Spis treści

1. Wprowadzenie.

Celem pracy jest porównanie zdolności predykcyjnych modeli sieci neuronowych opartych na tekście osadzonym w notacji Word2Vec dla embeddingu zbudowanego przy pomocy rekursywnych sieci neuronowych na zbiorze danych recenzji lokalnych biznesów Yelp[1] oraz osadzeniu przy pomocy słownika globalnego wyszkolonego w ramach biblioteki Pythona nltk poprzez porównanie ze sobą różnych struktur sieci oraz optymalizatorów.

Drugim zagadnieniem jest zbadanie zależności, czy model uczy się lepiej na zbiorze danych pozytywnych recenzji oraz negatywnych.

2. Przygotowanie zbioru danych.

I. Opis zbioru danych.

Zbiór danych zawiera 6 990 280 recenzji w języku angielskim, do której przypisana jest zmienna celu „stars” [1:5] reprezentująca opinie użytkowników. Ze względu na ograniczenia związane z przetwarzaniem zbioru w wersji surowej o wielkości 8,65 GB, zdecydowano się na wykorzystanie próby losowej o wielkości 60 tysięcy recenzji.

II. Model zewnętrzny.

W celu budowy modelu zewnętrznego zdecydowano się na użycie i zastosowanie modelu globalnego z biblioteki Pythona nltk. Słowo definiuje się jako podmiot stringa oddzielony spacjami.

Celem jest osadzenie zdań w przestrzeni 300 wymiarowej jako średnią z odpowiadających w modelu wektorów słów. Wpierw dokonano lematyzacji słów, zamiany dużych liter na małe oraz usunięcia z pomocą zewnętrznego słownika tak zwanych „stopwords”, czyli słów występujących powszechnie takich jak „and”, „or”, „the” , „ . ” itp.

Jeżeli słowo nie występuje w osadzeniu modelu zewnętrznego, traktowane jest ono jako wektor 0, sprowadzając osadzenie recenzji do 0.

[1] https://www.yelp.com/dataset

Tym samym uzyskano następujący rezultat:

Tab.1. Model zewnętrzny.

"Went to The Breakfast Club on the spur of the moment today (Saturday) just after noon. Despite the hour, we were looking for breakfast.\n\nThis place is terrific. I had a very basic order of scrambled eggs, hash browns and corned beef hash. All were among the best I've had. My wife tried their french toast and loved it. Plus, their coffee is fabulous. What more can one ask of a breakfast place?\n\nTheir lunch menu looks equally appealing, and we'll be giving lunch a go soon. The only problem is, to do that we'll have to pass on a delicious breakfast. Such a dilemma..."					
went breakfast club spur moment today saturday noon despite hour looking breakfast place terrific basic order scrambled egg hash brown corned beef hash among best ive wife tried french toast loved plus coffee fabulous one ask breakfast place lunch menu look equally appealing well giving lunch go soon problem well pas delicious breakfast dilemma					
V1	V2	V3 ...	... V298	V299	V300
0,15038	-0,0699	-0,0716 ...	... -0,0794	-0,0418	-0,0672

Źródło: Opracowanie własne.

III. Model na danych.

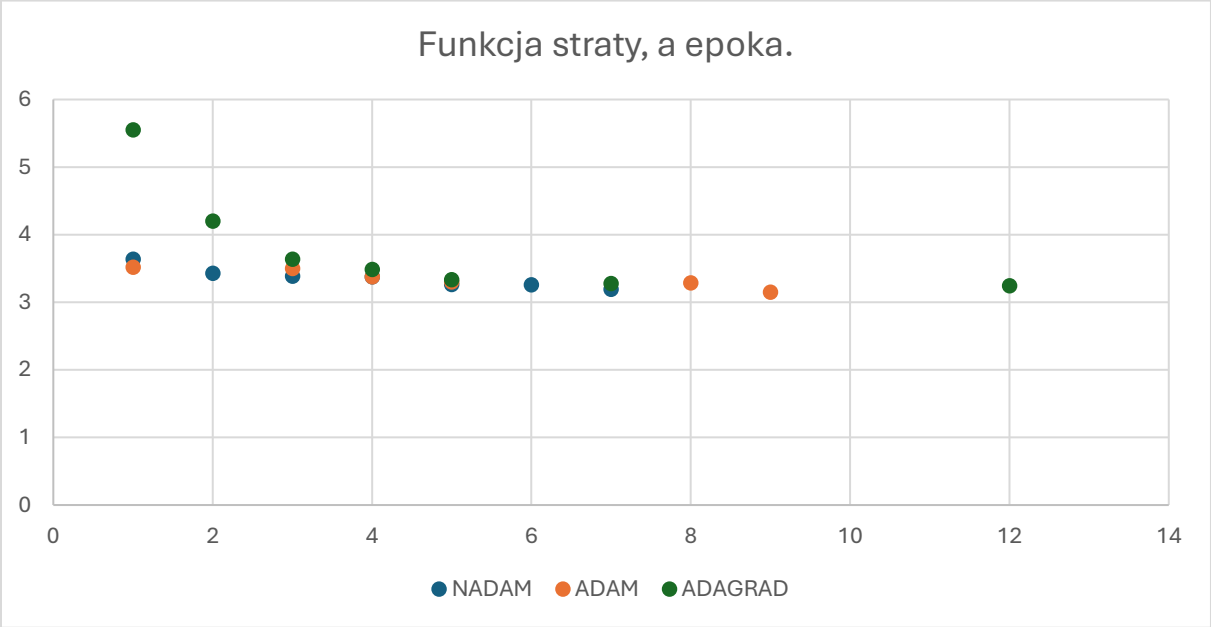
W celu budowy tego modelu nie dokonano lematyzacji słów, jednak zamieniono małe litery na duże, usunięto znaki interpunkcyjne oraz zastosowano subsampling w celu wykrycia zbędnych słów.

Tab. 2. Najlepsza wartość funkcji straty dla wybranych optymalizatorów.

Epoka	NADAM	ADAM	ADAGRAD
1	3,637101	3,517559	5,549912
2	3,425587		4,196521
3	3,381998	3,498553	3,63268
4	3,375733	3,376444	3,481462
5	3,262917	3,298976	3,332732
6	3,254674		
7	3,191887		3,274103
8		3,28398	
9		3,145199	
10			
11			
12			3,240215

Źródło: Opracowanie własne.

Wykres 1. Optymalizacja embeddingu.



Źródło: Opracowanie własne.

Najlepszym optymalizatorem w przypadku embeddingu okazał się być zwykły ADAM, jednak ADAM z pędem nestora szybciej dochodzi do rozwiązania. AdaGrad zaczął z kolei z najwyższego poziomu funkcji straty, co utrudniło proces uczenia.

Tab. 3. Model na danych

"Went to The Breakfast Club on the spur of the moment today (Saturday) just after noon. Despite the hour, we were looking for breakfast.\n\nThis place is terrific. I had a very basic order of scrambled eggs, hash browns and corned beef hash. All were among the best I've had. My wife tried their french toast and loved it. Plus, their coffee is fabulous. What more can one ask of a breakfast place?\n\nTheir lunch menu looks equally appealing, and we'll be giving lunch a go soon. The only problem is, to do that we'll have to pass on a delicious breakfast. Such a dilemma..."					
["went", "to", "the", "breakfast", "on", "the", "of", "the", "moment", "saturday", "just", "after", "noon", "despite", "the", "hour", "we", "were", "looking", "for", "breakfast", "this", "place", "is", "i", "had", "a", "very", "order", "of", "hash", "and", "beef", "hash", "all", "were", "the", "i", "ve", "had", "my", "wife", "their", "french", "toast", "and", "it", "plus", "their", "coffee", "is", "fabulous", "what", "more", "can", "one", "ask", "of", "a", "breakfast", "place", "their", "lunch", "menu", "looks", "equally", "and", "we", "ll", "be", "giving", "lunch", "a", "go", "soon", "the", "only", "is", "to", "do", "that", "we", "ll", "have", "to", "pass", "on", "a", "delicious", "breakfast", "such", "a"]					
V1	V2	V3 ...	... V298	V299	V300
0.0801	0.0385	-0.0287…	... -0.0014	0.0257	-0.0254

Źródło: Opracowanie własne.

Osadzenia recenzji są wyraźnie bliższe przestrzeni zerowej, co wynikać może z ograniczonego kontekstu. Dodatkowo metoda subsamplingu nie wychwyciła wielu „stopwords”, co może zaburzać jakość embeddingu.

3. Modele na notacji word2vec.

I. Dane własne.

Do modeli użyto funkcji straty crossentroprii w celu uniknięcia przeuczenia oraz funkcję końcową softmax zwracającą prawdopodobieństwa przynależności do klasy. Precyzję (accuracy) zdefiniowano w najprostszy sposób, używając maksymalnego prawdopodobieństwa jako predykcji w porównaniu do rzeczywistej wartości (oceny 1:5). Do porównania wybrano trzy optymalizatory:

-ADAM: klasyczny wybów

-AdaGrad: mały gradient

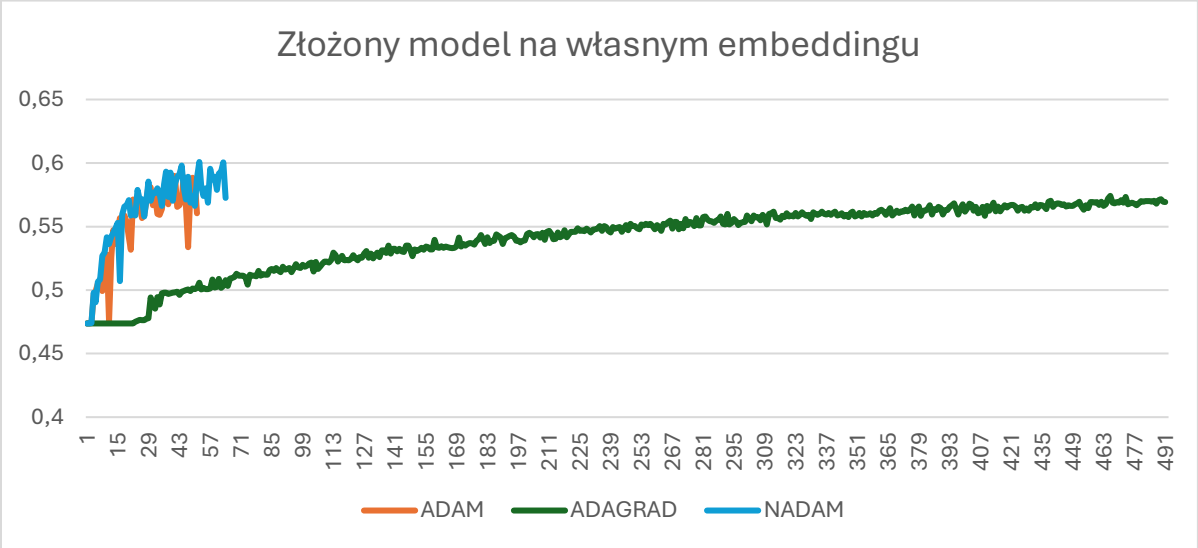
-NADAM: ADAM z pędem nestora, zwiększa średni gradient.

A. Złożona sieć neuronowa.

Model I opisano jako:

Sieć trzywarstwowa (300 =>128 =>64 => 5) z funkcją aktywacji ReLU oraz funkcją przekształcającą softmax.

Wykres 2. Accuracy złożonych modeli na własnym zbiorze walidacyjnym.



Źródło: Opracowanie własne.

Zarówno ADAM jak i NADAM wykazują się silnym zjawiskiem catastrophic interference, co skutkuje wysokimi spadkami gradientu. Funkcja AdaGrad ma go z kolei bardzo niski.

Tab. 4. Poprawna predykcja na zbiorze testowym.

ADAM	AdaGrad	NADAM
0,5625	0,57	0,6

Źródło: Opracowanie własne.

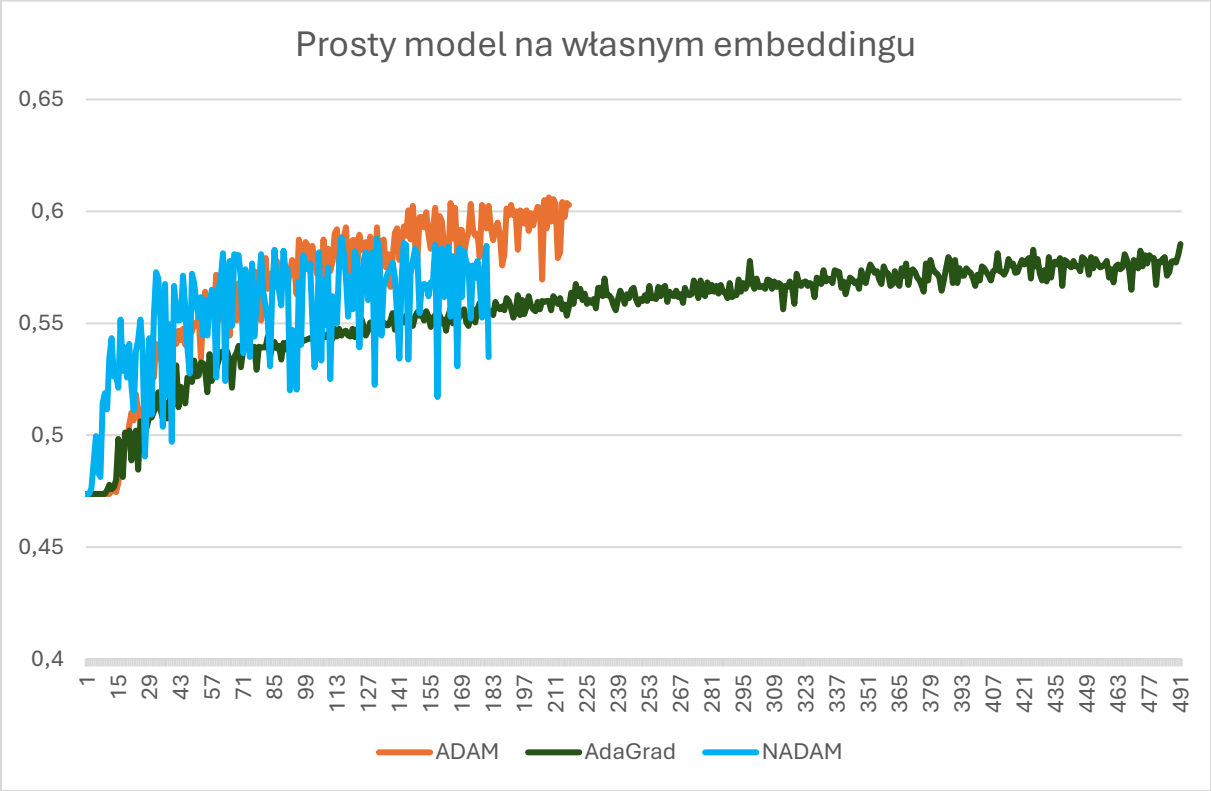
Pomimo problemów z dużymi wahaniami gradientu ADAM z pędem Nestora osiągnął największy wynik na próbie testowej, a ADAM zbyt szybko został zbieżny.

B. Prosta sieć neuronowa.

Model II opisano jako:

Sieć dwuwarstwowa (300 =>32 => 5) z funkcją aktywacji ReLU oraz funkcją przekształcającą softmax.

Wykres 3. Accuracy prostych modeli na własnym zbiorze walidacyjnym.



Źródło: Opracowanie własne.

Zwykły ADAM okazuje się być lepszy od ADAMU z pędem Nestora, jednak również popada w problem catastrophic interference. Pod koniec próby epok również AdaGrad zaczął odnotowywać nagłe wahania gradientów.

Tab. 5. Poprawna predykcja na zbiorze testowym.

ADAM	AdaGrad	NADAM
0,601	0,574	0,553

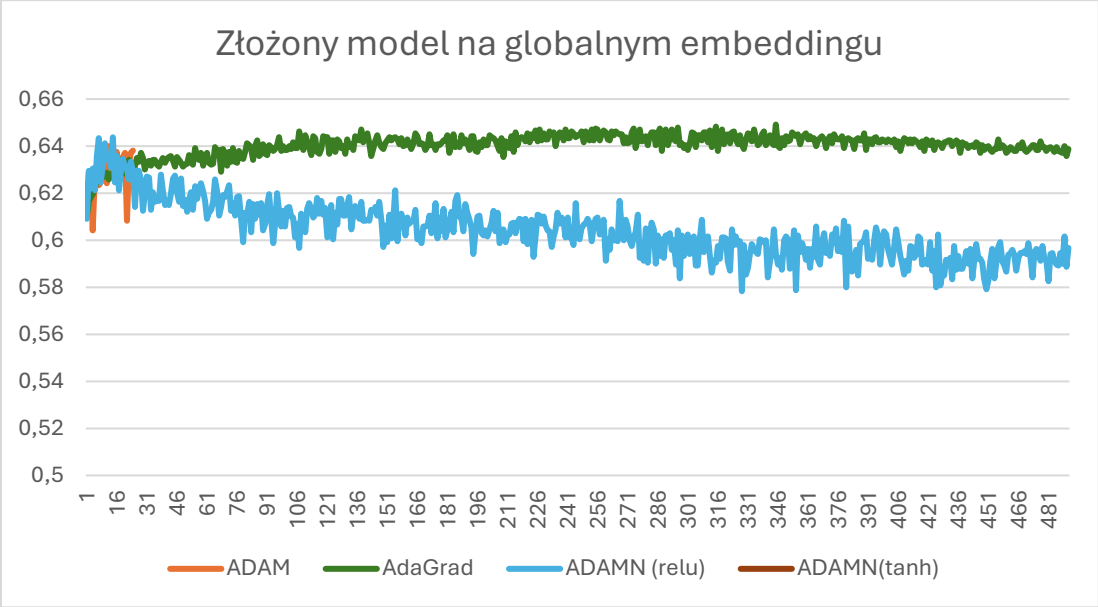
Źródło: Opracowanie własne.

Granicą dla wszystkich modeli okazuje się być accuracy na poziomie 0,6. Dla bardziej złożonego modelu, pęd dołączony do optymalizatora ADAM okazał się bardziej przydatny, a dla prostszego szkodliwy.

II. Zewnętrzny embedding.

A. Złożona sieć neuronowa.

Wykres 4. Accuracy złożonych modeli na zbiorze walidacyjnym.



Źródło: Opracowanie własne.

W przypadku optymalizatora ADAMN z funkcją aktywacji RELU zaobserwowano szybkie osiągnięcie maximum, po czym ciągły spadek accuracy na zbiorze walidacyjnym. Nagły spadek jakości modelu po najszybszym znalezieniu najwyższego accuracy na zbiorze walidacyjnym świadczy o przeuczeniu modelu. W tym celu porównano wartości z inną funkcją aktywacji (tangens hiperboliczny) i zauważono ten sam trend (tu zastosowano kryterium zbieżności modelu).

Tab. 6. Poprawna predykcja na zbiorze testowym.

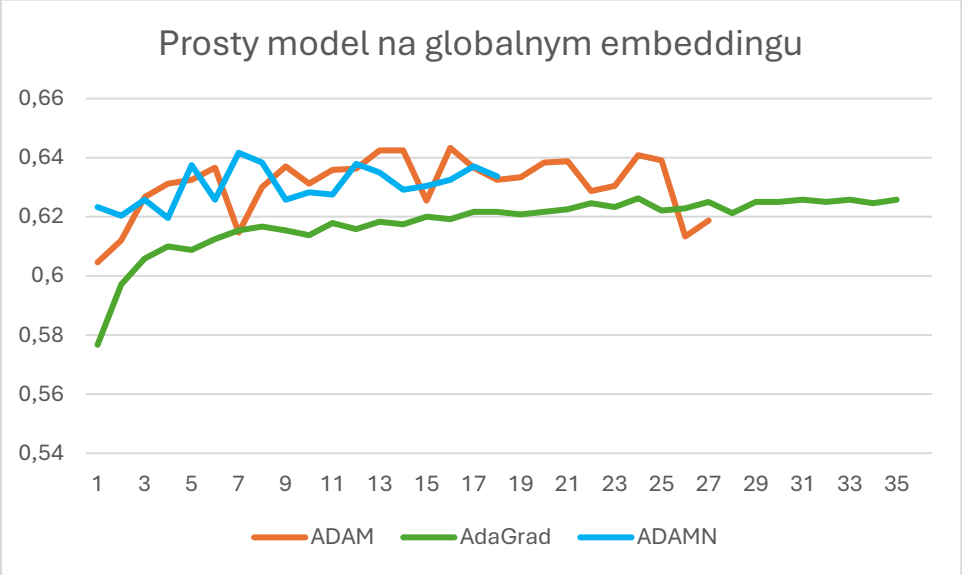
ADAM	AdaGrad	NADAM
0,638	0,643	0,636

Źródło: Opracowanie własne.

Najlepszym optymalizatorem okazał się być AdaGrad, jednak ze względu na mały gradient uczył się najdłużej jak w poprzednich przypadkach.

B. Prosta sieć neuronowa.

Wykres 5. Accuracy prostych modeli na zbiorze walidacyjnym.



Źródło: Opracowanie własne.

ADAMN osiągnął najszybszą zbieżność, AdaGrad po raz pierwszy nie uczył się przez cały zadany okres, a ADAM już na początku próby wykazywał się dużymi wahaniami gradientu.

Tab. 7. Poprawna predykcja na zbiorze testowym.

ADAM	AdaGrad	NADAM
0,641	0,637	0,643

Źródło: Opracowanie własne.

Nieznacznie lepszym od ADAMa okazał się być NADAM, a model prostszy okazał się być lepszy.

III. Wnioski.

Model globalnego embeddingu okazał się być nieznacznie lepszy od modelu zbudowanego na danych, co świadczy o słuszności wykorzystywania transfer - learningu.

Dla sieci neuronowych z optymalizatorami ADAM i NADAM okazał się być bardzo problematyczny catastrophic interference skutkujący ciągłym zapominaniem poprzednich wag.

Optymalizator NADAM najszybciej znajduje najlepsze rozwiązanie globalne, jednak równie szybko wpada w wyżej wspomniany problem oraz jest bardziej podatny na przeuczenie.

Najlepszym typem modelu okazała się prosta sieć neuronowa z dwoma warstwami na embeddingu zewnętrznym. Dla porównania jakość predykcji modelu Gradient Boostingu na zbiorze testowym wynosiła 0.57, a sieci z optymalizatorem NADAM 0,64.

4. Modele generatywne sieci rekurencyjnych.

Spośród wielu badanych kombinacji modeli wybrano najlepszy dla każdego z optymalizatorów.

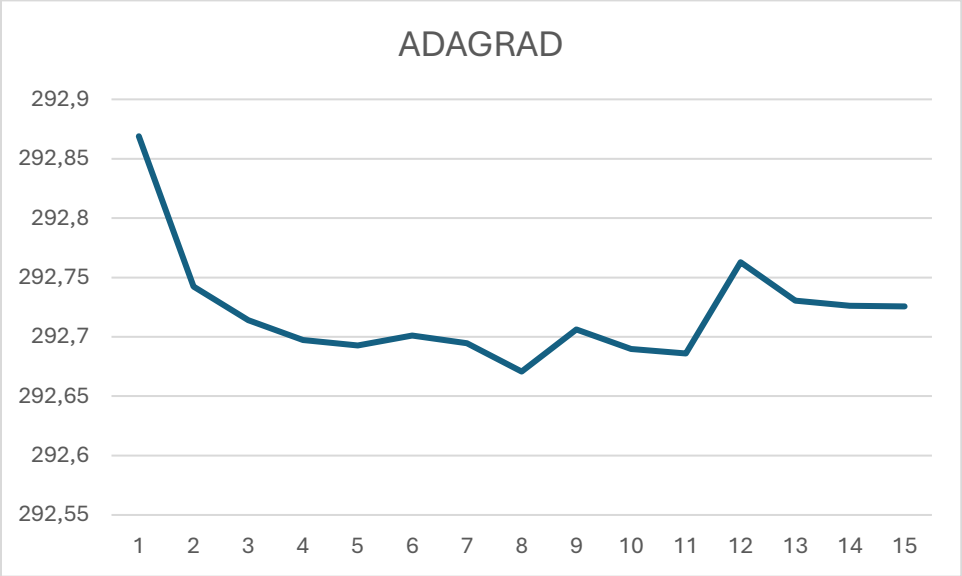
Za pozytywne opinie ocenia się te większe od 3, a negatywne pozostałe.

Wybrano 5000 losowych recenzji dla każdego z typów i poddano preprocessingowi, to znaczy zmianie liter na małe, usunięciu niełacińskich znaków poza interpunkcją oraz spacjami.

Ze względu na ograniczenia w mocy obliczeniowej oraz pamięci RAM (32GB) i wolnego miejsca na dysku SSD (110 GB) ograniczono się wyłącznie do początku próby epok dla pozytywnych recenzji.

I. AdaGrad.

Wykres 6. AdaGrad na ogólnej próbie.

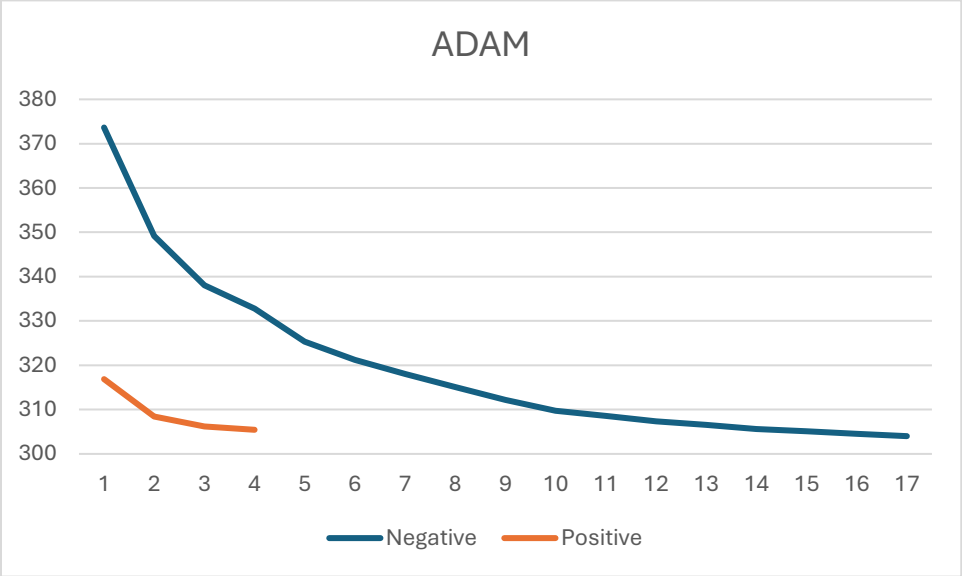


Źródło: Opracowanie własne.

AdaGrad nie jest dobrym optymalizatorem, ze względu na mały gradient bardzo szybko się zawiesza wokół minima lokalnego.

II. ADAM.

Wykres 7. ADAM na próbie 5000



Źródło: Opracowanie własne.

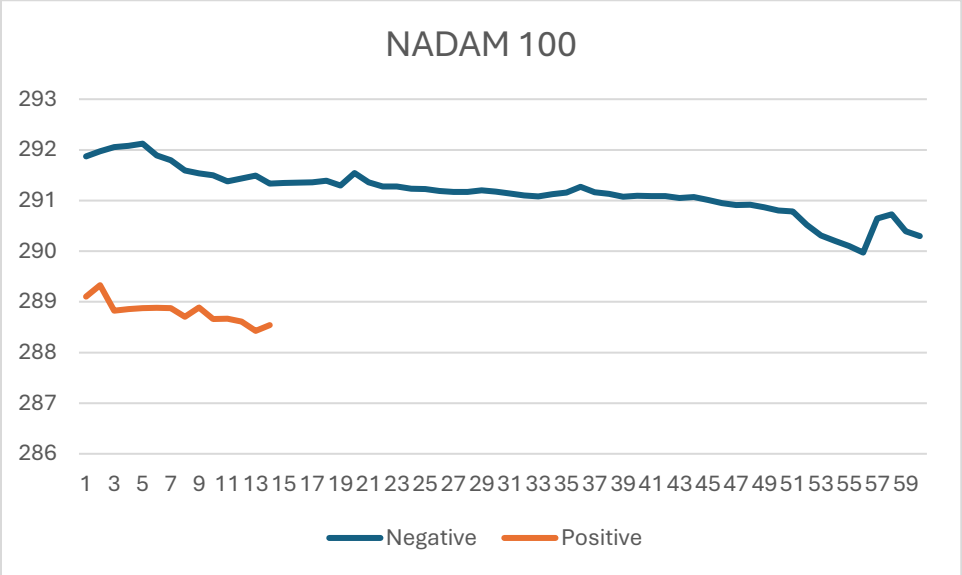
Widoczny jest wyraźny spadek na początku próby, po czym znaczne osłabienie siły gradientu, nie występuje jednak problem nagłych wahań jak w przypadku klasyfikacji.

III. NADAM.

Biorąc pod uwagę wysoki problem z bardzo małym gradientem pomimo manipulacji stopą uczenia na szerszą skalę zastosowano optymalizator NADAM.

A. NADAM na długości sekwencji 100 i wielkości próby 5000.

Wykres 8. NADAM na długości sekwencji 100



Źródło: Opracowanie własne.

Widoczna jest podobnie jak u generatora opartego o optymalizator ADAM wyraźna różnica w wartości funkcji straty pomiędzy recenzjami pozytywnymi oraz negatywnymi.

Pod koniec badania na próbie negatywnej NADAM gwałtownie zwiększył wartość funkcji straty, po nagłej obniżce, co może świadczyć o napotkaniu minima lokalnego.

Tab. 8. Najlepsze wartości funkcji straty.

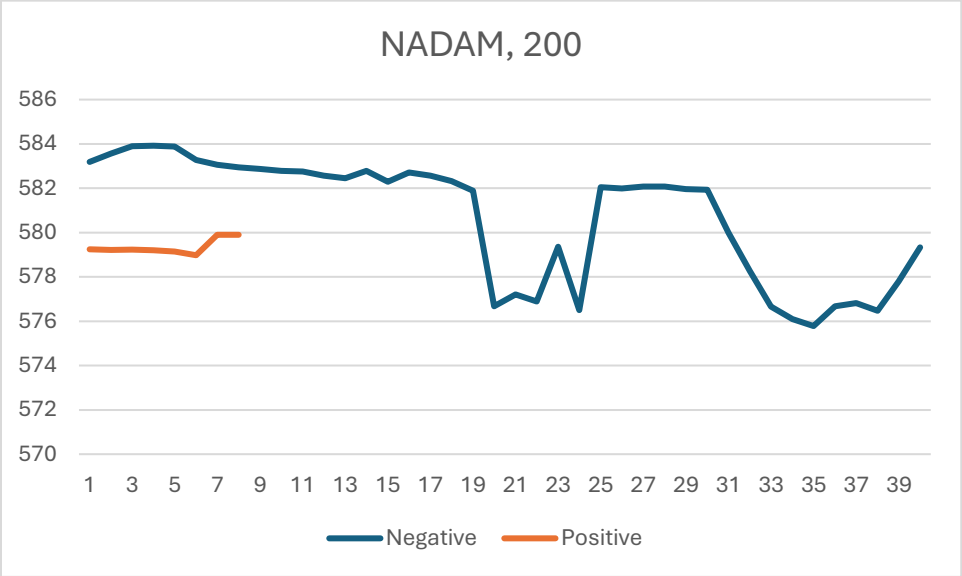
	Negative	Positive
Best	289.97	287.94

Źródło: Opracowanie własne.

A. NADAM na długości sekwencji 200 i wielkości próby 15000.

W celu naprawy problemu z niewielkim gradientem funkcji straty postanowiono zwiększyć próbę 3-krotnie oraz odpowiednio podwoić długość sekwencji by zwiększyć średni poziom straty.

Wykres 9. NADAM na długości sekwencji 200.



Źródło: Opracowanie własne.

Widoczne są pierwsze gwałtowne wzrosty i spadki wartości funkcji straty na próbie negative, co może świadczyć o powstawianiu problemu catastrophic inference.

Tab. 9. Najlepsze wartości funkcji straty.

	Negative	Positive
Best	582,86	576,49

Źródło: Opracowanie własne.

IV. Wnioski

Model generatywny oparty o rekurencyjną sieć neuronową niezależnie od optymalizatora boryka się z problemem niskiego gradientu funkcji, co w połączeniu z czasem uczenia epoki skutkuje bardzo nieefektywnym sposobem ucznia.

Zastosowanie metod zwiększających spadek funkcji celu poprzez pęd może prowadzić do utraty wiedzy z poprzednich iteracji, co w tym przypadku jest bardzo kosztowne czasowo.

Model uczy się lepiej na recenzjach nacechowanych pozytywnie niż negatywnie, co najprawdopodobniej wynika ze stylu wypowiedzi, który wynika z sentymentu.

5. Bibliografia.

1. Efficient Estimation of Word Representations in Vector Space [7 Sep 2013]: https://arxiv.org/pdf/1301.3781

2. Yelp Open Dataset [Dane] : https://www.yelp.com/dataset

3. NLT [Model word2vec]: https://www.nltk.org/