Отчет по задаче “LSTM для классификации 16s рРНК”

Выполнили студенты 471 группы:

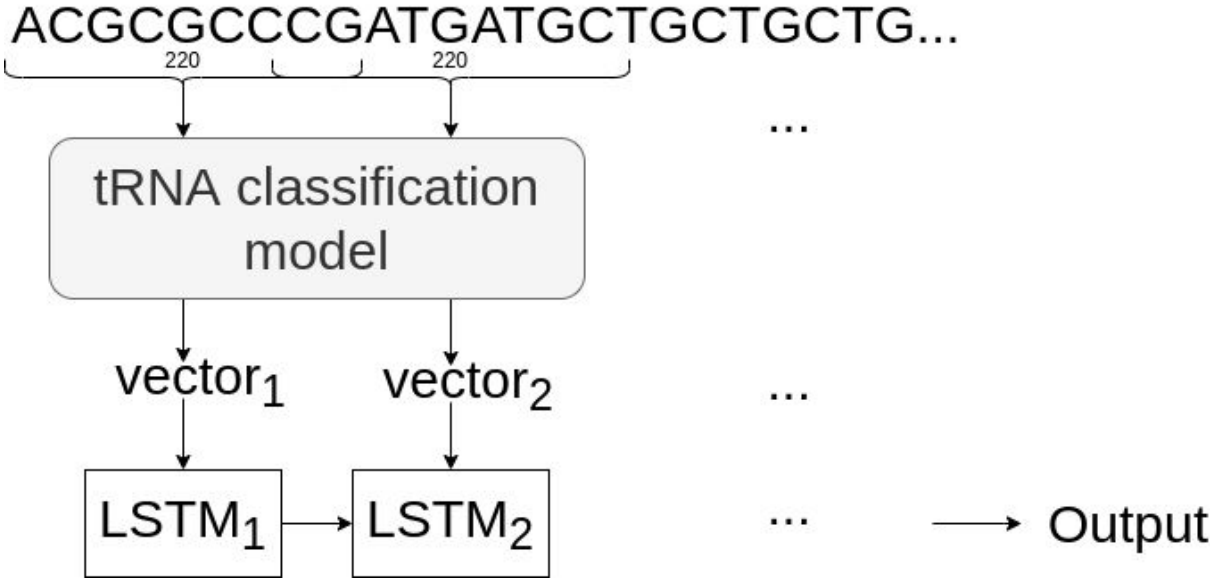
Кутленков Дмитрий - поиск и обработка данных, создание общего пайплайна
Келим Илья - эксперименты с архитектурой, подбор параметров

Введение

В рамках данной задачи предлагалось на основе существующей модели для классификации тРНК на 4 класса (Археи, Бактерии, Грибы и Растения) создать модель для предсказания этих классов по 16S РНК. Сложность задачи заключалась в том, что тРНК и 16S существенно различаются по длине - 16S РНК имеют длину от тысячи нуклеотидов, в то время как имеющаяся модель работает с последовательностями длины 220.

Идея решения

Идея решения заключается в разделении последовательности на пересекающиеся подпоследовательности длины 220 с некоторым шагом. К полученным последовательностям мы применяем исходную модель с помощью слоя TimeDistributed, а результаты обрабатываются с помощью LSTM сети.



Данные

Были использованы данные из базы SILVA версии 138.1. Для ускорения обучения модели и увеличения достоверности данных, был использован вариант NR99, который содержит только наиболее достоверные последовательности и за счет этого имеет меньший размер.
Так как последовательности имели разный размер, а на вход модели должны были подаваться одинаковые размерности, решено было обрезать концы последовательностей, которые превышали выбранный размер (около 1300 нуклеотидов, подобран эмпирически) и не включать в набор данных те, которые этого размера не достигают.
Производилось разбиение на train и test датасеты в соотношении 75%/25%. Разбиение на валидационный датасет было решено не делать ради увеличения числа данных в обучающем датасете. Это решение не повлияло на результаты экспериментов, так как при каждом изменении гиперпараметров во время обучения мы производили переразбиение датасета, а следовательно, не могли таким образом подобрать гиперпараметры под тестовый датасет (ссылка на код модели, с которой мы работали, указана в конце отчета).

Архитектура

Из изначальной модели были убраны последние слои (активация, полносвязный слой, нормализация по батчам). Таким образом, последним слоем остался полносвязный слой с 64 нейронами.
В лучшем решении были добавлены LSTM слои с полносвязным слоем между ними (LSTM(32) -> Dense(16) -> LSTM(16)).

Ход исследования

Были опробованы различные комбинации параметров и видов архитектуры. Лучшим оптимизатором оказался алгоритм ADAM с функцией потерь “Categorical Crossentropy”. Приведем некоторые полученные результаты.

Первым удовлетворительным результатом оказалось применение одного слоя LSTM с размерностью 16 на 6 эпохах. Здесь и далее в результатах сначала идет таблица, в которой отмечено количество объектов каждого царства отнесенных к каждому царству (здесь правильный результат - попадание на диагональ). Далее идут значения статистик precision и recall по каждому из царств, а в конце среднее значение этих статистик. В дальнейшем была также добавлена метрика F1-score.

```
  A   B   F   P
[704, 0, 0, 114] Archaea
[55, 1480, 265, 82] Bacteria
[15, 884, 1364, 90] Fungi
[102, 3, 1, 670] Plantae
prec(a) = 0.8036529680365296
prec(b) = 0.6252640473172792
prec(f) = 0.8368098159509203
prec(p) = 0.700836820083682
rec(a) = 0.8606356968215159
rec(b) = 0.7863974495217854
rec(f) = 0.5796855078623034
rec(p) = 0.8634020618556701
mean recision: 0.7416409128471028
mean recall: 0.7725301790153187
```

Однако при дальнейшем обучении на 12 эпохах результаты стали хуже.

```
[637, 3, 0, 178]
[24, 1643, 149, 66]
[15, 998, 1294, 46]
[230, 7, 0, 539]
prec(a) = 0.7030905077262694
prec(b) = 0.6197661259901924
prec(f) = 0.8967428967428968
prec(p) = 0.6501809408926418
rec(a) = 0.7787286063569682
rec(b) = 0.8730074388947928
rec(f) = 0.5499362515937102
rec(p) = 0.6945876288659794
mean recision: 0.7174451178380001
mean recall: 0.7240649814278626
```

Попытки изменить параметр batch_size так же привели к ухудшению качества модели. Ниже приведены результаты, полученные при выставлении batch_size равным 128:

```
[813, 2, 0, 12]
[95, 1274, 430, 75]
[56, 569, 1658, 95]
[356, 4, 1, 389]
prec(a) = 0.615909090909091
prec(b) = 0.6890210924824229
prec(f) = 0.7936811871708952
prec(p) = 0.681260945709282
rec(a) = 0.9830713422007256
rec(b) = 0.67982924226254
rec(f) = 0.6972245584524811
rec(p) = 0.5186666666666667
mean recision: 0.6949680790679228
mean recall: 0.7196979523956034
```

Был произведен эксперимент с добавлением двух слоев LSTM размерности 8, которые выдал значительно более точные результаты.

```
[835, 3, 0, 35]
[29, 1592, 177, 86]
[6, 950, 1271, 73]
[86, 3, 2, 681]
prec(a) = 0.8734309623430963
prec(b) = 0.6248037676609105
prec(f) = 0.876551724137931
prec(p) = 0.7782857142857142
rec(a) = 0.9564719358533792
rec(b) = 0.8450106157112527
rec(f) = 0.552608695652174
rec(p) = 0.8821243523316062
mean recision: 0.788268042106913
mean recall: 0.809053899887103
```

Добавление дополнительных слоев LSTM только ухудшило точность модели. Ниже приведены результаты, полученные при добавлении трех слоев LSTM с размерностями 16, 8 и 8.

```
[881, 2, 0, 5]
[12, 1702, 51, 69]
[13, 871, 1401, 56]
[726, 11, 0, 29]
prec(a) = 0.539828431372549
prec(b) = 0.6581593194122196
prec(f) = 0.9648760330578512
prec(p) = 0.18238993710691823
rec(a) = 0.9921171171171171
rec(b) = 0.9280261723009815
rec(f) = 0.5984621956428876
rec(p) = 0.037859007832898174
mean precision: 0.5863134302373846
mean recall: 0.6391161232234711
```

Были поставлены эксперименты с изменением размерностей добавленных слоев LSTM. Ниже приведены результаты, полученные после выставления размерности первого слоя на 16 и второго на 4.

```
[847, 2, 0, 39]
[61, 631, 1049, 93]
[37, 210, 2045, 49]
[416, 7, 7, 336]
prec(a) = 0.6223365172667157
prec(b) = 0.7423529411764705
prec(f) = 0.6594646888100613
prec(p) = 0.6499032882011605
rec(a) = 0.9538288288288288
rec(b) = 0.3440567066521265
rec(f) = 0.8735583084152072
rec(p) = 0.4386422976501306
mean recision: 0.668514358863602
mean recall: 0.6525215353865733
```

Однако при увеличении размерности первого слоя до 32, а второго до 16, были получены гораздо более точные результаты.

```
[849, 0, 0, 13]
[15, 1069, 701, 94]
[3, 175, 2072, 59]
[99, 1, 7, 672]
prec(a) = 0.8788819875776398
prec(b) = 0.8586345381526105
prec(f) = 0.7453237410071942
prec(p) = 0.801909307875895
rec(a) = 0.9849187935034803
rec(b) = 0.5689196381053752
rec(f) = 0.8973581637072325
rec(p) = 0.8626444159178434
mean precision: 0.8211873936533349
mean recall: 0.8284602528084829
```

Также были поставлены эксперименты с ограничением количества входных данных по каждому классу. При взятии 3000 элементов каждого класса и обучении на 30 эпохах были достигнуты результаты, приведенные ниже:

```
[753, 0, 1, 6]
[8, 681, 0, 70]
[5, 375, 309, 33]
[186, 0, 0, 573]
prec(a) = 0.7909663865546218
prec(b) = 0.6448863636363636
prec(f) = 0.9967741935483871
prec(p) = 0.8401759530791789
rec(a) = 0.9907894736842106
rec(b) = 0.8972332015810277
rec(f) = 0.4279778393351801
rec(p) = 0.7549407114624506
mean precision: 0.818200724204638
mean recall: 0.7677353065157172
```

При взятии 3400 элементов каждого класса результаты не улучшились.

```
[807, 0, 0, 4]
[17, 740, 2, 115]
[2, 432, 387, 40]
[339, 2, 0, 513]
prec(a) = 0.6927038626609442
prec(b) = 0.6303236797274276
prec(f) = 0.9948586118251928
prec(p) = 0.7633928571428571
rec(a) = 0.9950678175092479
rec(b) = 0.8466819221967964
rec(f) = 0.44947735191637633
rec(p) = 0.6007025761124122
mean precision: 0.7703197528391054
mean recall: 0.7229824169337081
mean f-score: 0.7459007935468578
```

Также были поставлены эксперименты по изменению шага, с которым происходила нарезка исходной последовательности. Результаты для шага 80 (12 эпох):

```
[981, 2, 0, 18]
[65, 1656, 70, 96]
[33, 803, 1318, 58]
[334, 11, 1, 1934]
prec(a) = 0.6942675159235668
prec(b) = 0.6699029126213593
prec(f) = 0.9488840892728582
prec(p) = 0.9183285849952516
rec(a) = 0.98001998001998
rec(b) = 0.8775834658187599
rec(f) = 0.5958408679927667
rec(p) = 0.8482456140350877
mean precision: 0.807845775703259
mean recall: 0.8254224819666486
mean f-score: 0.8165395514127762
```

Лучший результат был достигнут на такой конфигурации — два слоя LSTM с размерностями 32 и 16, шагом нарезки последовательности 160, оптимизатором ADAM с функцией потерь “Categorical Crossentropy” на 12 эпохах:

```
[1782, 0, 0, 17]
[21, 1648, 192, 88]
[13, 166, 2001, 35]
[578, 12, 2, 1618]
prec(a) = 0.7443609022556391
prec(b) = 0.9025191675794085
prec(f) = 0.911617312072893
prec(p) = 0.9203640500568828
rec(a) = 0.990550305725403
rec(b) = 0.8455618265777322
rec(f) = 0.9033860045146727
rec(p) = 0.7321266968325791
mean precision: 0.8697153579912059
mean recall: 0.8679062084125968
mean f-score: 0.8688098413909049
```

Выводы

В результате проведенного исследования архитектура с добавлением двух LSTM слоев размерности 32 и 16, использующая оптимизатор ADAM с функцией потерь “Categorical Cross-entropy” показала себя лучшей.
Число 160 оказалось оптимальным шагом нарезки исходной последовательности. Это

говорит о том, что слишком большое пересечение по последовательностям ухудшает результат.
Лучше всего модель предсказывает грибы (precision и recall около 0.9). Лучшей статистикой является recall у архей (0.99), что говорит о том, что практически все археи были найдены верно. Аномалией модели является большое количество растений, предсказанных как археи, что ухудшает precision первых и recall вторых.

Код модели

- Код лучшей полученной модели
- Веса лучшей модели
- Веса исходной модели