

1. Что такое L2 регуляризация, в чем отличие от L1?

L2 регуляризация предотвращает переобучение модели путём запрета на несоразмерно большие весовые коэффициенты. При L2 регуляризации дополнительный член является квадратичной функцией (при L1 регуляризации – модулем).

$$L(x, y) \equiv \sum_{i=1}^n (y_i - h_{\theta}(x_i))^2 + \lambda \sum_{i=1}^n \theta_i^2,$$

где  $\lambda$  является гиперпараметром, который контролирует интенсивность штрафа.

Когда  $\lambda \rightarrow 0$ , результаты аналогичны линейной регрессии.

Когда  $\lambda \rightarrow \infty$ , все особенности уменьшены до 0.

Различия:

- L1 штрафует сумму абсолютных значений весов, а L2 штрафует
- сумму квадратных весов.
- L1 имеет разреженное решение, L2 имеет не разреженное
- решение.
- L1 имеет несколько решений, у L2 есть одно решение.
- L1 имеет встроенный выбор функций, L2 не имеет выбора
- функций.
- L1 устойчив к выбросам, L2 не устойчив к выбросам.

- L1 генерирует модели, которые просты и понятны, но не могут выучить сложные шаблоны, а L2 дает лучший прогноз, когда выходная переменная является функцией всех входных функций.

2. Какие методы надо определить в пользовательском callback, чтобы он срабатывал в начале обучения и конце каждого пакета?

Необходимо определить 3 метода:

- on\_epoch\_end
- on\_train\_batch\_end
- on\_test\_batch\_end

3. Как в Keras callback получается доступ к истории ошибок?

Через использование словаря logs, который хранит ошибки и другие метрики. Взято из документации tensorflow.keras.

```
class LossAndErrorPrintingCallback(tf.keras.callbacks.Callback):  
  
    def on_train_batch_end(self, batch, logs=None):  
        print('For batch {}, loss is {:.2f}'.format(batch, logs['loss']))  
  
    def on_test_batch_end(self, batch, logs=None):  
        print('For batch {}, loss is {:.2f}'.format(batch, logs['loss']))  
  
    def on_epoch_end(self, epoch, logs=None):  
        print('The average loss for epoch {} is {:.2f}'.format(epoch,  
logs['loss']))
```