CHƯƠNG II: LÝ THUYẾT NỀN TẢNG

CHƯƠNG 2

* 1. Lý thuyết nền tảng của dịch máy

2.1.1. Định nghĩa

* + - 1. Định nghĩa dịch máy
      2. Word embeddings
         1. Word2vec
         2. GloVe
      3. Beam Search
      4. Bleu Score
    1. Lý thuyết nền tảng mạng nơ-ron(Neural Networks)

### Mạng nơ-ron (Neural Networks)

Mạng nơ-ron là một tập hợp các mô hình thống kê được đặc trưng bởi một tập các nút 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 thống kê phổ biến nhất được sử dụng cho các ứng dụng máy học trong một loạt các lĩnh vực bao gồm: thị giác máy tính, nhận dạng giọng nói, xạ hình y tế, ...

* + - 1. Giải phẫu tế bào mạng nơ-ron

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}.

𝑁

𝑦 = 𝑎 (∑ 𝑤𝑖𝑥𝑖 + 𝑏)

𝑖

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,

𝑎(𝑥) = { 1

𝑛ế𝑢 𝑥 ≥ 0

0 ℎ𝑜ặ𝑐 − 1 𝑛ế𝑢 𝑥 < 0

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

2.3 minh họa kiến trúc điển hình của một tế bào mạng nơ-ron.

A close up of a map

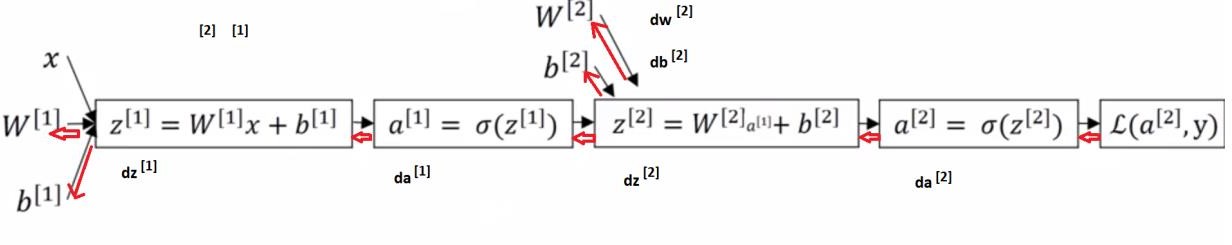
Description automatically generated

Hình 2.3: Minh họa kiến trúc điển hình của một tế bào mạng nơ-ron. (Nguồn: [13])

* + - 1. Lan truyền ngược (Backpropagation)

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:

∆𝑤𝑛 = −𝛼

𝑖𝑗

𝜕𝐸𝑛

𝜕𝑤𝑖𝑗

= −𝛼𝛿𝑛𝑦𝑖

𝑗

Trong đó:

* 𝛿𝑛: đã được định nghĩa ở công thức

𝑗

* 𝑦𝑖 : đầu ra của nơ-ron i

A close up of a wire fence

Description automatically generated

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: [13])

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 họa độ ảnh hưởng của tỉ lệ học trên độ giảm dốc. Tỉ lệ học quá nhỏ làm cho kết quả học rất chậm như 𝛼0, tuy nhiên tỉ lệ học quá lớn lại làm cho bước nhảy

xung quanh khu vực điểm cực tiểu và mất thời gian dài để tiếp cận điểm cực tiểu này như

𝛼2 và 𝛼3.

Để đạt được điểm cực tiểu cục bộ, tỉ lệ học cũng phải được giảm trong quá trình huấn luyện. Tuy nhiên, nếu giảm quá nhanh, nó có thể không bao giờ đạt được lưu vực gần với điểm cực tiểu, như với 𝛼0, ngược lại nếu giảm quá chậm, nó có thể mất một thời gian dài

để tiến vào khu vực này, như với 𝛼3.

Cân bằng việc cố gắng tìm một tỉ lệ học và chính sách học phù hợp không may là một phần "ma thuật đen" đằng sau việc huấn luyện DNN đến từ kinh nghiệm. Tuy nhiên Bottou (2012) và I. Goodfellow, Y.Bengio, và Courville (2016) là tài liệu tham khảo tuyệt vời về một số các tiếp cận phổ biến để làm công việc này dễ dàng hơn.

* + - 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, cho phép mạng nơ-ron nhiều lớp ẩn học các hàm phi tuyến phức tạp.

Trong lĩnh vực mạng nơ-ron, hàm kích hoạt được chọn một cách cổ điển là hàm sigmoid, ánh xạ các giá trị đầu vào vào khoảng biên từ 0 đến 1, phù hợp cho các mạng phân loại nhị phân. Có công thức theo phương trình:

𝑧[𝑖] = 𝑊[𝑖]𝑥 + 𝑏[𝑖]

𝑎 =

1

1 + 𝑒−𝑧

Trong đó:

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

A close up of a lamp

Description automatically generated

Hình 2.6 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 =𝑡𝑎𝑛ℎ (𝑧) =

𝑒𝑧 − 𝑒−𝑧

𝑒𝑧 + 𝑒−𝑧

A close up of a lamp

Description automatically generated

Hình 2.7: 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)

A picture containing object, clock

Description automatically generated

Hình 2.8 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 tới (Forward Propagation)
      2. Lan truyền ngược (Backpropagation)
      3. Phương pháp giảm độ dốc với Gradient desent và các biến thể
      4. Phương pháp giảm độ dốc với động lượng (Momentum)
    1. Các phương pháp huấn luyện mạng nơ-ron hiện đại
       1. Hàm kích hoạt đơn vị tuyết tính chỉnh lưu (Rectified Linear Unit)
       2. Phương pháp chuẩn hoá theo lô (Batch Normalization)
       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 nhơ 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 Recurrent Neural Network)
    3. Các kỹ thuật trong dịch máy
       1. Cơ chế Attention (Attention Mechanism)
  1. Hệ thống dịch máy