Sprawozdanie z projektu SSN

Predykcja wieku krabów

Łukasz Pyrek

# Opis problemu

Problem dotyczy określenia wieku krabów błotnych na podstawie fizycznych cech przedstawicieli gatunku, takich jak płeć, długość ciała, średnica zwierzęcia i jego wysokość, waga całkowita lub bez skorupy oraz waga organów wewnętrznych lub samej skorupy. Problem ten poza zastosowaniem go jako jeden z tematów do nauki sztucznych sieci neuronowych ma też zastosowanie ekologiczne oraz biznesowe. Wiele gatunków krabów jest zagrożonych wyginięciem, co stawia ekologów przed ciężkim zadaniem ich ochrony. Prawidłowe określenie wieku osobnika jest w stanie bardzo mocno ułatwić ten proces, m.in. poprzez ułatwienie określenia gotowości do rozmnażania czy też przewidywanej dalszej długości życia stworzenia. O ile kraby błotne nie są jednym z gatunków zagrożonych, to dysponując odpowiednimi danymi można w prosty sposób zmienić przedmiot nauki sieci, tak aby faktycznie odpowiadała temu problemowi. Jeśli chodzi o zastosowanie biznesowe, w tym przypadku badanie krabów błotnych jest jak najbardziej na miejscu, ponieważ stanowią one dużą część komercyjnie hodowanych skorupiaków. Poprawne określenie wieku zwierząt pozwala hodowcą odłowić je w optymalnym momencie, ponieważ odławianie zbyt młodych krabów powoduje, że nie dorastają do odpowiednich rozmiarów, co bezpośrednio oddziałuje na ich cenę. Podobnie odławianie osobników zbyt późno powoduje, że od pewnego momentu przestają istotnie rosnąć, a więc utrzymywanie takich osobników generuje niepotrzebne koszty i nie przynosi dodatkowego zysku.

# Dataset

Zbiór danych zawiera fizyczne cechy hodowanych krabów.

* Sex - Płeć kraba - Samiec, Samica i Nieokreślona.
* Length - Długość kraba (w stopach; 1 stopa = 30,48 cm)
* Diamete - Średnica kraba (w stopach; 1 stopa = 30,48 cm)
* Height - Wysokość kraba (w stopach; 1 stopa = 30,48 cm)
* Weight - Waga kraba (w uncjach; 1 funt = 16 uncji)
* Shucked Weight - Waga bez skorupy kraba (w uncjach; 1 funt = 16 uncji)
* Viascera Weight - Waga otaczająca organy brzuszne głęboko w ciele (w uncjach; 1 funt = 16 uncji)
* Shell Weight - Waga skorupy kraba (w uncjach; 1 funt = 16 uncji)
* Age - Wiek kraba (w miesiącach)

# Analiza parametrów

W celu predykcji wieku kraba będziemy korzystać z jednokierunkowej, wielowarstwowej sztucznej sieci neuronowej. Dla warstwy wyjściowej nie zostaje użyta żadna funkcja aktywacyjna. Trenowanie sieci następuje poprzez użycie metody gradientu prostego dla którego jeden krok w celu zmniejszenia funkcji kosztu jest obliczany z całego zbioru uczącego (batch gradient descent). Zbiór treningowy i testowy jest podzielony w proporcjach 80%:20%. Oceną skuteczności modelu jest błąd średniokwadratowy (MSE) dla zbioru testowego. Dane zostały znormalizowane.

W celu znalezienia modelu z najmniejszym błędem średniokwadratowym dla zbioru testowego zostaną przeanalizowane różne wartości następujących parametrów:

- ilość warstw ukrytych – [1, 2, 3, 4]

- ilość neuronów – [4, 8, 16, 32]

- funkcje aktywacji – [*ReLU, Leaky ReLU, Softplus, Sigmoid]*

Dla każdej kombinacji parametrów model zostanie nauczony dla szybkości uczenia równemu 0.0001 Uczenie sieci neuronowych jest procesem stochastycznym dlatego każdy model zostanie nauczony 4 razy.

Na początku znajdziemy optymalną ilość warstw i neuronów, w tym celu sprawdzimy 64 modeli (16 kombinacji parametrów, każda po 4 razy). Trenując modele skorzystamy z najpopularniejszej funkcji aktywacji – ReLU oraz 500 epok.

Tabela 1. 10 najlepszych modeli dla różnych ilości neuronów oraz warstw

|  |  |  |  |
| --- | --- | --- | --- |
| **Loss** | **Layers** | **Neurons** | **Activation** |
| 4,622642 | 4 | [8, 32, 32, 1] | ReLU |
| 4,694714 | 5 | [8, 32, 32, 32, 1] | ReLU |
| 4,88239 | 4 | [8, 32, 32, 1] | ReLU |
| 4,920865 | 5 | [8, 32, 32, 32, 1] | ReLU |
| 4,975367 | 5 | [8, 32, 32, 32, 1] | ReLU |
| 4,977194 | 4 | [8, 32, 32, 1] | ReLU |
| 5,068371 | 5 | [8, 32, 32, 32, 1] | ReLU |
| 5,128494 | 4 | [8, 16, 16, 1] | ReLU |
| 5,158242 | 6 | [8, 32, 32, 32, 32, 1] | ReLU |
| 5,249231 | 6 | [8, 16, 16, 16, 16, 1] | ReLU |

Osiągamy najlepsze rezultaty, stosując modele o architekturze posiadającej 2-3 warstwy ukryte, z każdą warstwą zawierającą po 32 neurony. Analizując pozostałe modele, zauważamy, że im mniejsza liczba neuronów w warstwie, tym model charakteryzuje się większą stratą. Jeśli chodzi o ilość warstw ukrytych, modele posiadające jedną warstwę nigdy nie osiągają satysfakcjonujących wyników, natomiast dla większej liczby warstw trudno dostrzec jakiekolwiek wyraźne zależności.

Znając potencjalne architektury dla których strata będzie najmniejsza przeanalizujemy teraz różne funkcje aktywacji dla 2,3 oraz 4 warstw ukrytych, każda posiadająca po 32 neurony. W tym celu nauczymy 48 modele (12 kombinacji parametrów każda po 4 razy). Tym razem dla osiągnięcia finalnego, najskuteczniejszego modelu trenować będziemy dla 1000 epok.

Tabela 2. 64 modele dla różnych funkcji aktywacji

Najlepszy wynik dostaliśmy dla modelu z 3 warstwami ukrytymi oraz dla funkcji aktywacji *leaky* *ReLU*. Wynik ten nie wiele się różni dla modeli z inną ilością warstw ukrytych oraz dla funkcji aktywacji *ReLU*. Uczenie sieci jest procesem stochastycznym, więc nie możemy stwierdzić jednoznacznie przewagi tej architektury nad pozostałymi z funkcjami aktywacji z rodziny *ReLU*.

Znaczna różnica pojawia się natomiast w przypadku użycia funkcji aktywacji *softplus*, gdzie uzyskane wyniki są istotnie gorsze, a najniższe wyniki notujemy przy zastosowaniu funkcji *sigmoid*. Warto zauważyć, że najgorsze wyniki, które znacząco odbiegają od reszty występują tylko dla największej ilości warstw ukrytych (4), może to wskazywać na występowanie problemu zanikającego gradientu. Problem ten wynika z tego, że podczas propagacji wstecznej gradienty mają tendencję maleć w miarę przechodzenia przez kolejne warstwy sieci, zwłaszcza gdy używane są funkcje aktywacji której pochodne mogą posiadać niskie wartości dla wysokich na moduł argumentów. Z tego powodu funkcje aktywacyjne takie jak ReLU są bardziej odporne na ten problem od funkcji sigmoid.