
Sprawozdanie laboratorium 1

Maciej Urbaniak

Abstract

Wielowarstwowa sieć jednokierunkowa MLP w Tensorflow – celem ćwiczenia jest wprowadzenie do biblioteki Tensorflow i przypomnienie podstawowej sieci MLP i wpływu hiperparametrów na uczenie i jakość otrzymywanych wyników.

1 Regularyzacje

Testy przeprowadzono dla następujących hiperparametrów:

| | | |
|-----------------|---------------------|---------------|
| Epoki | 30 | |
| AdamOptimizer | learning rate:0.005 | |
| Zbiór trenujący | 30000 | |
| Wielkość batchy | 1000 | |
| Warstwy ukryte | 5 (relu) | wielkość: 256 |

1.1 Brak regularyzacji

Wyniki tego eksperymentu będą odniesieniem dla kolejnych.
Najlepszy wynik na zbiorze testowym nastąpił w 16 epoce i wyniósł 27,5%.

1.2 Regularyzacja L2

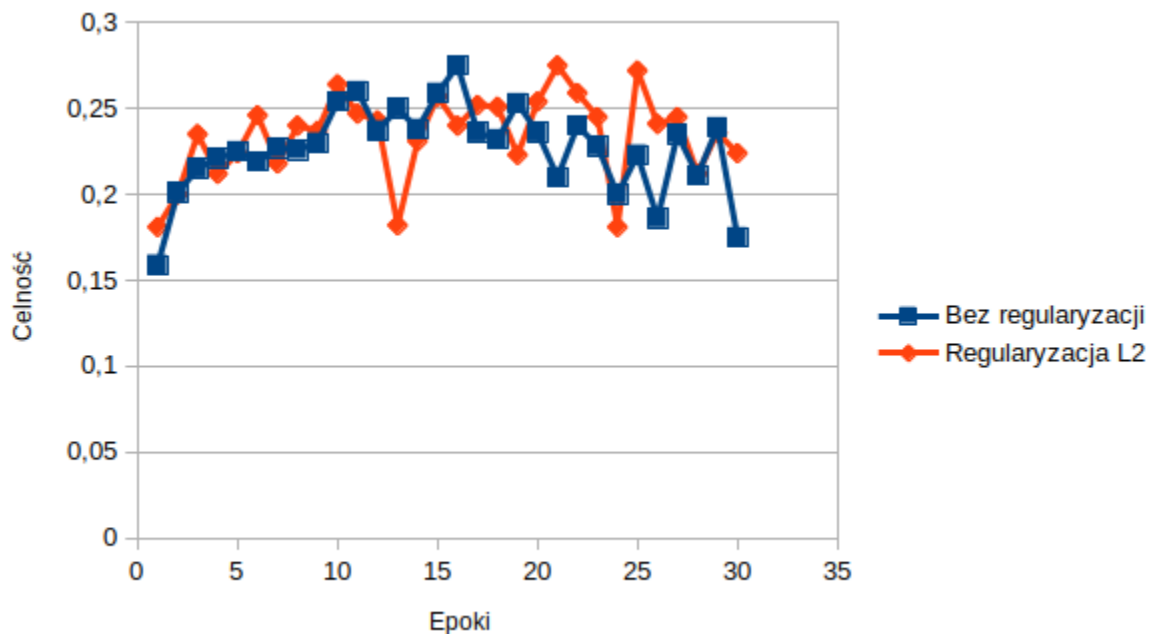
Regularyzację L2 stosujemy do uniknięcia dopasowania sieci do danych trenujących (ang. overfitting).
Najlepszy wynik wyniósł 27,5% dla 21 epoki.

Na rysunku 1 widać pozytywny wpływ na szybkość zbieżności do większej celności predykcji, co ma znaczenie dla obliczeń przy niewielkiej liczbie epok. Można też zauważyć mniejsze przeuczenie dla późniejszych epok.

Regularyzacja ta będzie wykorzystywana w dalszych eksperymentach.

1.3 Dropout

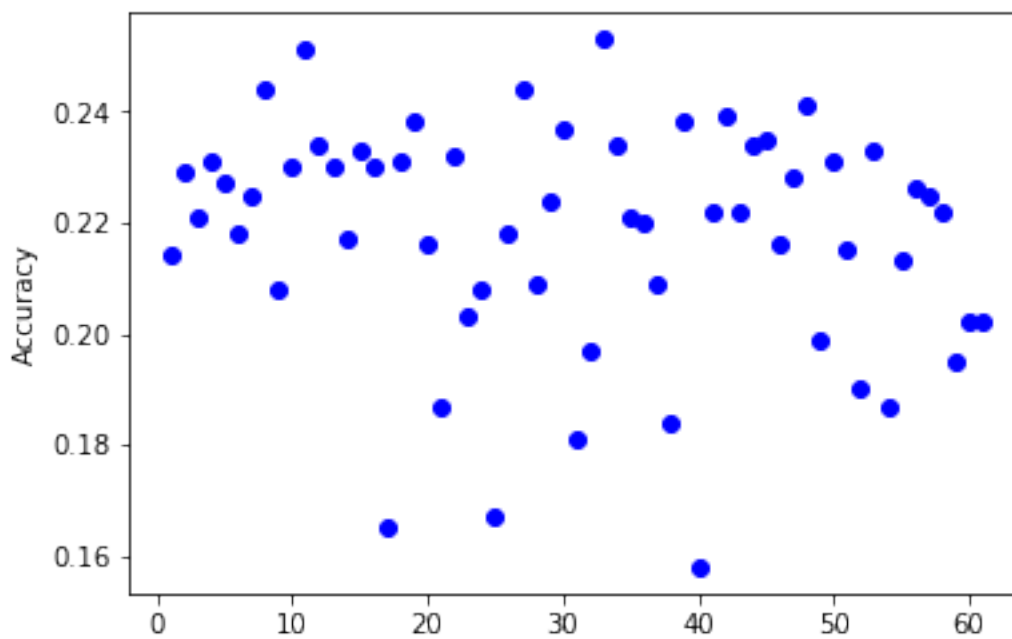
Testy przeprowadzono dla modelu PP (poprawionego). Dodano dropout o wartości 0.8 w drugiej warstwie ukrytej.



Rysunek 1: Porównanie celności predykcji dla regularyzacji L2 i jej braku.

2 Różna liczba iteracji w procesie uczenia

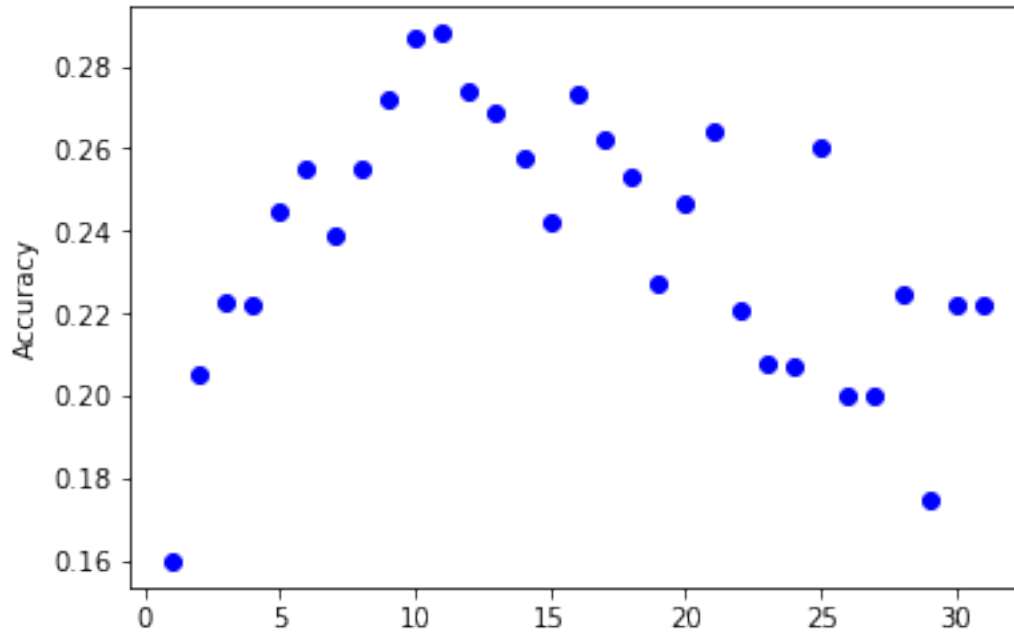
Eksperyment dla różnej liczby iteracji został przeprowadzony "przy okazji" eksperymentu pierwszego. Wraz ze zwiększaniem się epok następuje wzrost przeuczenia, który wpływa na wyniki szczególnie przy braku regularyzacji. Wraz z kolejnymi epokami celność zaczyna stopniowo spadać, co można zaobserwować na rysunku 2 (zbadano do 60 epoki).



Rysunek 2: Celność dla dalszych epok.

3 Liczba warstw

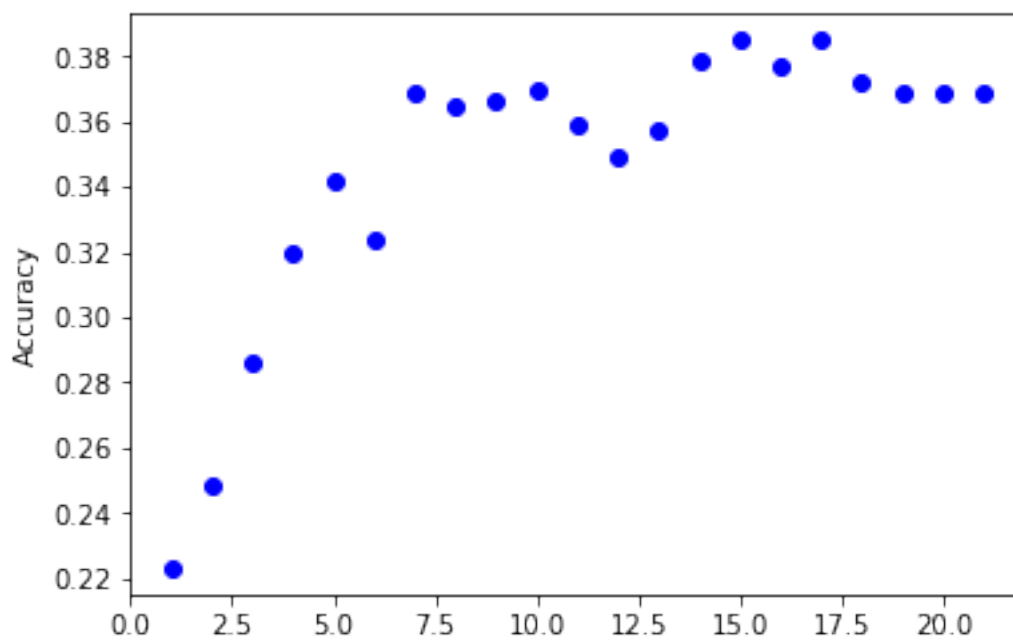
Więcej warstw wpływa na czas obliczeń jak i użycie pamięci. Przy eksperymencie przeprowadzonym dla sieci o 2 warstwach ukrytych, można zauważyć gwałtowny wzrost a następnie postępujący spadek celności predykcji (rysunek 3). Przeprowadzono również test dla 10 warstw, uzyskane wyniki były w swej strukturze podobne do sieci 5 warstwowej.



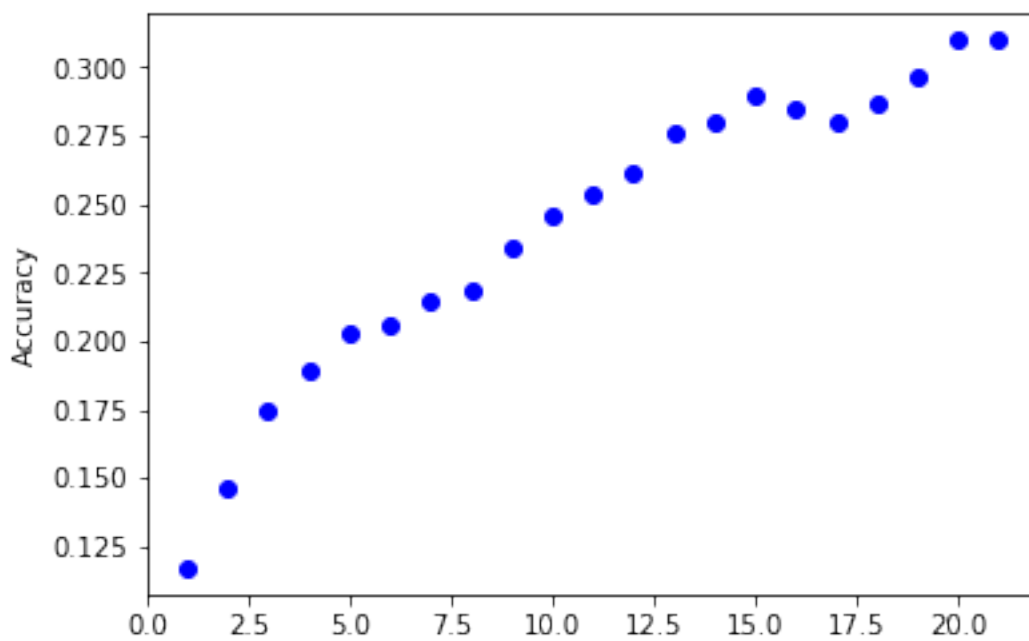
Rysunek 3: Celność dla 2 warstw.

4 Różne licznosci paczek (mini batch)

Dla paczek o wielkości 100 następuje znaczne wydłużenie obliczeń. Celność predykcji dla dużych paczek (10000) zwiększa się stopniowo z każdą epoką bez występowania nagłych spadków, ale rośnie ona znacznie wolniej. Poniższe wyniki uzyskano na modelu PP (poprawionym) o zmniejszonej liczbie epok (20).



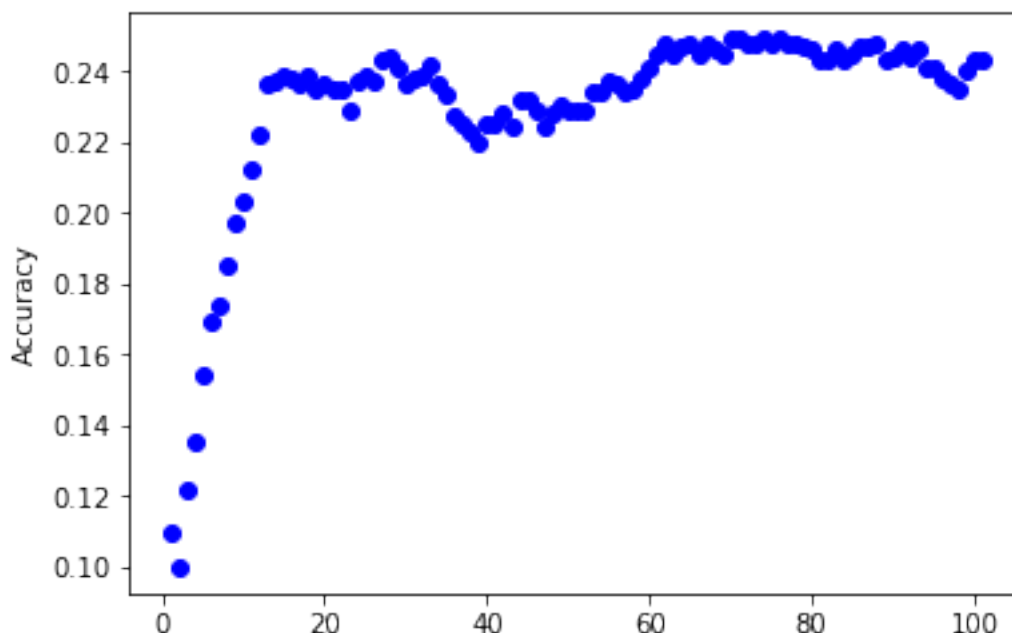
Rysunek 4: Celność model PP batch 100 elementów.



Rysunek 5: Celność model PP batch 1000 elementów.

5 Wypróbuj różne metody optymalizacji współczynnika uczenia

Przeprowadzono badania z użyciem "AdamOptimizer", "AdagradOptimizer" oraz "AdadelatOptimizer" ze współczynnikiem uczenia 0.005. Najlepsze wyniki to kolejno dla powyższych: 27,5%, 24,7% i 7,9%. Do testów badających wpływ współczynnika uczenia wybrano "AdamOptimizer". Zwiększanie wartości powodowało spadek celności natomiast przy wartości 0,0001 odnotowano niewielki stały wzrost. Po zwiększeniu liczby epok do 100 wzrost ustabilizował się co można zobaczyć na rysunku 4. Wiąże się to z wpadnięciem w minimum lokalne.



Rysunek 6: "AdamOptimizer" o współczynniku uczenia 0,0001

6 Najlepszy model

Ostatecznie modelem z najwyższą odnotowaną celnością predykcji był ten o następujących hiperparametrach.

Regularyzacja L2

30 epok

AdamOptimizer z learning rate o wartości 0.005

Wielkość zbioru trenującego 30000

Wielkość batchy 1000

5 warstw ukrytych (każda o rozmiarze 256)

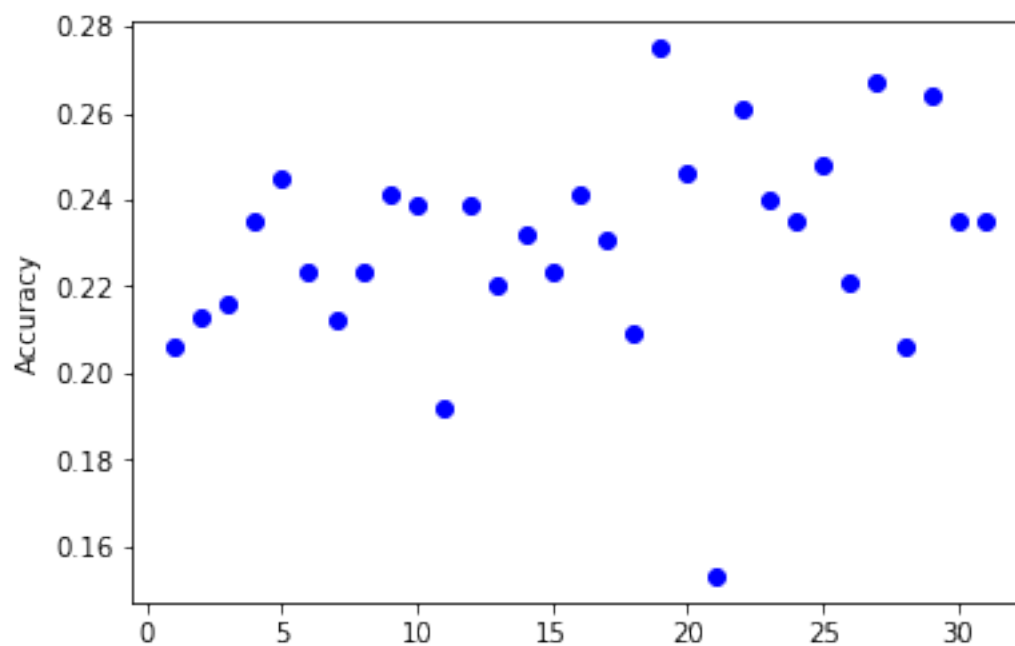
Klasa najczęściej nierozpoznawana: żaba (frog)

Klasy z największą liczbą obiektów: samolot (airplane), jelen (deer) oraz statek (ship)

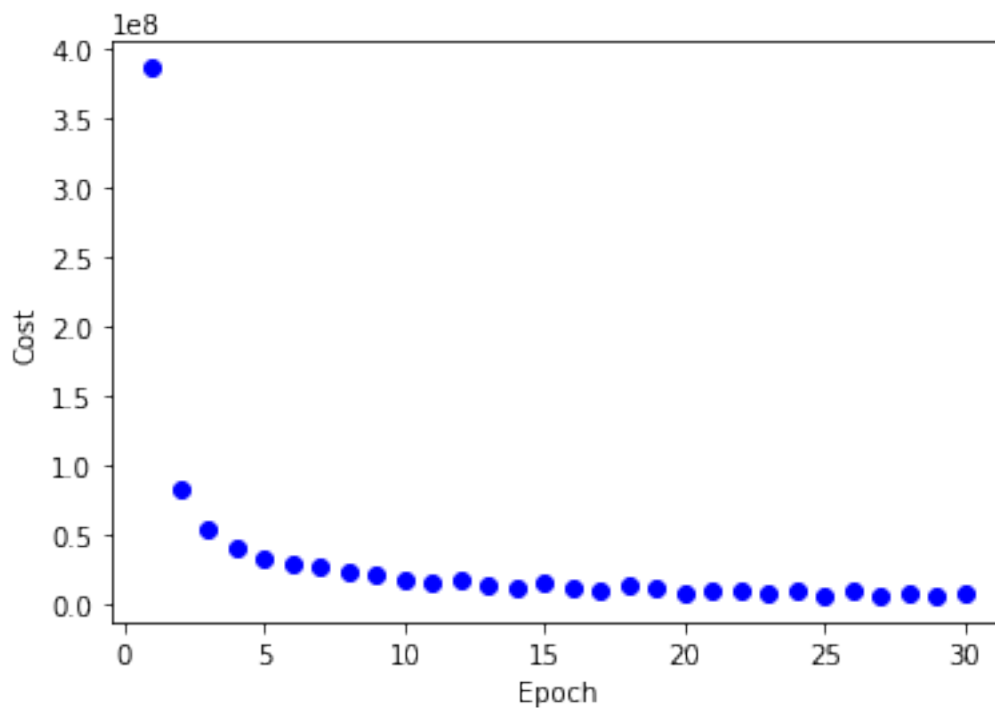
Klasa najlepiej rozpoznawana: statek (ship)

Macierz pomyłek dla klas "airplane", "automobile", "bird", "cat", "deer", "dog", "frog", "horse", "ship", "truck":

| | | | | | | | | | |
|----|----|---|----|----|----|---|---|----|----|
| 43 | 4 | 3 | 0 | 12 | 3 | 1 | 1 | 31 | 5 |
| 13 | 12 | 2 | 5 | 10 | 6 | 0 | 4 | 20 | 17 |
| 32 | 1 | 6 | 3 | 31 | 6 | 0 | 0 | 16 | 5 |
| 18 | 4 | 4 | 13 | 31 | 10 | 0 | 4 | 11 | 8 |
| 17 | 0 | 5 | 3 | 42 | 1 | 0 | 4 | 10 | 8 |
| 15 | 3 | 3 | 6 | 28 | 10 | 0 | 0 | 14 | 7 |
| 20 | 4 | 8 | 2 | 38 | 17 | 1 | 4 | 12 | 6 |
| 17 | 3 | 6 | 6 | 37 | 5 | 0 | 7 | 14 | 7 |
| 24 | 1 | 2 | 1 | 7 | 2 | 0 | 0 | 66 | 3 |
| 19 | 8 | 6 | 4 | 7 | 2 | 0 | 2 | 26 | 35 |



Rysunek 7: Wykres celności wybranego modelu.



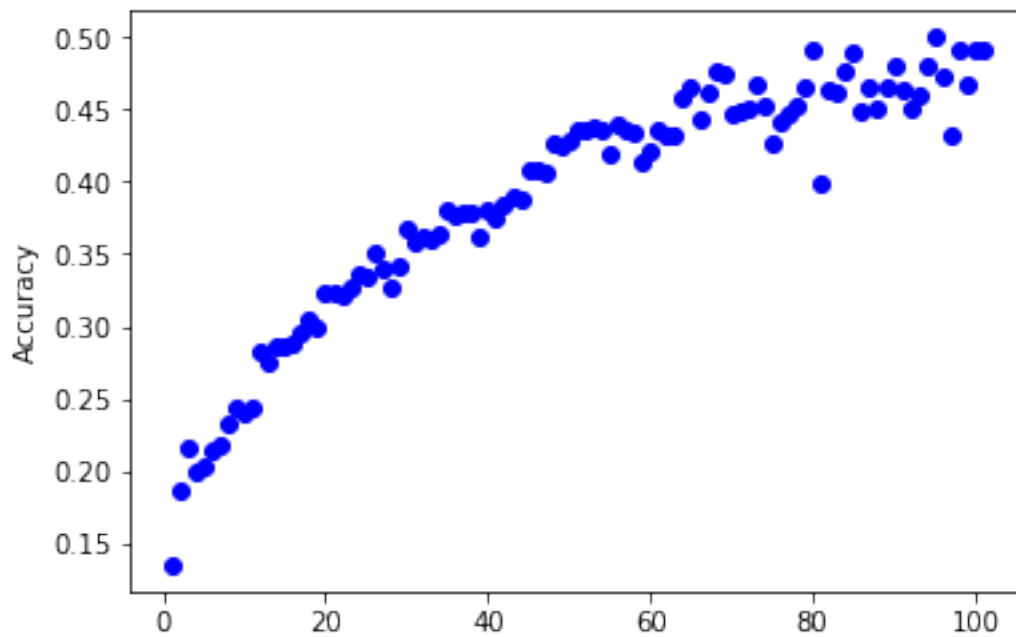
Rysunek 8: Wykres funkcji kosztu wybranego modelu.

7 Poprawiony model

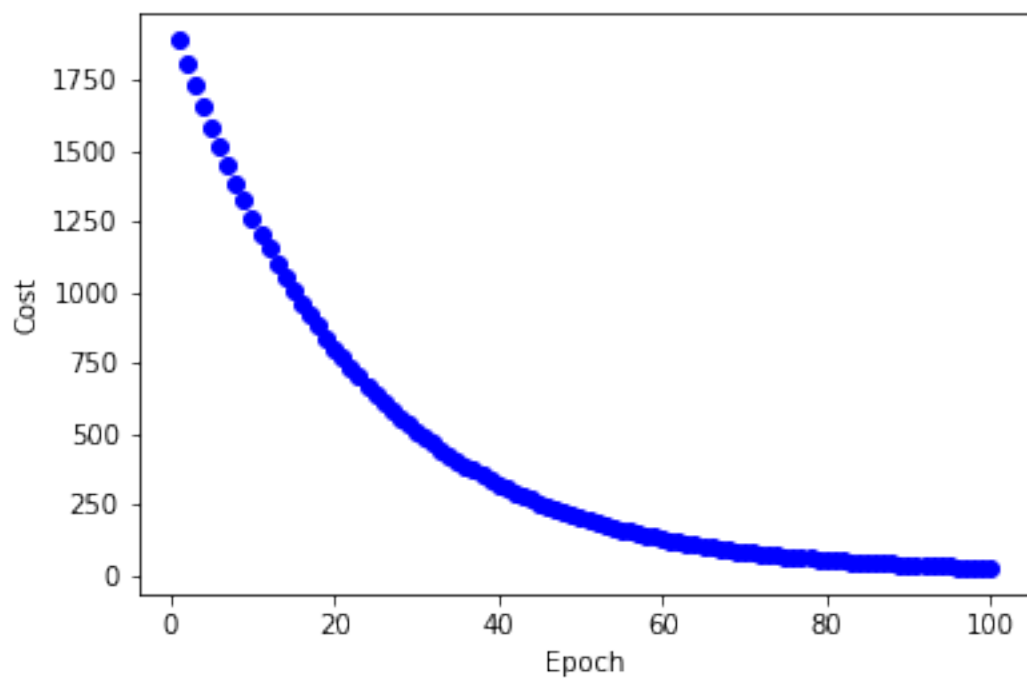
W celu poprawy klasyfikacji znormalizowano wartości kolorów poprzez podzielenie przez 255. Następnie zmieniono warstwy aktywacji na sigmoidalne. Model ten oznaczany będzie przez PP.

Ostateczne hiperparametry:

| | | |
|-----------------|---------------------|-----------------------------------|
| Epoki | 100 | |
| AdamOptimizer | learning rate:0.001 | |
| Zbiór trenujący | 30000 | |
| Wielkość batchy | 1000 | |
| Warstwy ukryte | 5 (sigmoid) | wielkość: 1024, 512, 256, 128, 64 |



Rysunek 9: Wykres celności poprawionego modelu PP.



Rysunek 10: Wykres funkcji kosztu poprawionego modelu PP.

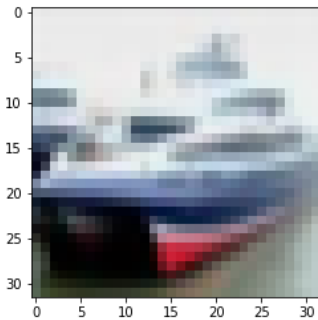
7.1 Analiza wybranych przypadków klasyfikacji

Poniżej znajduje się macierz pomyłek dla modelu PP.

| | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|
| 44 | 0 | 3 | 2 | 5 | 2 | 7 | 2 | 33 | 5 |
| 8 | 11 | 0 | 1 | 0 | 4 | 5 | 2 | 31 | 27 |
| 10 | 0 | 24 | 5 | 23 | 10 | 14 | 5 | 8 | 1 |
| 4 | 0 | 6 | 21 | 9 | 22 | 25 | 5 | 6 | 5 |
| 7 | 0 | 4 | 3 | 39 | 7 | 17 | 5 | 5 | 3 |
| 0 | 0 | 3 | 10 | 11 | 37 | 13 | 6 | 4 | 2 |
| 1 | 0 | 1 | 4 | 18 | 6 | 78 | 0 | 3 | 1 |
| 5 | 1 | 1 | 8 | 14 | 16 | 11 | 41 | 4 | 1 |
| 8 | 0 | 0 | 1 | 2 | 1 | 0 | 1 | 89 | 4 |
| 7 | 0 | 0 | 2 | 1 | 5 | 5 | 3 | 25 | 61 |

Najczęściej poprawnie klasyfikowanymi obrazkami były statki. Na rysunku 11 zobaczyć można przykład takiej klasyfikacji oraz duży współczynnik pewności modelu dla tego przyporządkowania.

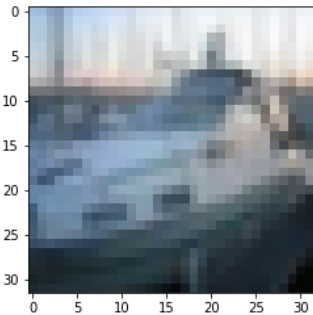
```
Class probability: [[3.1266403e-03 3.3920526e-03 5.6741510e-06 5.4547381e-06 1.6256422e-04  
6.5986702e-04 1.1340605e-05 1.7007494e-03 9.9093401e-01 1.6404074e-06]]  
Model output [ True]
```



Rysunek 11: Poprawna klasyfikacja statku

Przykład niepoprawnej klasyfikacji na rysunku 12 jako frog (żaba) dla statku. Najbardziej prawdopodobną przyczyną niepoprawnej klasy w tym przypadku zdaje się być kształt łodzi, który może przypominać żabę.

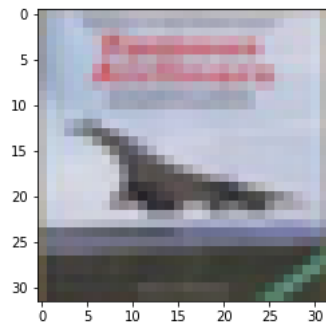
```
Class probability: [[2.7455338e-10 3.9343140e-05 9.3411791e-06 7.2382405e-02 1.4885637e-07  
1.9227996e-02 9.0816671e-01 7.7310411e-08 7.3078986e-08 1.7397539e-04]]  
Model output [False]
```



Rysunek 12: Niepoprawna klasyfikacja statku

Poniżej przedstawiono obrazek samolotu przyporządkowany do klasy samochodu. Prawdopodobna przyczyna to bliskość obiektu do ziemi.

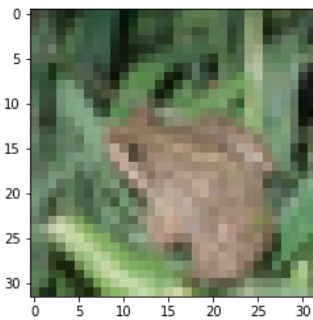
```
Class probability: [[1.2948040e-02 6.8871665e-01 1.7130747e-05 2.6229268e-02 1.1685564e-03
2.6687539e-01 3.1375568e-03 9.0214994e-04 4.0455579e-06 1.1967248e-06]]
Model output [False]
```



Rysunek 13: Niepoprawna klasyfikacja samolotu

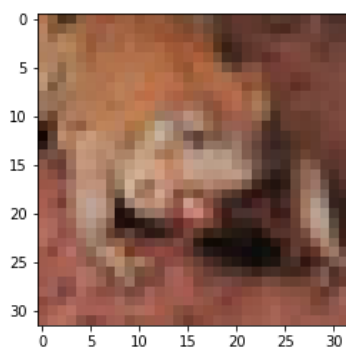
Poniżej przedstawiono zaklasyfikowanie dwóch żab jako samoloty. Klasa frog charakteryzowała się tym że zebrała w sobie wiele różnych obiektów. Może to wynikać z różnego kształtu i koloru co można zobaczyć na dwóch poniższych rysunkach.

```
Class probability: [[9.9854553e-01 1.5343316e-07 1.3985431e-09 8.3431378e-10 1.4370821e-03
7.2175023e-11 1.7080198e-05 2.6122422e-07 8.0997724e-09 4.1678154e-09]]
Model output [False]
```



Rysunek 14: Niepoprawna klasyfikacja żaby

```
Class probability: [[8.74456048e-01 1.53316284e-06 5.37285954e-03 1.84592463e-05  
2.65914946e-06 1.20055586e-01 1.78878508e-07 8.47364208e-05  
1.23400568e-07 7.78246522e-06]]  
Model output [False]
```



Rysunek 15: Niepoprawna klasyfikacja żaby