Практическое задание №8

Вариант №6

Задание: Необходимо реализовать собственной CallBack, и провести обучение вашей модели из практического занятия №6 с написанным CallBack'ом. То, какой CallBack необходимо реализовать определяется вариантом.

Условие: Построение и сохранение таблицы со следующими данными: номер эпохи, номер наблюдения с наименьшей точностью классификации на заданной эпохе, к какому классу принадлежит наблюдение, точность классификации, значение ошибки. Каждая строчка должна рассчитываться с заданным пользователем интервалом начиная с 0 эпохи, а также на самой последней.

Выполнение: Был реализован класс CallBack `MinAccJournal`, использующий методы `def on_train_begin(self, logs={})`, внутри которого происходит инициализация массивов-журналов точности и потерь в начале обучения, метод `def on_batch_end(self, batch, logs={})` внутри которого происходит добавление результатов одного наблюдения (в данном случае размер батча = 1 наблюдению) в журналы точности и ошибок и метод `def on_epoch_end(self, epoch, logs=None)` внутри которого происходит нахождения необходимых данных для внесения в таблицу, а также обнуление массивов точности и ошибок.

```
class MinAccJournal(keras.callbacks.Callback):
    def on_train_begin(self, logs={}):
        self.losses = []
        self.accuracy = []

    def on_batch_end(self, batch, logs={}):
        self.losses.append(logs.get('loss'))
        self.accuracy.append(logs.get('accuracy'))

    def on_epoch_end(self, epoch, logs=None):
        min_acc = min(self.accuracy)
        min_acc_ind = self.accuracy.index(min_acc)
```

```
                min_loss = self.losses[min_acc_ind]
                obs_class = labels[min_acc_ind]
                self.losses = []
                self.accuracy = []
                if epoch % interval == 0 or epoch == num_epochs - 1:
                    table.append([epoch,  min_acc_ind,  obs_class[0],  min_acc,
min_loss])
```