

Optymalizacja Ciągła
Automatyczne różniczkowanie
wygenerowano: 23 kwietnia 2019

1 Treść zadania

Zadanie polega na automatycznym liczeniu pochodnych i gradientów podanych funkcji. Zbiór testów składa się z wielu instancji, z których każda zawiera wejście i wyjście w określonym formacie. Wejście składa się z funkcji i nazw zmiennych oraz ich wartości. Oczekiwane wyjście zawiera wartości gradientów/pochodnych w wyspecyfikowanych punktach. Więcej o formacie wejścia i wyjścia w dalszych akapitach.

2 Ocenianie

Instancje problemu podzielone są na publiczne i prywatne. Dla instancji publicznych pliki wejściowe są udostępnione, a wyniki będą widoczne na bieżąco. Dodatkowo, dla niektórych instancji publicznych udostępnione są pliki wyjściowe abyś mógł lokalnie przetestować swoją implementację. Instancje prywatne nie będą udostępnione, a wyniki będą ukryte - będą jednak one zawierać funkcje o ekwiwalentnej trudności jak w instancjach publicznych.

Zadania należy zgłaszać na platformę Optil.io. Rozwiązanie powinno czytać instancje ze standardowego wejścia (stdin) i wypisywać wyniki na standardowe wyjście (stdout). Oceną za zadanie będzie średnia wyników uzyskanych na instancjach **prywatnych** w dniu **16.06.2019** o godzinie **23:59**. Wynik dla instancji będzie średnią poprawnie wykonanych testów wewnątrz instancji. W celu weryfikacji konta należy też wysłać maila do odpowiadającego prowadzącego z kodem programu i loginem.

3 Format wejścia

Pierwsza linia zawiera liczbę całkowitą **n** oznaczającą liczbę przypadków w danej instancji. Każdy z **n** przypadków testowych zawiera:

- linię z listą **zmiennych** oraz ich wymiarowościami oddzielonych przecinkami w formacie NAZWA: (D0,D1,D2) gdzie DX to wymiary (np. x: (1), y: (3,2) definiuje zmienną x, która jest skalarą i zmienną y, która jest macierzą o 3 wierszach i 2 kolumnach),
- linię z listą parametrów oraz ich wymiarowościami w takim samym formacie jak zmienne - w przypadku braku parametrów linia zawiera **None**,
- linię z wyrażeniem będącym funkcją, z której mamy policzyć gradient, np. " $x^2 + 2 * x + 1$ ",
- linię z liczbą całkowitą **m** danych wejściowych dla tej funkcji, dla których trzeba policzyć gradient,
- m** linii zawierających dane wejściowe jako oddzielone przecinkami pary NAZWA: WARTOŚĆ (koniecznie spacja po dwukropku), w przypadku wektorów i macierzy wartości są zapisane za pomocą standardowej notacji zagnieżdżonych list (kwadratowe nawiasy i wartości oddzielone przecinkami). Wszystkie wartości zapisane są w jednej linii.

4 Format wyjścia

Dla każdego z **n** przypadków i **m** wartości zmiennych/parametrów należy wypisać linię zawierającą wartości pochodnych/gradientów **TYLKO PO ZMIENNYCH** w formacie identycznym jak lista wartości zmiennych i parametrów z wejścia. Kolejność zmiennych nie ma znaczenia.

Wartości liczbowe powinny być podane z dokładnością do **trzeciego** miejsca znaczącego, odpowiednio zaokrąglone (efekt ten osiągnąć można przez ustawienie precyzji wyświetlania do 3). Do obliczeń zalecane jest używanie 64-bitowych floatów. W razie wątpliwości co do formatu i dokładności numerycznej najlepiej skonsultować się z plikiem **random_solution.py**.

5 Przykładowe wejście/wyjście

Udostępniamy pliki wejściowe dla instancji publicznych (oraz niektóre pliki wyjściowe) np. **0_sample.in** i **1_sample.out**. Należy jednak zauważyć, że wejście/wyjście nie będą dostarczane w postaci plików, lecz w postaci wejścia i wyjścia standardowego.

6 Przykładowe rozwiązanie

Plik **random_solution.py** pokazuje przykładowe rozwiązanie zwracające gradienty o losowych wartościach, ale poprawnych rozmiarach.

7 Przetwarzanie wejścia

W celu ułatwienia wykonywania zadania dostarczamy gotowy kod wczytujący funkcję w formie stringa i zwracający drzewo, które w wierzchołkach ma operacje, a w liściach argumenty (parser.py). Format ten jest inspirowany reprezentacją używaną w programowaniu genetycznym

8 Możliwe funkcje i operacje

W przypadku funkcji, które zwyczajowo przyjmują jako argument jeden skalar należy założyć, że w przypadku podania wektora/macierzy funkcja ta zostanie zastosowana na każdym elemencie wektora/macierzy osobno, a wynikiem będzie wektor/macierz o oryginalnym kształcie (np. max, abs, “*”).

W sytuacji gdy pochodna funkcji jest nieciągła w punkcie należy zwrócić średnią wartość pochodnej lewostronnej i prawostronnej. W przypadku gdy pochodna może przyjąć więcej wartości niż jedną (np. dla max) należy wybrać średnią gradientów.

- dodawanie, odejmowanie,
- podnoszenie do potęgi ($^$) - można przyjąć, że wykładnik zawsze będzie stałą,
- mnożenie macierzowe (matmul(x,y)),
- mnożenie i dzielenie (* i /),
- transpozycja (.T),
- suma po danej osi (sum(x, axis)),
- średnia po danej osi (mean(x, axis)),
- funkcja wykładnicza exp(x),
- logarytm naturalny ln(x),
- sin(x), cos(x), tg(x),
- wartość bezwzględna (abs(x)),
- max i min z dwóch wartości (wartości mogą być wektorami),
- złożenie funkcji,

9 Dodatkowe uwagi

- Zabronione jest korzystanie z jakichkolwiek zewnętrznych pakietów do automatycznego liczenia pochodnych (choć nie powinno być to raczej możliwe na środowisku testowym).
- Zgłoszenie czyjegoś kodu skutkować będzie niezdaniem przedmiotu z brakiem możliwości poprawy.
- W przypadku wykrycia nieścisłości i/lub błędów proszę zwracać uwagi na michal.kempka@cs.put.poznan.pl