép tính trên số nguyên. Để

khắc phục các nhược điểm trên và kết hợp các ưu điểm của các nghiên cứu trên,

trong đề tài này, nhóm thực hiện xây dựng mô hình CNN theo cách tiếp cận HDL ở

nền tảng FPGA, có khả năng tái cấu hình mạng và sử dụng các phép tính trên

Floating-point để có tầm biểu diễn số rộng hơn.

## Hướng nghiên cứu đề tài

Dựa trên ý tưởng phát triển mạng CNN hoạt động như một end-device, xử lý ngay sau khi thu thập dữ liệu từ camera, nhóm đề xuất một lõi IP CNN, có thể tự động tạo mã Verilog dựa trên các tham số đầu vào như (kích thước ảnh đầu vào, kích thước của bộ lọc, số bộ lọc, số nốt đầu ra, ...) và xử lý trên số dấu chấm động có độ chính xác đơn theo chuẩn IEEE 754.

Trọng số của mạng được huấn luyện trước trên GPU và sau đó được trích xuất để nạp vào FPGA. Cuối cùng, thiết kế sẽ được tổng hợp và nạp vào board FPGA để thực thi.

## Mục tiêu của đề tài

Trong khoá luận này, đầu tiên, nhóm sẽ nghiên cứu, phân tích và tìm hiểu lý thuyết về các thành phần của mô hình CNN. Tiếp theo, nhóm tiếp tục nghiên cứu, phân tích và thiết kế IP có thể hiện thực được một mô hình CNN cơ bản (giới hạn với một lớp Fully Connected), sử dụng ngôn ngữ Verilog thiết kế theo kiến trúc đường ống (pipeline). Sau cùng, hiện thực thiết kế trên board FPGA Virtex 7 VC707 bằng cách tạo mạng nhận diện chữ số viết tay (MNIST) cho mục đích mô phỏng, kiểm tra và so sánh kết quả xử lý trên Python.

# CƠ SỞ LÝ THUYẾT

## Convolutional Neural Network – Mạng Nơron tích chập

Mạng nơron tích chập được lấy cảm hứng từ vỏ não thị giác. Mỗi khi chúng ta nhìn thấy một vật, một loạt các lớp tế bào thần kinh được kích hoạt, và mỗi lớp thần kinh sẽ phát hiện một tập hợp các đặc trưng như đường thẳng, cạnh, màu sắc, ... của đối tượng, lớp thần kinh càng cao sẽ phát hiện các đặc trưng phức tạp hơn để nhận ra những gì chúng ta đã thấy. Mạng nơron tích chập là mạng bao gồm một lớp đầu vào và một lớp đầu ra, ngoài ra, ở giữa còn có nhiều lớp ẩn (hidden layers). Các lớp ẩn của mạng thường bao gồm nhiều lớp tích chập (convolutional), lớp tổng hợp (pooling), lớp kết nối đầy đủ (Fully Connected) và theo sau các lớp ẩn là các hàm kích hoạt (ReLU, Softmax, …).



Hình .: Kiến trúc mạng CNN VGG16

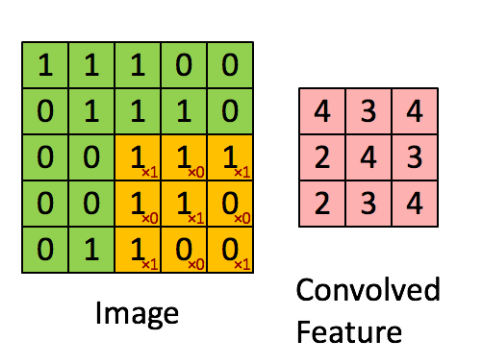
Các mạng nơron tích chập nổi bậc như là LeNet - một trong những mạng CNN lâu đời nổi tiếng nhất được Yann LeCun phát triển vào năm 1998, AlexNet - mạng CNN đã dành chiến thắng trong cuộc thi ImageNet LSVRC-2012 năm 2012 với large margin (15.3% VS 26.2% error rates), VGGNet (Hình 2.1) - có tỉ lệ sai (error rate) nhỏ hơn AlexNet trong ImageNet Large Scale Visual Recognition Challenge (ILSVRC) năm 2014, ResNets phát triển bởi Microsoft vào năm 2015, thắng giải ImageNet ILSVRC competition 2015 với tỉ lệ sai 3.57%, …

### Convolutional Layer – Lớp tích chập

Mỗi lớp tích chập trong mạng nơron sẽ có đầu vào là một tensor với kích thước (Số ảnh đầu vào) x (chiều rộng ảnh) x (chiều cao ảnh) x (chiều sâu của ảnh). Lớp tích chập sẽ thực hiện nhân chập với một kernel (ma trận lọc), chiều dài và rộng của kernel sẽ được chọn theo dạng tham số, còn chiều sâu thì phải bằng với chiều sâu của ảnh, sau đó, kết quả của lớp này sẽ được truyền sang lớp tiếp theo.

Để tính tích chập ta cần quan tâm đến hai thông số là Stride và Padding. Kernel sẽ được trượt qua ảnh, với mỗi lần trượt ta sẽ tính kết quả tại vị trí đầu ra. Số bước ở mỗi lần trượt kernel được gọi là Stride, thông thường, trong tích chập, Stride bằng 1. Nếu tích chập Same Padding, sẽ tạo các điểm ảnh bằng 0 ở rìa của ảnh, đặc trưng đầu ra sẽ có kích thước không đổi so với ảnh đầu vào. Còn Valid Padding, đặc trưng đầu ra sẽ giảm K - 1 so với ảnh đầu vào, với K là kích thước kernel.

Hình 2.2 là ví dụ về nhân tích chập 2 chiều trên ảnh kích thước 5x5, kích thước kernel là 3x3, stride bằng 1 và Valid Padding. Sau đó, đặc trưng thu được có kích thước 3x3.



Hình .: Ví dụ về tích chập 2 chiều Stride 1, Valid Padding

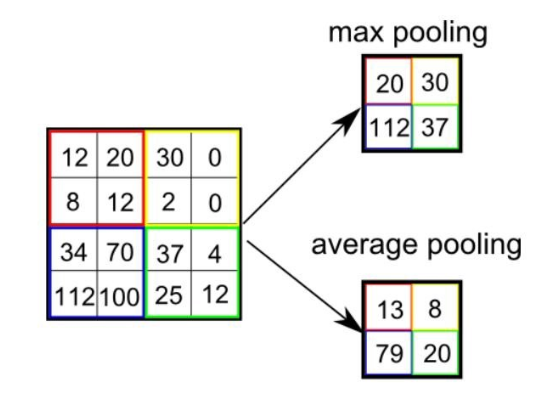
### Pooling Layer – Lớp tổng hợp

Lớp tổng hợp chịu trách nhiệm giảm kích thước không gian của đặc trưng sau khi tích chập. Nhờ đó, lớp tổng hợp giúp giảm số lượng tính toán của dữ liệu. Hơn nữa, lớp này rất hữu ích trong việc lọc ra những đặc trưng “trội”, giúp cho model được huấn luyện tốt và dễ hơn. Lớp tổng hợp cũng có hai thông số Stride và Padding, Stride được định nghĩa giống với Stride ở lớp tích chập. Với Padding Same, kích thước đầu ra sẽ được xác định với công thức (2.1), với Padding Valid, kích thước đầu ra sẽ được xác định với công thức (2.2), gọi H, W là kích thước đầu vào, H’, W’ là kích thước đầu ra, K là kích thước của cửa sổ trượt.

(.)

(.)

Có hai loại Pooling là Max Pooling (trả về giá trị lớn nhất trong cửa sổ trượt) và Average Pooling (trả về giá trị trung bình trong cửa sổ trượt). Hình 2.3 là ví dụ của Max Pooling và Average Pooling với cửa sổ trượt kích thước 2x2, đầu vào có kích thước 4x4, stride bằng 2.



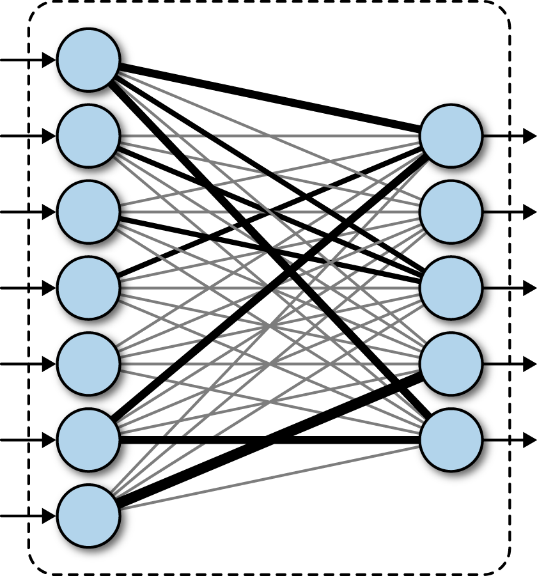
Hình .: Lớp Pooling khi Stride bằng 2

### Fully Connected Layer – Lớp kết nối đầy đủ

Sau khi thực hiện các lớp tích chập và tổng hợp, mạng CNN thường kết thúc bằng lớp kết nối đầy đủ - kết nối tất cả nốt của lớp này với tất cả nốt của lớp khác. Lớp này thường dung trong mục đích để phân loại ảnh và để dữ liệu dữ đoán có thể hiểu được, hàm kích hoạt Softmax sẽ được áp dụng để đưa dữ liệu về dạng xác suất.

Đầu vào của lớp sẽ được “làm phẳng” thành ma trận một chiều, tất cả các nốt của ma trận sẽ được nhân với một trọng số để tạo ra một nốt đầu ra. Lớp này được tính bằng công thức (2.3), với N là số nốt đầu ra. Hình 2.4 mô tả lớp kết nối đầy đủ với 7 nốt đầu vào và 5 nốt đầu ra.

(.)



Hình .: Lớp kết nối đầy đủ với 7 nốt đầu vào và 5 nốt đầu ra.

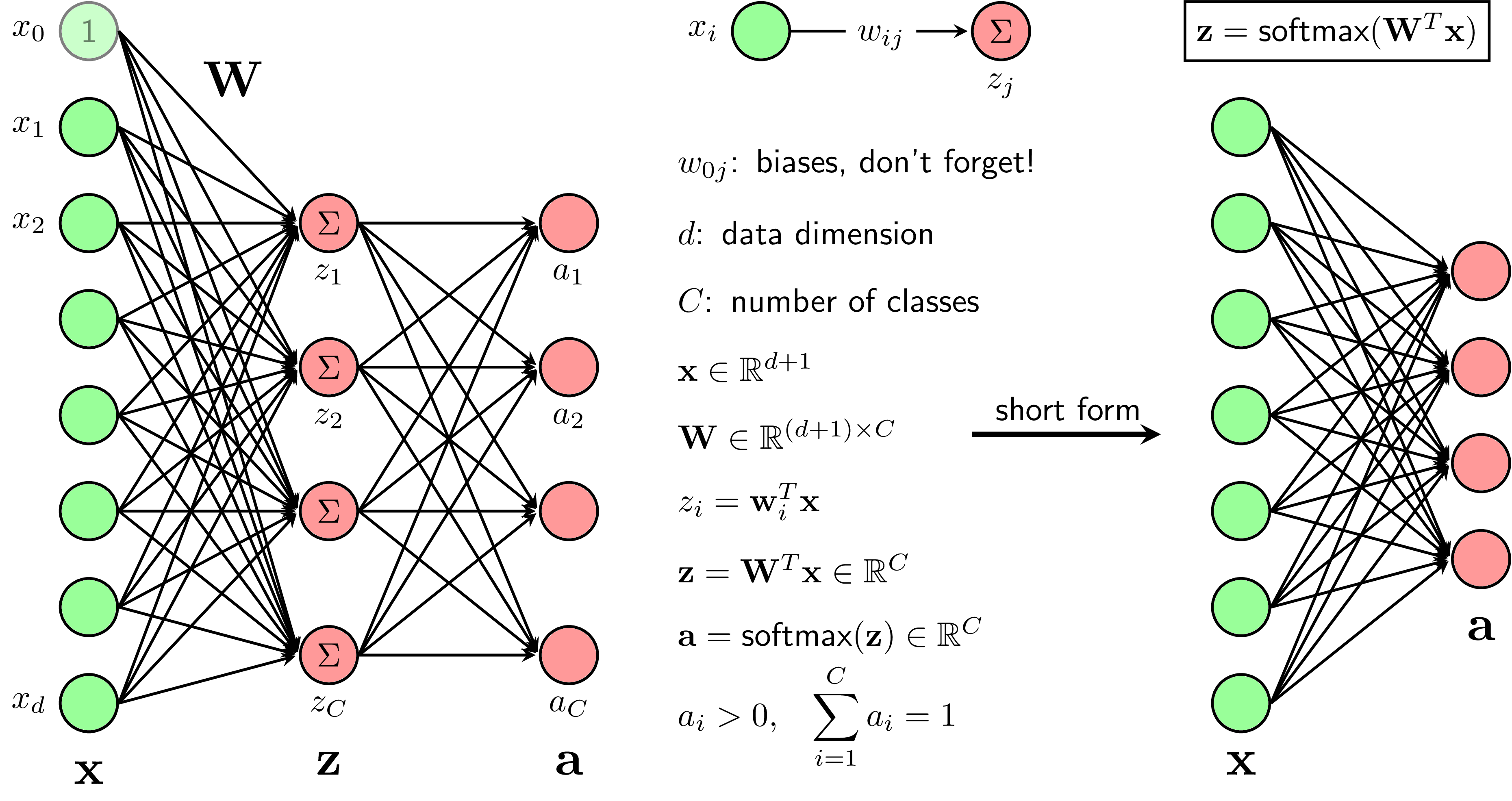
### Các hàm kích hoạt thường dùng

#### ReLU

ReLU, đầy đủ là Rectified Linear Unit, loại bỏ các giá trị âm bằng cách đặt nó về 0, làm tăng các đặc trưng phi tuyến tính của mạng tổng thể mà không ảnh hưởng trực tiếp đến lớp tích chập. Công thức đầy đủ của hàm kích hoạt ReLU được thể hiện ở (2.4).

f(x) = max(0, x) (.)

#### Softmax

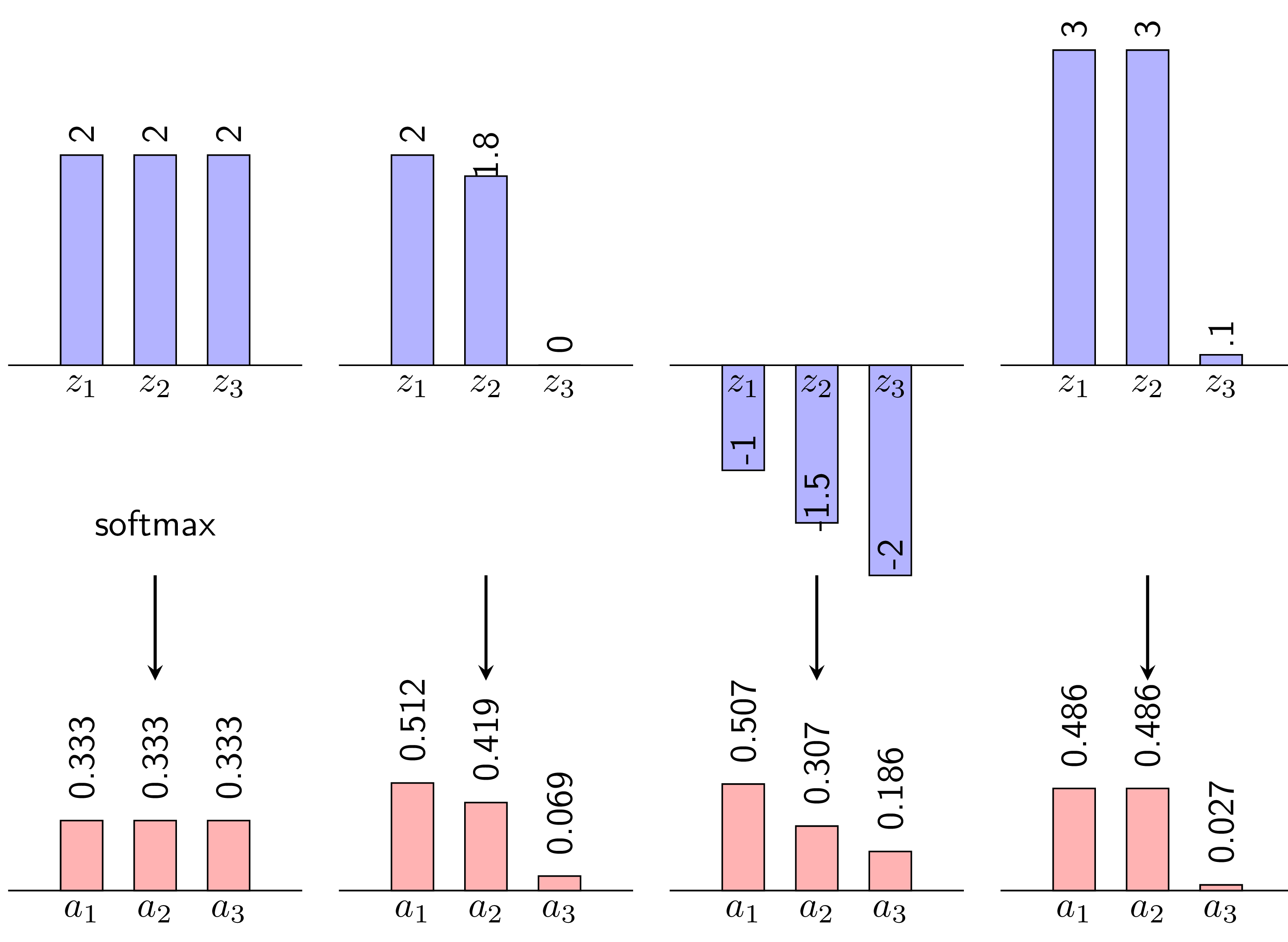


Hình .: Hàm kích hoạt Softmax kết nối với lớp kết nối đầy đủ

Hàm kích hoạt Softmax thường được dùng sau khi thực hiện FC Layer, mục đích để giá trị được chuẩn hoá dưới dạng xác suất, Hình 2.5 mô tả lớp kết nối đầy đủ và theo sau là hàm kích hoạt Softmax.

Công thức tổng quát của hàm kích hoạt Softmax, với là kết quả đề cập ở trên, là xác suất dự đoán của từng lớp được thể hiện như công thức (2.5).

(.)



Hình .: Ví dụ về kết quả của hàm kích hoạt Softmax

Với những đầu vào bằng nhau, xác suất sẽ bằng nhau, càng lớn sẽ cho xác suất càng lớn, càng bé sẽ cho xác suất càng bé. Hình 2.6 minh hoạ cơ bản ví dụ về Softmax.