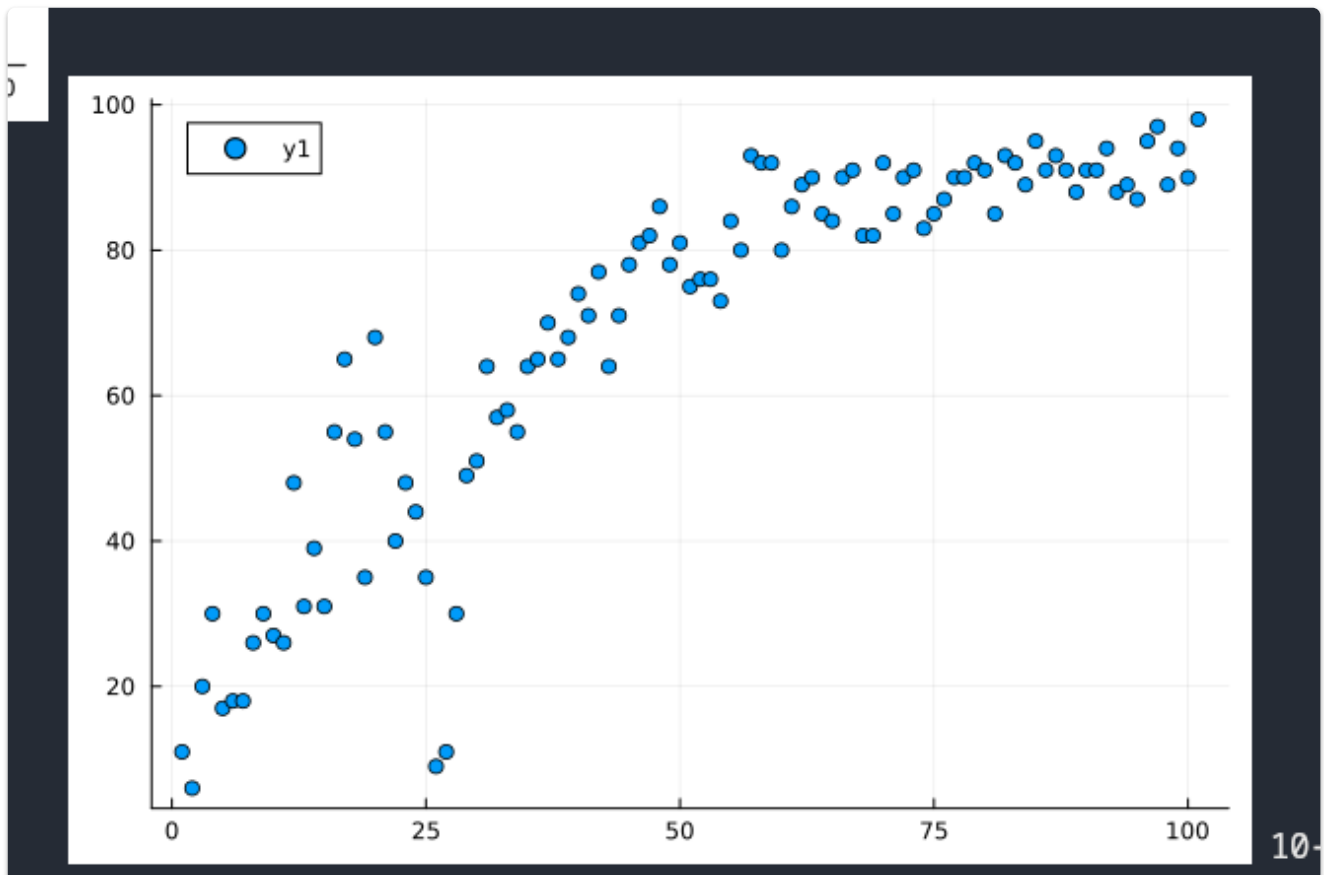


Podsumowanie Kamienia Milowego 2

Na poniższym zrzucie ekranu przedstawiona została skuteczność sieci na przestrzeni 100 batchy o rozmiarze 100. W tym przykładzie skuteczność była liczona jedynie dla wycinka zbioru testowego (za każdym razem zbiór testowy był mieszany), by uzyskać skrócić czas działania.



Mówiąc o czasie działania jest on następujący.

```
0.001048 seconds (13.51 k allocations: 2.891 MiB)
0.006966 seconds (51.02 k allocations: 6.749 MiB)
```

Pierwszy wpis odpowiada przejściu sieci w przód, drugi odpowiada za przejście sieci w tył.

Sama struktura sieci bazuje na pierwszej architekturze z pliku referencyjnego ostatecznie jednak lekko od niej odbiega

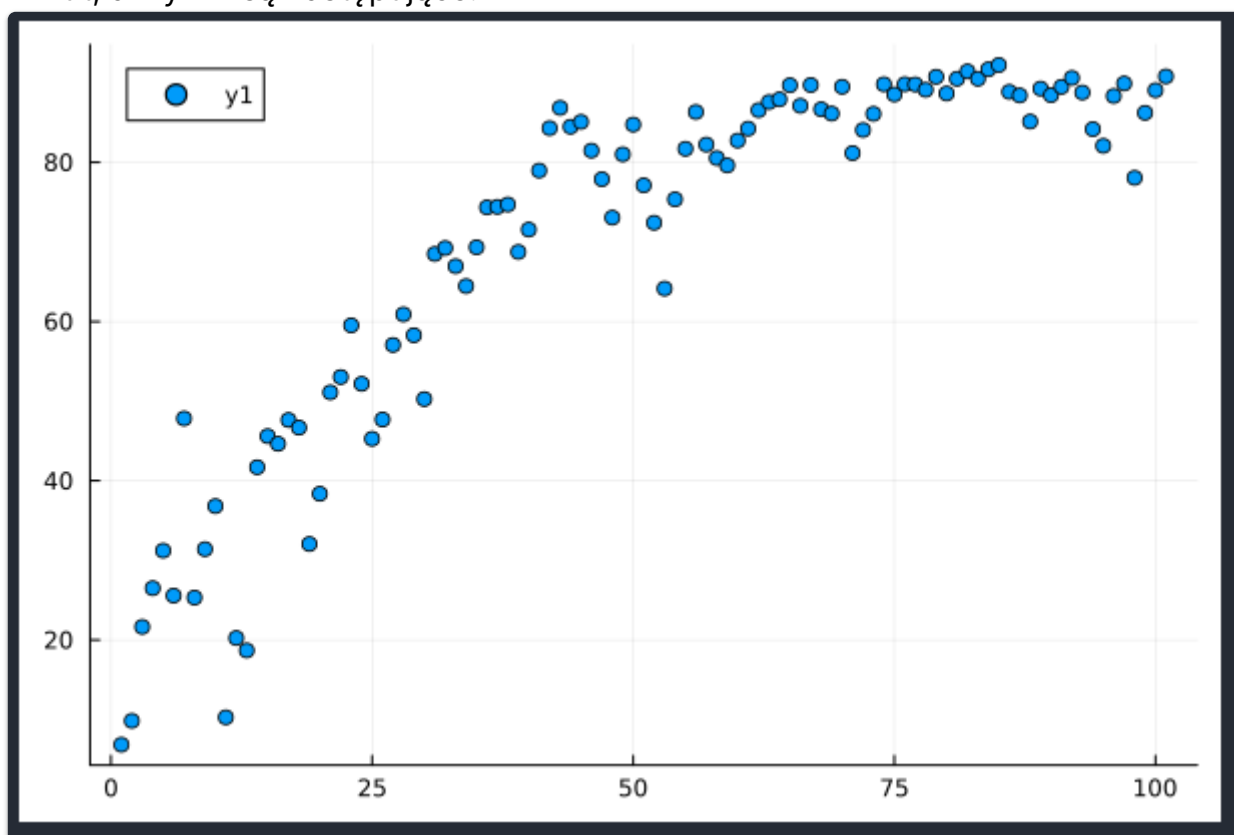
Całościowy czas nauki sieci (wraz z liczeniem accuracy sieci dla wycinka zbioru testowego o rozmiarze 100) 100 batch'ów każdy o rozmiarze 100 wynosi około 2 minut 17 sekund.

Jak łatwo zauważyć przerabiany jest zaledwie wycinek zbioru treningowego bo 10000 z 60000 próbek. Żeby sieć została uczona przez 3 pełne epoki (tak jak w rozwiązaniu referencyjnym) trwałoby to około 40 minut.

W związku z nie optymalnością kodu musiałem pójść na pewne ustępstwa związane z samymi parametrami sieci, w tym m.in. zrezygnowałem na ten moment na wykorzystanie pełnych epok. By uczyła się ona szybciej zwiększyłem krok z 0.01 do 0.1. Ponadto sam gradient nie jest średnią z całego batch'a a sumą. Te wszystkie przedsięwzięcia pozwoliły mi na osiągnięcie przedstawionej na pierwszym wykresie skuteczności.

Podczas pracy zauważyłem znaczną różnicę działania sieci, kiedy jej wagi inicjowane były za pomocą funkcji `randn()`, a kiedy za pomocą własnoręcznie zaimplementowanej funkcji `glorot_uniform()`. Przy pierwszej opcji, wartości często były ogromne, co najczęściej prowadziło do uzyskania NaN'ów. Przy pomocy `glorot_uniform`, problem ten zniknął.

W pracy pozwoliłem sobie jednorazowo dla aktualnej wtedy struktury i implementacji sieci wykorzystać cały zbiór testowy do wyznaczania accuracy, cały proces trwał 39 minut, a wyniki są następujące.



By wypróbować implementację należy uruchomić plik `finalNetwork_1.jl` znajdujący się w katalogu `./src/data/` razem z innymi sieciami testowymi powstałymi w czasie pracy nad projektem

Podsumowując, pod kątem optymalizacji projekt wymaga sporo poprawy, natomiast pomimo tego jestem zadowolony z całego procesu tworzenia tego projektu oraz efektu tej części projektu.

Jakub Maciejewski

Nr albumu: 311343