AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE
Wydział Zarządzania

# Projekt II
# Sztuczne Sieci Neuronowe -
# Klasyfikacja otyłości

Autorzy: Marcin Mika, Jakub Sornat
Kierunek studiów: Informatyka i Ekonometria

# 1. Opis

Tematem projektu jest przypisanie osób do odpowiedniej kondycji fizycznej na podstawie informacji o mieszkańcach Meksyku, Peru i Kolumbii dotyczących ich stylu życia. Dane zastosowane pochodzą ze strony https://www.kaggle.com/datasets/mandysia/obesity-dataset-cleaned-and-data-sinthetic.
Użyte zmienne:

- Płeć: mężczyzna lub kobieta
- Wiek
- Historia rodziny z nadwagą: Czy członek rodziny cierpiał lub cierpi na nadwagę? - tak lub nie
- FAVC: Częste spożywanie wysokokalorycznych potraw - tak lub nie
- FCVC: Częstotliwość spożywania warzyw - Nigdy, Czasami, Zawsze
- NCP: Liczba głównych posiłków - 1, 2, 3, 4
- CAEC: Spożycie jedzenia między posiłkami - Nie, Czasami, Często, Zawsze
- SMOKE: Czy osoba pali? - tak lub nie
- CH2O: Dzienne spożycie wody - Mniej niż litr, między 1 a 2 l, więcej niż 2 l
- SCC: Monitorowanie spożycia kalorii - tak lub nie
- FAF: Częstotliwość aktywności fizycznej - 0, 1 do 2, 2 do 4, 4 do 5
- TUE: Czas korzystania z urządzeń technologicznych - 0 do 2, 3 do 5, >5
- CALC: Spożycie alkoholu - nie, czasami, często, zawsze
- MTRANS: używany środek transportu- samochód, motocykl, rower, transport publiczny, chodzenie

Grupy kondycji fizycznej (dopasowane na podstawie BMI – Waga/(Wzrost)^2):

- Niedowaga ( BMI do ~18,5)
- Waga normalna (BMI do ~25)
- Nadwaga - poziom I (BMI do ~27)
- Nadwaga - poziom II (BMI do ~30)
- Otyłość typu I (BMI do ~34)
- Otyłość typu II (BMI do ~39)
- Otyłość typu III (BMI od ~39)

*Na granicach przedziałów klasyfikacja zależy jeszcze od innych zmiennych, np. 26 letni mężczyzna z BMI 24,91 ma normalną wagę, a 16 letnia dziewczyna z BMI 24,9 jest klasyfikowana jako osoba z nadwagą (poziom I).

Nie skorzystaliśmy ze zmiennych Waga, Wzrost i BMI ponieważ z nich wynika dopasowanie do grupy

## 2. Przegląd literatury

Predykcja stanu otyłości była analizowana między innymi w artykule opublikowanym 18 marca ubiegłego roku pod tytułem ”Estimation of Obesity Levels with a Trained Neural Network Approach optimized by the Bayesian Technique”. Głównym celem tego badania było opracowanie wytrenowanego modelu ML opartego na sieciach neuronowych do przewidywania poziomów otyłości na podstawie informacji socjogeograficznych, statusu aktywności fizycznej oraz różnych nawyków żywieniowych, co zdecydowanie pokrywa się z wybranym przez nas tematem. Okazało się, że korzystaliśmy z prawie identycznego zestawu danych, co autorzy tego artykułu. Jedyna różnica to ilość obiektów (u nas około 2000, w artykule około 500) - badana była nawet ludność tych samych państw. Średnia dokładność modelu została oceniona na aż **93.06%** (na zbiorze testowym). Jest to zdecydowanie lepszy wynik od naszego. Wpłynęło na to wiele czynników, np. lepszy dobór zmiennych objaśniających.

Kolejnym badaniem przewidującym otyłość jest artykuł opublikowany w 2021 pod tytułem „Classification and Prediction on the Effects of Nutritional Intake on Overweight/Obesity, Dyslipidemia, Hypertension and Type 2 Diabetes Mellitus Using Deep Learning Model: 4–7th Korea National Health and Nutrition Examination Survey”. Prognozowano tu również wystąpienie innych chorób takich jak dyslipidemia czy cukrzyca na podstawie Koreańskiego Krajowego Badania Zdrowia i Żywienia. Model zaprezentowany w artykule składa się z jednej warstwy wejściowej z 7 węzłami, trzech warstw ukrytych z 30, 12 i 8 węzłami w każdej warstwie oraz jednej warstwy wyjściowej z jednym węzłem. Dokładność przewidywania  nadwagi/otyłości wyniosła 62,496%. Wyniki porównano z regresja logistyczna i drzewem decyzyjnym – okazało się, że DNN jest najbardziej dokładne.

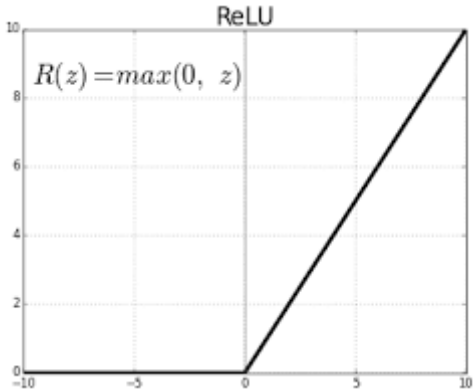
## 3. Analiza wpływu różnych parametrów

### Model bazowy:

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 32)                480

 dense_1 (Dense)             (None, 32)                1056

 dense_2 (Dense)             (None, 7)                 231

=================================================================
Total params: 1767 (6.90 KB)
Trainable params: 1767 (6.90 KB)
Non-trainable params: 0 (0.00 Byte)
```

**Opis Modelu:**

1. **Warstwa wejściowa:** Model rozpoczyna się od warstwy wejściowej, która ma 14 neuronów odpowiadających 14 kolumnom wejściowym danych.
2. **Pierwsza warstwa ukryta:** Po warstwie wejściowej znajduje się warstwa ukryta z 32 neuronami. Aktywacja tego warstwy jest realizowana przez funkcję aktywacji ReLU (Rectified Linear Unit), co pozwala na nieliniową transformację danych wejściowych.
3. **Druga warstwa ukryta:** Następnie mamy kolejną warstwę ukrytą z 32 neuronami, również z funkcją aktywacji ReLU. Wykres funkcji wygląda następująco:




https://www.researchgate.net/figure/Graph-of-ReLu-activation-function_fig3_343675998

4. **Warstwa wyjściowa:** Ostatnia warstwa ma 7 neuronów, które reprezentują 7 różnych klas (Niedowaga, Waga normalna, Nadwaga - poziom I, Nadwaga - poziom II, Otyłość typu I, Otyłość typu II, Otyłość typu III). Aktywacja tej warstwy jest realizowana przez funkcję aktywacji softmax, co pozwala na uzyskanie prawdopodobieństw przynależności do poszczególnych klas.
5. **Funkcja straty:** Model używa funkcji straty kategorycznej entropii krzyżowej (**categorical_crossentropy**), która jest często używana w problemach klasyfikacji wieloklasowej.

6. **Optymalizator:** Wybraliśmy optymalizator SGD (Stochastic Gradient Descent). SGD to popularny algorytm optymalizacji, który aktualizuje wagi modelu w kierunku, który zmniejsza wartość funkcji straty. interfejs Keras przeprowadzi algorytm propagacji wstecznej (tzn. odwrotne różniczkowanie automatyczne wraz z algorytmem gradientu prostego)
7. **Metryki:** Do oceny wydajności modelu używana jest **dokładność (accuracy):** Mierzy procent poprawnie sklasyfikowanych próbek.
8. **Liczba epok:** Model jest uczony przez 100 epok. Epoka oznacza, że każda próbka w zestawie treningowym została użyta raz do aktualizacji wag modelu.
9. **Podział danych:** Dane zostały podzielone na trzy zestawy: treningowy (70% danych), walidacyjny (15% danych) i testowy (15% danych).

Na zbiorze treningowym model uzyskał około 74,75% dokładności, a na walidacyjnym 68%

```
Epoch 100/100

 1/47 [..............................] - ETA: 0s - loss: 0.5613 - accuracy: 0.7812
47/47 [==============================] - 0s 1ms/step - loss: 0.7059 - accuracy: 0.7475 - val_loss: 0.9092 - val_accuracy: 0.6845
```

Na zbiorze testowym model wykazał 75,08% dokładności (lepiej niż na zbiorze treningowym, co nie zdarza się często):

```
 1/10 [==>...........................] - ETA: 0s - loss: 1.0252 - accuracy: 0.7188
10/10 [==============================] - 0s 889us/step - loss: 0.8959 - accuracy: 0.7508
```

Jako ciekawostkę, sprawdźmy prognozę dla 5 obiektów z zbioru testowego:
Przewidywana procentowa przynależność do danej kategorii:

| | insufficient_w | normal_w | obesity_t1 | obesity_t2 | obesity_t3 |
|---|---|---|---|---|---|
| 0 | 0.92 | 0.07 | 0.00 | 0.00 | 0.00 |
| 1 | 0.73 | 0.18 | 0.04 | 0.00 | 0.03 |
| 2 | 0.35 | 0.35 | 0.00 | 0.01 | 0.00 |
| 3 | 0.99 | 0.01 | 0.00 | 0.00 | 0.00 |
| 4 | 0.00 | 0.00 | 0.01 | 0.00 | 0.98 |

| | overweight_l1 | overweight_l2 |
|---|---|---|
| 0 | 0.00 | 0.01 |
| 1 | 0.01 | 0.00 |
| 2 | 0.06 | 0.22 |
| 3 | 0.00 | 0.00 |
| 4 | 0.00 | 0.00 |

Rzeczywiste wartości:

| insufficient_w | normal_w | obesity_t1 | obesity_t2 | obesity_t3 |
|---|---|---|---|---|
| True | False | False | False | False |
| False | False | True | False | False |
| True | False | False | False | False |
| False | True | False | False | False |
| False | False | False | False | True |

| overweight_l1 | overweight_l2 |
|---|---|
| False | False |
| False | False |
| False | False |
| False | False |
| False | False |

Można zauważyć, że algorytm prawidłowo dopasował obiekt do grupy w 3 z 5 przypadkach (0,2 i 4), jednak tylko w jednym przypadku wskazał prawdopodobieństwo aż 98% przynależności do odpowiedniej grupy, a dla 2 tylko 35%. Dla obiektu 3 wskazał 99% prawdopodobieństwo

przynależności do grupy „Niedowaga”, a w rzeczywistości osoba należała do grupy osób o normalnej wadze – jednak nie jest to wielka pomyłka, gorzej byłoby gdyby osoba w rzeczywistości miała otyłość

## Analiza liczby warstw ukrytych w modelu:

### Model z jedną warstwą ukrytą

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_3 (Dense)             (None, 32)                480

 dense_4 (Dense)             (None, 7)                 231

=================================================================
Total params: 711 (2.78 KB)
Trainable params: 711 (2.78 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

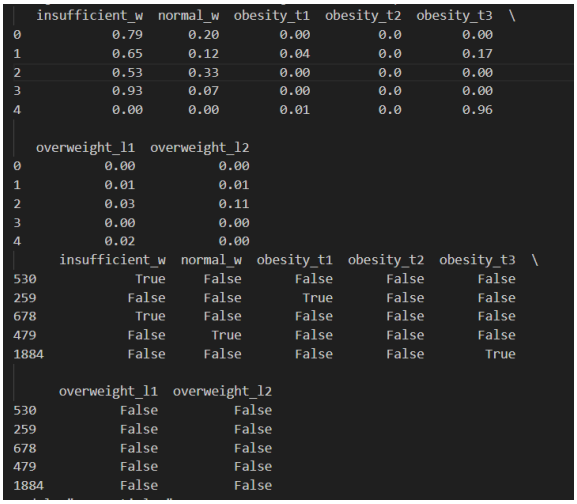
Na zbiorze treningowym model uzyskał 70,95% dokładności, a na walidacyjnym 65,3% - dokładność na zbiorze walidacyjnym jest mniejsza, więc nie sugeruje to przetrenowania

```
 1/47 [..............................] - ETA: 0s - loss: 0.7513 - accuracy: 0.7188
47/47 [==============================] - 0s 1ms/step - loss: 0.8517 - accuracy: 0.7095 - val_loss: 0.9841 - val_accuracy: 0.6530
```

Na zbiorze testowym model uzyskał 70,35% dokładności – jest to dobry wynik, ponieważ nie odbiega bardzo od dokładności na zbiorze treningowym:

```
 1/10 [==>...........................] - ETA: 0s - loss: 0.9675 - accuracy: 0.6562
10/10 [==============================] - 0s 778us/step - loss: 0.9342 - accuracy: 0.7035
```

Prognoza dla 5 obiektów:



```
   insufficient_w  normal_w  obesity_t1  obesity_t2  obesity_t3  \
0            0.79      0.20        0.00         0.0        0.00
1            0.65      0.12        0.04         0.0        0.17
2            0.53      0.33        0.00         0.0        0.00
3            0.93      0.07        0.00         0.0        0.00
4            0.00      0.00        0.01         0.0        0.96

   overweight_l1  overweight_l2
0           0.00           0.00
1           0.01           0.01
2           0.03           0.11
3           0.00           0.00
4           0.02           0.00
      insufficient_w  normal_w  obesity_t1  obesity_t2  obesity_t3  \
530             True     False       False       False       False
259            False     False        True       False       False
678             True     False       False       False       False
479            False      True       False       False       False
1884           False     False       False       False        True

      overweight_l1  overweight_l2
530           False          False
259           False          False
678           False          False
479           False          False
1884          False          False
```

Model również wskazał poprawne grupy dla obiektu 0,2 i 4 (dla obiektu nr 2 prawdopodobieństwo dopasowania do odpowiedniej grupy wzrosło do 53% w porównaniu z modelem bazowym). Dla obiektu 3 model lekko zbliżył się do odpowiedniej grupy, jednak prawdopodobieństwo wynoszące 7% jest wciąż bardzo małe

## Model z trzema warstwami ukrytymi

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_5 (Dense)             (None, 32)                480

 dense_6 (Dense)             (None, 32)                1056

 dense_7 (Dense)             (None, 32)                1056

 dense_8 (Dense)             (None, 7)                 231

=================================================================
Total params: 2823 (11.03 KB)
Trainable params: 2823 (11.03 KB)
Non-trainable params: 0 (0.00 Byte)
```

Dokładność na zbiorze treningowym wynosi 79,62% - jest to najwyższa wartość do tej pory, a na walidacyjnym 68,14%

```
Epoch 100/100

 1/47 [..............................] - ETA: 0s - loss: 0.9180 - accuracy: 0.5625
47/47 [==============================] - 0s 1ms/step - loss: 0.6098 - accuracy: 0.7962 - val_loss: 0.9059 - val_accuracy: 0.6814
```

Dokładność na zbiorze testowym wynosi 71,29% - nie jest to najlepszy wynik, więc możliwe że model nadmiernie przystosował się do danych treningowych

```
 1/10 [==>...........................] - ETA: 0s - loss: 0.9877 - accuracy: 0.6250
10/10 [==============================] - 0s 889us/step - loss: 0.8809 - accuracy: 0.7129
```

Prognoza

```
     insufficient_w  normal_w  obesity_t1  obesity_t2  obesity_t3  \
0              0.93      0.06        0.00         0.0         0.0
1              0.67      0.18        0.12         0.0         0.0
2              0.27      0.24        0.00         0.0         0.0
3              0.97      0.02        0.00         0.0         0.0
4              0.00      0.00        0.00         0.0         1.0

     overweight_l1  overweight_l2
0             0.00           0.01
1             0.02           0.01
2             0.02           0.47
3             0.00           0.00
4             0.00           0.00
      insufficient_w  normal_w  obesity_t1  obesity_t2  obesity_t3  \
530             True     False       False       False       False
259            False     False        True       False       False
678             True     False       False       False       False
479            False      True       False       False       False
1884           False     False       False       False        True

      overweight_l1  overweight_l2
530           False          False
259           False          False
678           False          False
479           False          False
1884          False          False
```

Model ponownie przewidział odpowiednią grupę dla tych samych 3 modeli

## Model z czterema warstwami ukrytymi

```
Model: "sequential_3"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_9 (Dense)             (None, 32)                480

 dense_10 (Dense)            (None, 32)                1056

 dense_11 (Dense)            (None, 32)                1056

 dense_12 (Dense)            (None, 32)                1056

 dense_13 (Dense)            (None, 7)                 231

=================================================================
Total params: 3879 (15.15 KB)
Trainable params: 3879 (15.15 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Model uzyskał 79,15% dokładności na zbiorze treningowym, a na walidacyjnym 64,35%

```
 1/47 [..............................] - ETA: 0s - loss: 0.6015 - accuracy: 0.7812
47/47 [==============================] - 0s 1ms/step - loss: 0.6011 - accuracy: 0.7915 - val_loss: 0.9673 - val_accuracy: 0.6435
```

Dokładność na zbiorze testowym wynosi 70,35% - wartość spadła względem modeli o mniejszej ilości warstw i  dokładności na zbiorze treningowym  co sugeruje overfitting (nadmierne przywiązanie się modelu do zbioru treningowego)

```
 1/10 [==>...........................] - ETA: 0s - loss: 0.9470 - accuracy: 0.6562
10/10 [==============================] - 0s 889us/step - loss: 0.8863 - accuracy: 0.7035
```

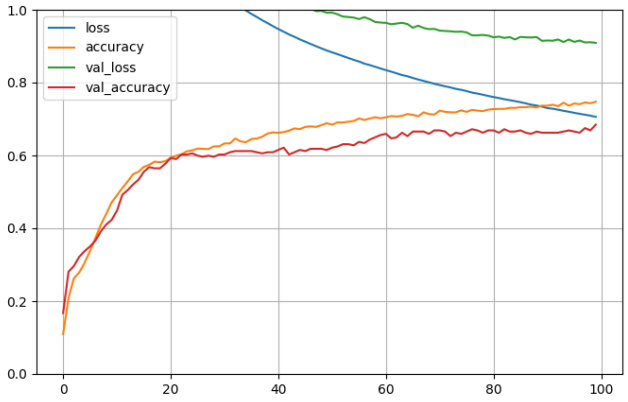
Prognoza:

```
   insufficient_w  normal_w  obesity_t1  obesity_t2  obesity_t3  \
0            0.91      0.05        0.00         0.0         0.0
1            0.30      0.39        0.23         0.0         0.0
2            0.64      0.14        0.01         0.0         0.0
3            0.97      0.03        0.00         0.0         0.0
4            0.00      0.00        0.00         0.0         1.0

   overweight_l1  overweight_l2
0           0.01           0.03
1           0.07           0.01
2           0.01           0.21
3           0.00           0.00
4           0.00           0.00
      insufficient_w  normal_w  obesity_t1  obesity_t2  obesity_t3  \
530             True     False       False       False       False
259            False     False        True       False       False
678             True     False       False       False       False
479            False      True       False       False       False
1884           False     False       False       False        True

      overweight_l1  overweight_l2
530           False          False
259           False          False
678           False          False
479           False          False
1884          False          False
```

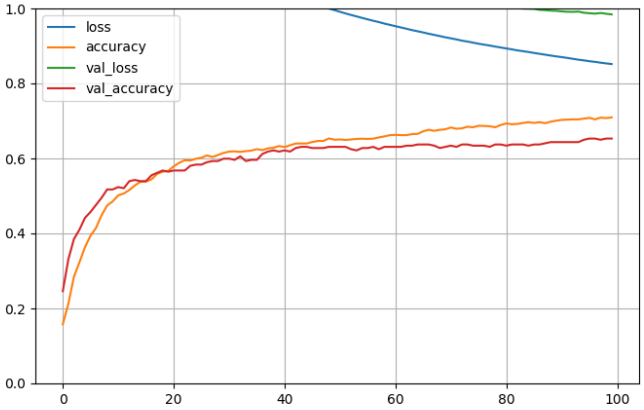
Model ponownie dopasował poprawne grupy dla 3 obiektów (pojawiło się nawet prawdopodobieństwo 100%). Dla obiektu o numerze id 1 wzrosło prawdopodobieństwo przynależności do poprawnej grupy (względem poprzednich obiektów), jednak dalej wynik 23% jest niesatysfakcjonujący. Prognoza dla 5 obiektów nie jest miarodajna, więc wyniki traktujemy raczej jako ciekawostkę.

## Analiza wszystkich wartości parametru

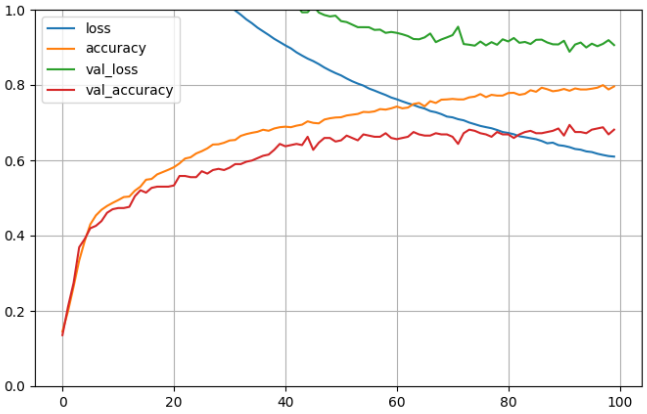
Model bazowy – dwie warstwy ukryte



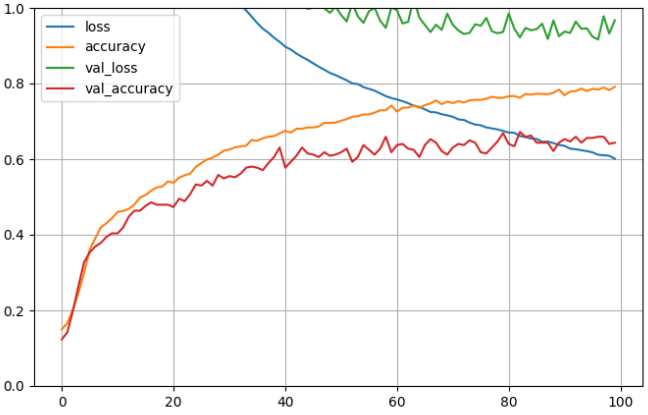

Model z jedną warstwą ukrytą




Model z trzema warstwami ukrytymi




Model z czterema warstwami ukrytymi




Dokładność na zbiorze testowym:
{'base': 0.7507886290550232, 'number of layers: 1': 0.7034700512886047, 'number of layers: 3': 0.7129337787628174, 'number of layers: 4': 0.7034700512886047}

Podsumowanie:
W modelu z czterema warstwami funkcja straty na zbiorze walidacyjnym nie jest malejąca, co sugeruje wspomniany wcześniej overfitting (w modelu z trzema warstwami ukrytymi również lekko widać to zjawisko oraz dokładność na zbiorze testowym jest zdecydowanie niższa od tej na zbiorze treningowym). Model bazowy z dwiema warswami ukrytymi ma większą dokładność na zbiorze testowym od modelu z jedna taka warstwą, więc to on jest najlepszy z tej grupy.

Wniosek:
W przypadku wielu problemów perceptrony z jedną warstwą ukrytą dają satysfakcjonujące wyniki. W naszym przypadku dodanie kolejnej warstwy poprawiło dokładność modelu o około 5 punktów procentowych%. Dodanie kolejnych warstw powoduje przetrenowanie modelu. Czas uczenia jest

zbliżony we wszystkich przypadkach (najszybciej w modelu z jedną warstwą ukrytą, co jest dosyć logiczne)

## Analiza liczby neuronów w warstwach ukrytych

Bazowy model ma 32 neutrony w warstwach ukrytych. Zobaczymy jak wyniki zmienią się przy innych wartościach tego parametru

### Model z 100 neuronami w jednej warstwie ukrytej

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_14 (Dense)            (None, 100)               1500

 dense_15 (Dense)            (None, 100)               10100

 dense_16 (Dense)            (None, 7)                 707

=================================================================
Total params: 12307 (48.07 KB)
Trainable params: 12307 (48.07 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Model uzyskał około 80,64 proc dokładności na zbiorze treningowym co jest lepszym wynikiem od modelu bazowego

```
 1/47 [..............................] - ETA: 0s - loss: 0.6615 - accuracy: 0.8125
47/47 [==============================] - 0s 1ms/step - loss: 0.5750 - accuracy: 0.8064 - val_loss: 0.8996 - val_accuracy: 0.6909
```

Dokładność na zbiorze testowym wynosi około 74,76%. Czas uczenia jest zbliżony do modelu bazowego

```
 1/10 [==>...........................] - ETA: 0s - loss: 0.8668 - accuracy: 0.7188
10/10 [==============================] - 0s 778us/step - loss: 0.8378 - accuracy: 0.7476
```

Prognoza:

| | insufficient_w | normal_w | obesity_t1 | obesity_t2 | obesity_t3 | overweight_l1 | overweight_l2 |
|---|---|---|---|---|---|---|---|
| 0 | 0.88 | 0.11 | 0.00 | 0.0 | 0.00 | 0.00 | 0.01 |
| 1 | 0.36 | 0.33 | 0.19 | 0.0 | 0.06 | 0.03 | 0.03 |
| 2 | 0.51 | 0.24 | 0.01 | 0.0 | 0.00 | 0.02 | 0.23 |
| 3 | 0.98 | 0.01 | 0.00 | 0.0 | 0.00 | 0.00 | 0.00 |
| 4 | 0.00 | 0.00 | 0.00 | 0.0 | 0.99 | 0.00 | 0.00 |

| | insufficient_w | normal_w | obesity_t1 | obesity_t2 | obesity_t3 | overweight_l1 | overweight_l2 |
|---|---|---|---|---|---|---|---|
| 530 | True | False | False | False | False | False | False |
| 259 | False | False | True | False | False | False | False |
| 678 | True | False | False | False | False | False | False |
| 479 | False | True | False | False | False | False | False |
| 1884 | False | False | False | False | True | False | False |

Model poraz kolejny przewidział grupę dla tych samych trzech obiektów

## Model z 200 neuronami w warstwach ukrytych

```
Model: "sequential_5"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_17 (Dense)            (None, 200)               3000

 dense_18 (Dense)            (None, 200)               40200

 dense_19 (Dense)            (None, 7)                 1407

=================================================================
Total params: 44607 (174.25 KB)
Trainable params: 44607 (174.25 KB)
Non-trainable params: 0 (0.00 Byte)
```

Model uzyskał 82,13% dokładności na zbiorze treningowym i około 70,66% na zbiorze walidacyjnym, co jest najlepszym wynikiem do tej pory.

```
Epoch 100/100

 1/47 [..............................] - ETA: 0s - loss: 0.5293 - accuracy: 0.8438
47/47 [==============================] - 0s 1ms/step - loss: 0.5341 - accuracy: 0.8213 - val_loss: 0.8515 - val_accuracy: 0.7066
```

Dokładność na zbiorze testowym wynosi 76,34% jest to jeden z lepszych wyników. Czas uczenia wykonywania jest trochę większy niż w poprzednich przypadkach – 1ms/step

```
 1/10 [==>...........................] - ETA: 0s - loss: 0.8326 - accuracy: 0.7500
10/10 [==============================] - 0s 1ms/step - loss: 0.8013 - accuracy: 0.7634
```

Prognoza:

```
      insufficient_w  normal_w  obesity_t1  obesity_t2  obesity_t3  \
0               0.90      0.08        0.00         0.0        0.00
1               0.38      0.42        0.12         0.0        0.03
2               0.62      0.14        0.00         0.0        0.00
3               0.97      0.03        0.00         0.0        0.00
4               0.00      0.00        0.00         0.0        0.99

      overweight_l1  overweight_l2
0              0.00           0.02
1              0.04           0.01
2              0.01           0.22
3              0.00           0.00
4              0.00           0.00
      insufficient_w  normal_w  obesity_t1  obesity_t2  obesity_t3  \
530             True     False       False       False       False
259            False     False        True       False       False
678             True     False       False       False       False
479            False      True       False       False       False
1884           False     False       False       False        True

      overweight_l1  overweight_l2
530           False          False
259           False          False
678           False          False
479           False          False
1884          False          False
```

Model poraz kolejny przewidział grupę dla tych samych trzech obiektów

## Model z 300 neuronami w warstwach ukrytych

```
Model: "sequential_6"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_20 (Dense)            (None, 300)               4500

 dense_21 (Dense)            (None, 300)               90300

 dense_22 (Dense)            (None, 7)                 2107

=================================================================
Total params: 96907 (378.54 KB)
Trainable params: 96907 (378.54 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Model ma  82,67% dokładności na zbiorze treningowym

```
Epoch 100/100

 1/47 [..............................] - ETA: 0s - loss: 0.3499 - accuracy: 0.8750
47/47 [==============================] - 0s 1ms/step - loss: 0.5241 - accuracy: 0.8267 - val_loss: 0.8414 - val_accuracy: 0.7035
```

Dokładność na zbiorze testowym wynosi 75,39%, co jest bardzo zbliżonym wynikiem do poprzedniego

```
 1/10 [==>...........................] - ETA: 0s - loss: 0.8609 - accuracy: 0.7500
10/10 [==============================] - 0s 904us/step - loss: 0.8085 - accuracy: 0.7539
```

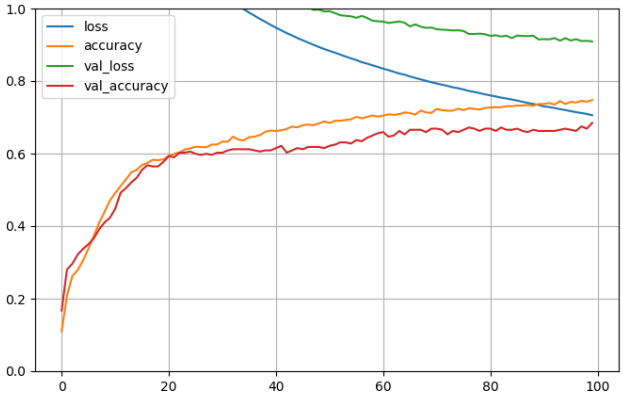
Prognoza:

```
   insufficient_w  normal_w  obesity_t1  obesity_t2  obesity_t3  \
0            0.94      0.05        0.00         0.0        0.00
1            0.40      0.45        0.08         0.0        0.04
2            0.69      0.11        0.00         0.0        0.00
3            0.98      0.02        0.00         0.0        0.00
4            0.00      0.00        0.00         0.0        0.99

   overweight_l1  overweight_l2
0           0.00           0.01
1           0.02           0.02
2           0.01           0.19
3           0.00           0.00
4           0.00           0.00
      insufficient_w  normal_w  obesity_t1  obesity_t2  obesity_t3
530             True     False       False       False       False
259            False     False        True       False       False
678             True     False       False       False       False
479            False      True       False       False       False
1884           False     False       False       False        True

      overweight_l1  overweight_l2
530           False          False
259           False          False
678           False          False
479           False          False
1884          False          False
```

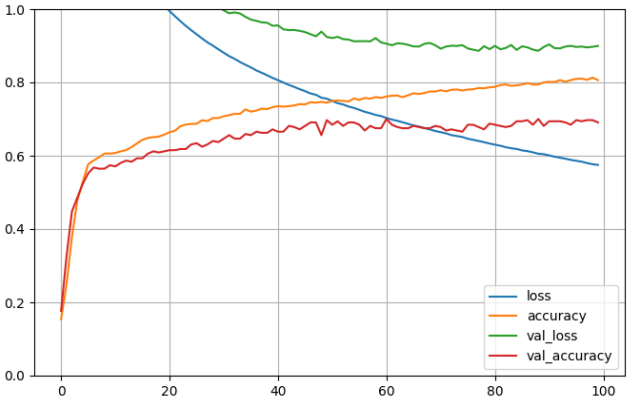
Model poprawnie dopasował te same obiekty co wcześniej.

## Analiza wszystkich wartości parametru

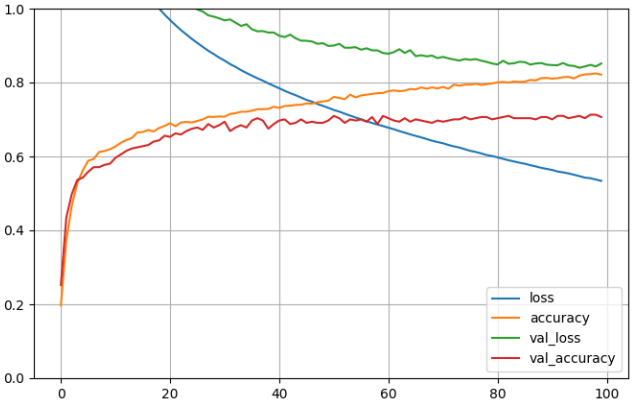
Model bazowy – 32 neurony w warstwie



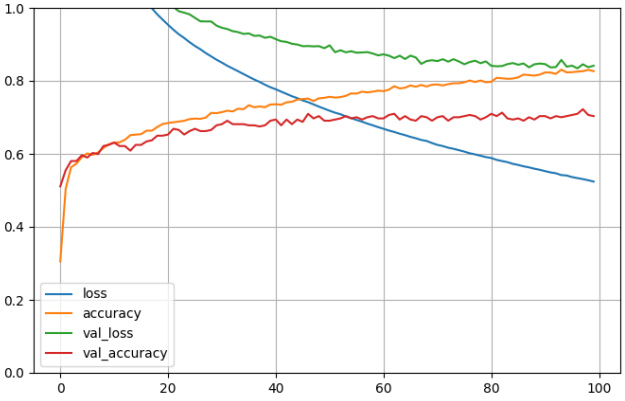

Model z 100 neuronami w warstwie




Model z 200 neuronami w warstwie




Model z 300 neuronami w warstwie




{'base': 0.7507886290550232, 'number of neurons: 100 and 100': 0.7476340532302856, 'number of neurons: 200 and 200': 0.7634069323539734, 'number of neurons: 300 and 300': 0.7539432048797607}

Podsumowanie:

W naszym przypadku zmiany liczby neuronów w warstwie ukrytej nie dały większych zmian – dla 200 neuronów w warstwie dokładność na zbiorze testowym wzrosła tylko o 1,3 punktu procentowego, dla 300 neuronów o 0,3. Są to bardzo niewielkie różnice. Różnica dokładności na zbiorach treningowych jest większa, lecz nie to jest naszym celem.

Warstwy ukryte tworzono kiedyś na wzór piramidy, jednak porzucono to rozwiązanie, ponieważ taka sama liczba neuronów w każdej warstwie daje takie same, a nawet lepsze rezultaty.

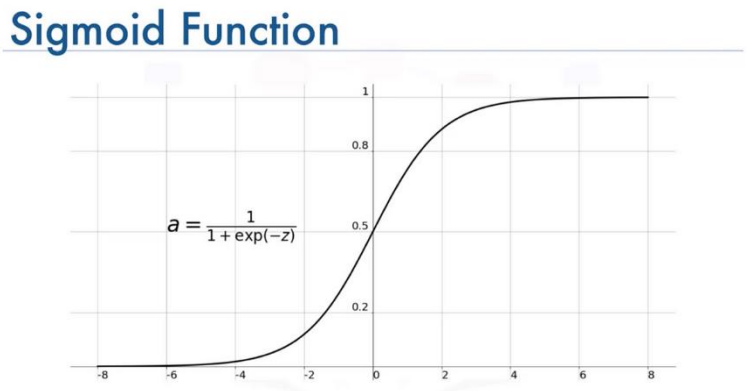
## Analiza użycia różnych funkcji aktywacji

Funkcja wyjściowa zależy od rodzaju zadania – jej nie będziemy zmieniać. Jednak funkcje aktywacji w warstwach ukrytych można zmieniać, więc sprawdzimy która z tych najbardziej popularnych jest najlepsza. W każdym  modelu jego podsumowanie będzie wyglądać tak samo, dlatego dla przypomnienia zaprezentujemy je tylko tutaj jeszcze raz:

```
 Layer (type)                Output Shape              Param #
=================================================================
 dense_23 (Dense)            (None, 32)                480

 dense_24 (Dense)            (None, 32)                1056

 dense_25 (Dense)            (None, 7)                 231

=================================================================
Total params: 1767 (6.90 KB)
Trainable params: 1767 (6.90 KB)
Non-trainable params: 0 (0.00 Byte)
```

### Funkcja Sigmoid

Tak prezentuje się wykres funkcji:




https://medium.com/@toprak.mhmt/activation-functions-for-deep-learning-13d8b9b20e

Funkcja ta spowodowała, że dokładność modelu jest fatalna – na zbiorze treningowym wynosi tylko 46,72%. W przypadku sieci neuronowych funkcje aktywacji, takie jak ReLU, są bardziej preferowane, ponieważ są centrowane wokół zera, co pomaga w uczeniu wag w modelu. Funkcja Sigmoid ma natomiast wartości z przedziału (0,1), co również jest minusem.

```
Epoch 100/100

 1/47 [..............................] - ETA: 0s - loss: 1.5335 - accuracy: 0.4375
47/47 [==============================] - 0s 1ms/step - loss: 1.5559 - accuracy: 0.4672 - val_loss: 1.5609 - val_accuracy: 0.4416
```

Na zbiorze testowym dokładność wynosi tylko 40%

```
 1/10 [==>...........................] - ETA: 0s - loss: 1.5600 - accuracy: 0.4375
10/10 [==============================] - 0s 919us/step - loss: 1.5562 - accuracy: 0.4069
```

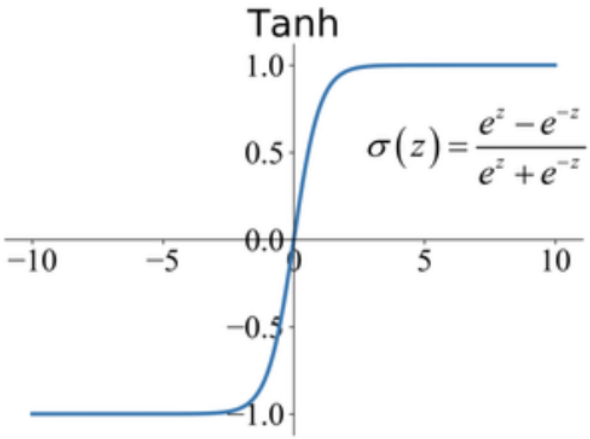
Prognoza:

```
   insufficient_w  normal_w  obesity_t1  obesity_t2  obesity_t3  \
0            0.29      0.29        0.05        0.03        0.13
1            0.18      0.14        0.08        0.06        0.36
2            0.25      0.29        0.06        0.04        0.11
3            0.27      0.24        0.05        0.03        0.23
4            0.09      0.07        0.14        0.10        0.41

   overweight_l1  overweight_l2
0           0.14           0.06
1           0.13           0.06
2           0.16           0.08
3           0.14           0.04
4           0.11           0.08
      insufficient_w  normal_w  obesity_t1  obesity_t2  obesity_t3
530             True     False       False       False       False
259            False     False        True       False       False
678             True     False       False       False       False
479            False      True       False       False       False
1884           False     False       False       False        True

      overweight_l1  overweight_l2
530           False          False
259           False          False
678           False          False
479           False          False
1884          False          False
```

Tylko dla jednego obiektu (4) model wskazał poprawnie grupę (największa wartość procentowa) , a i ta prawdopodobieństwo przynależności do tej grupy wynosi tylko 41%

## Funkcja Tanh

Wykres tej funkcji wygląda następująco




https://paperswithcode.com/method/tanh-activation

Model uzyskał 72,85% dokładności na zbiorze treningowym – jest to nienajgorszy wynik.

```
Epoch 100/100

 1/47 [..............................] - ETA: 0s - loss: 0.7551 - accuracy: 0.7188
47/47 [==============================] - 0s 1ms/step - loss: 0.8084 - accuracy: 0.7285 - val_loss: 0.9648 - val_accuracy: 0.6845
```

Dokładność na zbiorze testowym wyniosła 70,03% - zdecydowanie lepszy wynik od funkcji sigmoid i trochę gorszy od reLu (5 punktów procentowych). W porównaniu z funkcją sigmoidalną, funkcja tanh może pomóc w redukcji problemu zanikającego gradientu w głębokich sieciach neuronowych w pewnym stopniu, choć nie jest to idealne rozwiązanie.

```
 1/10 [==>...........................] - ETA: 0s - loss: 1.0156 - accuracy: 0.6875
10/10 [==============================] - 0s 1ms/step - loss: 0.9182 - accuracy: 0.7003
```

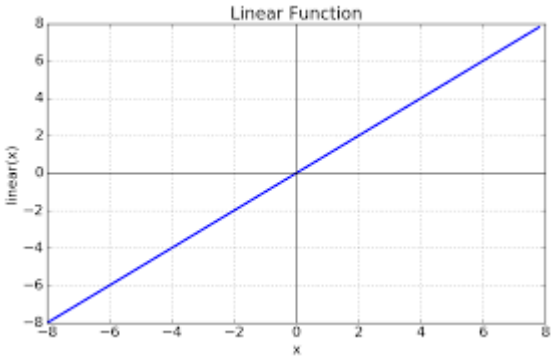
Prognoza:

```
   insufficient_w  normal_w  obesity_t1  obesity_t2  obesity_t3  \
0            0.79      0.19        0.00         0.0        0.00
1            0.51      0.36        0.02         0.0        0.07
2            0.65      0.22        0.00         0.0        0.00
3            0.95      0.04        0.00         0.0        0.00
4            0.00      0.00        0.01         0.0        0.96

   overweight_l1  overweight_l2
0           0.01           0.01
1           0.02           0.01
2           0.06           0.06
3           0.00           0.00
4           0.01           0.01
      insufficient_w  normal_w  obesity_t1  obesity_t2  obesity_t3
530             True     False       False       False       False
259            False     False        True       False       False
678             True     False       False       False       False
479            False      True       False       False       False
1884           False     False       False       False        True

      overweight_l1  overweight_l2
530           False          False
259           False          False
678           False          False
479           False          False
1884          False          False
```

Model prawidłowo przewidział grupę dla 3 obiektów(0, 2 i 4)

## Funkcja Linear

Wykres funkcji wygląda tak:




https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6

Model z funkcją liniową jako funkcja aktywacji uzyskał 58,63% dokładności na zbiorze treningowym i 59,31% dokładności na zbiorze walidacyjnym – jest to słaby wynik w porównaniu do modeli z funkcją ReLu czy tanh. Funkcja liniowa jest ograniczona w zdolności do modelowania nieliniowych relacji w danych.

```
Epoch 100/100

 1/47 [..............................] - ETA: 0s - loss: 1.3365 - accuracy: 0.4688
47/47 [==============================] - 0s 1ms/step - loss: 1.1250 - accuracy: 0.5863 - val_loss: 1.1409 - val_accuracy: 0.5931
```

Na zbiorze testowym model uzyskał tylko 56,47% dokładności

```
 1/10 [==>...........................] - ETA: 0s - loss: 1.1759 - accuracy: 0.6250
10/10 [==============================] - 0s 889us/step - loss: 1.1811 - accuracy: 0.5647
```

Prognoza:

| | insufficient_w | normal_w | obesity_t1 | obesity_t2 | obesity_t3 | \ |
|---|---|---|---|---|---|---|
| 0 | 0.80 | 0.19 | 0.00 | 0.0 | 0.00 | |
| 1 | 0.51 | 0.08 | 0.01 | 0.0 | 0.37 | |
| 2 | 0.52 | 0.36 | 0.00 | 0.0 | 0.00 | |
| 3 | 0.96 | 0.04 | 0.00 | 0.0 | 0.00 | |
| 4 | 0.00 | 0.00 | 0.03 | 0.0 | 0.92 | |

| | overweight_l1 | overweight_l2 |
|---|---|---|
| 0 | 0.00 | 0.00 |
| 1 | 0.02 | 0.01 |
| 2 | 0.09 | 0.03 |
| 3 | 0.00 | 0.00 |
| 4 | 0.03 | 0.01 |

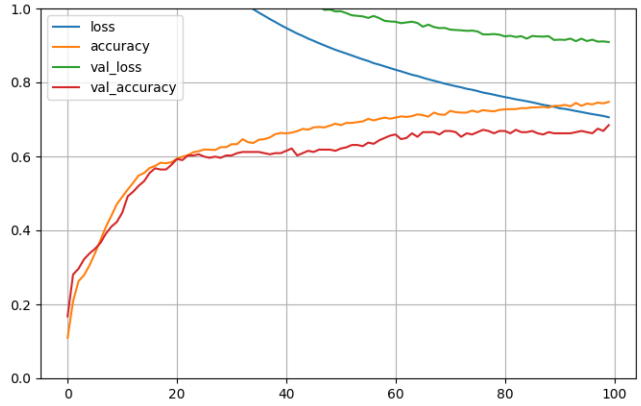
| | insufficient_w | normal_w | obesity_t1 | obesity_t2 | obesity_t3 | \ |
|---|---|---|---|---|---|---|
| 530 | True | False | False | False | False | |
| 259 | False | False | True | False | False | |
| 678 | True | False | False | False | False | |
| 479 | False | True | False | False | False | |
| 1884 | False | False | False | False | True | |

| | overweight_l1 | overweight_l2 |
|---|---|---|
| 530 | False | False |
| 259 | False | False |
| 678 | False | False |
| 479 | False | False |
| 1884 | False | False |

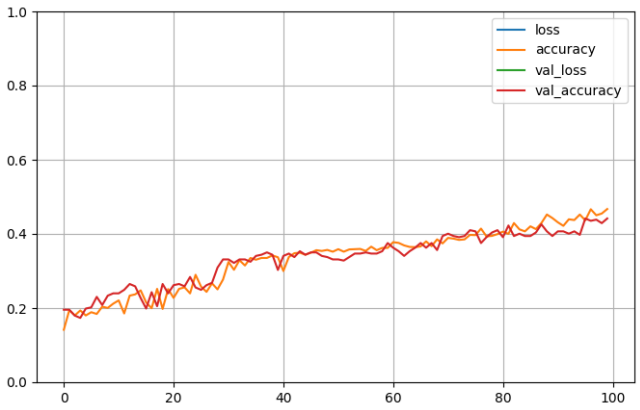
Model prawidłowo przewidział grupę dla trzech obiektów, jednak prawdopodobieństwo przynależności obiektu o id 2 do prawidłowej grupy wynosi tylko 52%

## Analiza wszystkich wartości parametrów

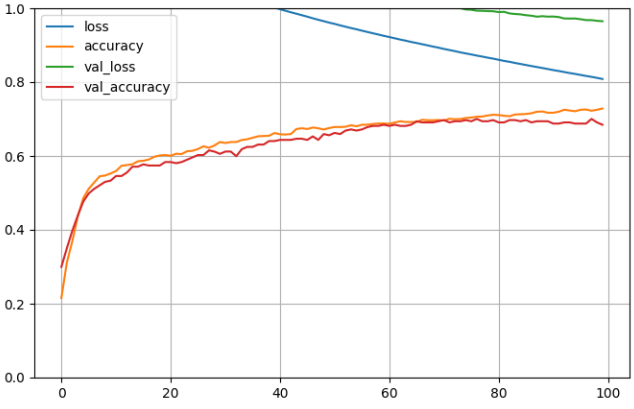
Model bazowy - funkcja relu



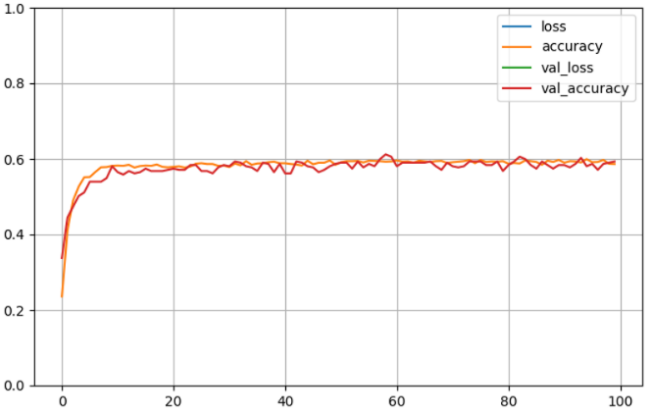

Model z funkcja sigmoid




Model z funkcja tanh




Model z funkcja linear



{'base': 0.7507886290550232,  'activaction function: sigmoid': 0.4069400727748871, 'activaction function: tanh': 0.7003154754638672, 'activaction function: linear': 0.5646687746047974}

Podsumowanie:

W sieciach neuronowych używa się nieliniowych funkcji aktywacji, takich jak funkcja ReLU (Rectified Linear Unit) czy tangens hiperboliczny. Modele z funkcja aktywacji sigmoid i liniową osiągnęły bardzo słabą wartość

## Analiza ilości epok

Epoka odnosi się do jednego pełnego przejścia przez cały zestaw danych treningowych, a liczba epok określa, ile razy cały zestaw danych treningowych został użyty do aktualizacji wag modelu podczas procesu uczenia. W modelu bazowym użyliśmy 100 epok. W każdym modelu jego podsumowanie będzie wyglądać tak samo, dlatego dla przypomnienia pokażemy je jeszcze raz:

```
Model: "sequential_10"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_32 (Dense)            (None, 32)                480

 dense_33 (Dense)            (None, 32)                1056

 dense_34 (Dense)            (None, 7)                 231

=================================================================
Total params: 1767 (6.90 KB)
Trainable params: 1767 (6.90 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

## Model- 10 epok

Model z 10 epokami spodziewanie osiągnął bardzo słaby wynik. Na zbiorze treningowym to tylko 49,7% a walidacyjnym 50,16%

```
Epoch 10/10
 1/47 [..............................] - ETA: 0s - loss: 1.5072 - accuracy: 0.4375
47/47 [==============================] - 0s 1ms/step - loss: 1.4383 - accuracy: 0.4970 - val_loss: 1.4476 - val_accuracy: 0.4700
```

Dokładność na zbiorze testowym wyniosła 50,16%. Uczenie sieci nie jest procesem deterministycznym, a 10 epok to zdecydowanie za mało  żeby dobrze wytrenować model.

```
 1/10 [==>...........................] - ETA: 0s - loss: 1.3819 - accuracy: 0.5312
10/10 [==============================] - 0s 778us/step - loss: 1.4423 - accuracy: 0.5016
```

Prognoza

```
   insufficient_w  normal_w  obesity_t1  obesity_t2  obesity_t3  \
0            0.47      0.19        0.06        0.08        0.06
1            0.20      0.12        0.11        0.08        0.28
2            0.24      0.29        0.08        0.08        0.04
3            0.35      0.21        0.10        0.11        0.06
4            0.03      0.03        0.05        0.02        0.66

   overweight_l1  overweight_l2
0           0.08           0.05
1           0.16           0.05
2           0.09           0.18
3           0.10           0.09
4           0.16           0.05
      insufficient_w  normal_w  obesity_t1  obesity_t2  obesity_t3  \
530             True     False       False       False       False
259            False     False        True       False       False
678             True     False       False       False       False
479            False      True       False       False       False
1884           False     False       False       False        True

      overweight_l1  overweight_l2
530           False          False
259           False          False
678           False          False
479           False          False
1884          False          False
```

Model poprawnie przewidział grupę dla 2 obiektów, ale dla obiektu o id 10 prawdopodobieństwo wynosi tylko 47%

## Model- 200 epok

Model z 200 epokami osiągnął na zbiorze treningowym dokładność aż 81,25% (więcej od bazowego z około 75%), a walidacyjnym 69,09%

```
Epoch 200/200

 1/47 [..............................] - ETA: 0s - loss: 0.7183 - accuracy: 0.8125
47/47 [==============================] - 0s 1ms/step - loss: 0.5369 - accuracy: 0.8301 - val_loss: 0.9311 - val_accuracy: 0.6909
```

Dokładność na zbiorze testowym wyniosła 73,19% - jest to całkiem dobry wynik w porównaniu do kilku poprzednich modeli, ale o około 2 punkty procentowe gorszy od modelu bazowego, co może świadczyć o lekkim przetrenowaniu modelu.

```
 1/10 [==>...........................] - ETA: 0s - loss: 0.7962 - accuracy: 0.7500
10/10 [==============================] - 0s 864us/step - loss: 0.8865 - accuracy: 0.7319
```

Prognoza

| | insufficient_w | normal_w | obesity_t1 | obesity_t2 | obesity_t3 | \ |
|---|---|---|---|---|---|---|
| 0 | 0.99 | 0.01 | 0.00 | 0.0 | 0.0 | |
| 1 | 0.46 | 0.36 | 0.05 | 0.0 | 0.0 | |
| 2 | 0.90 | 0.06 | 0.00 | 0.0 | 0.0 | |
| 3 | 0.81 | 0.19 | 0.00 | 0.0 | 0.0 | |
| 4 | 0.00 | 0.00 | 0.00 | 0.0 | 1.0 | |

| | overweight_l1 | overweight_l2 |
|---|---|---|
| 0 | 0.00 | 0.00 |
| 1 | 0.09 | 0.03 |
| 2 | 0.01 | 0.03 |
| 3 | 0.00 | 0.00 |
| 4 | 0.00 | 0.00 |

| | insufficient_w | normal_w | obesity_t1 | obesity_t2 | obesity_t3 |
|---|---|---|---|---|---|
| 530 | True | False | False | False | False |
| 259 | False | False | True | False | False |
| 678 | True | False | False | False | False |
| 479 | False | True | False | False | False |
| 1884 | False | False | False | False | True |

| | overweight_l1 | overweight_l2 |
|---|---|---|
| 530 | False | False |
| 259 | False | False |
| 678 | False | False |
| 479 | False | False |
| 1884 | False | False |

Model poprawnie dopasował 3 obiekty do odpowiedniej grupy, a prawdopodobieństwa te wynoszą ponad 90%.

## Model- 2000 epok

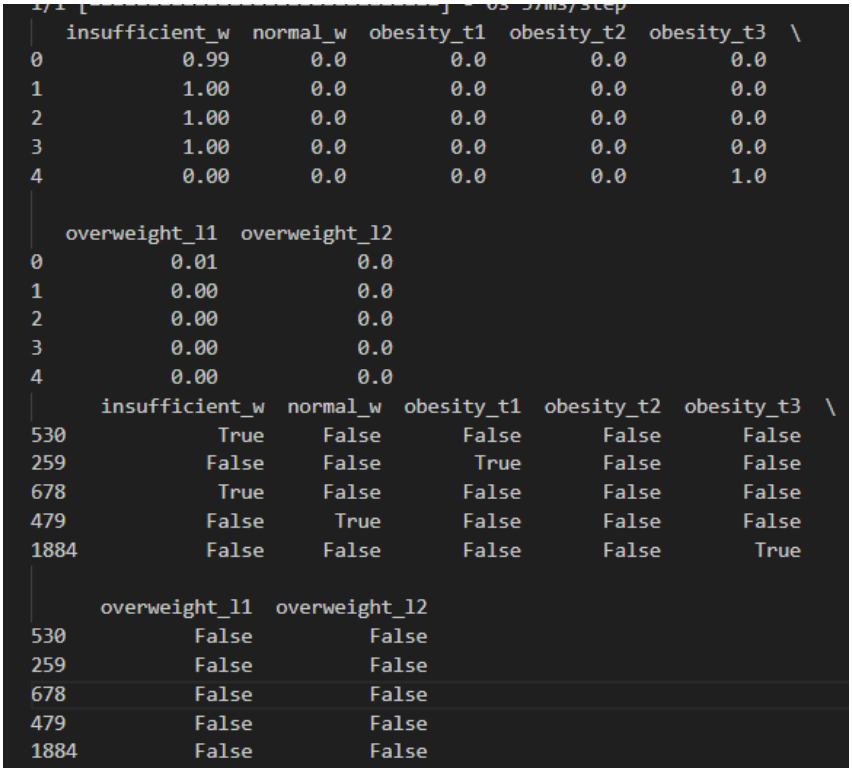
Model zdobył rekordowe 93,75% dokładności na zbiorze treningowym, jednak dokładność na zbiorze walidacyjnym jest zdecydowanie niższa, co może być objawem nadmiernego dopasowania do zbioru treningowego.

```
Epoch 2000/2000

 1/47 [..............................] - ETA: 0s - loss: 0.3094 - accuracy: 0.9375
47/47 [==============================] - 0s 1ms/step - loss: 0.0782 - accuracy: 0.9695 - val_loss: 2.5978 - val_accuracy: 0.7003
```

Na modelu testowym dokładność jest zbliżona do tej na zbiorze walidacyjnym, co również sugeruje overfitting.

```
 1/10 [==>...........................] - ETA: 0s - loss: 2.5504 - accuracy: 0.7500
10/10 [==============================] - 0s 889us/step - loss: 3.3112 - accuracy: 0.7256
```

Prognoza



| | insufficient_w | normal_w | obesity_t1 | obesity_t2 | obesity_t3 | \ |
|---|---|---|---|---|---|---|
| 0 | 0.99 | 0.0 | 0.0 | 0.0 | 0.0 | |
| 1 | 1.00 | 0.0 | 0.0 | 0.0 | 0.0 | |
| 2 | 1.00 | 0.0 | 0.0 | 0.0 | 0.0 | |
| 3 | 1.00 | 0.0 | 0.0 | 0.0 | 0.0 | |
| 4 | 0.00 | 0.0 | 0.0 | 0.0 | 1.0 | |

| | overweight_l1 | overweight_l2 |
|---|---|---|
| 0 | 0.01 | 0.0 |
| 1 | 0.00 | 0.0 |
| 2 | 0.00 | 0.0 |
| 3 | 0.00 | 0.0 |
| 4 | 0.00 | 0.0 |

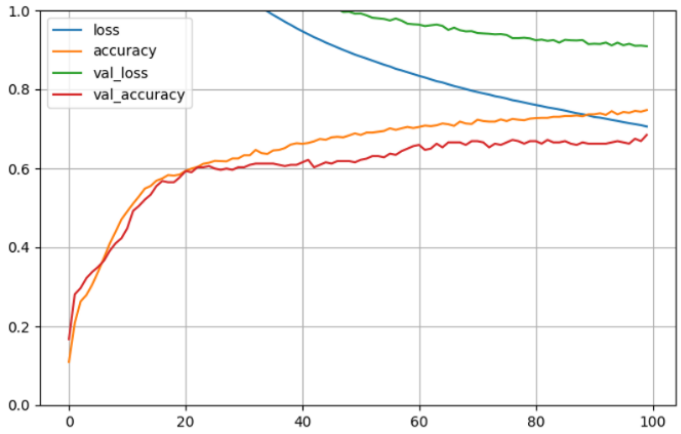
| | insufficient_w | normal_w | obesity_t1 | obesity_t2 | obesity_t3 | \ |
|---|---|---|---|---|---|---|
| 530 | True | False | False | False | False | |
| 259 | False | False | True | False | False | |
| 678 | True | False | False | False | False | |
| 479 | False | True | False | False | False | |
| 1884 | False | False | False | False | True | |

| | overweight_l1 | overweight_l2 |
|---|---|---|
| 530 | False | False |
| 259 | False | False |
| 678 | False | False |
| 479 | False | False |
| 1884 | False | False |

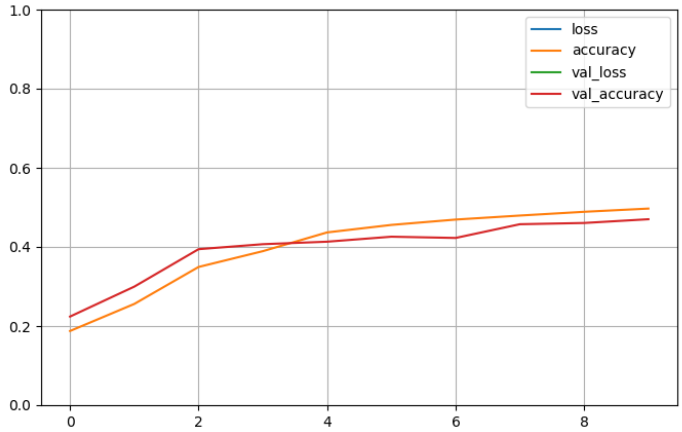
Model dobrze przewidział dopasowanie dla 3 obiektów, jednak dla pozostałych dwóch przy błędnych grupach prawdopodobieństwo dopasowania wynosi 100% co potwierdza nadmierne dopasowanie do zbioru treningowego.

## Analiza wszystkich wartości parametru:

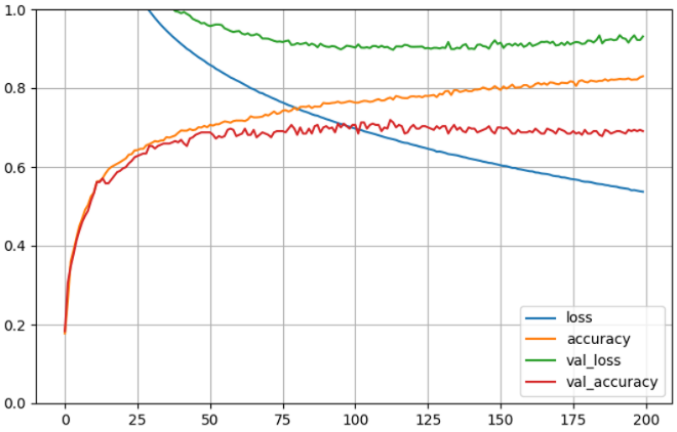
Model bazowy – 100 epok



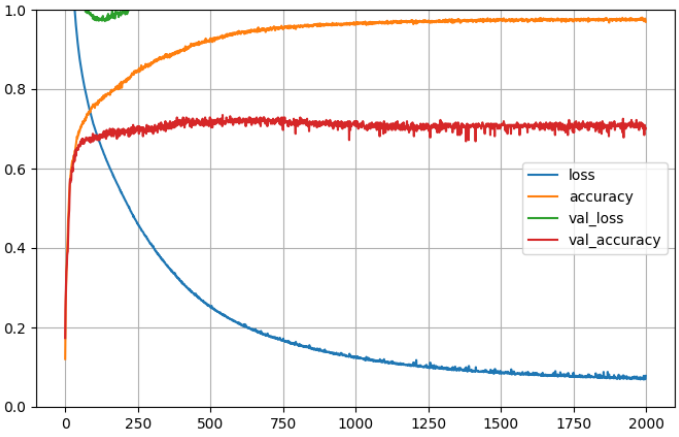

Model - 10 epok




Model – 200 epok




Model - 2000 epok




{'base': 0.7507886290550232, 'number of epocs: 10': 0.5015772581100464, 'number of epocs: 200': 0.7318611741065979, 'number of epocs: 2000': 0.7255520224571228}

Podsumowanie:

Liczbę epok należy dobrze wyważyć, ponieważ model uczony 10 epokami to zdecydowanie za mało – dokładność na zbiorze testowym to tylko około 50%, a 2000 epok to zdecydowanie za dużo – dokładność na zbiorze treningowym zdecydowanie odbiega od tej na zbiorze walidacyjnym oraz funkcja straty nie jest funkcją malejącą (dobrze to widać na wykresie wyżej) - overfitting. W modelu uczonym 200 epokami dokładności na zbiorach walidacyjnym i testowym bardziej się rozbiegają niż w modelu bazowym, oraz funkcja straty nie jest funkcją malejącą – są to oznaki overfittingu.

## Analiza podziału zestawu danych na zbiory: treningowe, walidacyjne i testowe

Wszystkie modele w analizie tego parametru będą miały te same podsumowania, więc dla przypomnienia zaprezentujemy je raz poniżej:

```
Model: "sequential_13"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_41 (Dense)            (None, 32)                480

 dense_42 (Dense)            (None, 32)                1056

 dense_43 (Dense)            (None, 7)                 231

=================================================================
Total params: 1767 (6.90 KB)
Trainable params: 1767 (6.90 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

## Zbiór treningowy – 50%, testowy 25%, walidacyjny 25%

Gdy dane podzielono na 50% zbiór treningowy i liczące po 25% zbiory walidacyjne i testowe, model uzyskał 74,22% dokładności na zbiorze treningowym i 69,32% dokładności na zbiorze walidacyjnym (są to wyniki zbliżone do modelu bazowego, gdzie zbiory liczyły odpowiednio 70, 15 i 15 proc.)

```
Epoch 100/100

 1/33 [..............................] - ETA: 0s - loss: 0.6676 - accuracy: 0.8125
33/33 [==============================] - 0s 2ms/step - loss: 0.7591 - accuracy: 0.7422 - val_loss: 0.8973 - val_accuracy: 0.6932
```

Dokładność na zbiorze testowym z kolei jest dość niska – wynosi mniej niż 70%

```
 1/17 [>.............................] - ETA: 0s - loss: 1.1360 - accuracy: 0.6562
17/17 [==============================] - 0s 823us/step - loss: 0.9554 - accuracy: 0.6742
```

Prognoza

| | insufficient_w | normal_w | obesity_t1 | obesity_t2 | obesity_t3 | \ |
|---|---|---|---|---|---|---|
| 0 | 0.77 | 0.19 | 0.00 | 0.0 | 0.00 | |
| 1 | 0.68 | 0.24 | 0.03 | 0.0 | 0.02 | |
| 2 | 0.64 | 0.24 | 0.00 | 0.0 | 0.00 | |
| 3 | 0.92 | 0.07 | 0.00 | 0.0 | 0.00 | |
| 4 | 0.00 | 0.00 | 0.00 | 0.0 | 0.98 | |

| | overweight_l1 | overweight_l2 |
|---|---|---|
| 0 | 0.01 | 0.03 |
| 1 | 0.01 | 0.02 |
| 2 | 0.03 | 0.09 |
| 3 | 0.00 | 0.00 |
| 4 | 0.01 | 0.01 |

| | insufficient_w | normal_w | obesity_t1 | obesity_t2 | obesity_t3 |
|---|---|---|---|---|---|
| 530 | True | False | False | False | False |
| 259 | False | False | True | False | False |
| 678 | True | False | False | False | False |
| 479 | False | True | False | False | False |
| 1884 | False | False | False | False | True |

| | overweight_l1 | overweight_l2 |
|---|---|---|
| 530 | False | False |
| 259 | False | False |
| 678 | False | False |
| 479 | False | False |
| 1884 | False | False |

Model poprawnie dopasował grupę dla 3 obiektów jednak prawdopodobieństwa dla dwóch z nich są dość niskie

## Zbiór treningowy 50%, zbiór testowy 50%

Gdy dane podzielono na 50% zbiór treningowy i  50% zbiór testowy (bez zbioru walidacyjnego), model uzyskał 75,17% dokładności na zbiorze treningowym, co jest bardzo podobnym wynikiem do poprzedniego modelu.

```
Epoch 100/100

 1/33 [..............................] - ETA: 0s - loss: 1.0020 - accuracy: 0.6250
33/33 [==============================] - 0s 750us/step - loss: 0.7298 - accuracy: 0.7517
```

Dokładność na zbiorze testowym wynosi 67,99% co również jest bardzo zbliżonym wynikiem do poprzedniego modelu. Wnioskujemy, że pomimo tego, że zbiór walidacyjny pomaga w ocenie, jak model będzie działał na nowych, niewidzianych wcześniej danych, to zrezygnowanie z niego nie wpłynęło znacząco na wynik naszych modeli.

```
 1/33 [..............................] - ETA: 2s - loss: 0.7935 - accuracy: 0.6875
33/33 [==============================] - 0s 779us/step - loss: 0.9259 - accuracy: 0.6799
```

Prognoza:

| | insufficient_w | normal_w | obesity_t1 | obesity_t2 | obesity_t3 | \ |
|---|---|---|---|---|---|---|
| 0 | 0.75 | 0.17 | 0.00 | 0.0 | 0.00 | |
| 1 | 0.36 | 0.24 | 0.02 | 0.0 | 0.31 | |
| 2 | 0.42 | 0.22 | 0.00 | 0.0 | 0.00 | |
| 3 | 0.93 | 0.07 | 0.00 | 0.0 | 0.00 | |
| 4 | 0.00 | 0.00 | 0.00 | 0.0 | 0.98 | |

| | overweight_l1 | overweight_l2 |
|---|---|---|
| 0 | 0.00 | 0.08 |
| 1 | 0.02 | 0.06 |
| 2 | 0.01 | 0.34 |
| 3 | 0.00 | 0.00 |
| 4 | 0.01 | 0.00 |

| | insufficient_w | normal_w | obesity_t1 | obesity_t2 | obesity_t3 | \ |
|---|---|---|---|---|---|---|
| 530 | True | False | False | False | False | |
| 259 | False | False | True | False | False | |
| 678 | True | False | False | False | False | |
| 479 | False | True | False | False | False | |
| 1884 | False | False | False | False | True | |

| | overweight_l1 | overweight_l2 |
|---|---|---|
| 530 | False | False |
| 259 | False | False |
| 678 | False | False |
| 479 | False | False |
| 1884 | False | False |

Model znowu poprawnie przewidział grupę dla 3 z 5 grup

## Zbiór treningowy 90%, zbiór testowy 5%, zbiór walidacyjny 5%

Gdy dane podzieliliśmy na zbiory zawierające odpowiednio 90, 5 i 5 proc. model uzyskał 75,3% dokładności na zbiorze treningowym i 71,7% na zbiorze walidacyjnym, więc nie sugeruje to przetrenowania modelu.

```
Epoch 100/100

 1/60 [..............................] - ETA: 0s - loss: 0.6326 - accuracy: 0.7188
60/60 [==============================] - 0s 1ms/step - loss: 0.6869 - accuracy: 0.7530 - val_loss: 0.8759 - val_accuracy: 0.7170
```

Dokładność na zbiorze testowym wynosi 74,53%, co jest bardzo zbliżonym wynikiem do modelu bazowego.

```
1/4 [======>.......................] - ETA: 0s - loss: 0.8699 - accuracy: 0.7188
4/4 [==============================] - 0s 1ms/step - loss: 0.8238 - accuracy: 0.7453
```

Prognoza:

| | insufficient_w | normal_w | obesity_t1 | obesity_t2 | obesity_t3 | \ |
|---|---|---|---|---|---|---|
| 0 | 0.87 | 0.11 | 0.00 | 0.00 | 0.00 | |
| 1 | 0.49 | 0.37 | 0.07 | 0.02 | 0.01 | |
| 2 | 0.64 | 0.20 | 0.00 | 0.00 | 0.00 | |
| 3 | 0.86 | 0.13 | 0.00 | 0.00 | 0.00 | |
| 4 | 0.00 | 0.00 | 0.00 | 0.00 | 0.99 | |

| | overweight_l1 | overweight_l2 |
|---|---|---|
| 0 | 0.00 | 0.01 |
| 1 | 0.03 | 0.01 |
| 2 | 0.00 | 0.15 |
| 3 | 0.00 | 0.00 |
| 4 | 0.00 | 0.00 |

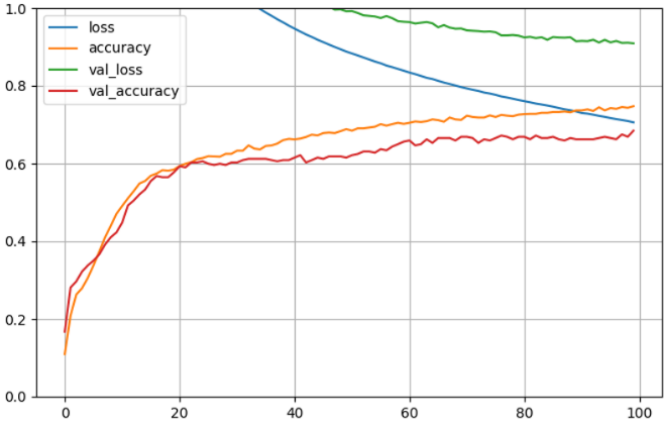
| | insufficient_w | normal_w | obesity_t1 | obesity_t2 | obesity_t3 | \ |
|---|---|---|---|---|---|---|
| 530 | True | False | False | False | False | |
| 259 | False | False | True | False | False | |
| 678 | True | False | False | False | False | |
| 479 | False | True | False | False | False | |
| 1884 | False | False | False | False | True | |

| | overweight_l1 | overweight_l2 |
|---|---|---|
| 530 | False | False |
| 259 | False | False |
| 678 | False | False |
| 479 | False | False |
| 1884 | False | False |

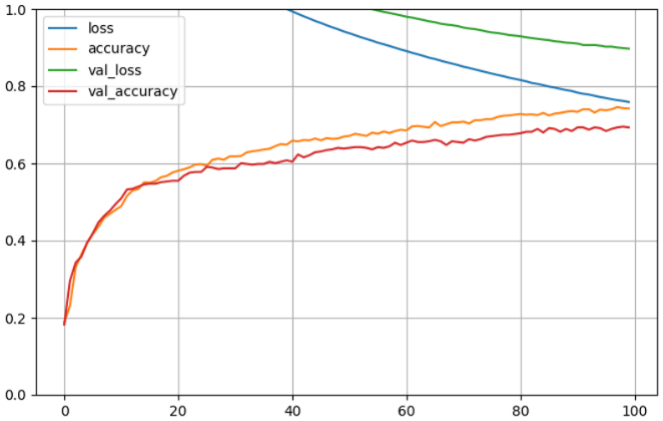
Model poprawnie przewidział grupę dla 3 obiektów i prawdopodobieństwa przynależności do nich są całkiem wysoki

### Analiza wszystkich wartości parametru

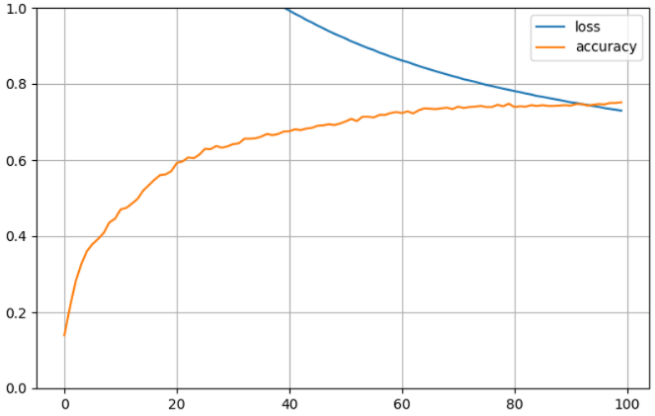
Model bazowy

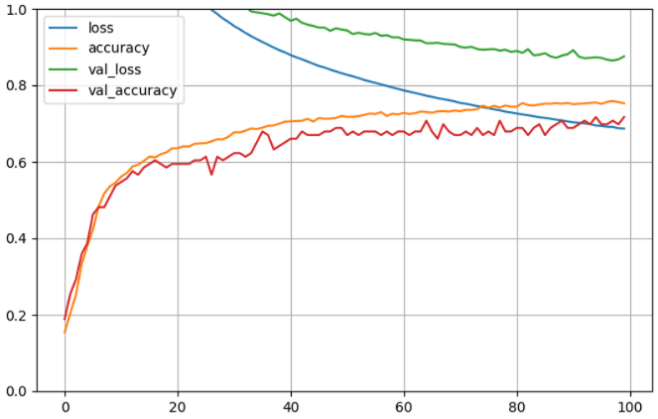



Model – 50%, 25%, 25%




Model – 50%, 50% (val=0%)




{'base': 0.7507886290550232, 'set: 50% training, 25%val, 25%test': 0.6742424368858337, 'set: 50% training, 50%test': 0.6799242496490479, 'set: 90% training, 5%val, 5%test': 0.7452830076217651}

Podsumowanie:
Na pierwszy rzut oka wykresy sa bardzo podobne, no może poza wykresem dotyczącym modelu bez zbioru walidacyjnego. Jednak patrząc na dokładności na zbiorach testowych, można dojść do wniosku, że rozmiary zbiorów: treningowego, testowego i walidacyjnego istotnie wpływają na dokładność modelu.

## 4. Podsumowanie projektu

Celem projektu było nauczenie modelu potrafiącego przypisać osoby do odpowiedniej kondycji fizycznej na podstawie informacji dotyczących ich stylu życia. Najlepszy model uzyskał 76% dokładności i była to wersja z 200 neuronami w warstwie, jednak nie jest to znacznie lepszy wynik od modelu bazowego (75%).

## 5. Porównanie z innymi modelami ekonometrycznymi

Jako dodatek przeprowadziliśmy porównanie SNN z regresją liniową, regresją wielomianową, regresją logistyczną oraz nieliniową klasyfikacją SVM. Modele były trenowane na tych samych zestawach danych.

W porównaniu z regresją liniową oraz wielomianową obliczyliśmy MSE i wartość dla modelu sztucznych sieci neuronowych jest mniejsza o odpowiednio około 0,04 i 0,03. Wyniki wskazują, że sztuczne sieci neuronowe mogą lepiej radzić sobie z przewidywaniem.

```
MSE SNN:  0.060539596
MSE Regresja liniowa:  0.09553960906792969
MSE Regresja wielowymiarowa:  0.08908523817647979
```

W porównaniu SNN z regresją logistyczną i SVM zmierzyliśmy dokładność modeli

```
Dokładność SNN:  0.7507886290550232
Dokładność regresji logistycznej:  0.5741324921135647
Dokładność modelu SVM: 0.7665615141955836
PS C:\Users\Marcin\Desktop\ProjektII>
```

Okazało się, że nieliniowa klasyfikacja SVM zdobyła trochę lepszy wynik od sztucznych sieci neuronowych. Zarówno SVM, jak i SNN mają różne hiperparametry, które można dostosować, aby zoptymalizować wydajność modelu, więc bardzo prawdopodobne, że lepiej „dostrojony” model sieci neuronowych uzyskałby lepsze wartości.

## 6. Bibliografia


http://drogaprogramisty.pl/2021/04/07/od-machine-learning-do-deep-learning/

https://www.mdpi.com/2076-3417/13/6/3875

https://www.mdpi.com/1660-4601/18/11/5597

Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. Concepts, Tools, and Techniques to Build Intelligent Systems – Aurelien Geron