Kosturek Michał, 220885
piątek, 7:30

10 listopada 2017 r.

Sieci Neuronowe
Wielowarstwowa Sieć Neuronowa

Sprawozdanie nr 2

Spis treści

1 Wstęp **3**
1.1 Propagacja wsteczna 3

2 Przebieg i badane parametry **4**

3 Badania **5**
3.1 Szybkość uczenia w zależności od liczby neuronów w warstwie ukrytej 5
3.1.1 Rezultaty 5
3.1.2 Wnioski 5
3.2 Współczynnik uczenia 6
3.2.1 Rezultaty 6
3.2.2 Wnioski 6
3.3 Momentum 7
3.3.1 Rezultaty 7
3.3.2 Wnioski 7
3.4 Rozmiar zbioru uczącego 8
3.4.1 Rezultaty 8
3.4.2 Wnioski 8
3.5 Inicjalizacja wag początkowych 9
3.5.1 Rezultaty 9
3.5.2 Wnioski 9

Literatura **9**

1 Wstęp

W ramach ćwiczenia laboratoryjnego została zaimplementowana biblioteka umożliwiająca modułowe komponowanie wielowarstwowych sieci neuronowych uczonych metodą propagacji wstecznej wykorzystując wybraną funkcję straty oraz liczbę, rozmiary i aktywacje kolejnych warstw. Następnie zostały przeprowadzone badania wpływu poszczególnych parametrów sieci i algorytmów jej uczenia na szybkość i jakość nauki.

1.1 Propagacja wsteczna

Wyzwaniem pojawiającym się przy budowaniu wielowarstwowych sieci jest określenie błędu popełnianego przez neurony warstw ukrytych, który jest niezbędny do odpowiedniej aktualizacji wag. W warstwie wyjściowej, tak jak w prostych perceptronach, zadanie to jest proste - błąd może być określony tak prosto jak np. kwadrat różnicy wartości oczekiwanej na wyjściu neuronu i rzeczywistej wartości tego wyjścia. Metoda propagacji wstecznej tego właśnie błędu warstwy wyjściowej pozwala uzyskiwać wartości błędu we wcześniejszych warstwach sieci. Jest ona wyznaczana jako ważona przez wagi synaps wyjściowych suma błędów neuronów z warstwy następnej.

Tak więc błąd w warstwie ukrytej i-tej (gdzie warstwa 0 - warstwa wejściowa, warstwa n -wyjściowa) $(0 < i < n)$ można określić jako:

$$\delta_i = \frac{\partial}{\partial net_i} f_i\big(net_i(x)\big) \circ W_{i+1}^T \delta_{i+1}, \tag{1}$$

gdzie w_{i+1} to wagi wyjściowe warstwy i-tej (wejściowe $i+1$-szej).

Natomiast dla warstwy wyjściowej błąd ten jest uzależniony od wybranych funkcji straty L i aktywacji f_n w tej warstwie (pochodna straty po całkowitym pobudzeniu wymaga uwzględnienia postać aktywacji):

$$\delta_i = \frac{\partial}{\partial net_n} L\Big(f_n\big(net_n(x)\big),\ y\Big) \tag{2}$$

gdzie y jest wartością oczekiwaną na wyjściu sieci.

Tak obliczone wartości błędów w poszczególnych warstwach są wykorzystywane do aktualizacji wag w kolejnych przebiegach algorytmu uczącego:

$$W_i[t+1] = W_i[t] + \eta \circ \delta_i z_i^T \tag{3}$$

gdzie:

η - to współczynnik uczenia,

z_i - to wejście do warstwy i-tej, na podstawie którego sieć dokonała predykcji, dla której wyznaczono błąd. Dla pierwszej warstwy ukrytej będą to wartości wejścia sieci, dla każdej kolejnej - aktywacje z poprzedniej warstwy.

Korekcja wag wyrażona powyższym wzorem wykonywana jest dla każdego wzorca (lub *batcha* przy algorytmie stochastycznego gradientu zstępującego) z całego zbioru uczącego.

2 Przebieg i badane parametry

Model sieci, dla której prowadzono badania:

1. Warstwa wejściowa:
   - 70 neuronów,
   - inicjalizacja wag w zakresie $[-0.1; +0.1]$
2. Warstwa ukryta:
   - 50 neuronów,
   - inicjalizacja wag w zakresie $[-0.1; +0.1]$,
   - bias,
   - aktywacja: Softplus
3. Warstwa wyjściowa:
   - 70 neuronów,
   - inicjalizacja wag w zakresie $[-0.1; +0.1]$,
   - bias,
   - aktywacja: Sigmoid

Badaniom podlegały następujące parametry i wpływ ich wartości na długość uczenia (w epokach):

- Zakres możliwych wartości przy inicjalizacji wag
- Współczynnik uczenia
- Funkcja aktywacji

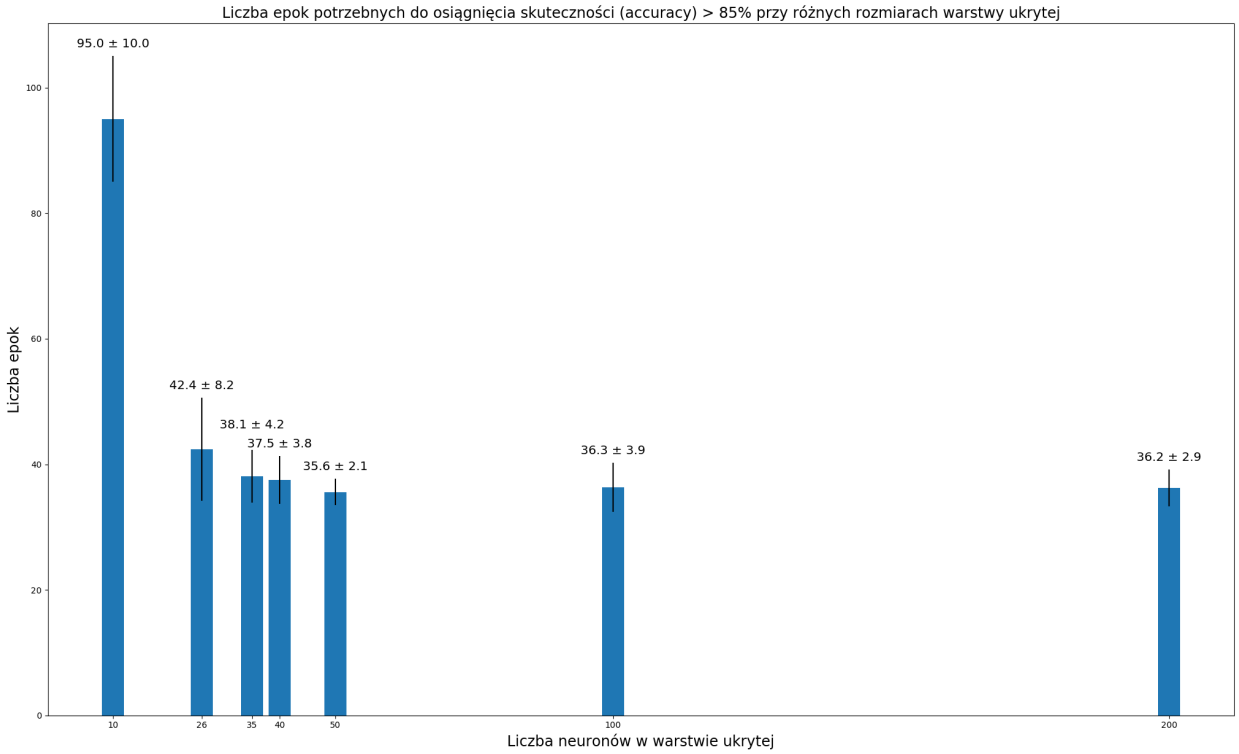
Przy każdym uruchomieniu uczenia sieci losowane są jej wagi, a w każdej epoce losowo zmieniana jest kolejność obserwacji w zbiorze uczącym. W konsekwencji tej losowości każde z badań zostało przeprowadzone dziesięciokrotnie - prezentowane wyniki reprezentują średnią arytmetyczną uzyskanych rezultatów wraz z ich odchyleniem standardowym.

3 Badania

3.1 Szybkość uczenia w zależności od liczby neuronów w warstwie ukrytej

Pierwsze badanie ma na celu sprawdzenie ile neuronów w warstwie ukrytej jest potrzebne do skutecznego działania sieci. W tym celu sprawdzono 7 różnych rozmiarów warstwy ukrytej: 10, 35, 50, 100, 200, a także 26 i 40 - stanowiące geometryczną i arytmetyczną średnią liczby neuronów warstw wejściowej i wyjściowej.

3.1.1 Rezultaty



Rysunek 1: Zależność liczby epok nauki od rozmiaru warstwy ukrytej

Rozmiar warstwy ukrytej	Liczba epok nauki
10	95.0 ± 10.0099
26	42.4 ± 8.2122
35	38.1 ± 4.2297
40	37.5 ± 3.8275
50	35.6 ± 2.1071
100	36.3 ± 3.9000
200	36.2 ± 2.9257

Tabela 1: Zależność liczby epok nauki od rozmiaru warstwy ukrytej

3.1.2 Wnioski

Ustalenie zbyt małej liczby neuronów w warstwie ukrytej powoduje bardzo nieefektywne działanie sieci. Z drugiej strony - ustawianie zbyt dużej liczby neuronów od pewnego momentu nie zmienia już czasu uczenia sieci.

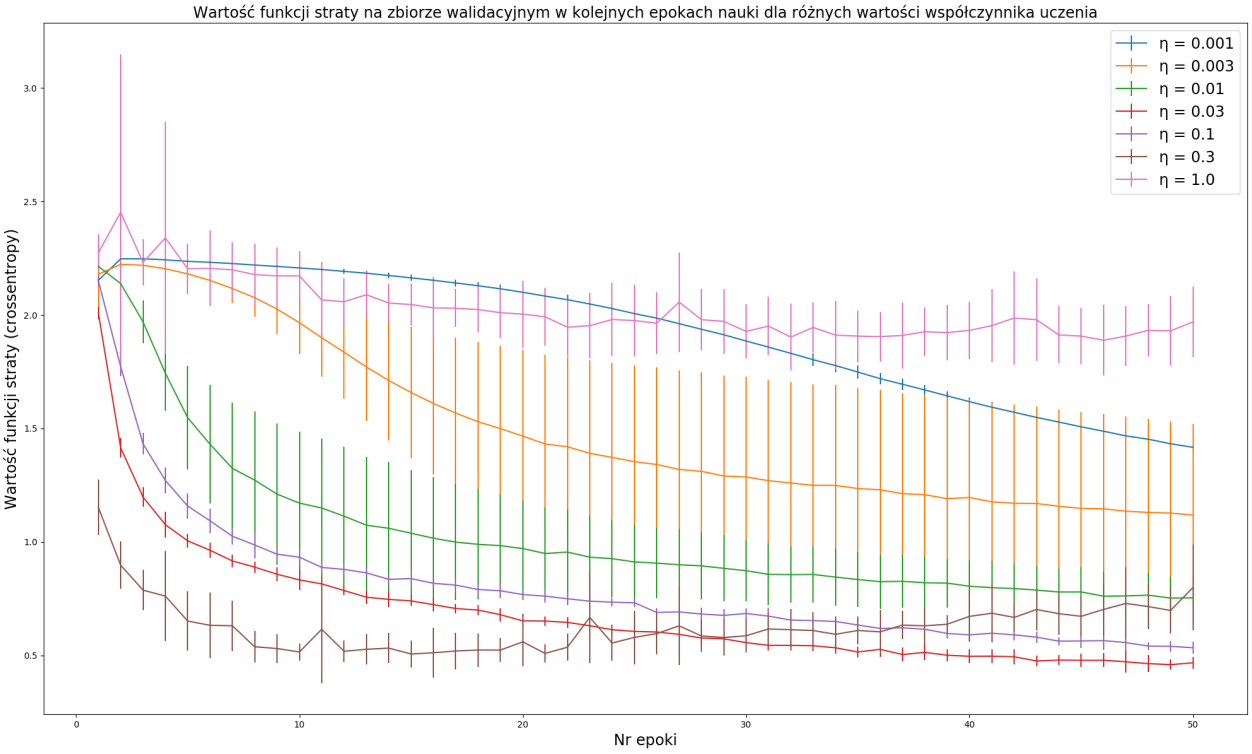
Wartości wyznaczane jako średnia geometryczna lub arytmetyczna pozwalają wyznaczyć liczbę neuronów zbliżoną do tej granicznej, powyżej której dodatkowe neurony nie wnoszą nic jeśli chodzi o szybkość uczenia. W tym przypadku lepiej w tym kontekście zadziałała średnia arytmetyczna.

3.2 Współczynnik uczenia

W ramach eksperymentu zbadano 7 wartości współczynnika uczenia: $\eta \in \{0.001; 0.003; 0.01; 0.03; 0.1; 0.3; 1\}$

W badaniu przeprowadzono uczenie sieci używając metody stochastycznego gradientu, przy rozmiarze mini-batcha równym 10, na zbiorze uczącym liczącym około 5000 obserwacji i dla zbioru walidacyjnego o liczności ok. 400. Uczenie takie przeprowadzono dla każdej z testowanych wartości współczynnika uczenia.

3.2.1 Rezultaty



Rysunek 2: Wartość straty na zb. walidacyjnym podczas nauki dla różnych współczynników uczenia

3.2.2 Wnioski

Łatwo zauważyć, że zbyt duże wartości współczynnika uczenia powodują szybkie początkowe zbieganie wartości straty, jak w przypadku $\eta = 1$, a następnie (albo od razu jak dla $\eta = 0.3$) wartość straty wręcz rośnie. Ponadto duże wahania wartości straty pomiędzy epokami wskazują na dość chaotyczny sposób eksploracji przestrzeni wag.

Zbyt małe wartości współczynnika (jak $\eta = 0.001$ lub $\eta = 0.003$) prowadzą do bardzo systematycznego, ale powolnego zbiegania wartości straty.

Najlepszym wyborem wydają się wartości $\eta = 0.03$ i $\eta = 0.1$, przy których wartość straty zbiegła szybko i systematycznie, a po 50 epokach osiągnęła najniższą spośród wszystkich badanych konfiguracji wartość funkcji straty.

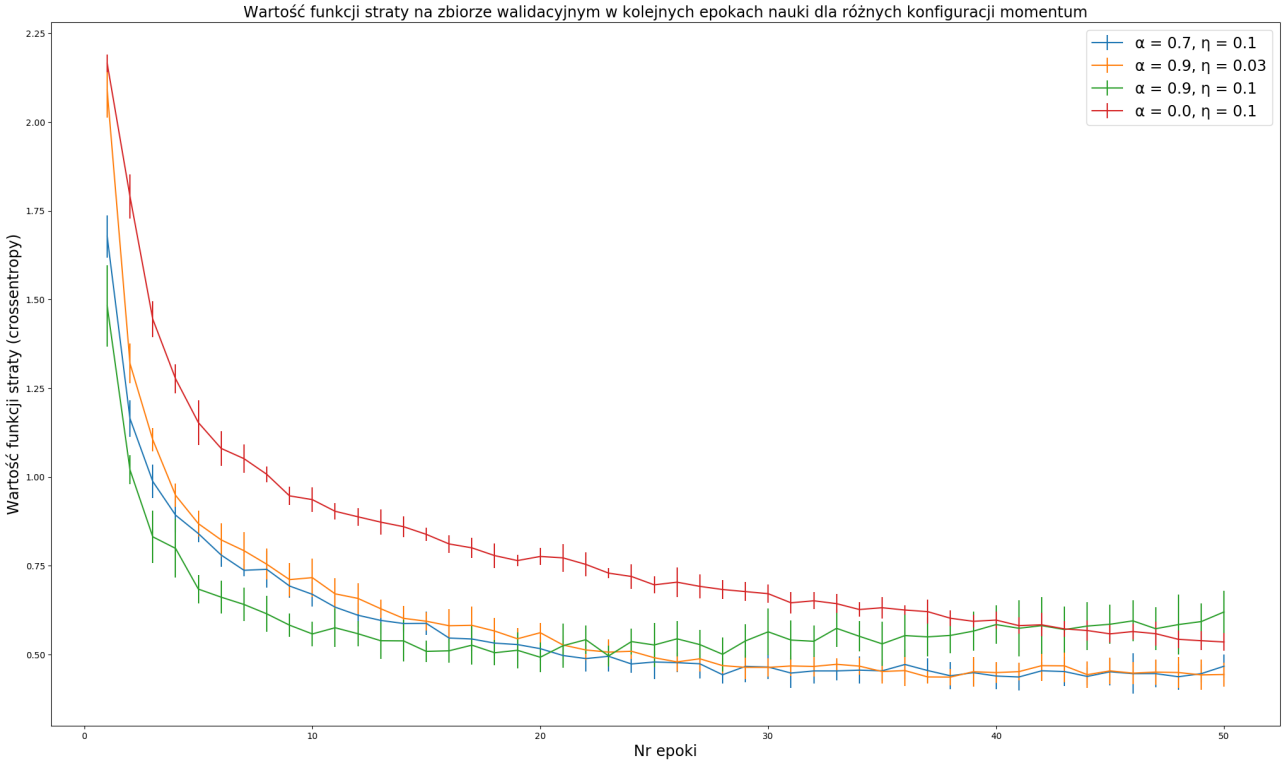
3.3 Momentum

Tym razem zbadano jak przebiega nauka sieci przy wykorzystaniu momentum i bez niego, przy różnych wartościach współczynnika momentum α i współczynnika uczenia η.

Sprawdzono cztery konfiguracje:

- bez momentum ($\alpha = 0$), $\eta = 0.1$
- $\alpha = 0.7$, $\eta = 0.1$,
- $\alpha = 0.9$, $\eta = 0.1$,
- $\alpha = 0.9$, $\eta = 0.03$,

3.3.1 Rezultaty



Rysunek 3: Wartość straty na zbiorze walidacyjnym podczas nauki przy różnych konfiguracjach momentum

3.3.2 Wnioski

Już na pierwszy rzut oka widać, że zastosowanie momentum, nawet bez dostosowywania wartości współczynnika uczenia, znacząco poprawia jakość działania rozumianą przez szybkość zbiegania wartości funkcji straty.

Ustawienie zbyt wysokiego współczynnika momentum bez dostosowania współczynnika uczenia powoduje bardzo szybkie zbieganie w początkowej fazie uczenia, ale później strata zaczyna rosnąć. Da się zauważyć też dość 'chaotyczny' przebieg wartości straty w zależnośći od epoki - podobny jak przy zbyt wysokiej wartości współczynnika uczenia.

Zmniejszenie współczynnika momentum lub dostosowanie współczynnika uczenia do zastosowanego momentum pozwala na znaczącą poprawę jakości działania takiego optymalizatora - funkcja straty zbiega szybcieej niż przy braku momentum, a charakterystyka przebiegu takiej nauki wskazuje na dużo mniejszy 'chaotyzm' przeszukiwania przestrzeni wag.

3.4 Rozmiar zbioru uczącego

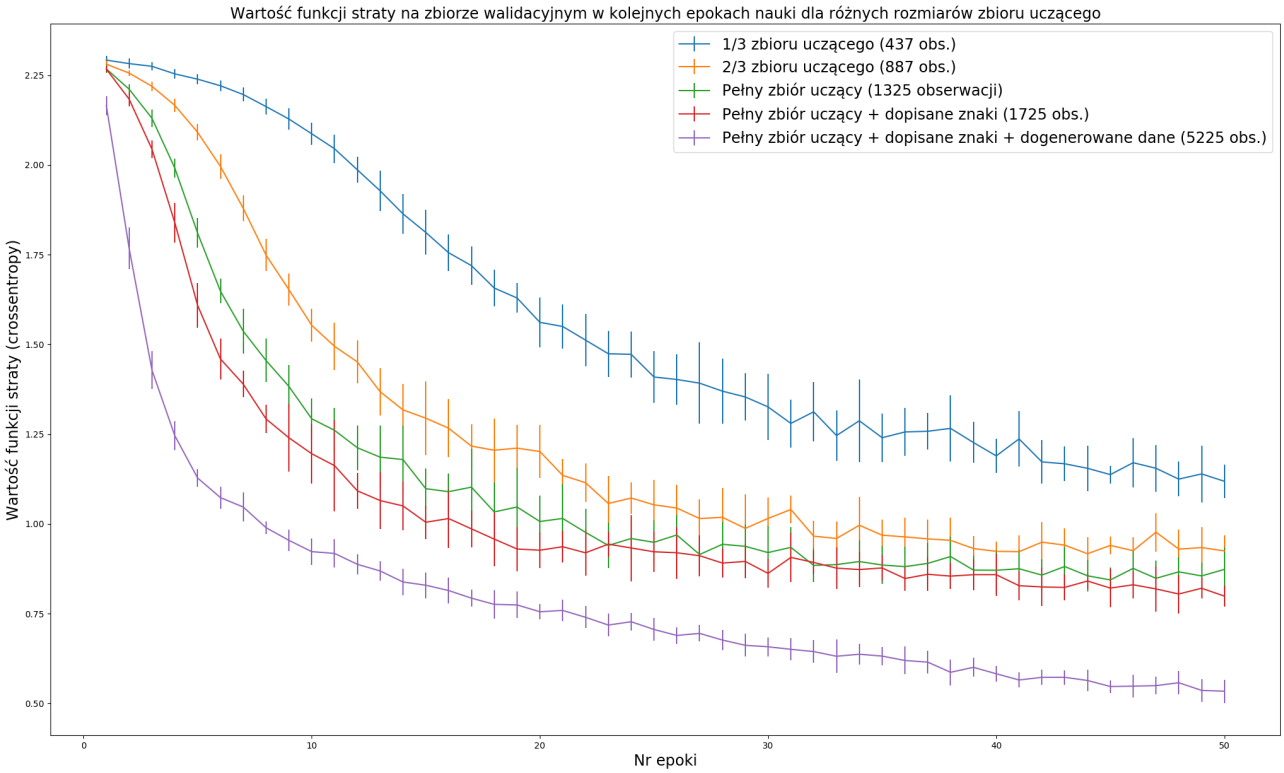
Zbadany został wpływ rozmiaru zbioru uczącego na szybkość nauki w tej samej sieci. Dodatkowo został zbadany wpływ powiększenia zbioru uczącego o przykłady sztucznie wygenerowane w oparciu o modyfikacje znanych przykładów ze zbioru uczącego.

Zbadano uczenie sieci dla następujących zbiorów uczących:

- 1/3 zbioru uczącego - 437 obserwacji
- 2/3 zbioru uczącego - 887 obserwacji
- Pełny zbiór uczący - 1325 obserwacji
- Pełny zbiór powiększony o manualnie 'dorysowane' przykłady - 1725 obserwacji
- Zbiór uczący z dodatkowymi przykładami i przykładami sztucznie wygenerowanymi - 5225 obserwacji

Warto zauważyć, że w przypadku dwóch pierwszych zbiorów (1/3 i 2/3) w każdym z 10 powtórzeń mogły one zawierać różne, losowo wybrane przykłady z całego zbioru.

3.4.1 Rezultaty



Rysunek 4: Wartość straty na zbiorze walidacyjnym podczas nauki przy różnych rozmiarach zbioru uczącego

3.4.2 Wnioski

Wniosek z przeprowadzonego badania nasuwa się sam: im większy zbiór uczący - tym szybciej zbiega wartość straty zbiorze walidacyjnym.

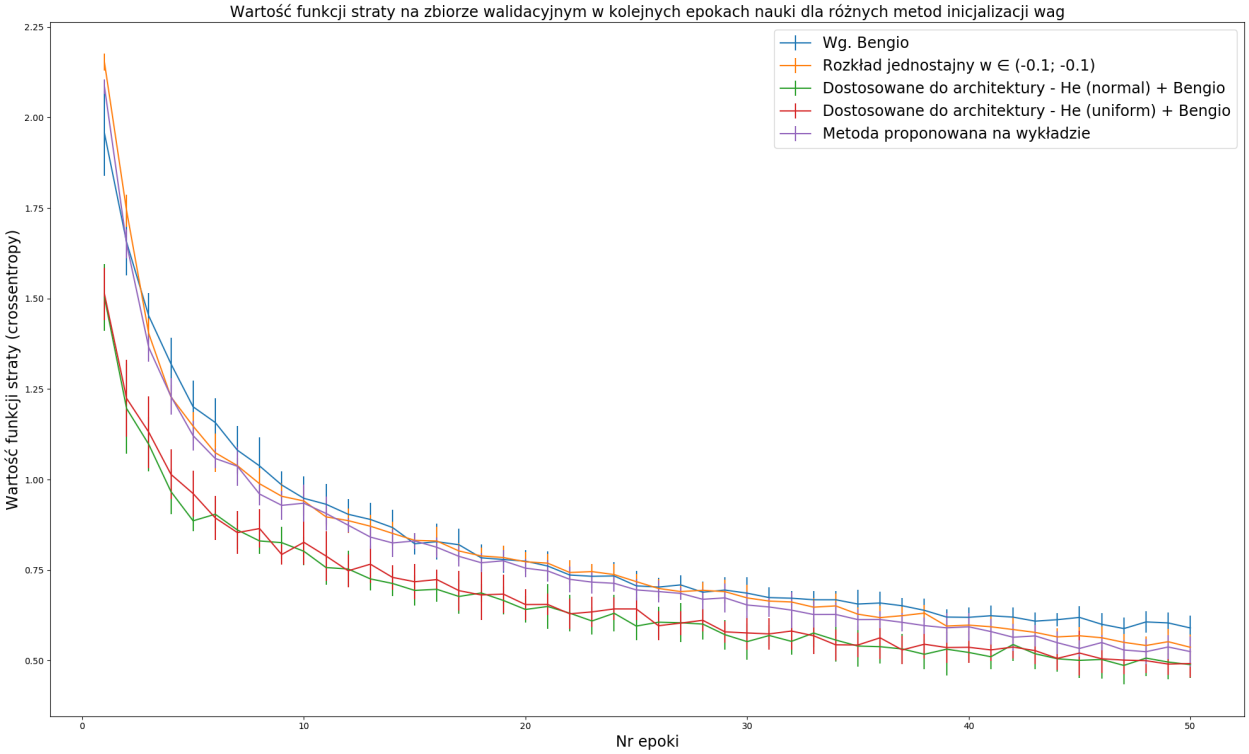
Wygenerowanie dodatkowych obserwacji z użyciem sztucznych przekształceń obrazów pozwoliło znacząco poprawić jakość nauki - wartość straty zbiega szybciej i po 50 epokach nie utrzymuje się już na stałym poziomie, jak dla mniejszych rozmiarów zbioru uczącego, ale jej przebieg wskazuje na potencjał do dalszego zbiegania.

3.5 Inicjalizacja wag początkowych

Ostatnim przeprowadzonym badaniem było sprawdzenie wpływu sposobu inicjalizacji początkowych wag w sieci na przebieg jej nauki. Wykorzystano 5 metod inicjalizacji:

- rozkład jednostajny w zakresie od -0.1 do 0.1
- metoda proponowana na wykładzie - rozkład jednostajny w zakresie od $-\frac{1}{\sqrt{n_{in}}}$ do $\frac{1}{\sqrt{n_{in}}}$
- metoda proponowana przez [Glorot and Bengio, 2010]
- metoda dostosowana do architektury sieci (do f. aktywacji) - metoda proponowana przez [He et al., 2015] z rozkładem jednostajnym dla warstwy z aktywacją Softplus i metoda [Glorot and Bengio, 2010] dla warstwy wyjściowej (Sigmoid)
- analogicznie do powyższej - z użyciem rozkładu normalnego w metodzie [He et al., 2015]

3.5.1 Rezultaty



Rysunek 5: Wartość straty na zbiorze walidacyjnym podczas nauki przy różnych inicjalizacjach wag

3.5.2 Wnioski

Zastosowanie odpowiedniej inicjalizacji wag pozwala uzyskać lepszy punkt startowy do uczenia sieci, co daje szansę na szybsze zbieganie funkcji straty. Dobór sposobu inicjalizacji odpowiedniego dla uczonej sieci okazuje się być kluczowy. Użycie tego samego sposobu inicjalizacji dla każdej warstwy, niezależnie od stosowanej w niej funkcji aktywacji okazało się dawać bardzo zbliżone rezultaty, z lekką przewagą inicjalizacji proponowanej na wykładzie nad metodą losową całkowicie niezależną od rozmiaru sieci. Jednak zastosowanie metody inicjalizacji proponowanej dla warstw z aktywacją typu ReLu (a więc także Softplus) - metody He ([He et al., 2015]), pozwoliło dać sieci dużo lepszy punkt startowy i osiągnąć lepsze wyniki uczenia.

Literatura

[Glorot and Bengio, 2010] Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In Teh, Y. W. and Titterington, M., editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.

[He et al., 2015] He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852.