**embedding regularzed autoencoder** (то, что мы называем ванильный)

**Архитектура**

Идея этого автоэнкодера в том, что он сохраняет метрическое расстояние между векторами в исходном и скрытом пространствах. Я работала с евклидовой метрикой. Мотивация выбрать эту модель - возможное улучшение последующей классификации.

Нейросеть была написана на свободной библиотеке Pytorch. Сама сеть содержит по 3 полносвязных линейных слоя в энкодере и декодере, на вход подаем MS-профиль размерности 12001, размерности следующих слоев 6000 и 750, соответственно, на выходе (из енкодера) получаем вектор размерности 50, между всеми слоями нелинейная функция активации – RelU. Архитектура декодера симметрична описанной выше. В данном случае размерность скрытого пространства – 50. Функция потерь – сумма MSE и следующей величины:

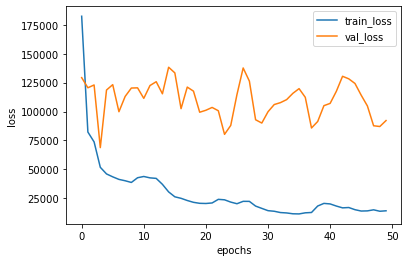
Где – все вектора скрытого и исходного пространств, соответственно, в текущем батче. Оптимайзер – Adam, параметры – стандартные (learning rate = 3\*1e-4 параметр для первого экспоненциального скользящего среднего = 0,9, для второго = 0,99). Переобучения модели не наблюдалось, так что dropout слоев нету, в некоторых случаях он ухудшал дальнейшую классификацию (брался по одному слою на кодер и декодер, частота = 0.25).

**Тренировка, синтез данных**

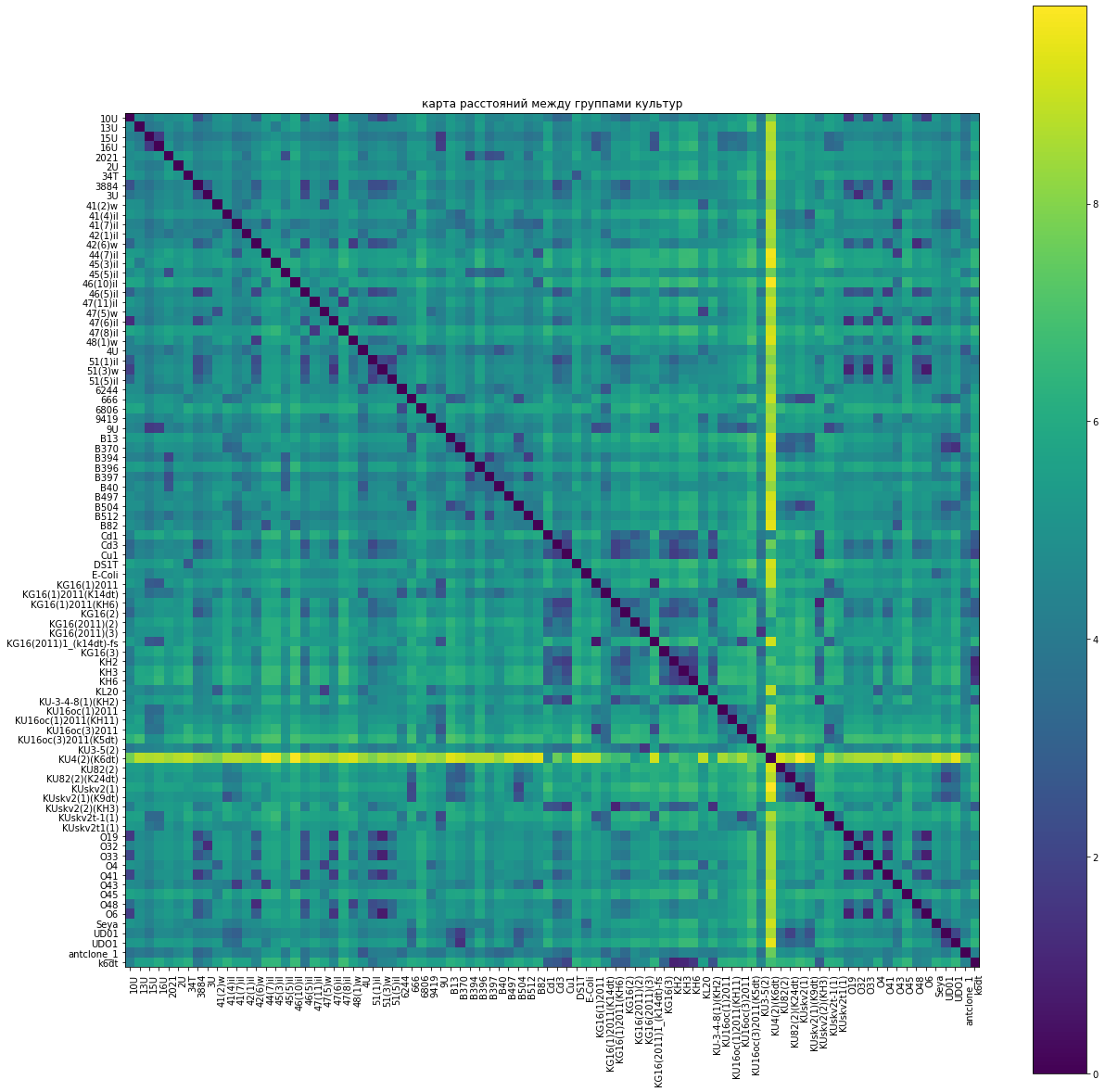
Для тренировки был предложен очень маленький датасэт – всего 151 профиль. Новые профили генерировались зашумлением имеющихся. Было рассмотрено 5 датасэтов: к каждой из половины случайно выбранных компонент прибавлялась случайная величина в интервале (-1e-2, 1e-2), так, что бы вес не стал отрицательным (далее как half noise 1e-2), так же для всего профиля(далее как full noise 1e-2), два аналогичных датасэта для интервала (-1e-1, 1e-1) и еще один где профиль суммировался со случайном вектором, умноженным на noise factor = 0,5, у которого компоненты имели нормальное распределение(далее как norm noise). В каждом датасэте было 3171 профилей. Эпох в обучении – 50, размер батча – 16, размер тренировочной выборки = 70% от исходного датасэта. На графиках функции потерь видно, что для full/half noise 1e-2 лосс на инференсе сходится к нулю, а для интервала (-1e-1, 1e-1) сходимости нет, но, что удивительно, классификация все равно будет достаточно хороша. Для профилей с norm noise нет как сходимости, так и хорошей классификации.

|  |  |  |
| --- | --- | --- |
|  | Half noise | Full noise |
| 1е-2 |  |  |
| 1e-1 |  |  |

Norm noise



Для самого лучшего лосса - 1е-2 full noise была построена heatmap для наглядного представления структуры скрытого пространства. Здесь представлены расстояния между средними, для соответствующей бактерии, векторами, а на диагонали – среднее внутри-бактериальное расстояние.



Видно, что наименьшие расстояния – межгрупповые и для разных бактерий расстояния до других отличаются, т.е. присутствуют выраженные в полосах представители. Следовательно, можно предположить, что классификация будет достаточно точной.

**Классификация**

Скрытые состояния были проклассифицированы по группам с помощью random forest (был использован готовый метод из свободной библиотеки sklearn). Лесу на тренировку отдавалось 70% от value сэта. Оба датасэта для (-1e-2, 1e-2) показали 100% точности. Для (-1e-1, 1e-1) с половиной измененных координат - 94%, а полностью измененные – 80%. Сгенерированный вектором с нормальным распределением – 24%.

**Denoising autoencoder**

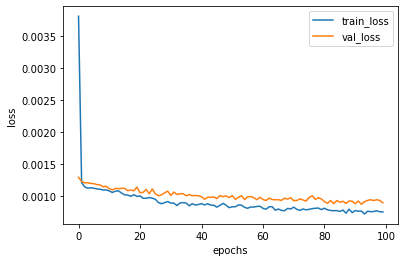
**Архитектура**

Идея этой модели, в том, что кодеру мы отдаем зашумленный вектор, но лосс считаем между реконструированным и чистым. Идея его использования в работе с шумом, которого в наших датасэтах очень много.

Структура во многом такая же как у embedding regularzed autoencoder, за исключением лосс функции, здесь это просто MSE.

**Тренировка, синтез данных**

Для тренировки были взяты исходные 151 профиль, перед тем как отдать его модели, весь батч зашумлялся, (прибавлялся вектор, умноженный на noise\_factor = 0,5, у компонент которого нормальное распределение). Затем лосс вычислялся между реконструированным вектором и чистым. Таких эпох было 100, (т. е. уникальных векторов было уже 151\*100), размер батча – 16.



**Классификация**

Скрытые состояния были проанализированы тем же RF. Лесу отдавались на тренировку 70% от полных датасэтов для моделей(т. е. 70% от 3171 профиля). На norm noise точность – 94%, для full noise 1e-1 – 96%, что лучше чем у embedding regularzed autoencoder.

**Variational autoencoder**

**Архитектура**

Идея этой модели, в том, что распределение скрытого состояния мы максимально приближаем к нормальному. Это дает возможность получить адекватную интерпретацию для любой точки скрытого пространства.

Архитектура во многом та же, за исключением лосса и выхода кодера. На выходе кодера – вектор размерности 100, первые 50 координат – мат. ожидание пространства скрытого состояния (), оставшиеся – логарифм дисперсии в квадрате (). Во время тренировки на вход декодеру отдаем случайный вектор из полученного распределения, он вычисляется по формуле , где – случайный вектор с нормальным распределением, на инференсе берем средний из них, т. е. просто . Лосс функция – сумма MSE и дивергенции Кульбика-Лейблера. Это критерий различия двух распределений, в нашем случае записывается следующей формулой:

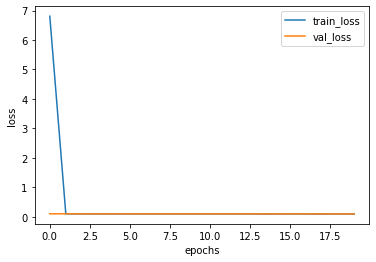
(комм: общий вид формулы(не для нормального распределения, а для любого) другой, если надо будет - вставим его, плюс математическое доказательство этого выражения я не могу понять, но знаю, статью на habr на которую можно сослаться(если на habr, вообще можно ссылаться))

**Тренировка**

Проблема этой модели была в том, что она моментально приводила скрытое пространство к нормальному распределению, т.е. все скрытые профили становились нулевыми (мат. ожидание нормального распределения), поэтому, чтобы хоть как-то проверить классификацию полученного пространства, на инференсе для декодера брались случайные вектора из распределения, плюс в обучении было всего 20 эпох, размер батча – такой же. Опыты я провела для трех сэтов, half noise 1e-2, full noise 1e-1 и norm noise.

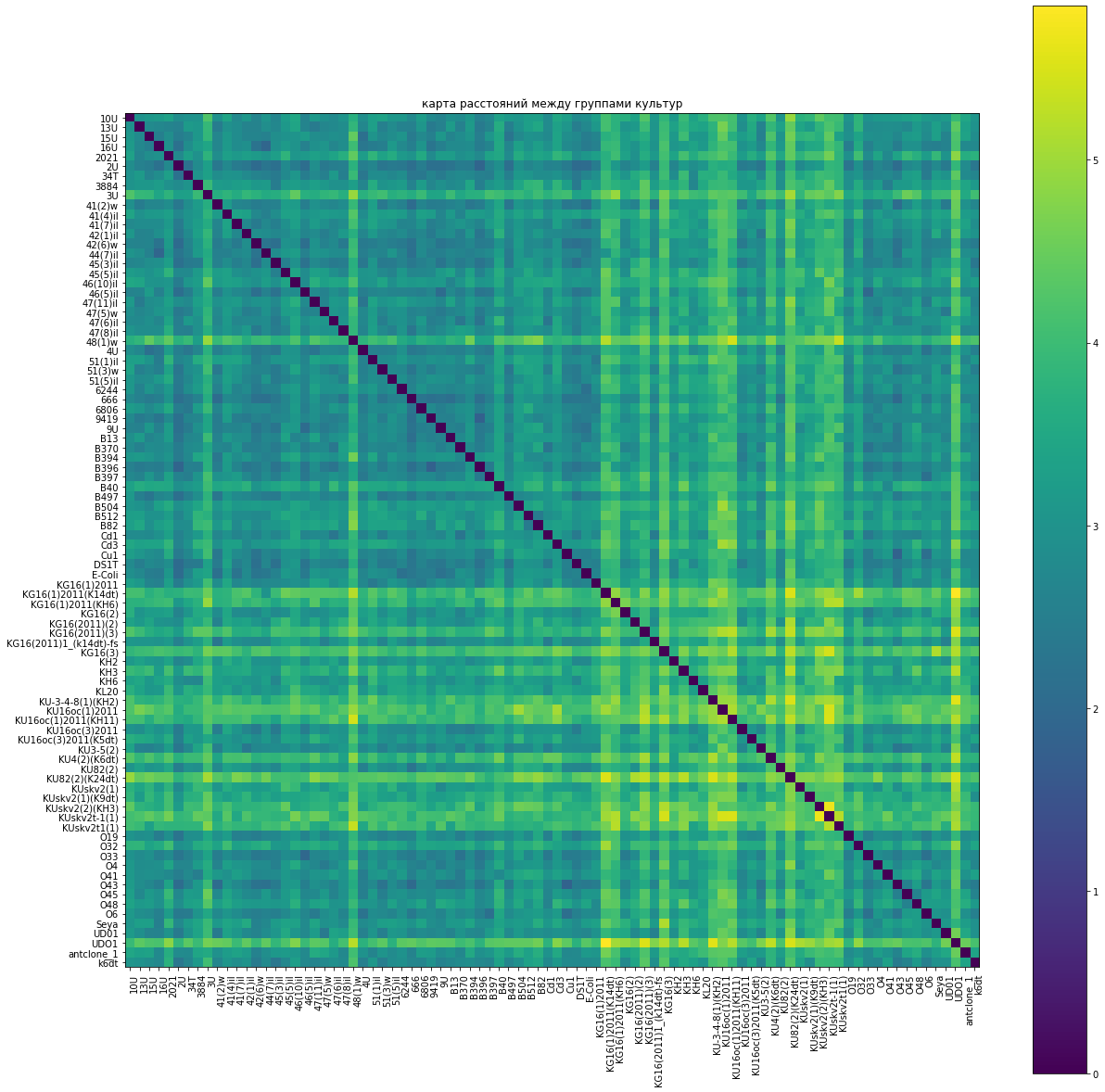
|  |  |
| --- | --- |
| Half noise 1e-2 | Full noise 1e-1 |
|  |  |

Norm noise



Не смотря, на эти приемы, сеть все равно слишком быстро учится. Скорее всего, это происходит из-за шума: модель по распределению распознает не 3171 сгенерированный профиль, а всего 151 чистых, а этого для качественного обучения нейросети недостаточно.

Для full noise 1e-1 была построена heatmap, снова видно, что самые маленькие величины – ср. внутри-бактериальные расстояния, но меж-бактериальные для всех бактерий примерно одинаковые, т.е. нет выраженных представителей, как в случае с embedding regularzed autoencoder. Таким образом, можно предположить, что эта модель не очень хороша (позже это подтвердиться) для классификации.



**Классификация**

Снова использовался RF. Для half noise 1e-2 точность – 19%, для full noise 1e-1 и norm rand точность – 17%. Конечно, результаты точно ухудшило то, что на инференсе брались случайные вектора из распределения, а не средние, но чисто структурно в VAE нет предпосылок к улучшению классификации. Но что бы это проверить экспериментально, скорее всего исходный датасэт должен быть намного больше.

**Заключение**

Эти модели обучены весьма искусственно и при реальном анализе точность классификации может упасть. Что бы улучшить модель нужно увеличить базу данных. Но пока для сервиса можно брать либо embedding regularzed либо denoising autoencoder. А когда на нем соберется база данных покрупнее, нужно будет повторить исследование для нее и этих моделей.