MPUM mini projekt 1 - Regresja liniowa

Tymoteusz Siemieniuk

Kwiecień 2024

1 Podział danych

Dane zostały podzielone w sposób:

- 60% - zbiór treningowy,
- 20% - zbiór walidacyjny,
- 20% - zbiór treningowy.

Do danych została również dodana kolumna stała równa 1.

2 Implementacje rozwiązań regresji

Rozwiązanie analityczne zostało zaimplementowane w standardowy sposób dla funkcji straty MSE (opcjonalne z regularyzacją L2) przy użyciu biblioteki numpy.

Rozwiązanie gradientowe zostało zaimplementowane dla każdej funkcji kosztu, która ma zaimplementowany gradient. Zaimplementowałem gradient dla MSE, MAE, a także dla tych funkcji z regularyzacjami L1, L2, L1+L2. Rozszerzyłem nieco klasyczną procedurę szukania minimum funkcji przy użyciu gradientu, o następujący pomysł, który jak się okaże czasem zwiększa prędkość zbieżności:

Załóżmy, że mając aktualne parametry równe θ, chcemy przejść do następnego kroku używając gradientu funkcji straty w punkcie θ - oznaczmy go przez ∇_θ .

Klasycznie, wykonalibyśmy: $\theta := \theta - \nabla_\theta * lr$ Jednak takie podejście jest podatne na rozbieżność, gdy lr jest za duże, bądź na zbyt wolną zbieżność, gdy lr jest zbyt małe. Proponuję więc następujące rozwiązanie:

Oznaczmy przez L optymalizowaną funkcję straty.

1. Zaczynamy w punkcie θ_1

2. obliczamy $\theta_2 := \theta_1 - \nabla_{\theta_1} \cdot lr$
3. Dopóki $L(\theta_2) > L(\theta_1)$ zmniejszamy $lr := lr/2$ i na nowo obliczamy θ_2
4. $\theta_p := \theta_2$
5. Dopóki $L(\theta_2) \leqslant L(\theta_p)$ zwiększamy $lr := lr \cdot 2$, ustawiamy $\theta_p := \theta_2$ i na nowo obliczamy θ_2
6. Robimy binsearch na funkcji straty pomiędzy θ_p a θ_2, wynik to θ_3
7. ustawiamy $\theta_1 := \theta_3$ i przechodzimy do początku - wykonaliśmy właśnie jeden krok po gradiencie.

Zaimplementowałem również early stoping: obliczam dwie średnie kroczące na stracie na zbiorze walidacyjnym. Nazwijmy te średnie stara i nowa. Stara to domyślnie średnia z 30 poprzednich strat, a nowa z 10. Jeśli $|\frac{nowa}{stara} - 1| < m$ to zatrzymuję spadek gradientu. Parametr m można ustalić, domyślna wartość wynosi 0.00001.

3 Sucha regresja

Zwykła regresja na nieprzetworzonych danych testowych otrzymuje wynik MSE równy 138k.

4 Regresja na danych znormalizowanych

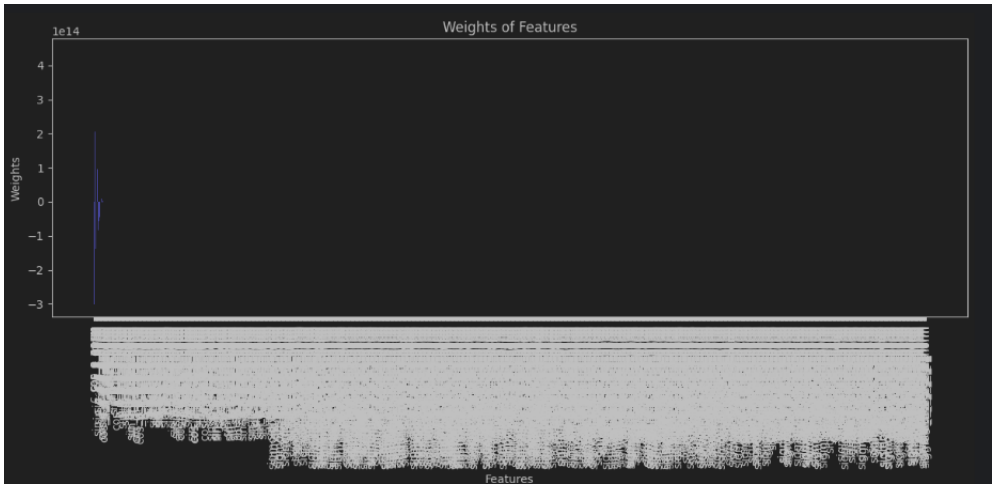
Po odjęciu od każdej kolumny (również od zmiennej objaśnianej (!)) jej średniej a następnie podzieleniu przez odchylenie standardowe i dopasowaniu modelu do takich danych otrzymujemy MSE równe 0.96, jednak jest to tylko pozorne polepszenie, gdyż znormalizowaliśmy y. Po pomnożeniu MSE przez $\sigma(y)^2$ otrzymujemy 136k - mniej więcej tyle co ostatnio. Nie powinno to dziwić, gdyż nasze modele w zasadzie się niczym nie różnią - dokonaliśmy jedynie *liniowych* transformacji na danych. Jest to jednak dobre sanity check dla modelu regresji.

5 Po dodaniu funkcji bazowych

Spróbujmy przetworzyć dane w następujący sposób:

1. Dodajmy dane będące wielomianami stopnia co najwyżej 5 (na przykład dane postaci $(x_0)^5$ lub $(x_0)^2(x_4)^3$
2. Znormalizujmy każdą kolumnę danych: $x_i := \frac{(x_i - \overline{x_i})}{\sigma(x_i)}$
3. Dodajmy inne transformacje: sin, cos, sigmoid

Otrzymujemy w sumie 1845 kolumn...
Po dopasowaniu modelu do danych treningowych otrzymujemy następujące straty: 995.7, 5582.6, odpowiednio na zbiorze treningowym i walidacyjnym. Nie jest to najlepszy model. Zobaczmy jak wyglądają jego wagi posortowane po wartości bezwzględnej:

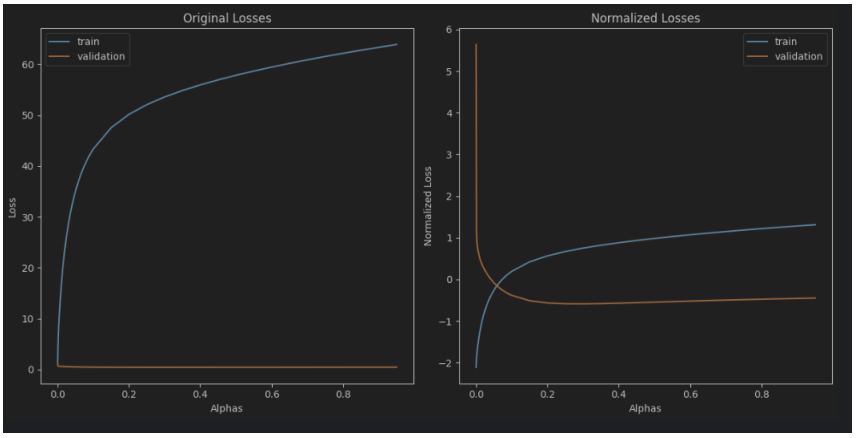


Te białe bazgroły na dole to nazwy zmiennych. Widać, że zdecydowana większa zmiennych jest niepotrzebna. Ponad to, model ten działa gorzej niż model bez dodanych zmiennych - strata jest mierzona w tysiącach, podczas gdy model używający mniejszej liczby zmiennych ma stratę bliską 1 (wszystko jest liczone na znormalizowanym y). Podejrzewam, że wrost straty wynika z niedokładności w zapisie liczb zmiennoprzecinkowych, co może powodować rozbieżność w operacjach macierzowych.

6 Regularyzacja

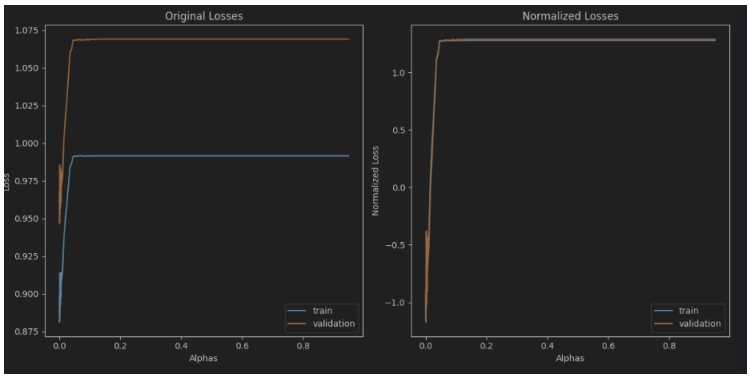
6.1 L2

Dla różnych regularyzacji L2 otrzymujemy następujący wykres funkcji straty (dodałem wykres znormalizowanych strat, żeby trendy były bardziej widoczne. Najlepszy parametr regularyzacji wynosi 0.3, co daje stratę równą 0.438 na zbiorze walidacyjnym.



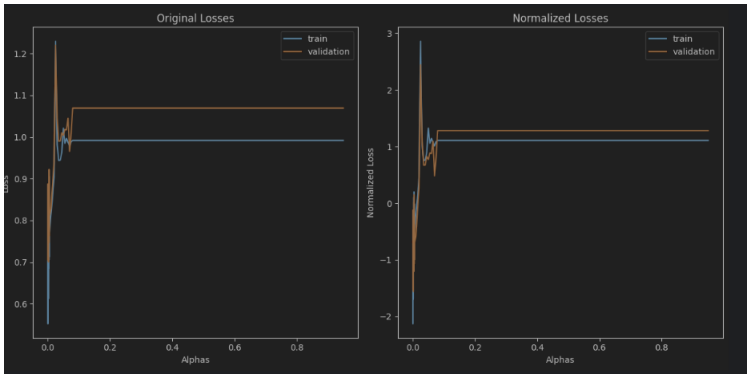
6.2 L1

Dla różnych regularyzacji L1 otrzymujemy natomiast poniższy wykres (parametry zostały obliczone za pomocą metody gradientowej, z hiperparametrami: `lr=1e-4, epochs=5000, optim=False, batch_size=256, min_improvement=1e-4`, gdzie `optim=False` oznacza, że nie wykonujemy omawianej przeze mnie wcześniej optymalizacji).



Rozpatrzyłem 86 różnych wartości parametru regularyzacji z przedzialu $[0, 1]$. Te 86 modeli liczyło się w sumie 6 minut i 17 sekund. Najlepszy z modeli osią-

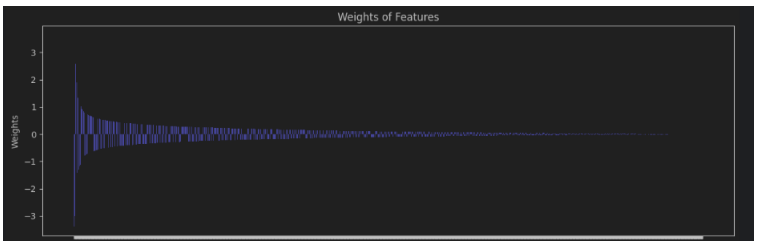
gnął MSE=0.94 na zbiorze walidacyjnym. Poniżej wyniki tych samych obliczeń, tylko że z hiperparametrem `optim=True` (czyli wykonujemy opisaną przeze mnie optymalizację spadku gradientu):

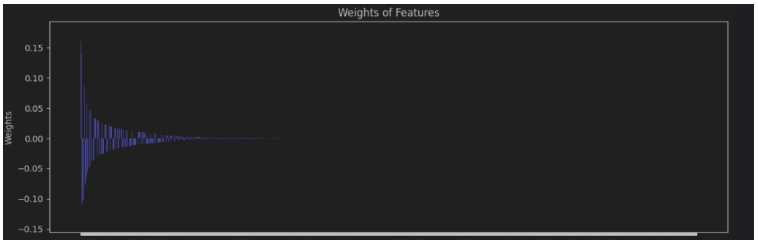


Teraz 86 modeli liczyło się 3 minuty 8 sekund, a najlepszy z modeli osiągnął MSE=0.70 na zbiorze walidacyjnym. Jest to moim zdaniem znacząca poprawa, choć MSE wciąż pozostawia wiele do życzenia.

6.3 L1 i L2

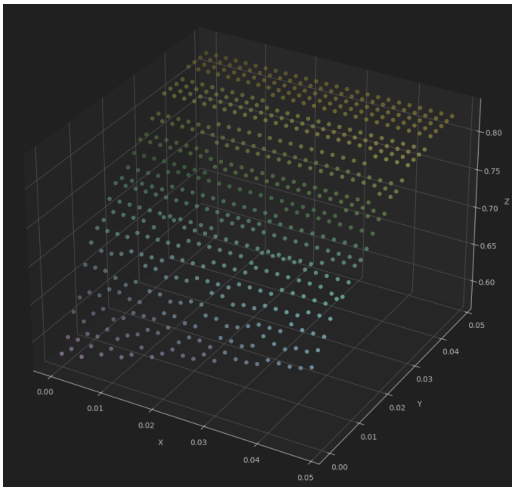
Zanim zacznę dobierać wagi dla regularyzacji z siecią elastyczną, nieco zmniejszę liczbę zmiennych. Oto wykresy wartości wag odpowiednio dla najlepszych modeli L2 i L1 wyliczonych w podpunktach 6.1 i 6.2:





Widać tutaj wyraźnie różnicę w tych regularyzacjach. Wybierzmy teraz 100 najlepszych zmiennych z obu modeli. Dodając do siebie te zbiory atrybutów

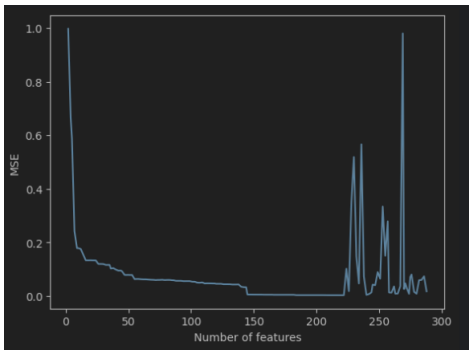
otrzymaliśmy w sumie 174 zmienne. Na tych zmiennych będziemy rozważać model z regularyzacją L1 i L2. Oto wykres MSE od parametrów regularyzacji:



Jak widać, najlepszy model to model, który w ogóle nie używa regularyzacji, osiągając MSE=0.57. Skoro i tak nie używamy to spróbujmy dopasować ten model analitycznie. Osiągamy w ten sposób MSE=0.004 - w końcu w miarę zadowalający wynik. Nie wiem skąd wynika tak słabe zachowanie metody gradientowej - testowałem dla kilku różnych szybkości uczenia, z moją optymalizacją oraz bez niej.

7 Modele na różnej liczbie najlepszych parametrów

Zobaczmy jak się zachowuje strata na zbiorze walidacyjnym w zależności od liczby parametrów używanych przez model (najlepsze parametry wybieramy patrząc na wartości bezwzględne modelów z sekcji 6.1 oraz 6.2):



Wzrost funkcji straty w okolicach 220 zmiennych prawdopodobnie wynika z niedokładności operacji macierzowych.

8 Najlepszy model

Najlepszy model używa 174 najbardziej istotnych zmiennych wyznaczonych przez modele z regularyzacją L1 i L2, dopasowane na zbiorze wszystkich 1825 wygenerowanych zmiennych. Osiąga on stratę MSE równą 0.0017 na zbiorze treningowym, 0.004 na walidacyjnym i 0.003 na testowym. Oto krzywa uczenia dla tego modelu: