

Rekursywne sieci neuronowe (analiza sentymentu)

Agnieszka Ławrynowicz

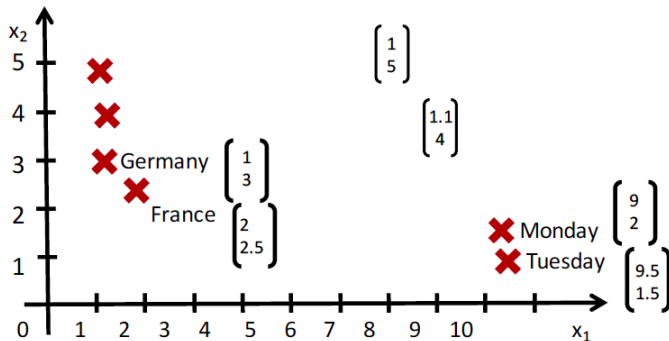
Wydział Informatyki Politechniki Poznańskiej

7 maja 2019

Rekursywne sieci neuronowe

Jak możemy reprezentować dłuższe zdania?

Poprzez odwzorowanie ich do tej samej przestrzeni wektorowej



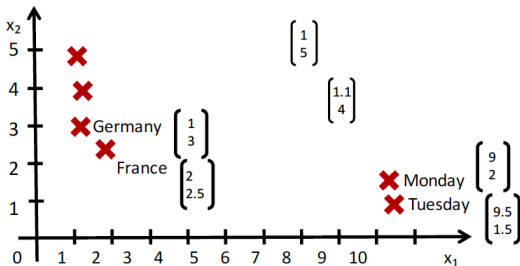
the country of my birth
the place where I was born

Jak możemy odwzorować frazy do przestrzeni wektorowej?

Zasada kompozycyjności

Znaczenie (wektor) zdania jest zdeterminowany przez:

- 1 znaczenie jego wyrazów
- 2 reguły, które je łączą.



the country of my birth
the place where I was born

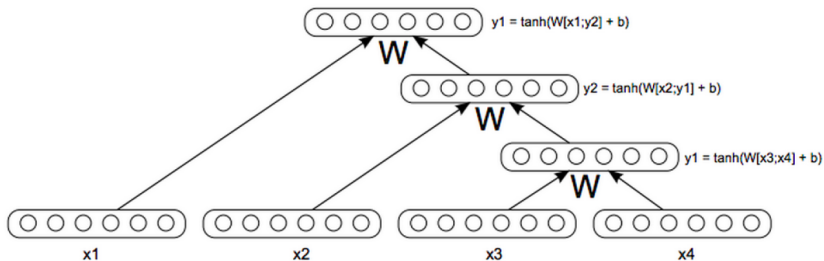
Rekursywne sieci neuronowe

- nadzbiór wcześniej omówionych rekurencyjnych sieci neuronowych (RNN)
- idealne w sytuacjach, gdzie występuje zagnieżdżona hierarchia i rekursywna wewnętrzna struktura

Rekursywne sieci neuronowe c.d.

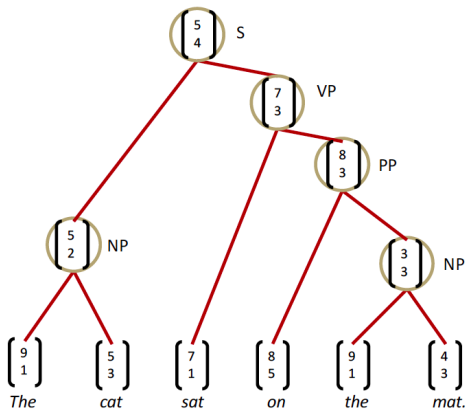
- skupienie się na uczeniu się 'kompozycyjnych' reprezentacji:
 - hierarchicznej struktury, cech i predykcji
 - różnych kombinacji: celu uczenia, funkcji kompozycji, struktury drzewiastej

Standardowa rekursywna sieć neuronowa



źródło: Richard Socher i inni. Kurs *CS224d: Deep Learning for Natural Language Processing*, Uniwersytet Stanforda.

Uczenie się struktury i reprezentacji



źródło: Richard Socher i inni. Kurs CS224d: Deep Learning for Natural Language Processing, Uniwersytet Stanforda.

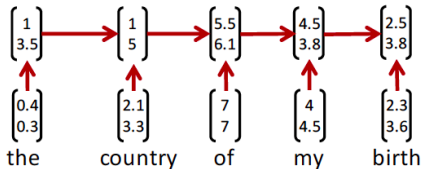
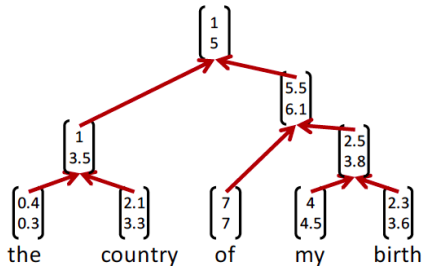
Sieć rekursywna - definicje

$$score = U^T p$$

$$p = \tanh(W \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} + b^{(1)})$$

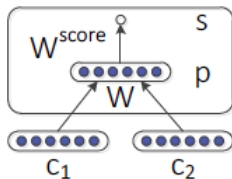
Takie same wagi W we wszystkich węzłach drzewa

Sieć rekursywna versus sieć rekurencyjna



źródło: Richard Socher i inni. Kurs *CS224d: Deep Learning for Natural Language Processing*, Uniwersytet Stanforda.

Prosta jednowarstwowa sieć rekursywna

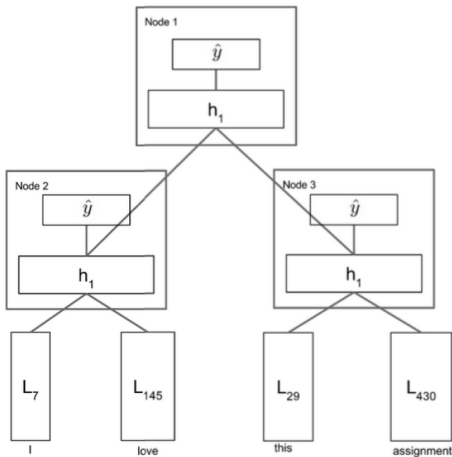


Funkcja kompozycji taka sama dla wszystkich kategorii składniowych.

źródło: Richard Socher i inni. Kurs *CS224d: Deep Learning for Natural Language Processing*, Uniwersytet Stanforda.

Prosta jednowarstwowa sieć rekursywna c.d.

Aplikacja do sparsowanego zdania: *"I love this assignment"*



źródło: Richard Socher i inni. Kurs *CS224d: Deep Learning for Natural Language Processing*, Uniwersytet Stanforda.

Prosta jednowarstwowa sieć rekursywna

c.d.

Aplikacja do sparsowanego zdania: “*I love this assignment*”

- najpierw uwzględniamy na wejściu drzewo wyprowadzenia zdania (parse tree) i wektory wyrazów zdania i rozpoczynamy 'wchodzenie' po drzewie
- “this assignment”
 - najniższy węzeł w grafie to *Node 3*, więc łączymy L_{29} i L_{430} , aby utworzyć wektor w \mathbb{R}^{2d} i dostarczyć go naszej sieci aby obliczyć:
$$h^{(1)} = \tanh(W^{(1)} \begin{bmatrix} L_{29} \\ L_{430} \end{bmatrix} + b^{(1)})$$
 - ponieważ $W^{(1)} \in \mathbb{R}^{d \times 2d}$ i $b^{(1)} \in \mathbb{R}^d$, to $h^{(1)} \in \mathbb{R}^d$
 - $h^{(1)}$ - punkt w tej samej przestrzeni wektorowej jak bigram “this assignment” (nie nauczyliśmy się tej reprezentacji osobno, ale wyznaczyliśmy z wektorów składowych wyrazów)
 - bierzemy $h^{(1)}$ i 'przepuszczamy' poprzez warstwę softmax aby obliczyć sentyment (pozytywny/negatywny, 5 klas)

Prosta jednowarstwowa sieć rekursywna

c.d.

Aplikacja do sparsowanego zdania: *"I love this assignment"*

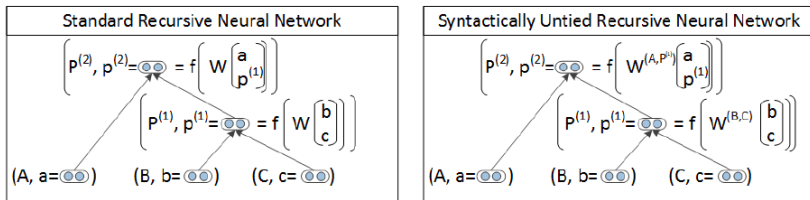
- "I love"
 - postępujemy podobnie w przypadku "I" i "love" aby wyznaczyć wektor $h^{(1)}$ dla frazy "I love"
 - ponownie obliczamy wynik dotyczący klas semantycznych dla tej frazy
- łączenie "I love" i "this assignment"
 - łączymy frazy wyrazów (a nie wektory wyrazów!) poprzez łączenie wektorów $h^{(1)}$ i obliczanie:
$$h^{(1)} = \tanh(W^{(1)} \begin{bmatrix} h_{Left}^{(1)} \\ h_{Right}^{(1)} \end{bmatrix} + b^{(1)})$$
 - otrzymujemy wektor w przestrzeni wektorowej wyrazów, reprezentujący ""I love this assignment"
 - możemy 'przepuścić' ten wektor przez warstwę softmax i policzyć prawdopodobieństwa klas sentymentu

Prosta sieć rekursywna - problemy

- Czy możemy efektywnie użyć tej samej macierzy W aby złączyć wszystkie wyrazy razem i dostać bardzo ekspresyjne $h^{(1)}$ a następnie użyć znowu tej samej macierzy W aby złączyć wszystkie wektory fraz?
- użycie tej samej macierzy aby łączyć wektory fraz rzeczownikowych i fraz czasownikowych aby uzyskać frazę zdaniową wydaje się intuicyjnie złym rozwiązaniem

Sieć rekursywna SU-RNN

'Rozsupłana' syntaktycznie rekursywna sieć neuronowa



Użycie różnych macierzy W do różnych kategorii wejściowych

źródło: Richard Socher i inni. Kurs *CS224d: Deep Learning for Natural Language Processing*, Uniwersytet Stanforda.

Sieć rekursywna SU-RNN

- ocena tego z jaką kategorią mamy do czynienia jest wykonywana z użyciem prostej probabilistycznej gramatyki bezkontekstowej, wyuczonej np. na podstawie zbiorczych statystyk na *Penn Tree Bank*
- macierze W są inicjalizowane jako macierze jednostkowe
- domyślnie zadaniem jest uśrednianie wektorów dwóch wyrazów, a model uczy się z czasem, który wektor jest ważniejszy, a także dowolnego obrotu lub skalowania wektorów, które poprawi wydajność

Sieć rekursywna SU-RNN - problemy

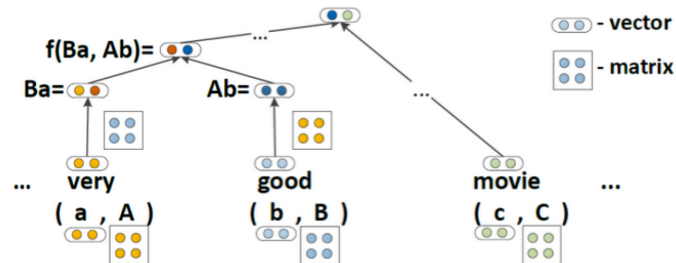
- W jaki sposób mamy traktować przysłówki typu “very”?
- W jaki sposób możemy uzyskać wektor, który wzmacnia jakikolwiek inny następujący wektor, gdy wykonujemy wyłącznie interpolację liniową?
- W jaki sposób możemy skonstruować wektor, który skaluje jakikolwiek inny wektor w ten sposób?
- Nie da się tego zrobić, musimy mieć inną formę mnożenia wyrazu przez wyraz

Sieć rekursywna MV-RNN

Macierzowo-wektorowa sieć rekursywna

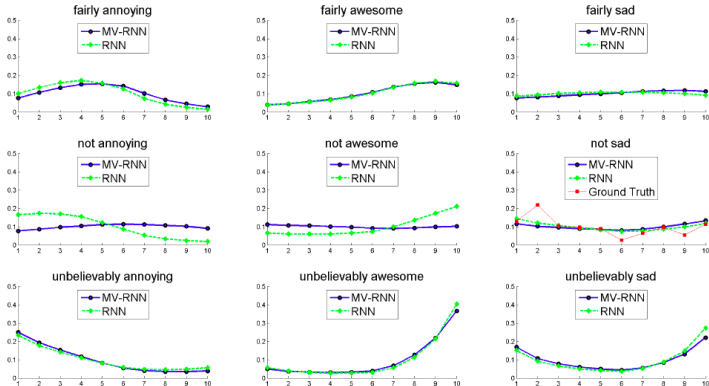
- powiększmy reprezentację wyrazów, aby zawierała nie tylko wektory wyrazów, ale także macierze wyrazów
- wyraz “very” będzie mieć odpowiedni wektor wyrazu $v_{very} \in \mathbb{R}^d$ oraz $v_{very} \in \mathbb{R}^{d \times d}$
- to daje nam możliwość wyrażania nie tylko tego, co dany wyraz znaczy, ale także uczenia się jak wyrazy **modyfikują** inne wyrazy

Sieć rekursywna MV-RNN c.d.



źródło: Richard Socher i inni. Kurs *CS224d: Deep Learning for Natural Language Processing*, Uniwersytet Stanforda.

Sieć rekursywna MV-RNN c.d.

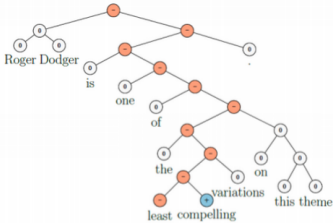
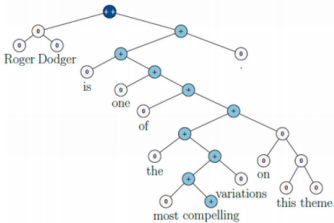


źródło: Richard Socher i inni. Kurs *CS224d: Deep Learning for Natural Language Processing*, Uniwersytet Stanforda.

Sieć rekursywna MV-RNN c.d.

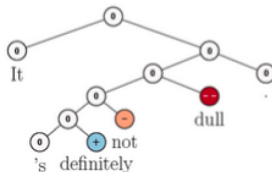
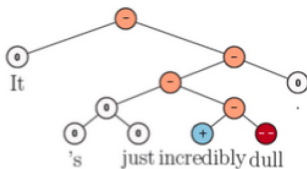
- aby podać dwa wyrazy, a i b , do RNN, bierzemy ich macierze wyrazów A i B , aby utworzyć wektor wejściowy x jako konkatenację wektora Ab i Ba
- jednak ciągle jeszcze obserwujemy trzy główne klasy błędów: 'zanegowane pozytywne', 'zanegowane negatywne', 'X, ale Y'

Zanegowane pozytywne



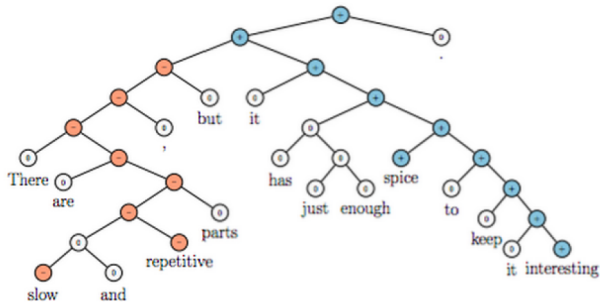
źródło: Richard Socher i inni. Kurs *CS224d: Deep Learning for Natural Language Processing*, Uniwersytet Stanforda.

Zanegowane negatywne



źródło: Richard Socher i inni. Kurs *CS224d: Deep Learning for Natural Language Processing*, Uniwersytet Stanforda.

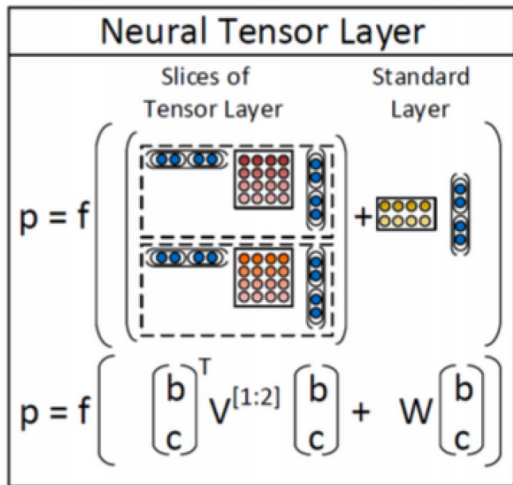
X, ale Y



źródło: Richard Socher i inni. Kurs *CS224d: Deep Learning for Natural Language Processing*, Uniwersytet Stanforda.

Sieć rekursywna RNTN

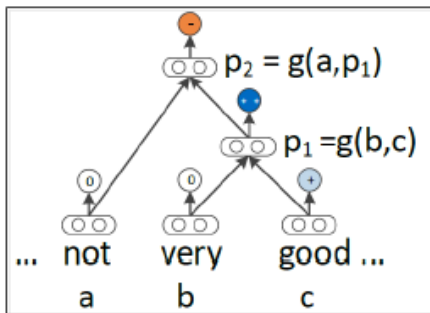
Sieć rekursywna neuronowych tensorów



Jeden 'plaster' RNTN (jest ich d)

źródło: Richard Socher i inni. Kurs CS224d: Deep Learning for Natural Language Processing, Uniwersytet Stanforda.

Sieć rekursywna RNTN c.d.



źródło: Richard Socher i inni. Kurs *CS224d: Deep Learning for Natural Language Processing*, Uniwersytet Stanforda.

Sieć rekursywna RNTN c.d.

- eliminuje pojęcie macierzy wyrazów oraz tradycyjną transformację \tanh / sigmoid
- aby skomponować dwa wektory wyrazów lub wektory fraz, ponownie łączymy je, aby utworzyć wektor w \mathbb{R}^{2d} , tym razem najpierw 'przepuszczamy' go przez funkcję kwadratową, później nieliniową, np.:

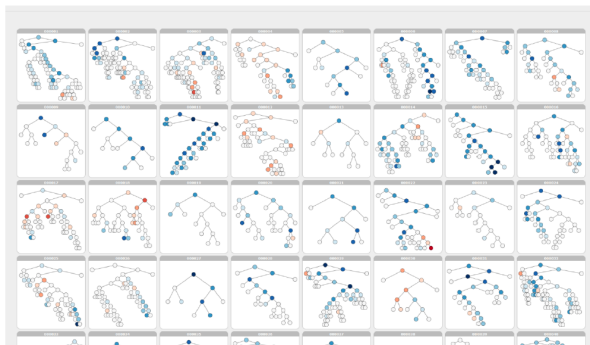
$$h^{(1)} = \tanh(x^T V x + W x)$$

gdzie V jest tensorem w $\mathbb{R}^{2d \times 2d \times d}$

- obliczamy $x^T V[i] \forall i \in [1, 2, \dots, d]$, 'plastry' tensora, obliczając wektor w \mathbb{R}^d
- następnie dodajemy Wx i 'przepuszczamy' przez funkcję nieliniową
- w ten sposób możemy pozwolić na multiplikatywny typ interakcji między wektorami wyrazów bez potrzeby utrzymywania i uczenia się macierzy wyrazów

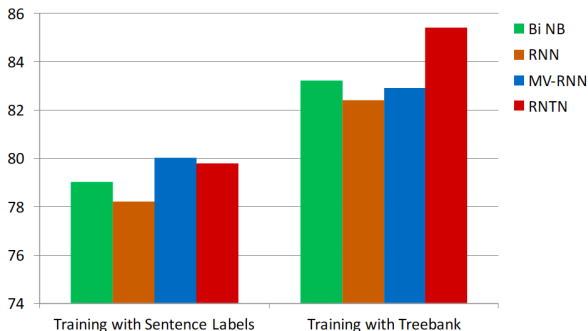
Bank drzew do analizy sentymentu

- drzewa wyprowadzenia dla 11 855 zdań
- 215 154 fraz z etykietami
- pozwala na uczenie i ewaluację biorąc pod uwagę informację 'kompozycyjną'



źródło: Richard Socher i inni. Kurs *CS224d: Deep Learning for Natural Language Processing*, Uniwersytet Stanforda.

Porównanie metod



źródło: Richard Socher i inni. Kurs *CS224d: Deep Learning for Natural Language Processing*, Uniwersytet Stanforda.

Dziękuję za uwagę!