SIwB lab04

Predykcja wartości miary RMSD

Zespół: Max Adamski, Sławomir Gilewski,
Patryk Jedlikowski, Mikołaj Sienkiewicz

Zbiór danych

- Na podstawie wszystkich plików XML utworzono tabelę z wartościami RMSD dla par plików PDB. W sumie w zbiorze danych znajdują się 34114 pary z określoną miarą RMSD.
- Jako zbiór testowy wybraliśmy folder pz10 (3718 par, czyli ok. 10% zbioru), który powinien być niezależny od reszty.
- Pliki z reszty folderów to nasz zbiór uczący (losowo wybrane 85% par) i walidacyjny (5% par). Chcieliśmy zminimalizować ilość danych nieużywanych do uczenia.
- **Użyliśmy tylko danych kartezjańskich**. Chcieliśmy zobaczyć jaki model uda się uzyskać korzystając tylko z nich.
- Usunięto nieprzydatne kolumny (identyfikatory atomu/łańcucha/segmentu/reszty) i zakodowano one-hot kategoryczne cechy (reszta A/C/U/G, atom N/O/C), co jest wymagane w sieciach neuronowych. W sumie wychodzi 12 cech (koordynaty x, y, z, occupancy temperature factor, one-hot reszta A/C/U/G, one-hot atom N/O/C). Wybrane cechy reprezentują unikalne informacje, więc dokonaliśmy dodatkowej selekcji cech.
- **Na potrzeby naszego małego eksperymentu zbiór danych ma odpowiednią wielkość**. Biorąc pod uwagę, że rozważamy pary struktur 3D RNA, to **zbiór danych można potencjalnie znacznie zwiększyć, obliczając (np. RNAQUA) wartość RMSD pomiędzy wszystkimi 2-kombinacjami struktur**. W naszej architekturze to by pomogło w uczeniu części wykonującej regresję. Dodanie kolejnych struktur pomogłoby w uczeniu enkodera opartego na rekurencyjnych sieciach.

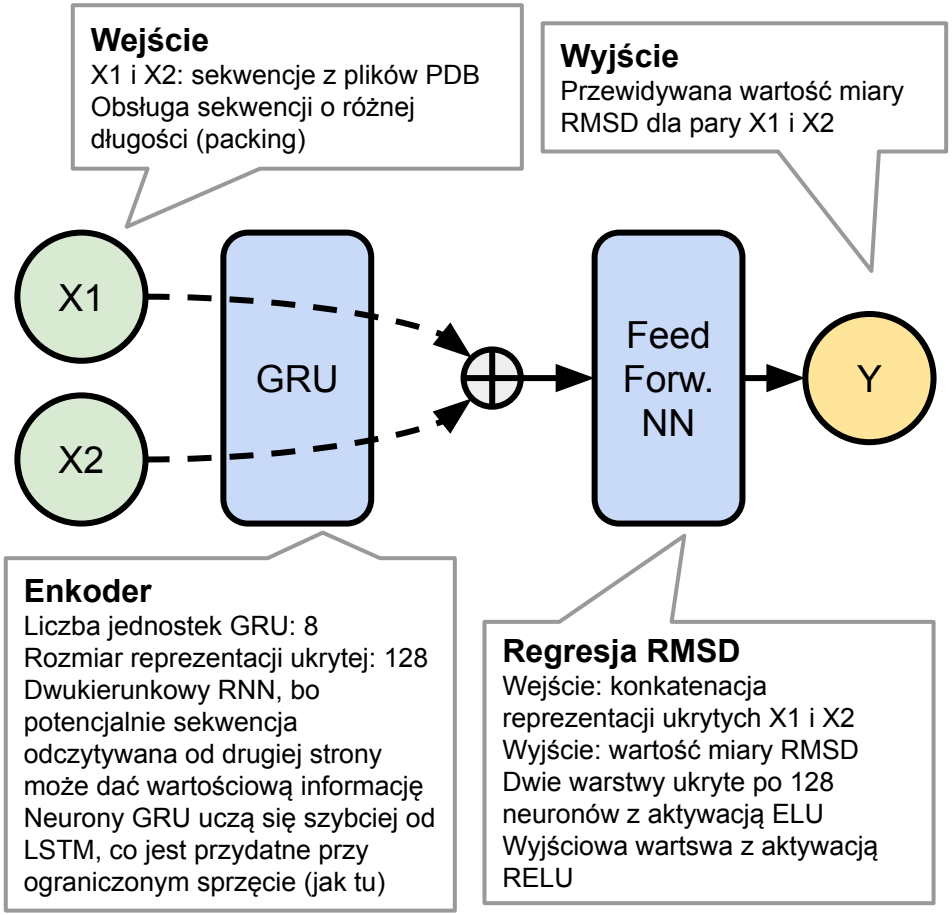
Architektura i uczenie

Koncepcja

- Nasza metoda tworzy **reprezentację ukrytą obu struktur RNA** przy użyciu rekurencyjnej sieci neuronowej, a następnie używa **sieci neuronowej feedforward do przewidzenia wartości RMSD**
- Wybraliśmy głębokie sieci neuronowe ponieważ musimy dokonywać regresji na podstawie sekwencji, do czego bardzo dobrze są przystosowane rekurencyjne sieci neuronowe. Część przewidująca wartości RMSD musi być siecią neuronową albo innym modelem umożliwiającym propagację wsteczną

Proces uczenia

- Funkcja straty: **MSE** (błąd średniokwadratowy), odpowiednia dla problemu regresji
- Optymalizator: **AdamW** [1], learning rate: 0.0001 (krzywa uczenia nie wskazała na potrzebę modyfikacji LR ani zmianę optymalizatora)
- 10 epok uczenia. Wielkość batcha: 32 [2]
- Optymalizacja hiperparametrów jest zbyt kosztowna (uczenie zajmuje ok. 4 godziny na GPU) dla modeli opartych na rekurencyjnych sieciach, dlatego jej nie wykonaliśmy. Potencjalnie można wykonać random search do optymalizacji, który ma lepsze właściwości niż grid search [X].



Wyniki i wnioski

- **RMSE na zbiorze testowym** (pierwiastek z błędu średniokwadratowego): 3.995
- **MAE na zbiorze testowym** (średni błąd bezwzględny): 3.065
- Z wykresu porównującego stratę na zbiorze uczącym i walidacyjnym wynika, że **nie nastąpiło przeuczenie** (stata walidacyjna nie rośnie w późniejszych epokach)
- Miara RMSE na zbiorze walidacyjnym spada w kolejnych epokach
- Wizualizujemy też rozkład błędów na zbiorze testowym (y - Y). Wynika z niego, że model raczej niedoszacowuje wartość miary błędu RMSD. Nasz model mógłby być użyty do pierwszego odsiewu struktur RNA o zbyt dużym RMSD.
- Możliwe, że wybranie architektury o większej liczbie parametrów, lub dodanie informacji z reprezentacji torsyjnej zwiększyłoby moc predykcyjną
- Napotkane problemy: Praca z sekwencyjnymi danymi i sieciami RNN jest dużym wyzwaniem, **ponieważ obliczenia są sekwencyjne, co znacznie wydłuża czas uczenia i spowolniło iteracyjny rozwój architektury**. Reprezentacja PDB okazała się nieporęczna i początkowo popełniliśmy błąd wynikający z naiwnego parsowania tych plików.