Raport do zadania 5

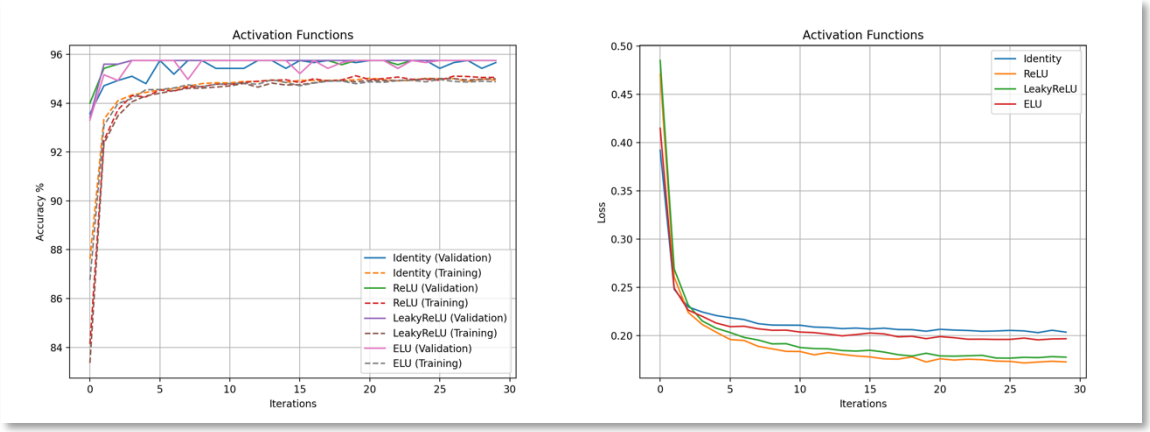
Sztuczne sieci neuronowe

Kacper Siemionek, Kajetan Witkowski

Numery indeksu: 331430, 331452

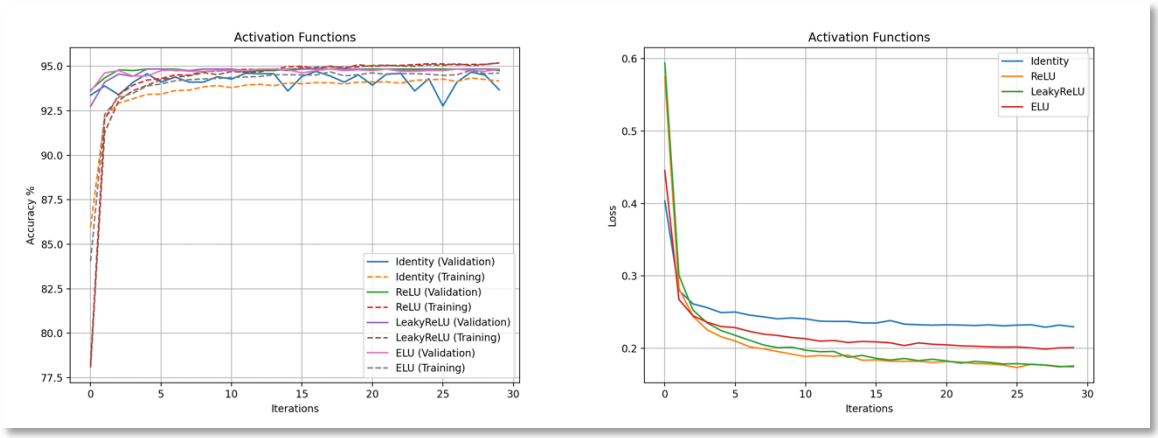
1. Porównanie funkcji aktywacji

Przeprowadzono testy na różnych ilościach warstw ukrytych porównujące następujące funkcje aktywacji: tożsamościowa (Identity), ReLU, LeakyReLU, ELU. W testach warstwy ukryte miały 64 neuronów, model uczył się przez 100 epok z krokiem uczącym o wartości 0,1.



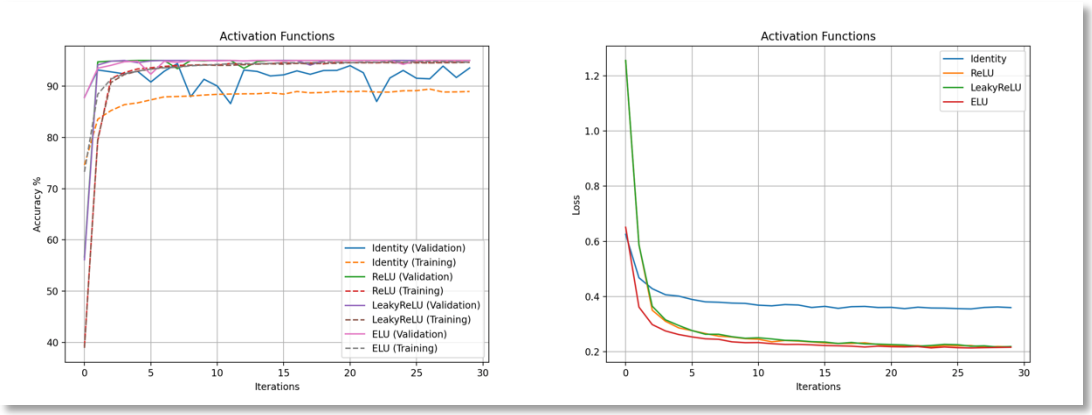
Rysunek 1.1 Porównanie funkcji aktywacji dla 1 warstwy ukrytej

Na powyższych wykresach możemy zauważyć, że w przypadku 1 warstwy ukrytej każda funkcja aktywacji była w stanie osiągnąć dobrą dokładność w podobnym czasie. Wykresy dokładności sprawdzanej na zbiorze treningowym nakładają się na siebie, co świadczy o bardzo podobnych wynikach. Na tym poziomie jedyne różnice, jakie możemy zauważyć to szybkość zmniejszania strat. Funkcje ReLU oraz LeakyReLU ostatecznie uzyskały najmniejsze straty w procesie uczenia.



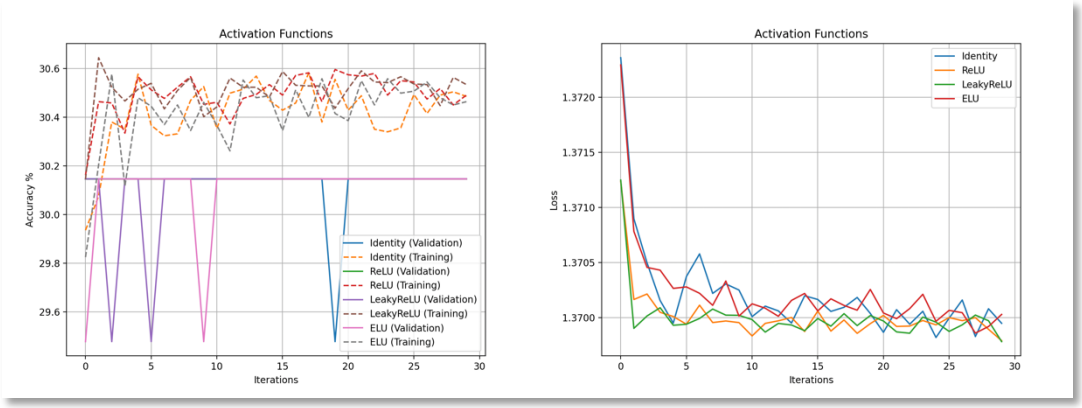
Rysunek 1.2 Porównanie funkcji aktywacji dla 2 warstw ukrytych

Zainicjowanie modelu z 2 warstwami ukrytymi sprawiło, że dokładność na zbiorach treningowych osiągała większe wartości. Jedynie wyniki funkcji tożsamościowej są minimalnie gorsze, dokładność walidacyjna lekko odstaje od pozostałych funkcji. To samo w przypadku drugiego wykresu – największe straty zanotowała funkcja tożsamościowa.



Rysunek 1.3 Porównanie funkcji aktywacji dla 5 warstw ukrytych

Zmiana na 5 warstw ukrytych nie wpłynęła znacznie na dokładność modelu, wciąż utrzymywała się w okolicach 95%. Model z funkcją tożsamościową wciąż odstawał od pozostałych pod względem dokładności walidacyjnej i treningowej oraz osiągał większe straty.

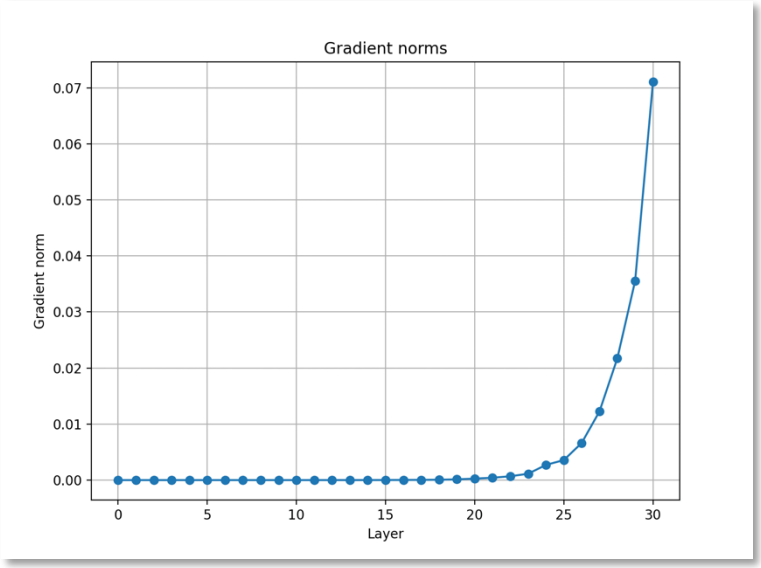


Rysunek 1.4 Porównanie funkcji aktywacji dla 30 warstw ukrytych

W ostatnim teście model uczył się z 30 warstwami ukrytymi, co negatywnie wpłynęło na wyniki. Straty każdego modelu były nieporównywalnie większe od tych w poprzednich testach, a ich dokładności nie przekraczały 31%

Tabela 1.1 Średnia norma gradientów dla 30 warstw ukrytych

Numer warstwy	Śr. norma gradientów	Numer warstwy	Śr. norma gradientów
1	$8{,}874^{-9}$	**17**	$2{,}814^{-5}$
2	$1{,}613^{-8}$	**18**	$4{,}174^{-5}$
3	$1{,}729^{-8}$	**19**	$6{,}487^{-5}$
4	$2{,}204^{-8}$	**20**	$1{,}282^{-4}$
5	$3{,}213^{-8}$	**21**	$2{,}584^{-4}$
6	$5{,}213^{-8}$	**22**	$4{,}168^{-4}$
7	$9{,}434^{-8}$	**23**	$6{,}813^{-4}$
8	$1{,}617^{-7}$	**24**	$1{,}139^{-3}$
9	$2{,}235^{-7}$	**25**	$2{,}711^{-3}$
10	$4{,}749^{-7}$	**26**	$3{,}572^{-3}$
11	$8{,}202^{-7}$	**27**	$6{,}622^{-3}$
12	$1{,}506^{-6}$	**28**	$1{,}226^{-2}$
13	$2{,}689^{-6}$	**29**	$2{,}172^{-2}$
14	$4{,}120^{-6}$	**30**	$3{,}556^{-2}$
15	$7{,}725^{-6}$	**31**	$7{,}108^{-2}$
16	$1{,}354^{-5}$	**-**	-



Rysunek 1.5 Średnia norma gradientów dla 30 warstw ukrytych

Z każdą kolejną warstwą sieci możemy zauważyć zwiększającą się normę gradientów. W początkowych warstwach wartości norm gradientów są na tyle małe, że możemy mówić o problemie znikania gradientów – zmiany w wagach są niewielkie, a sieć nie uczy się efektywnie. To tłumaczy, dlaczego w poprzednich testach modele z taką ilością warstw ukrytych nie były w stanie nauczyć się poprawnie.

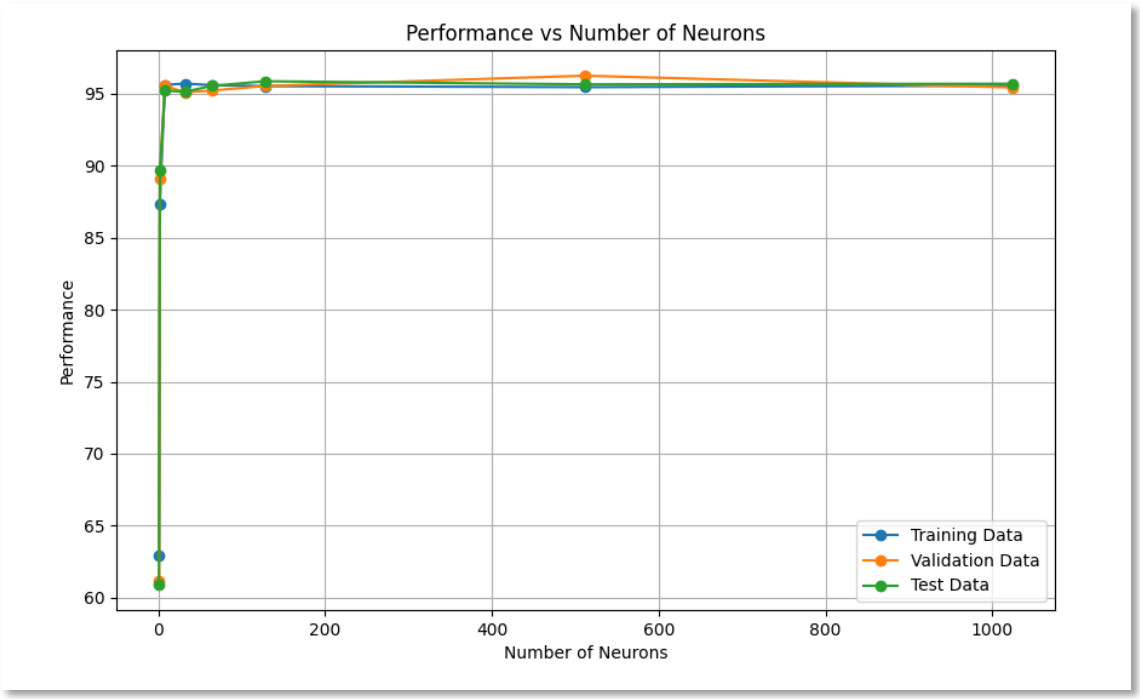
2. Wpływ liczby neuronów w ukrytych warstwach

W ramach tej części ćwiczenia pomiary wykonaliśmy dla 1 warstwy, 100 epok, kroku uczącego wynoszącego 0,1 oraz wsadów o wielkości 64.

Tabela 2.1 Wpływ liczby neuronów w ukrytych warstwach na dokładność modelu

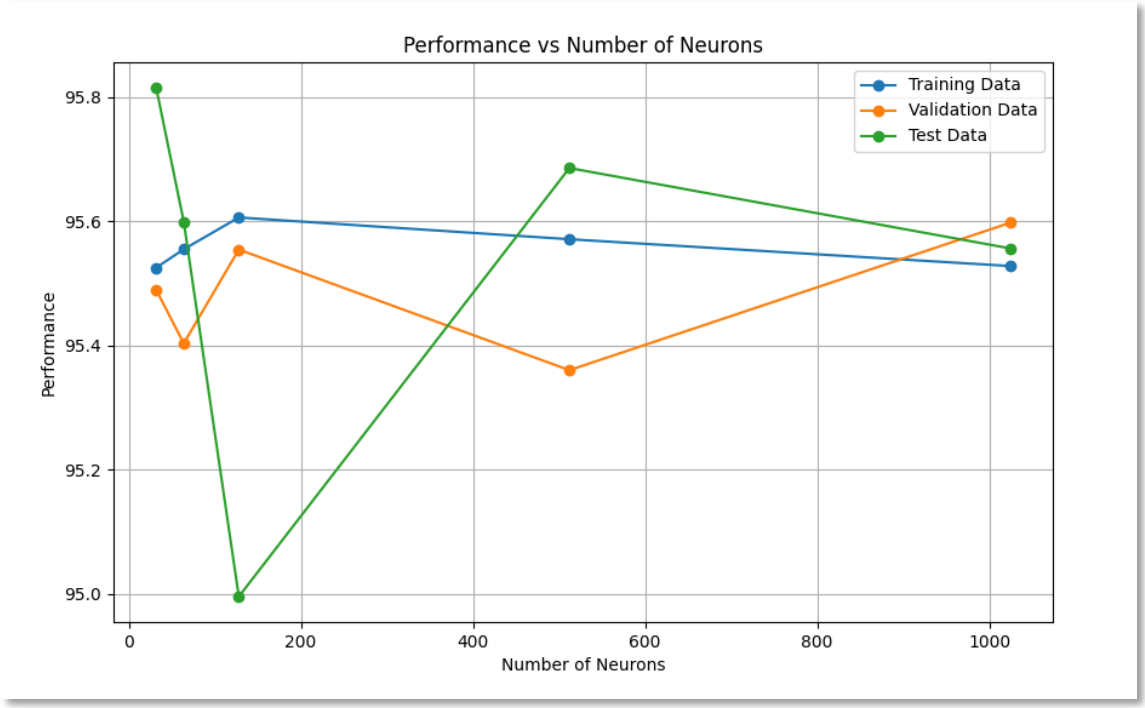
Liczba neuronów	Wyniki na zbiorze treningowym	Wyniki na zbiorze walidacyjnym	Wyniki na zbiorze testowym
1	62,95%	61,15%	60,91%
2	87,28%	89,07%	89,64%
8	95,66%	95,53%	95,21%
32	95,68%	95,17%	95,26%
64	95,65%	95,55%	95,22%
128	95,59%	95,56%	95,00%
512	95,59%	95,54%	95,24%
1024	95,51%	95,49%	95,27%

Otrzymane powyżej wyniki zostały zobrazowane na poniższych wykresach:



Rysunek 2.1 Wpływ liczby neuronów w warstwie ukrytej na dokładność modelu

W celu lepszej wizualizacji zachowania modelu dla większej liczby neuronów poniższy wykres nie zawiera danych o wynikach modelu z 1, 2 oraz 8 neuronami.

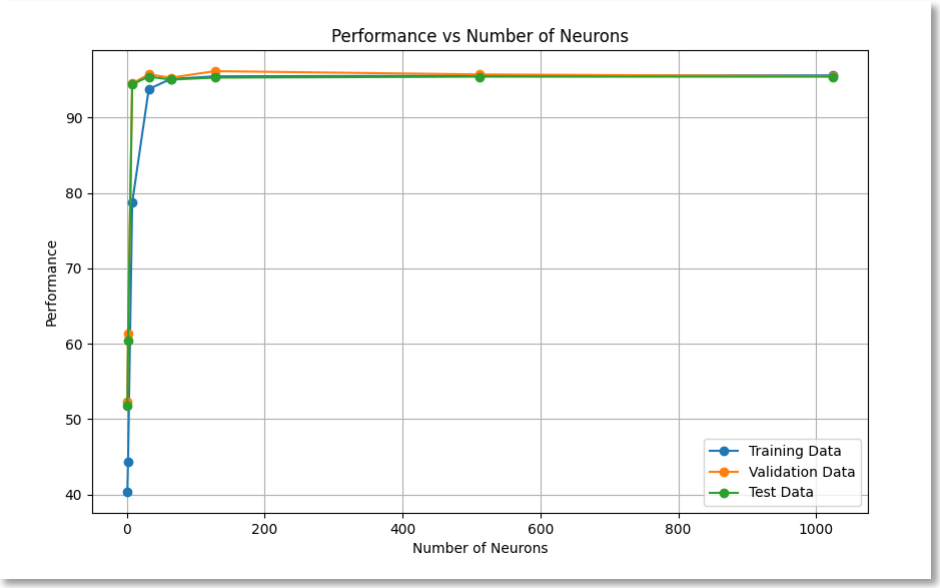


Rysunek 2.2 Wpływ liczby neuronów w warstwie ukrytej na dokładność modelu (mniejszy zakres)

Dla ilości neuronów 1, 2 oraz 8 model relatywnie słabą jakość nauki. Wynika to z faktu, że 12 atrybutów wejściowych przetwarzane było przez pojedyncze neurony, które nie były w stanie odpowiednio reprezentować danych. W miarę zwiększania liczy neuronów zwiększała się jakość modelu. Model osiągał najlepsze wyniki, gdy liczba neuronów znajdowała się w przedziale [64,1024] osiągając najbardziej zadowalający balans pomiędzy liczbą a wynikiem dla 128 neuronów w warstwie.

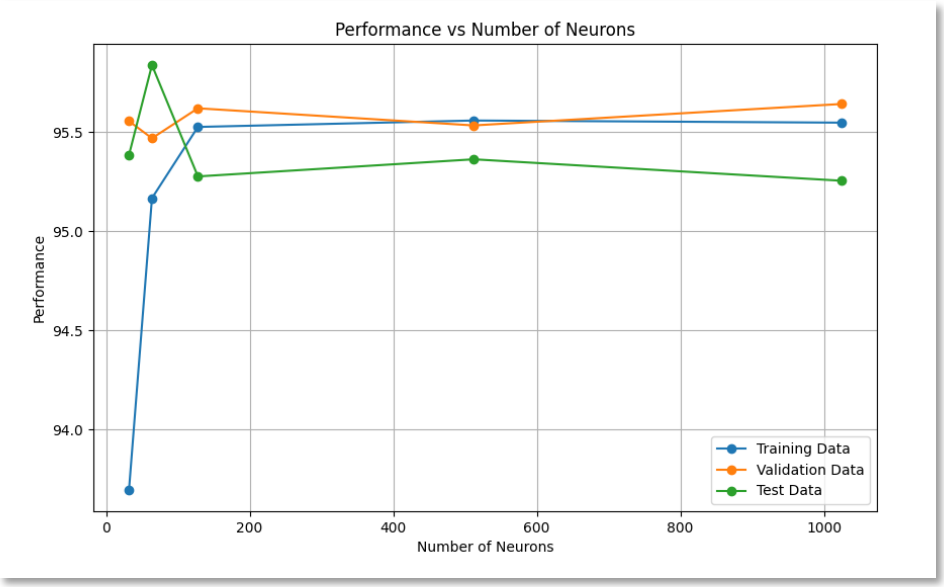
W załączonych przypadkach nie występuje przeuczenie, model nie dopasowuje się zbytnio do danych treningowych - świadczy o tym podobny poziom poprawnych wyborów ruchu co dla danych walidacyjnych oraz testowych. W przypadkach, gdy model trenowany jest na 1, 2 oraz 8 neuronach można zaobserwować zjawisko niedouczenia – model osiąga słabe wyniki na zbiorze treningowym. Wynika to ze słabych właściwości modelu do wnioskowania, które powstają poprzez słabe procesowanie danych – liczba neuronów jest zdecydowanie za mała.

Mimo braku przeuczenia sprawdzono wpływ dodania warstw odrzucających na wyniki. Rezultaty przedstawiono na wykresie poniżej:



Rysunek 2.3 Wpływ warstwy odrzucającej na dokładność modelu

Ponownie załączono wykres lepiej obrazujący wyniki dla większych liczb neuronów.



Rysunek 2.4 Wpływ warstwy odrzucającej na dokładność modelu (mniejszy zakres)

Jak przedstawiają wykresy, poprzez dodanie warstw odrzucających wyniki na danych testowych i walidacyjnych dla małych ilości neuronów znacznie szybciej rosły, przy podobnych wzrostach na danych treningowych. Wynika to z faktu, że w celu lepszego dobrania wag modelu w trakcie treningu niektóre neurony są wyłączane przez co podejście do każdego z nich jest bardziej indywidualne. W testowaniu wykorzystuje się ponownie wszystkie neurony. Największą różnicę widać dla 1 neuronu, gdzie tylko on może być wyłączany co negatywnie wpływa na wyniki redukując mu czas nauki. Zaobserwowano też mniejsze wahania wyników dla liczby 128 neuronów co korzystnie wpływa na wybór tego parametru dla najlepszego modelu.

3. Model osiągający najlepsze wyniki

Jako model osiągający najlepsze wyniki zakwalifikowano model o 128 neuronach, 5 warstwach ukrytych oraz funkcji aktywacji LeakyReLU. Algorytm wytrenowano w 100 epokach przy wykorzystaniu kroku uczącego 0,1. W 100 rozgrywkach model uzyskał średnio 16,7 punktu, czyli bardzo podobnie jak w przypadku ML.

Sporą różnicą pomiędzy ML a NN był fakt potrzeby znacznie większej ilości danych do wytrenowania modelu. Ze względu na większe skomplikowanie modelu zwiększenie ilości danych pozwoliło więcej razy przepropagować gradienty przez całą sieć ustawiając wagi na bardziej efektywne. Miało to tym większe znaczenie, że użyte zostały warstwy odrzucające, które wyłączały neurony w fazie treningu.

Log z trenowania wspomnianego modelu znajduje się w oddzielnym pliku (*trainlog.log*).