

РЕГУЛИРОВАНИЕ ПЕРЕОБУЧЕНИЯ НЕЙРОННОЙ СЕТИ ПРИ ПОМОЩИ БИБЛИОТЕКИ KERAS

Keras – открытая библиотека, написанная на языке Python и обеспечивающая взаимодействие с искусственными нейронными сетями. *Keras* позволяет интуитивно и просто создавать прототипы, исследовать и развертывать модели глубокого обучения, тем самым способствует уменьшению порога вхождения в разработку нейронных сетей, поэтому эта библиотека занимает лидирующие позиции в среде студентов и специалистов по глубокому обучению [1].

Влияние различных инструментов на результаты переобучения нейронной сети показано на модели классификации рукописных цифр. Для нейронной сети выбрана структура сети, содержащая один внутренний слой размером в 300 нейронов и функцией активации *Relu* и выходной слой с 10-ю нейронами и функцией активации *Softmax* (рис. 1). Сеть реализована с помощью объекта модели *keras Sequential()*, который создаёт последовательную модель с пошаговым добавлением слоёв в неё. *Dense*-слой отвечает за соединение нейронов из предыдущего и следующего слоя, основные гиперпараметры *Dense*-слоя это количество нейронов и функция активации. Функция активации выбирается на основе рекомендаций в зависимости от задачи обучения.

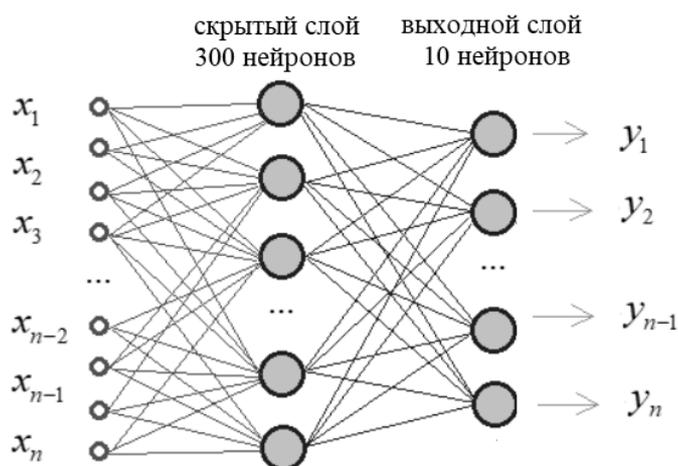
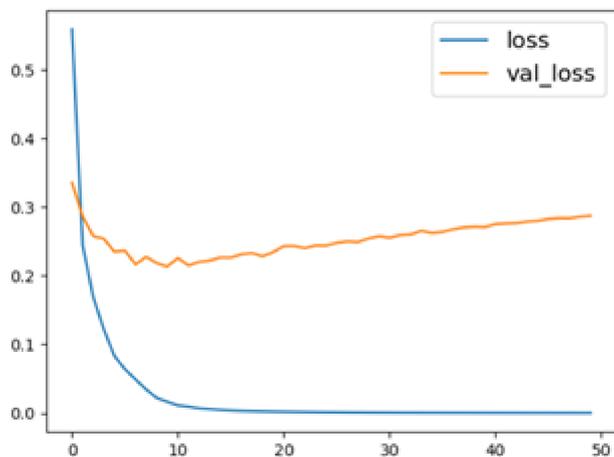


Рисунок 1 – Структура нейронной сети для распознавания рукописных цифр

Компиляция модели выполнена со следующими параметрами: *optimizer='adam'*, *loss='categorical_crossentropy'*, *metrics=['accuracy']*).

График потерь указывает на явное переобучение модели, так как

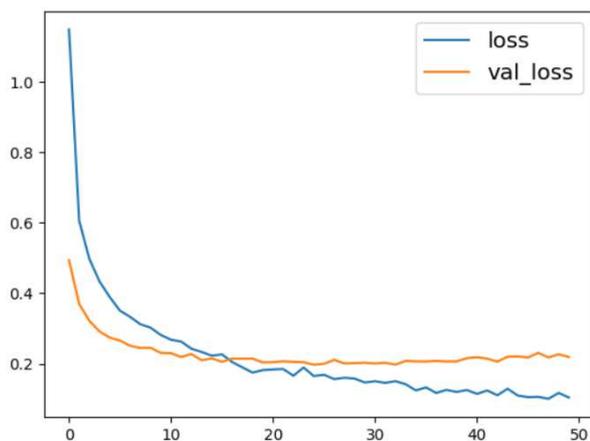
имеется расхождение в точности выходных значений между обучающей и тестовой выборками (рис. 2). Выходом может стать уменьшение количества нейронов во внутреннем слое, однако в ходе эксперимента уменьшение количества до 100 нейронов не повлияло на переобучение модели.



loss	7.4662e-05
accuracy	1.000
val_loss	0.2895
val_accuracy	0.9434

Рисунок 2 – График потерь и параметры модели

Для уменьшения переобучения использован метод *Dropout*, который позволяет исключать случайные нейроны (находящихся как в скрытых, так и видимых слоях) на разных итерациях (эпохах) во время обучения нейронной сети. Метод реализуется добавлением в структуру модели *Dropout*-слоя с параметром, указывающим на процент исключенных нейронов. Использование слоя *Dropout(0.8)* значительно уменьшило переобучение модели и улучшило точность на проверочных данных (рис. 3).

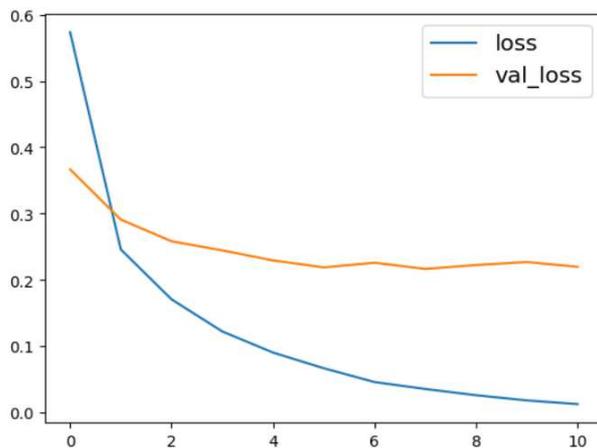


loss	0.1021
accuracy	0.9640
val_loss	0.2363
val_accuracy	0.9438

Рисунок 3 – График потерь с Dropout-слоем и параметры модели

Переобучения можно избежать, вовремя остановив обучение модели, для этого необходимо контролировать процесс обучения. Функции обратных вызовов позволяют оценивать состояние нейрон-

ной сети и своевременно выполнять управляющие воздействия. Обратный вызов имеет доступ к информации о состоянии модели, может вызываться на различных этапах обучения и способен производить следующие действия: прерывать обучение, если функция потерь не уменьшается, сохранять модель, загружать разные наборы весов или как-то иначе изменять состояние модели [2]. Использование обратного вызова позволило остановить обучение уже на 11 эпохе из заданных 50, однако точность модели получилась меньше, чем в случае использования *Dropout*-слоя (рис. 4).



loss	0.0126
accuracy	0.9994
val_loss	0.2197
val_accuracy	0.9390

Рисунок 4 – Список наиболее похожих слов с рассчитанным коэффициентом геометрического сходства

При комбинации двух рассмотренных методов точность на тестовых данных достигла 0.9398 на 23 эпохе.

Рассмотрено влияние двух распространённых способов регуляризации модели. На переобучение модели влияет также множество других параметров: выбор функции потерь и оптимизатора, структура нейронной сети, начальная инициализация весов и др. Обучение нейронных сетей можно назвать сложным эвристическим процессом, зависящим от множества факторов, и использование библиотеки *Keras* значительно упрощает этот процесс.

ЛИТЕРАТУРА

1. Keras documentation, <https://keras.io/> (дата обращения: 02.02.2024).
2. Шолле Франсуа, Глубокое обучение на Python. 2-е межд. издание. – СПб.: Питер, 2023. – 576 с.