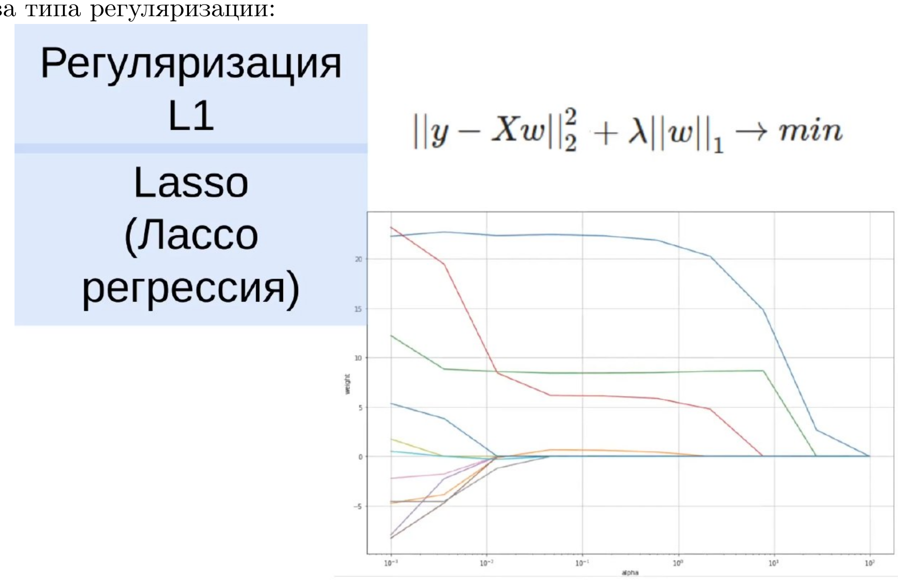
## Дополнительное задание 2.1

Регуляризация - это техника, используемая для предотвращения переобучения модели на обучающих данных, которая призвана бороться с гигантскими весами, делая модель не такой извилистой, а веса меньше по модулю. Существует два типа регуляризации:

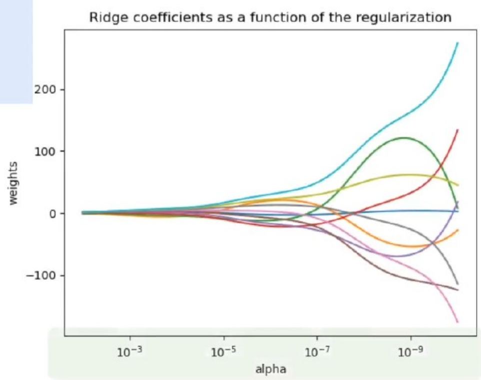



Основной плюс L1 регуляризации - это способность создавать разреженные модели, т.е. модели, где только небольшое количество признаков имеет ненулевые веса. Все это полезно в ситуациях, где признаков много, но только некоторые из них важны для прогнозирования.

**Регуляризация Тихонова**

**Ridge (гребневая регрессия)**

$$||y - Xw||_2^2 + \lambda||w||_2^2 \rightarrow min$$




В отличие от L1 регуляризации, L2 не склонна обнулять веса. Вместо этого она уменьшает веса постепенно, делая модель более "гладкой" и менее подверженной влиянию шума в данных. L2 регуляризация полезна в ситуациях, когда количество признаков в данных велико или когда они сильно коррелированы.

А также их объединение:

**Регуляризация L1 + L2**

**Elastic Net (Эластичная сеть)**

$$||y - Xw||_2^2 + \lambda_1||w||_1 + \lambda_2||w||_2^2 \rightarrow min$$

Elastic Net комбинирует преимущества L1 и L2 регуляризации. Требует настройки двух гиперпараметров (для L1 и L2 частей регуляризации), что может усложнить поиск оптимальных параметров.

Реализация:

```python
def SGD(X, y, a0, b0, c0, learning_rate, cnt_max_iterations, batch_size, learning_rate_scheduling, regularization=None, lambda_=0.01, alpha=0.5):
    coefs = np.array([a0, b0])
    c = c0

    for i in range(cnt_max_iterations):

        # создаем массивы данных размером batch_size
        # простыми словами: случайным образом сгенерировали батч точек, которые собираемся рассматривать
        indexes = np.random.randint(0, len(X) - 1, batch_size)

        # реальные данные (аргументы , реальные значения от этих аргументов)
        X_batch = X[indexes]
        y_batch = y[indexes]

        # считаем значение для текущих коэффициентов
        y_exec = np.dot(X_batch, coefs) + c

        # используем среднеквадратичную функцию потерь
        # (производные у квадратичной функции потерь) -> чтобы получить градиент
        # расчет градиентов вектора и свободного скаляра
        coefs_grad = -2 * np.dot(X_batch.T, y_batch - y_exec) / batch_size # 2* batch_size    (...) - batch_size
        #Np.sum так как по факту скалярное умножение (1....1) * error
        c_grad = -2 * np.sum(y_batch - y_exec) / batch_size

        if regularization == 'L1':
            coefs_grad += lambda_ * np.sign(coefs)
        elif regularization == 'L2':
            coefs_grad += lambda_ * coefs
        elif regularization == 'Elastic':
            coefs_grad += alpha * lambda_ * np.sign(coefs) + (1 - alpha) * lambda_ * coefs
        # идем вдоль антиградиента
        coefs -= learning_rate * coefs_grad
        c -= learning_rate * c_grad

        # меняем шаг
        if learning_rate_scheduling == "exponential":
            learning_rate *= 0.999
        elif learning_rate_scheduling == "stepwise":
            if i % 200 == 0:
                learning_rate *= 0.8

    return coefs, c
```

Процент использования L1 регуляризации относительно Elastic регуляризации

Параметр регуляризации L1 и L2

$$||y - Xw||_2^2 + \lambda||w||_1 \to min$$

$$||w||_1 = |w_1| + |w_2| + \ldots + |w_n|$$

$$||y - Xw||_2^2 + \lambda||w||_2^2 \to min$$

$$||a||^2 = a_1^2 + a_2^2 + \ldots + a_n^2$$

Проведённые испытания:

Входные данные: $y = ax_1^2 + bx_2^2 + c$, где $a = 3, b = 8, c = 50$; $\lambda_1 = 3, \lambda_2 = 0.000001, \alpha = 0.5$.

Результаты:

| | batch_size_start | learning_rate_scheduling_start | regularization_method | polinom_info | a_diff | b_diff | c_diff | count_function_operations | work_time (ms) | memory (bytes) | learning_and_test_diff: |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | None | None | [2, 2] | 0.084769 | 0.003486 | 0.269013 | 10000 | 170.331717 | 7882 | 0.138396 |
| 1 | 1 | None | L1 | [2, 2] | 0.035213 | 0.052348 | 0.295259 | 10000 | 255.471706 | 5220 | 0.116296 |
| 2 | 1 | None | L2 | [2, 2] | 0.061608 | 0.019534 | 0.319800 | 10000 | 198.287725 | 5580 | 0.198670 |
| 3 | 1 | None | Elastic | [2, 2] | 0.022436 | 0.082796 | 0.396855 | 10000 | 165.730953 | 4528 | 0.143541 |
| 4 | 1 | exponential | None | [2, 2] | 0.117057 | 0.106460 | 0.016323 | 10000 | 207.149029 | 4924 | 0.299938 |
| 5 | 1 | exponential | L1 | [2, 2] | 0.083181 | 0.087951 | 0.030333 | 10000 | 171.316862 | 4204 | 0.192409 |
| 6 | 1 | exponential | L2 | [2, 2] | 0.056861 | 0.036644 | 0.060381 | 10000 | 166.699886 | 4548 | 0.144413 |
| 7 | 1 | exponential | Elastic | [2, 2] | 0.033878 | 0.095972 | 0.109905 | 10000 | 166.776180 | 4540 | 0.208642 |
| 8 | 1 | stepwise | None | [2, 2] | 0.121018 | 0.168056 | 0.580119 | 10000 | 180.024147 | 4340 | 0.435188 |
| 9 | 1 | stepwise | L1 | [2, 2] | 0.082772 | 0.071895 | 0.453577 | 10000 | 178.383827 | 5092 | 0.416543 |
| 10 | 1 | stepwise | L2 | [2, 2] | 0.170472 | 0.109040 | 0.536714 | 10000 | 177.625656 | 4460 | 0.424892 |
| 11 | 1 | stepwise | Elastic | [2, 2] | 0.080689 | 0.116248 | 0.449423 | 10000 | 179.113626 | 4156 | 0.359718 |
| 12 | 300 | None | None | [2, 2] | 0.021701 | 0.019230 | 0.315878 | 3000000 | 170.766115 | 22352 | 0.091677 |
| 13 | 300 | None | L1 | [2, 2] | 0.023999 | 0.019340 | 0.320277 | 3000000 | 172.209024 | 22400 | 0.091341 |
| 14 | 300 | None | L2 | [2, 2] | 0.024321 | 0.020388 | 0.324457 | 3000000 | 164.633512 | 22680 | 0.091076 |
| 15 | 300 | None | Elastic | [2, 2] | 0.020919 | 0.020208 | 0.323713 | 3000000 | 166.602850 | 22384 | 0.090739 |
| 16 | 300 | exponential | None | [2, 2] | 0.070164 | 0.070095 | 0.066647 | 3000000 | 167.701006 | 22668 | 0.179334 |
| 17 | 300 | exponential | L1 | [2, 2] | 0.071161 | 0.067508 | 0.073350 | 3000000 | 169.302464 | 22736 | 0.181430 |
| 18 | 300 | exponential | L2 | [2, 2] | 0.069192 | 0.069274 | 0.064760 | 3000000 | 169.912577 | 22032 | 0.178538 |
| 19 | 300 | exponential | Elastic | [2, 2] | 0.066976 | 0.067483 | 0.064946 | 3000000 | 167.124987 | 22720 | 0.178348 |
| 20 | 300 | stepwise | None | [2, 2] | 0.123942 | 0.129797 | 0.550272 | 3000000 | 171.454430 | 22008 | 0.402312 |
| 21 | 300 | stepwise | L1 | [2, 2] | 0.120696 | 0.126980 | 0.563096 | 3000000 | 179.147243 | 23168 | 0.410755 |
| 22 | 300 | stepwise | L2 | [2, 2] | 0.129939 | 0.129841 | 0.559076 | 3000000 | 189.055443 | 23192 | 0.408185 |
| 23 | 300 | stepwise | Elastic | [2, 2] | 0.130305 | 0.130923 | 0.567862 | 3000000 | 180.036306 | 22604 | 0.413327 |
| 24 | 500 | None | None | [2, 2] | 0.024901 | 0.019674 | 0.320196 | 5000000 | 176.180601 | 35152 | 0.091493 |
| 25 | 500 | None | L1 | [2, 2] | 0.024659 | 0.020817 | 0.321915 | 5000000 | 177.058935 | 34248 | 0.091402 |
| 26 | 500 | None | L2 | [2, 2] | 0.024300 | 0.018739 | 0.320479 | 5000000 | 175.646305 | 35152 | 0.091333 |
| 27 | 500 | None | Elastic | [2, 2] | 0.025111 | 0.019849 | 0.324953 | 5000000 | 175.292253 | 34276 | 0.091121 |
| 28 | 500 | exponential | None | [2, 2] | 0.069147 | 0.068563 | 0.064917 | 5000000 | 170.299530 | 34616 | 0.178538 |
| 29 | 500 | exponential | L1 | [2, 2] | 0.068596 | 0.066499 | 0.061359 | 5000000 | 177.066326 | 34268 | 0.177319 |
| 30 | 500 | exponential | L2 | [2, 2] | 0.067733 | 0.068517 | 0.061272 | 5000000 | 167.766809 | 34616 | 0.177240 |
| 31 | 500 | exponential | Elastic | [2, 2] | 0.068557 | 0.067000 | 0.067955 | 5000000 | 170.584917 | 35120 | 0.179436 |
| 32 | 500 | stepwise | None | [2, 2] | 0.130381 | 0.131009 | 0.564243 | 5000000 | 179.528713 | 34160 | 0.411269 |
| 33 | 500 | stepwise | L1 | [2, 2] | 0.132043 | 0.130343 | 0.569388 | 5000000 | 173.947573 | 34272 | 0.414536 |
| 34 | 500 | stepwise | L2 | [2, 2] | 0.128839 | 0.128271 | 0.561048 | 5000000 | 181.626081 | 34504 | 0.409149 |
| 35 | 500 | stepwise | Elastic | [2, 2] | 0.128853 | 0.130082 | 0.562854 | 5000000 | 178.986549 | 34936 | 0.410174 |
| 36 | 1000 | None | None | [2, 2] | 0.025052 | 0.020373 | 0.321037 | 10000000 | 184.898376 | 64244 | 0.091501 |
| 37 | 1000 | None | L1 | [2, 2] | 0.021035 | 0.019207 | 0.319337 | 10000000 | 179.212809 | 64572 | 0.091236 |
| 38 | 1000 | None | L2 | [2, 2] | 0.026639 | 0.020553 | 0.322647 | 10000000 | 189.387083 | 65096 | 0.091724 |
| 39 | 1000 | None | Elastic | [2, 2] | 0.024282 | 0.018301 | 0.321073 | 10000000 | 177.776575 | 64264 | 0.091257 |
| 40 | 1000 | exponential | None | [2, 2] | 0.069869 | 0.068050 | 0.066622 | 10000000 | 180.360079 | 64800 | 0.179143 |
| 41 | 1000 | exponential | L1 | [2, 2] | 0.069218 | 0.068244 | 0.065431 | 10000000 | 182.488203 | 64300 | 0.178694 |
| 42 | 1000 | exponential | L2 | [2, 2] | 0.067677 | 0.067304 | 0.067595 | 10000000 | 182.533264 | 64456 | 0.179258 |
| 43 | 1000 | exponential | Elastic | [2, 2] | 0.070222 | 0.068179 | 0.067814 | 10000000 | 179.391146 | 64300 | 0.179567 |
| 44 | 1000 | stepwise | None | [2, 2] | 0.128243 | 0.128798 | 0.566217 | 10000000 | 185.571671 | 64160 | 0.412127 |
| 45 | 1000 | stepwise | L1 | [2, 2] | 0.126579 | 0.130085 | 0.568215 | 10000000 | 188.601494 | 65120 | 0.413136 |
| 46 | 1000 | stepwise | L2 | [2, 2] | 0.127650 | 0.129178 | 0.562988 | 10000000 | 196.115017 | 63976 | 0.410125 |
| 47 | 1000 | stepwise | Elastic | [2, 2] | 0.129864 | 0.129946 | 0.566607 | 10000000 | 183.279276 | 64880 | 0.412514 |