Chương 2: LÝ THUYẾT NỀN TẢNG

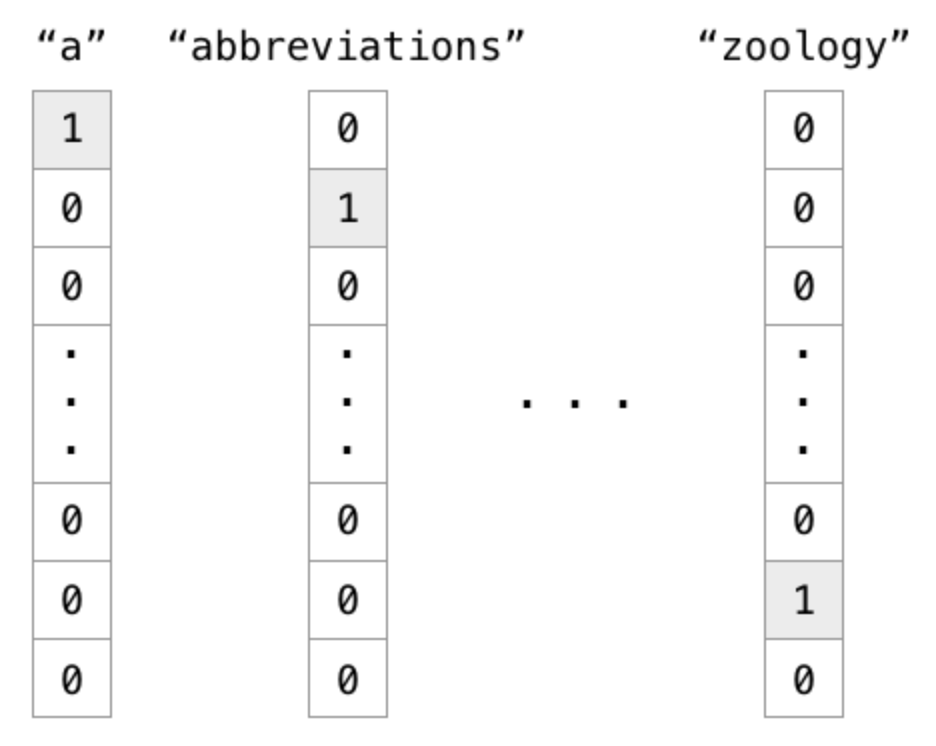
1. Lý thuyết nền tảng của dịch máy:
   1. Định nghĩa:

Phần này trình bày một số khái niệm cốt lõi về dịch máy, các mô hình ngôn ngữ và phương pháp học theo đặc trưng trong xử lý ngôn ngữ tự nhiên(NLP).

* + 1. Định nghĩa dịch máy:

Dịch máy (machine translation) là một quá trình thay đổi văn bản từ ngôn ngữ này sang ngôn ngữ khác (gọi là ngôn ngữ đích) một cách tự động, không có sự can thiệp của con người trong quá trình dịch.

* + 1. Word embeddings:
* Xử lý đầu vào cho bài toán dịch máy là một bước rất quan trọng, các thuật toán, vì các kiến trúc Machine learning, Deep learning chúng chỉ có thể hiểu được đầu vào ở dạng là số nên cần chuyển đầu vào ở dạng text sang dạng số để chúng có thể hiểu được.
* Nhưng nếu chỉ đơn giản biểu diễn từ bằng một con số có thể dẫn đến sai lệch mối quan hệ ngữ nghĩa giữa các từ.Ví dụ như nếu đánh dấu “mèo” là số 1 và “chó” là số 2, như vậy “mèo” + “mèo” = “chó”.
* Một kỹ thuật đơn giản được sử dụng để khác phục là One-hot vector, chúng chuyển các từ thành vector có số chiều bằng số từ của bộ từ vựng đầu vào, trong đó chỉ có duy nhất một phần tử bằng 1 (các phần tử khác bằng 0) tương ứng với vị trí từ đó trong bộ từ vựng. Tuy nhiên cách biểu diễn này là số chiều của vector lại rất lớn, ảnh hưởng đến quá trình xử lý và lưu trữ.



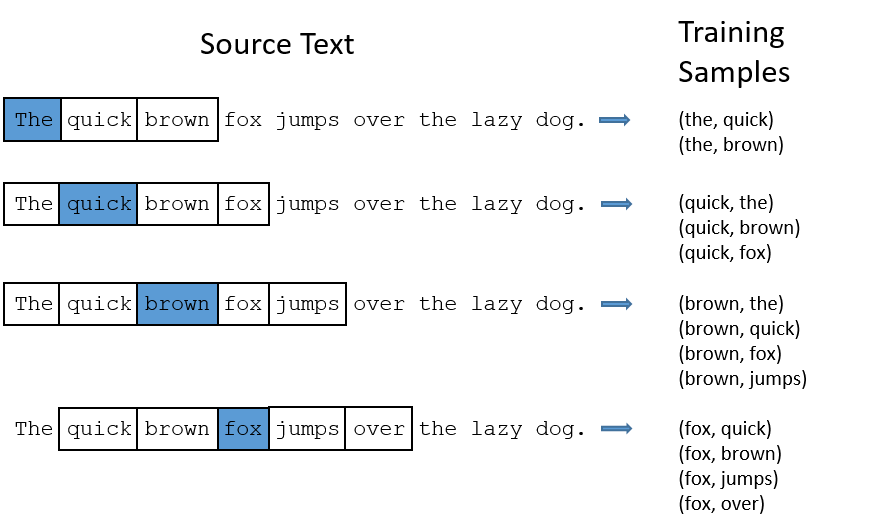
Hình 1: Hình mô tả cách mã hóa one-hot-vector

* Một cách khác là xử dụng vector ngẫu nhiên, mỗi từ được biểu thị bằng một vector có giá trị các chiều là ngẫu nhiên, mỗi từ là một điểm trong không gian 3D, do đó làm giảm số chiều vector, tuy nhiên nó lại không biểu diễn quan hệ tương đồng giữa các từ.
* Sử dụng **Word embeddings** được coi là cách tốt nhất để thể hiện các từ trong văn bản nó cũng gán mỗi từ với một vector nhưng các vector được tính toán để biểu diễn quan hệ tương đồng giữa các từ.
* Word embeddings có 2 model nổi tiếng là Word2vec và Glove.
  + 1. Word2vec:

Word2vec là một model unsupervised learning nó dùng để thể hiện mỗi quan hệ giữa các từ, nó được kết hợp từ hai thuật toán Skip-gram và Continuous bag of words (CBOW).

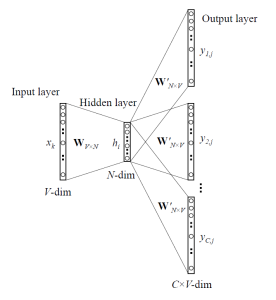
* + - 1. Skip-gram:

Ý tưởng chính của mô hình này là xác định các từ xung quanh từ mục tiêu trong một khoảng nhất định gọi là ‘window’.



Hình 2: Hình Mô Tả Trainning Với Window Bằng 2. (Nguồn: Leonardo Barazza)

Đối với Skip-gram, đầu vào là từ đích, trong khi đầu ra là các từ xung quanh từ đích. Tất cả dữ liệu đầu vào và đầu ra có cùng kích thước được mã hóa bằng one-hot. Mạng chứa một lớp ẩn có kích thước bằng kích thước nhúng, nhỏ hơn vector đầu vào và đầu ra. Ở cuối lớp đầu ra, một hàm kích hoạt softmax được áp dụng sao cho mỗi phần tử của vector đầu ra mô tả khả năng một từ cụ thể sẽ xuất hiện trong ngữ cảnh.



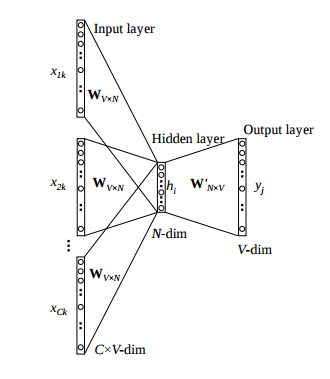
Hình 3: Hình mô tả cấu trúc mạng của Skip-gram

(Nguồn <https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/>)

Với skip-gram, kích thước biểu diễn từ giảm từ kích thước bằng số từ trong bộ từ vựng xuống bằng chiều dài lớp ẩn. Hơn nữa các vector có ý nghĩa nhiều hơn về mặt mô tả mối quan hệ giữa các từ.

* + - 1. Continuous bag of words (CBOW).

Ngược lại với Skip-gram nó hoán đổi đầu vào và đầu ra, ý tưởng của thuật toán CBOW là đưa ra một bối cảnh và cho biết từ nào có khả năng xuất hiện nhiều nhất trong đó.



Hình 4: Mô Tả Mạng CBOW

(Nguồn <https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/>)

Đối với CBOW, tất cả các ví dụ với từ đích là mục tiêu được vào mạng, sau đó lấy trung bình trong lớp ẩn. Giả sử chúng ta chỉ có hai câu, “Anh ấy là một chàng trai tốt” và “Cô ấy là một cô gái tốt”. Để tính toán đại diện cho từ “một”, chúng ta cần cung cấp hai ví dụ này cho mạng nơ-ral và lấy giá trị trung bình giá trị trong lớp ẩn.

* + 1. Greedy Search và Beam Search (Thuật toán tìm kiếm tham lam và thuật toán tìm kiếm chùm tia)

Thuật toán Greedy Search chọn một ứng cử viên tốt nhất làm chuỗi đầu vào cho mỗi bước thời gian(ứng cử viên có xác suất cao nhất). Chọn chỉ một ứng cử viên tốt nhất có thể phù hợp với bước thời gian hiện tại, nhưng khi xây dựng câu đầy đủ, nó có thể là một lựa chọn tối ưu. Tuy nhiên thì xác suất cao nhất ở bước hiện tại chưa chăc sẽ cho ra xác xuất cao nhất ở bước tiếp theo, vậy nên thay vì chỉ giữ 1 kết quả có xác xuất cao nhất chúng ta sẽ giữ lại k kết quả có xác suất cao nhất và đó chính là Beam Search.

Thuật toán Beam Search chọn nhiều lựa chọn thay thế cho chuỗi đầu vào tại mỗi dấu thời gian dựa trên xác suất có điều kiện. Số lượng nhiều lựa chọn thay thế phụ thuộc vào một tham số gọi là Beam Width B. Ở mỗi bước thời gian, tìm kiếm chùm tia chọn B số lựa chọn thay thế tốt nhất với xác suất cao nhất là lựa chọn khả dĩ nhất cho bước thời gian(ví dụ với Beam Width B = 3 thì tại mỗi bước thời gian sẽ chọn ra 3 ứng cử viên có xác suất cao nhất làm đầu vào cho bước thời gian tiếp theo(t). Sau đó lại tiếp tục chọn 3 ứng cử viên ở bước thời gian tiếp theo(t+1) làm đầu vào ở bước thời gian t+2. Cứ như vậy cho đến cuối cùng ta sẽ thu được 3 kết quả và chọn ra kết quả từ đó). Thuật toán Beam Search nếu có Beam Width B = 1 thì nó trở thành thuật toán Greedy Search. Beam Width B = 10 thường được sử dụng và mang lại hiệu quả đủ tốt.

* + 1. Bleu Score

BLEU là một thuật toán để đánh giá chất lượng văn bản đã được dịch bằng máy từ ngôn ngữ tự nhiên này sang ngôn ngữ tự nhiên khác. Chất lượng được coi là sự tương ứng giữa đầu ra của máy và của con người: "bản dịch máy càng gần với bản dịch chuyên nghiệp của con người thì càng tốt" - đây là ý tưởng trung tâm của BLEU.

Bilingual Evaluation Understudy Score hay ngắn gọn là BLEU score là một thang điểm được dùng phổ biến trong đánh giá Machine Translation. BLEU được Kishore Papineni và cộng sự đề xuất lần đầu vào năm 2002 qua bài nghiên cứu "a Method for Automatic Evaluation of Machine Translation".

BLEU được tính dựa trên số lượng n-grams[1] giống nhau giữa câu dịch của mô hình (output) với các câu tham chiếu tương ứng (reference) có xét tới yếu tố độ dài của câu.

Số n-grams tối đa của BLEU là không giới hạn, nhưng vì xét về ý nghĩa, cụm từ quá dài thường không có nhiều ý nghĩa, và nghiên cứu cũng đã cho thấy là với 4-gram, điểm số BLEU trung bình cho khả năng dịch thuật của con người cũng đã giảm khá nhiều nên n-grams tối đa thường được sử dụng là 4-gram.

* 1. Lý thuyết nền tảng mạng nơ-ron (Neural Networds)
     1. Mô Tả mạng nơ-ron:

Mạng nơ-ron là một tập hợp các mô hình toán học được được xây dựng dựa trên tập hợp các nút, được kết nối với các hàm kích hoạt phi tuyến tính cùng các tham số có khả năng học. Mạng nơ-ron hiện là mô hình phổ biến nhất được sử dụng cho các ứng dụng máy học trong một loạt các lĩnh vực như thị giác máy tính, nhận dạng giọng nói, xử lý ngôn ngữ tự nhiên,…

Một tế nào (nút) của mạng nơ-ron là một hàm của tập các trọng số tương ứng với các giá trị đầu vào (inputs) {𝑥0, … , 𝑥𝑁}.

Trong đó:

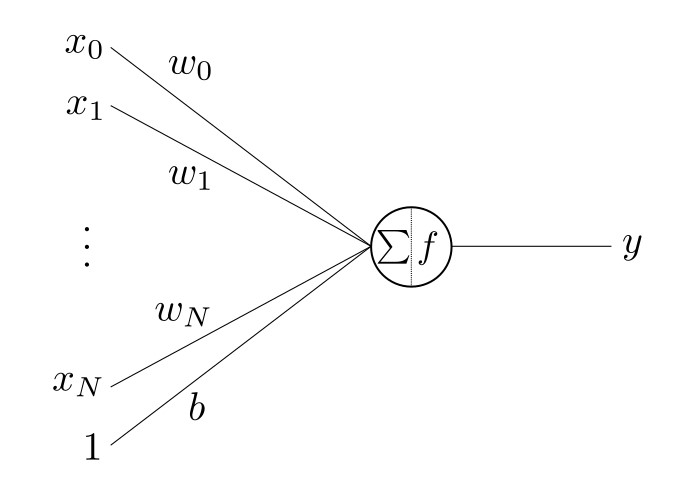
* 𝑤𝑖 : trọng số của đầu vào 𝑥𝑖
* 𝑎: hàm kích hoạt (activation function)
* 𝑏: độ sai lệch (bias)

Ta sẽ sử dụng kí hiệu ma trận để làm đơn giản cách thể hiện, trong đó mỗi tế bào nơ-ron bao gồm một vector đầu vào 𝑥 = {𝑥0, … , 𝑥𝑁}, một vector trọng số 𝑤 = {𝑤0, … , 𝑤𝑁} và một vector sai lệch 𝑏, khi đó đầu ra là:

𝑦 = 𝑎(𝑤𝑇𝑥 + 𝑏)

Nếu hàm kích hoạt a là một biến thể của hàm Heaviside,

thì tế bào nơ-ron này được gọi là một perceptron, một bộ phân loại nhị phân đơn giản, là một trong những phương pháp học kết nối sớm nhất được phát minh bởi Rosenblatt.



Hình 6: Minh họa kiến trúc điển hình của một tế bào mạng nơ-ron

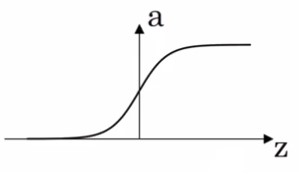
* + 1. Hàm kích hoạt (Activation function)

Hàm kích hoạt là phần rất quan trọng trong mạng nơ-ron, đặc biệt là mạng nơ-ron nhiều lớp ẩn. Nếu không có hàm kích hoạt phi tuyến tính, cho dù mạng nơ-ron có nhiều lớp ẩn đến cỡ nào thì cũng chỉ có sức mạnh đại diện cho phân loại tuyến tính, điều này tương đương với một mạng mà không có lớp ẩn nào. Vì bản chất tổng hợp các hàm tuyến tính là một hàm tuyến tính. Do đó, hàm kích hoạt 𝑎 là một hàm phi tuyến tính được áp dụng cho đầu ra tại mỗi nút và input data cho tầng tiếp theo, cho phép mạng nơ-ron nhiều lớp ẩn học các hàm phi tuyến phức tạp.

Hàm kích hoạt phổ biến là hàm sigmon, nó là một hàm phi tuyến với đầu vào là các số thực cho kết quả nằm trong khoảng từ 0 đến 1, phù hợp cho các mạng phân loại nhị phân (nổi tiếng như là thuật toán Logistic Regression), nó có công thức theo phương trình:

Trong đó:

* 𝑊[𝑖]: trọng số tại lớp thứ i
* 𝑏[𝑖]: tham số sai lệch tại lớp thứ i
* 𝑎: hàm kích hoạt



Hình 7: Minh họa hàm kích hoạt sigmoid. (Nguồn: Coursera Sequence Models)

Ngoài ra, có một hàm kích hoạt luôn hoạt động tốt hơn hàm sigmoid là hàm tiếp tuyến hyperbolic (hyperbolic tangent function), ánh xạ các giá trị đầu vào vào khoảng biên từ -1 đến 1. Có công thức là:

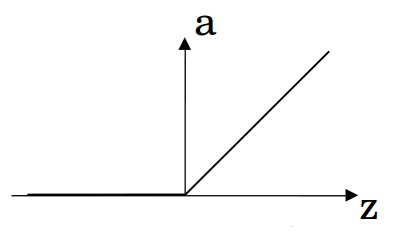
A picture containing table, white, clock, room

Description automatically generated

Hình 8 Minh họa hàm kích hoạt tanh. (Nguồn: Coursera Sequence Models)

Tuy nhiên, một vấn đề với hàm kích hoạt sigmoid và tanh nếu z quá lớn hoặc quá nhỏ thì độ dốc của hàm sẽ rất nhỏ, điều này làm chậm quá trình tìm điểm cực tiểu của hàm chi phí, dẫn đến làm chậm quá trình học. Vì lý do này, dựa vào các kết quả thực nghiệm được cải thiện, mạng nơ-ron hiện đại có xu hướng sử dụng hàm kích hoạt đơn vị tuyến tính chỉnh lưu (ReLu - Rectified Linear Unit). Có công thức như sau:

𝑎 = 𝑚𝑎𝑥 (0,𝑧)



Hình 9: Minh họa hàm kích hoạt ReLu. (Nguồn: Coursera Sequence Models)

Vì vậy, đạo hàm luôn bằng 1 nếu z dương, và bằng 0 nếu z âm. Dựa trên thực nghiệm, sử dụng hàm kích hoạt ReLu, mạng nơ-ron sẽ học nhanh hơn so với khi dùng với hàm sigmoid hoặc hàm tanh. Lý do chính là có ít hơn sự ảnh hưởng của độ dốc hàm bằng 0 làm chậm việc học. Vì mặc dù, có một nửa phạm vi của z làm độ dốc hàm ReLu bằng 0, nhưng trong thực tế, đủ các đơn vị ẩn thì ta sẽ có z lớn hơn 0, vì vậy việc học vẫn khá nhanh với hầu hết các ví dụ đào tạo.

* + 1. Lan truyền ngược (Back propagation)

Các thuật toán học sâu tương phản với các thuật toán học nông bởi số biến đổi được tham số hóa một tín hiệu gặp phải khi nó lan truyền từ các lớp đầu vào đến các lớp đầu ra. Mỗi chuỗi các biến đổi từ đầu vào đến đầu ra gọi là một đường gán kế thừa (CAP - Credit Assignment Path). Vấn đề gán kế thừa (Credit Assignment Problem) được giải quyết với khám phá lan truyền ngược (backpropagation), cho phép học với mạng nơ-ron nhiều lớp. Sau đây, nhóm sinh viên sẽ trình bày ý tưởng của quá trình lan truyền ngược.

Hình 2.4 minh họa một mạng nơ-ron 2 lớp, gồm một lớp đầu vào, một lớp ẩn và một lớp đầu ra.



Hình 2.4: Minh họa một mạng nơ-ron 2 lớp (Nguồn: Coursera Sequence Model)

Lưu ý ở lan truyền tiến, các bước tính toán là như sau: đầu tiên ta tính toán 𝑧[1], sau đó tính toán 𝑎[1], rồi tính toán 𝑧[2] ghi chú là 𝑧[2] cũng phụ thuộc vào các tham số 𝑊[2] và 𝑏[2], sau đó dựa vào 𝑧[2] tính toán 𝑎[2] và cuối cùng là tính toán chi phí. Đối với lan truyền ngược, ta sẽ đi tính toán theo chiều ngược lại, cụ thể là tính toán 𝑑𝑎[2], sau đó tính 𝑑𝑧[2], quay ngược lên tính 𝑑𝑤[2] và 𝑑𝑏[2], tương tự ta tính toán tiếp cho các biến 𝑑𝑎[1], 𝑑𝑧[1], 𝑑𝑊[1], 𝑑𝑏[1]. Thông thường, ta sẽ bỏ qua tính đạo hàm của 𝑑𝑎[2], thay vào đó nhập lại thành một bước là tính trên 𝑑𝑧[2]. Sau cùng ta rút ra các công thức sau:

𝑑𝑎[2] = 𝑎[2] – 𝑦

𝑑𝑊[2] = 𝑑𝑧[2]𝑎[1]𝑇

𝑑𝑏[2] = 𝑑𝑧[2]

𝑑𝑧[1] = 𝑊[2]𝑇𝑑𝑧[2] ∗ 𝑔[1]′(𝑧[1]) 𝑑𝑊[1] = 𝑑𝑧[1]𝑥𝑇

𝑑𝑏[1] = 𝑑𝑧[1]

* + 1. Học với lan truyền ngược

Học với lan truyền ngược giống như quy tắc delta, các độ nhạy được dùng để điều chỉnh trọng số tỷ lệ với một hằng số tỉ lệ học (learning rate) 𝛼. Việc cập nhật trọng số thực hiện theo công thức sau:

A picture containing knife

Description automatically generated

Trong đó:

* δjn: đã được định nghĩa ở công thức
* yi: đầu ra ở nơ-ron i



Hình 2.5 Minh họa sự ảnh hưởng của tỉ lệ học và chính sách học lên độ hội tụ với lan truyền ngược. (Nguồn: [2])

Lan truyền ngược là một phương pháp gốc dốc nhất (steepest descent). Hình 2.5 minh họa quy tắc học với lan truyền ngược, đặc trưng cho kích thước mỗi bước là tham số tỉ lệ học. Tham số tỉ lệ học thay đổi kích thước bước hay độ lớn của vector thay đổi trọng số. Hình 2.5 cũng minh hoạ độ ảnh hưởng của tỉ lệ học trên độ giảm dốc. Tỉ lệ học quá nhỏ làm cho kết quả học rất chậm như a0, tuy nhiên tỉ lệ học quá lớn lại làm cho bước nhảy xung quanh khu vực điểm cực tiểu và làm mất thời gian để tiếp cận điểm cực tiểu này như a2 và a3.

Để đạt được điểm cực tiểu cục bộ, tỉ lệ học cũng nên giảm dần trong quá trình huấn luyện. Tuy nhiên nếu giảm quá nhanh, nó có thể không đạt được lưu vực gần với điểm cực tiêu như a0, ngược lại nếu giảm quá chậm, nó có thể mất một khoảng thời gian dài để tiến vào lưu vực này như a2 và a3.

* + 1. Phương pháp giảm độ dốc với Gradient desent và các biến thể
  1. Các phương pháp huấn luyện mạng nơ-ron hiện đại
     1. Hàm kích hoạt đơn vị tuyến tính chỉnh lưu (Rectified Linear Unit - RLU)
     2. Phương pháp chuẩn hóa theo lô (Batch Nomalization)
     3. Phương pháp cắt giảm (Dropout)
  2. Các kiến trúc mạng nơ-ron hồi quy:
     1. Mạng nơ-ron hồi quy (RNN – Recurrent Neural Network)
     2. Mạng bộ nhớ dài ngắn (LSTM - Long Short Term Memory)
     3. Mạng nơ-ron hồi quy hai chiều (BIRNN - Bidirectional Recurrent Neural Network)
     4. Mạng nơ-ron hồi quy học sâu (Deep RNN – Deep Reccurent Neural Network)
  3. Các kỹ thuật trong dịch máy

Cơ chế Attention (Attention Mechanism)

1. Hệ thống dịch máy

**TÀI LIỆU THAM**

[1] Hovy E.H.: Toward finely differentiated evaluation metrics for machine translation. Proceedings of the Eagles Workshop on Standards and Evaluation, Pisa, Italy, 1999.

[2] Yani Andrew Ioannou. “Structural Priors in Deep Neural Networks” September 2017.