Hack jak Brno

PathoScan roadmap

immediate

Hospitalovo Plavidlo

Obsah

1. Prvotní myšlenky

Od začátku nás lákala myšlenka využít pro identifikaci patogenů metody hlubokého učení, která (jak jsme zjistili) není pro tento problém zatím hojně využívaná. První nápady směřovaly na použití konvolučních neuronových sítí a převodu sekvenačního čtení na obrázky pomocí k-mer frekvenční matice či dalších používaných funkcí. Tento nápad se nám zdál ale poměrně výpočetně a logicky náročný a potřebovaly bychom k němu vytvořit rozsáhlý anotovaný dataset. Záhy nás napadlo vytvořit nějaké mapování, které by mohlo převádět sekvenční čtení na číselnou vektorovou reprezentaci a převést problém vyhledávání podřetězce na vyhledávání nejbližšího vektoru ve vektorové databázi. Tento nápad se nám zdál natolik zajímavý, že jsme se ho rozhodli plně následovat. Navíc se nabízela možnost využít vektorové databáze IRIS, kterou poskytuje firma InterSystems.

2. Možnosti embeddingu sekvenčního čtení

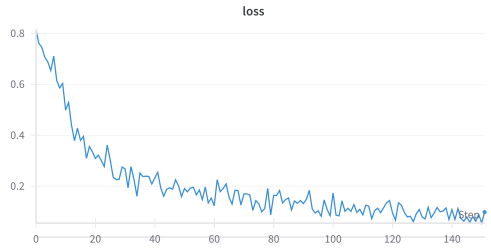
Pro mapování genových řetězců do vektorové reprezentace nemůžeme použít klasické encoder-based transformery, jelikož musíme nějak vzít v úvahu že nepracujeme s klasickými řetězci, nýbrž s řetězci nad omezenou abecedou $= \{A, C, G, T\}$, které jsou často neúplné (vlivem například chybou skenování). Některé čtení se navíc překrývají a nesou tak "stejnou"informaci. Po základní rešerši jsme našli dvě architektury, které by pro náš problém mohli fungovat, konkrétně se jedná o model **DNABert** [1] a **RDE transformer** [2].

3. DNABert

Jako první jsme se rozhodli použít model **DNABert**, který tokenizuje nějaké čtení a převádí jej do vektorového prostoru $\mathbf{R}^{768}$. Výhodou použití této architektury je jednoduchý přístup k předtrénovaným vahám přes prostředí huggingface. S modelem jsme embeddnuli přes 40 000 vektorů a vyzkoušeli jeho přesnost při hledání nejbližších vektorů k embeddutým patogenův. Ačkoliv model není přímo určen pro použití na úlohu hledání substringů, tak jsme zaznamenali pozitivní výsledky a motivovalo nás to na natrénovaní a použití více task specifického modelu.

4. RDE encoder-transformer

Jako druhý model jsme se rozhodli použít **Reference free DNA encoder**. Model následuje stejně jako **DNABert** encoder only transformer architekturu, ale narozdíl od druhého zmiňovaného modelu je přímo určen pro použití v úkolu celogenomového sekvenování. RDE model se totiž snaží minimalizovat ztrátovou funkci která reprezentuje vzdálenost dvou řetězců, kde první koresponduje se sekvencí genu a druhý s hledaným patogenem. Model se učí na neolabelovaných datech a poté sám hledá, podle určitých parametrů, v datech dvojice $< sekvence >, < target >$ a snaží se naučit data embeddovat do prostoru $\mathbf{R}^{768}$ tak aby se minimalizovala jejich odchylka. Architekturu modelu jsme přebrali od autorů originálního článku z minulého roku (2023 v době psaní tohoto dokumentu) a upravili některého jeho části tak aby korespondovali s naším problémem. Jednalo se především o zvětšení dimenze jednotlivých vrstev, jelikož je **RDE** určen primárně pro kratší sekvence. Zároveň jsme upravili způsob hledání $< targetu >$ tak aby více odpovídal našim datům, tedy aby byl například v průměru 1000 znaků dlouhý se směrodatnou odchylkou 200 znaků. Kritické části modelu, jako je tokenizer, jsme nechali původní jelikož je model schopný pracovat v podstatě s jakýmkoli problémem v rámci celogenomového sekvenování! Model jsme natrénovali na datech volně dostupných na internetu, které odpovídají požadavkům modelu. Konkrétně je model natrénován na lidském DNA. Ovšem díky chytré architektuře a kvalitnímu tokenizeru jde model hned použít i na DNA bakterii! Model jsme trénovali cca 4-5 hodin s *batch_size*= 32, s cosinovým *lr_schedulerem* a s akumulací gradientů nastavených na každé 4 kroky.



Obrázek 1. Trénovací loss function

5. Finální pipeline

Finální pipeline celého projektu poté stojí na vektorové databázi a vector searchi. Na vstupu bude model dostávat sekvence od sekvenátoru. Tyto sekvence okamžitě projdou inferencí modelu a výsledný embeddinge bude vložen do vektorové databáze. Jednou za x přidaných hodnot do databáze je spuštěn vector search, který pro každý patogen najde K nejbližších vektorů (my jsme měli nějvetší úspěch s volbou $K = 100$). Tyto nejbližší sousedé jsou poté testování pomocí klasických algoritmů jako například Pairwise2 a Smithova Watermanova vzdálenost aby se potvrdilo či vyvrátilo nalezení patogenu. Celý proces se tedy velmi urychlí jelikož se nemusí porovnávat všechny řetězce, ale pouze K nejbližších. Pipeline je ovšem náchylná na problémy při sekvenování jako velké překrytí řetězců či velmi rozdílné délky čtení. Věříme ovšem, že se tyto problémy dají předejít kvalitnějším trénováním modelu na více datech a zapojení lehkého předzpracování dat před vstupem do modelu.

Odkazy

[1] Y. Ji, Z. Zhou, H. Liu a R. V. Davuluri, „DNABERT: pre-trained Bidirectional Encoder Representations from Transformers model for DNA-language in genome“, *Bioinformatics*, roč. 37, č. 15, s. 2112–2120, ún. 2021, ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btab083. eprint: https://academic.oup.com/bioinformatics/article-pdf/37/15/2112/57195892/btab083.pdf. URL: https://doi.org/10.1093/bioinformatics/btab083.

[2] P. Holur, K. C. Enevoldsen, S. Rajesh et al., *Embed-Search-Align: DNA Sequence Alignment using Transformer Models*, 2024. arXiv: 2309.11087 [q-bio.GN]. URL: https://arxiv.org/abs/2309.11087.