Word2Vec tutorial - the skip -gram model

Hướng dẫn này bao gồm skip gram neural network và kết trúc cho word2vec. Ý định với hướng dần này đã bỏ qua hầu hết phần giới thiệu thông thường và hiểu biết sâu sắc về word2vec và nhận được nhiều chi tiết hơn. Đặc biệt tôi ở đây để tìm hiểu sâu hơn và skip gram neural network model

# The model

Skip-gram neural network model thật sự đơn giản đến bất ngờ trong hầu hết các form. Tôi nghĩ có một và tỉnh chỉnh nhỏ và làm lộn xộn các lời giải thích.

Hãy bắt đầu cái nhìn sâu sắc nơi chúng ta đang đi. Word2Vec sử dụng mto65 mẹo bạn có thể đã thấy đâu đó trong machine learning. Chúng ta sẽ huấn luyện một mạng nơ ron đơn giản với một lớp ẩn duy nhất để thực hiện một nhiệm vụ nào đó, nhưng sau đó chúng ta sẽ không sử dụng mạng nơron cho nhiệm vụ mà chúng ta đã huấn luyện! Thay vào đó, mục tiêu thực sự chỉ để học các trọng số của lớp ẩn - chúng ta sẽ thấy rằng các trọng số này thực sự là "từ vectơ" mà chúng ta đang cố gắng để tìm hiểu.

Một nay khác bạn có thể đã thấy mẹo này là chức năng trong học không giám xác. Nơi bạn huấn luyện một trình tự động mã hóa để nén một vector đầu vào trong một lớp ẩn và giải nén nó trở lại nguyên bản trong một lớp đầu ra. Sau khi huấn luyện nó, bạn gỡ bỏ lớp đầu ra (bước giải nén) và chỉ sử dụng lớp ẩn - đó là một mẹo để học các tính năng hình ảnh tốt mà không cần ghi nhãn dữ liệu huấn luyện.

# The fake task

Vì vậy bay giờ chúng ta cần nói về fake task thứ chúng ta sẽ xây dựng sử dụng mạng nơ ron. Và sau đó trở lại để bằng một cách gián tiếp chúng ta tạo ra word vector.

Chúng ta sẽ huấn luyện mạng nơ ron để tiếp tục. cho một đặc tả về từ ơ giữa câu, nhìn vào từ gần đó và chọn một từ ngẫu nhiên. Mạng sẽ nói cho chúng ta biết xác xuất cho mỗi từ in từ vựng của chúng ta gần từ đó.

Khi tôi nói gần, sự thật có 1 cái “window size” để tính toán. Một “wz” điển hình là 5, 5 nghĩa là là lấy đằng trước 5 từ và đằng sau 5 từ( tổng là 10)

Các xác suất đầu ra sẽ liên quan đến khả năng nó có thể tìm thấy mỗi từ từ ở gần từ đầu vào của chúng ta. Ví dụ, nếu bạn huấn luyện từ đầu vào “soviet”, xác xuất đầu sẽ cao hơn cho từ “union” và “russia” hơn là cho từ như “watermelon” và “kangaroo”.

Chúng ta sẽ huấn luyện mạng nơ-ron để làm điều này bằng cách cho vào từng cặp từ trong tài liệu của chúng tôi. Bên dưới là ví dụ cho thấy một vài huấn luyện ví dụ chúng ta lấy từ câu “…”. Tôi đã sử dụng wz nhỏ là 2 cho ví dụ này. Chữ được tô đậm bằng màu xanh là input word.

Mạng sẽ được huấn luyện để học thông kê từ số lần mỗi cặp từ xuất hiện. vì vậy trong ví dụ này mạng có lẽ sẽ lấy nhiều hơn tập huấn luyện (“soviet”,”union”) hơ là (“soviet”,”sasquatch”). Khi quá trình huấn luyện kết thúc, bạn có cho từ “soviet” như là đầu vào, sau đó đầu ra sẽ có xác xuất vao hơn cho “union” hoặc “russia” hơn “sasquatch”

# Model details

Như vậy làm sao nó biểu diễn

Trước hết, bạn biết bạn không thể cho word như là chuỗi vào neural network, vì vậy chúng ta cần trình bày word vào neural network. Để làm được điều này, đầu tiên chúng ta xây dụng một bộ từ vững của word từ tài liệu huấn luyện hãy giả sử chúng ta có 10,000 từ không trùng nhau.

Chúng ta tiến hành trình bày một input word như “ant” như một one-hot vector. Vector này sẽ có 10,000 thành phần (mto65 cho mỗi từ trong từ vựng của chúng ta) và chúng ta sẽ đặt một tại ví trí của “ants” và 0 vào tất cả các vị trí còn lại.

Đầu ra của mạng là một vector đơn chứa mọi từ trong từ vựng, xác xuất sẽ ngẫu nhiên được chọn từ gần trong từ vừng.

Kiến trúc của mạng neural

Khống có activation function trên lớp ẩn, nhưng output neurons sủa dụng softmax. Chúng t sẽ trở lại sau

Khi huấn luyện mạng này trên cặp từ, đầu vào là một on-hot vector biểu diện input word và đầu ra của quá trình huấn luyện cũng là một one-hot vector trình bay output-word. Nhưng khi bận tính toán trên mạng đã được huấn luyện, đầu ra thực sự sẽ là xác xuất phân phối

# Hidden layer

Cho ví dụ của chúng ta, chúng ta sẽ giả sử chúng ta sẹ học word vector với 300 features. Vì vậy hidden-layer sẽ trình bày ma trận weight của 10,000 dòng và 300 loclumn

300 features là cái Google đã sử dụng trên mô hình công bố của họ được huấn luyện trên bộ dự liệu Google news . số feature là một “hyper parameter” do đó bạn phải điều chỉnh trên ứng dụng của bạn ( thử các giá trị khác nhau để tìm ra cái tốt nhất)

Nếu bạn nhìn vào dòng của ma trận weight, nó thật ra là word vectors của chúng ta

Vì vậy mục đích cuối cùng của tất cả là học ma trận hidden layer weight, các lớp đầu ra sẽ chi tạo ra khi chúng ta hoàn thành.

Quay lại, mặc dù để làm việc thông qua định nghĩa của mô hình này chúng ta sẽ bắt huấn luyện.

Bây giờ có lẽ bạn tự hỏi ràng “ nó là one-hot vector gần như toàn là số 0 hiệu quả của nó là gì” nếu bạn nhân nó 1\*10000 one-hot vector với ma trận 10,000x300 , nó sẽ hiệu quả chỉ là chọn một dòng trong toàn bộ ma trận.

…

Điều này nghĩa là hidden layer của mô hình này thật ra chỉ hoạt động như một bảng tra cứu. đầu ra của hidden layer chỉ là một “word vector” cho input word

# Output layer

1x300 word vector cho từ “ants” sau đó cho vào vào output layer. Output layer là mto65 oftmax regression classifier. Có một hướng dẫn chuyên sâu cho softmax regression tại đây, nhưng ý chính của nó là mỗi nơ ron đầu ra ( một trên từ trong từ vựng của chúng ta) sẽ tạo ra một output giữa 0 và 1 và tổng của tất cả sẽ lên đến 1.

Đặc biệt, mỗi output neuron có một weight vector cái nó nhân một lần nữa với vector từ lớp ẩn, sau đó áp dụng hàm exp để có kết quả. Cuối cùng kết quả đầu ra sẽ được tổng hợp lên đến 1, chúng ta sẽ chia kết quả này bằng tổng kết quả từ 10,000 output nodes.

Chú ý ràng neural network không biết bất kì thứ gì về vị trí của từ liên quan tới input. Nó không tìm hiểu một tập khác của xác suất cho từ trước khi đầu vào so với từ sau. Để hiểu hàm ý, hãy giả định ràng văn thể của chúng ta, tất của cùng xuất hiện độc lập của từ ‘York’ thì được đứng trước bời từ ‘New’. Điều này có nghĩa, là tất cả theo tập huấn luyến, có 100% xác xuất ràng từ ‘New’ sẽ đứng trước ‘York’. Tuy nhiên, néo chúng ta lấy 10 từ trong vùng lân cận của từ York và ngẫu nhiên chọn một trong số chúng, xác xuất của ‘New’ không còn là 100%; bạn có thể chọn mto65 từ khác trong vùng lân cận.

# Intuition

Ok, bạn đã sẵn sàn để đi sâu hơn chưa ?

Nếu 2 từ khách biệt có ngữ cảnh gần nhau ( thường xuyên suất hiện cạnh nhau) sau đó mô hình của chúng ta cần cho ra output gần giống nhau kết quả cho 2 từ. một cách để mạng đưa ra các tiên đoán ngữ cảnh tương tự cho hai từ này là nếu các vectơ từ tương tự nhau. Vì vậy, nếu 2 từ có cùng ngữ cảnh thì chúng ta có động cơ để học 2 từ này

Và cái gì là là cho 2 từ có ngữ cảnh giống tương tự nhau. Tôi nghĩ bạn có tưởng mong đợi từ động nghĩa như “intelligent” và “smart” có thể có ngữ cảnh tương tự nhau, hoặc 2 từ đó liên quan, như “engine” và “transmission” cũng có thể có những ngữ cảnh tương tự.

Điều này cũng có thể xử lý bắt nguồn cho bạn – the network will likely learn similar word vectors for the words “ant” and “ants” because these should have similar contexts.