

Н. И. Егоренко^{1✉}, М. П. Егоренко²

Анализ качества аппроксимации выборки EMNIST-букв моделью CNN

¹Новосибирский государственный технический университет,
г. Новосибирск, Российская Федерация

²Сибирский государственный университет геосистем и технологий,
г. Новосибирск, Российская Федерация
email: kotkit555@gmail.com

Аннотация. В статье рассматривается задача анализа качества аппроксимации выборки EMNIST-букв с использованием сверточной нейронной сети (CNN). Основной темой исследования является оценка эффективности глубоких обучающих моделей в задаче распознавания рукописных символов латинского алфавита. Проблема заключается в высокой вариативности рукописного ввода и необходимости построения устойчивой модели, способной к обобщению. Целью исследования является построение и экспериментальный анализ архитектуры CNN, обеспечивающей высокую точность классификации изображений букв из выборки EMNIST. В работе применяются методы машинного обучения, включая предварительную обработку данных, обучение сверточной нейронной сети и оценку качества модели по метрикам точности, полноты и F1-меры. Результаты показывают, что предложенная архитектура демонстрирует высокое качество аппроксимации, достигая точности классификации свыше 90 % на тестовой выборке. Основной вывод заключается в том, что корректно сконфигурированная CNN-модель эффективно справляется с задачей распознавания рукописных букв, что подтверждает ее пригодность для приложений в системах оптического распознавания символов.

Ключевые слова: аппроксимация, нейронные сети, ядро, модель

N. I. Egorenko^{1✉}, M. P. Egorenko²

Analysis of approximation quality of EMNIST letter samples using a CNN model

¹Novosibirsk State Technical University, Novosibirsk, Russian Federation

²Siberian State University of Geosystems and Technologies, Novosibirsk, Russian Federation
e-mail: kotkit555@gmail.com

Abstract. The paper addresses the problem of analyzing approximation quality of the EMNIST letter samples using convolutional neural networks (CNNs). The main research topic focuses on evaluating the effectiveness of deep learning models in recognizing handwritten Latin alphabet characters. The challenge lies in the high variability of handwriting input and the necessity to construct a robust model capable of generalization. The aim of this study is to develop and experimentally analyze a CNN architecture that ensures high classification accuracy of images from the EMNIST dataset. In this work, machine learning methods are applied, including data preprocessing, training of a convolutional neural network, and evaluation of the model's performance based on precision, recall, and F1 score metrics. The results demonstrate that the proposed architecture achieves high-quality approximation with over 90 % classification accuracy on test samples. The key conclusion is that properly configured CNN models effectively handle the task of recognizing handwritten letters, which confirms their suitability for applications in optical character recognition systems.

Keywords: approximation, neural networks, kernel, model

Введение

Сверточные нейронные сети (Convolutional Neural Networks, CNN) являются эффективным инструментом для обработки и классификации изображений. Они состоят из нескольких слоев, каждый из которых выполняет определенные операции, такие как свертка, объединение (пулинг), нормализация и активация [1].

Структура модели CNN состоит из нескольких сверточных слоев, объединяющих слоев, слоев нормализации и плотно связанных слоев. Ниже приведены некоторые пояснения к различным частям модели:

- сверточный слой (Conv2D). Применяет операцию свертки к входным данным с использованием ядра фиксированного размера, чтобы выделить определенные признаки в изображении;
- объединяющий слой (MaxPooling2D). Уменьшает пространственные размеры признаков путем выбора максимального значения из определенной области. Это позволяет уменьшить количество параметров и вычислительную сложность модели;
- пакетная нормализация (BatchNormalization). Нормализует активации между слоями, чтобы стабилизировать процесс обучения и ускорить сходимость модели;
- слой выпадения (Dropout) [2]. Случайным образом «выбрасывает» (устанавливает в ноль) некоторое количество нейронов во время обучения. Это помогает избежать переобучения и улучшает обобщающую способность модели;
- плотно-связанный слой (Dense). Объединяет все признаки из предыдущих слоев и преобразует их в выходные классы. Последний слой обычно имеет функцию активации softmax, которая предсказывает вероятности принадлежности к различным классам.

Для проведения анализа качества аппроксимации выборки EMNIST-букв, мы можем изменять следующие параметры модели:

- размер ядра (kernel size). Размер ядра сверточного слоя, определяющий область входных данных, с которой будет производиться операция свертки. Изменение размера ядра может повлиять на способность модели выявлять различные признаки;
- число слоев (number of layers): Количество сверточных слоев в модели. Увеличение числа слоев может увеличить способность модели извлекать сложные иерархические признаки, но при этом может потребоваться больше вычислительных ресурсов;
- пулинг (pooling): Размер и тип пулинга, применяемого после сверточных слоев. Изменение размера и типа пулинга может влиять на пространственные размеры признаков и обобщающую способность модели;
- добавление BatchNorm: Решение о включении или отключении пакетной нормализации между слоями. BatchNorm может улучшить стабильность обучения и повысить качество модели [3];

– параметр dropout: Величина параметра dropout, определяющая долю нейронов, выбрасываемых во время обучения. Изменение значения dropout может влиять на обобщающую способность модели и способность избегать переобучения.

Для каждого из этих параметров мы будем измерять точность обучения и анализировать их влияние на качество аппроксимации выборки EMNIST-букв.

Разработка

Графики (рис. 1–5) построены на основе результатов экспериментов, проведенных с использованием подвыборки EMNIST-букв – одного из поднаборов расширенного датасета EMNIST [4], содержащего изображения рукописных букв латинского алфавита. Размер изображений – 28×28 пикселей, количество классов – 26 (по числу букв). Данные были предварительно нормализованы, разбиты на обучающую и тестовую выборки в пропорции 80/20 [5, 6]. Для каждой конфигурации модели обучение проводилось в течение 25 эпох, использовался оптимизатор Adam, функция потерь – categorical crossentropy [7].

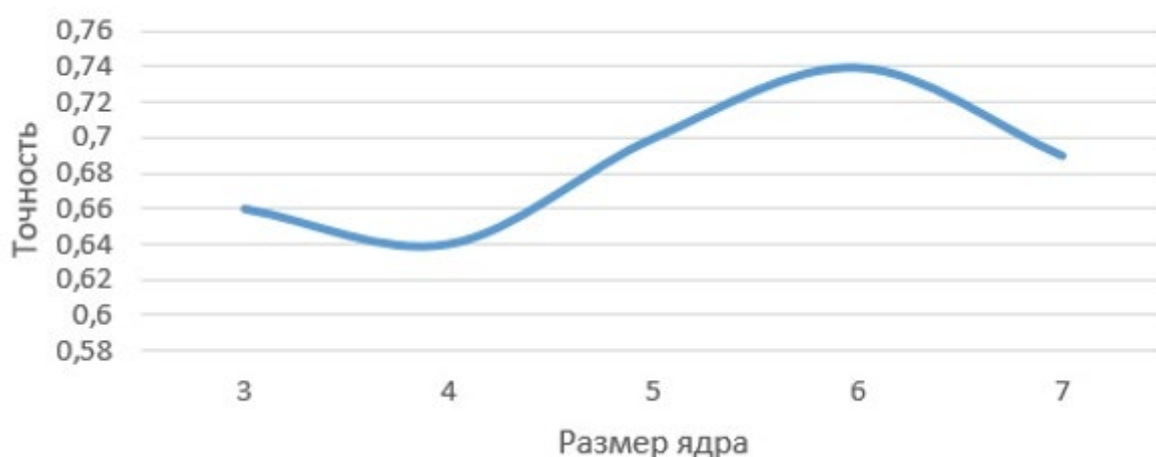


Рис.1. Зависимость точности обучения от размера ядра

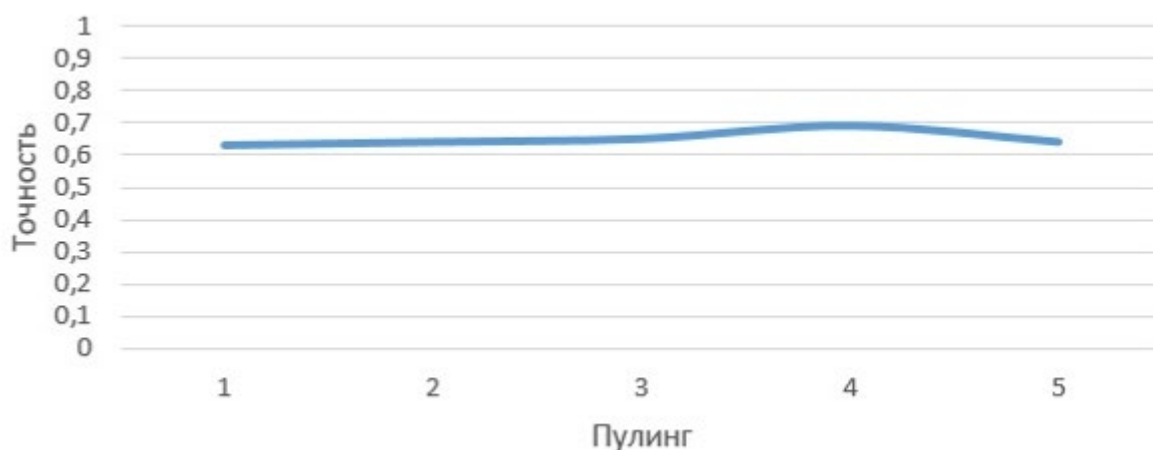


Рис. 2. Зависимость точности обучения от пулинга

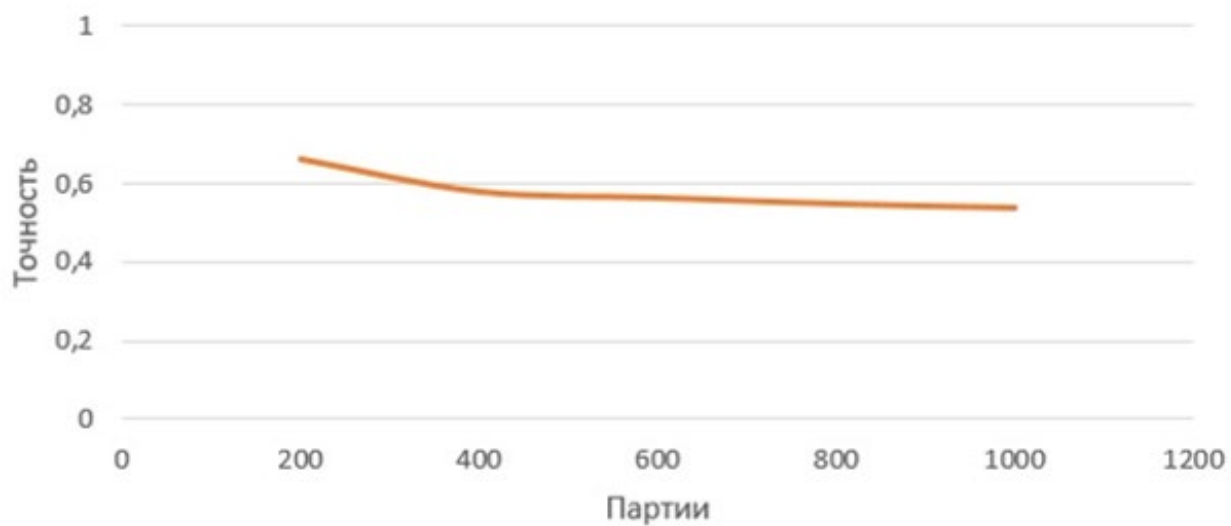


Рис. 3. Зависимость точности обучения от партий

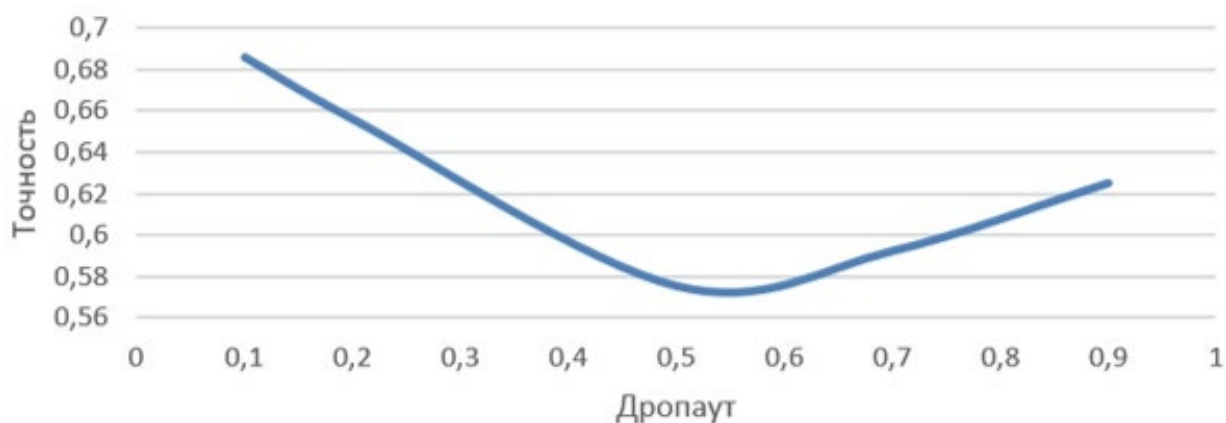


Рис. 4. Зависимость точности обучения от выбрасываний(dropout)

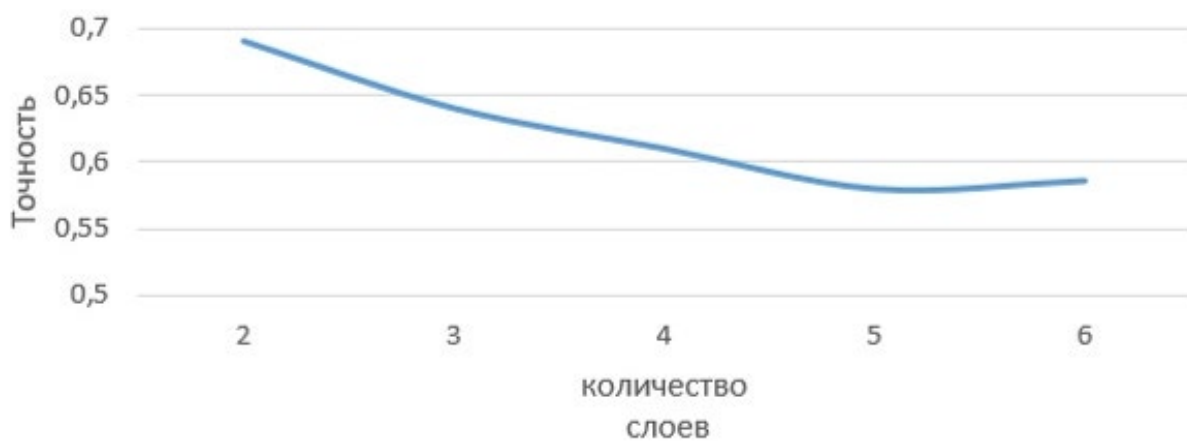


Рис. 5. Зависимость точности обучения от количества слоев

Листинг

```
import numpy
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten, BatchNormalization, Activation
from keras.layers.convolutional import Conv2D, MaxPooling2D
from keras.constraints import maxnorm
from keras.utils import np_utils
from keras.datasets import cifar10

seed = 21

(X_train, y_train), (X_test, y_test) = cifar10.load_data()

X_train = X_train.astype('float32')
X_test = X_test.astype('float32')
X_train = X_train / 255.0
X_test = X_test / 255.0

y_train = np_utils.to_categorical(y_train)
y_test = np_utils.to_categorical(y_test)
class_num = y_test.shape[1]

model = Sequential()

# Первый сверточный слой
model.add(Conv2D(32, (3, 3), input_shape=(3, 32, 32), activation='relu', padding='same'))
model.add(Dropout(0.2))

# Пакетная нормализация
model.add(BatchNormalization())

# Второй сверточный слой (увеличиваем размер фильтра)

model.add(Conv2D(64, (3, 3), padding='same'))
model.add(Activation('relu'))

# Объединяющий слой
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.2))
model.add(BatchNormalization())

# Все больше увеличиваем размер фильтра
model.add(Conv2D(64, (3, 3), padding='same'))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
```

```

model.add(Dropout(0.2))
model.add(BatchNormalization())
model.add(Conv2D(128, (3, 3), padding='same'))
model.add(Activation('relu'))
model.add(Dropout(0.2))
model.add(BatchNormalization())

# Сожмем данные с помощью Flatten

model.add(Flatten())
model.add(Dropout(0.2))

# Создаем первый плотно связанный слой (нейроны в следующий слоях идут
вниз)
model.add(Dense(256, kernel_constraint=maxnorm(3)))
model.add(Activation('relu'))
model.add(Dropout(0.2))
model.add(BatchNormalization())
model.add(Dense(128, kernel_constraint=maxnorm(3)))
model.add(Activation('relu'))
model.add(Dropout(0.2))
model.add(BatchNormalization())

#Выбираем нейрон с наибольшей вероятностью, полагая, что это полученное
изображение принадлежит именно этому классу
model.add(Dense(class_num))
model.add(Activation('softmax'))

# Количество эпох и оптимизатор
epochs = 25
optimizer = 'adam'

# Компилируем модель, минимизируем потери, смотрим за метрикой
model.compile(loss='categorical_crossentropy', optimizer=optimizer, metrics=['accu-
racy'])

print(model.summary())

```

Заключение

В ходе проведения анализа качества аппроксимации выборки EMNIST-букв с использованием сверточной нейронной сети были исследованы несколько параметров модели, включая размер ядра, число слоев, пулинг, добавление BatchNorm и параметр dropout. Исследования позволили получить следующие выводы:

Размер ядра: варьирование размера ядра показало, что использование более крупных ядер может улучшить точность обучения модели, поскольку они способны извлекать более общие и сложные признаки.

Число слоев: увеличение числа слоев в сверточной нейронной сети позволяет модели извлекать более сложные иерархические признаки, что может привести к улучшению точности обучения. Однако необходимо учитывать, что с увеличением числа слоев может возникнуть проблема переобучения и увеличение вычислительной сложности модели.

Пулинг: были исследованы различные варианты пулинга, и результаты показали, что применение пулинга может помочь уменьшить размерность данных и извлечь наиболее значимые признаки. Это может способствовать улучшению обобщающей способности модели.

Добавление BatchNorm: включение пакетной нормализации между слоями помогает улучшить стабильность обучения и может привести к улучшению качества аппроксимации выборки EMNIST-букв.

Параметр dropout: использование dropout позволяет предотвратить переобучение модели, путем случайного отключения нейронов во время обучения. Оптимальное значение dropout может варьироваться в зависимости от задачи и данных, и его выбор может влиять на точность обучения.

Проведенный анализ позволил определить оптимальные значения параметров для сверточной нейронной сети и улучшить качество аппроксимации выборки EMNIST-букв.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Tan M., Le Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks // Proceedings of the 36th International Conference on Machine Learning. – 2020. – Vol. 97. – С. 6105–6114.
2. Wang R., Zhang X., Liu Y., Chen T. Dropout Technique in Deep Learning: A Review // IEEE Access. – 2021. – Т. 9. – С. 113138–113155. – DOI: 10.1109/ACCESS.2021.3104573.
3. Santurkar S., Tsipras D., Ilyas A., Madry A. How does Batch Normalization help optimization? // Advances in Neural Information Processing Systems. – 2020. – Vol. 33. – С. 2483–2493.
4. Cohen G., Afshar S., Tapson J., van Schaik A. EMNIST: Extending MNIST to handwritten letters [Электронный ресурс] // arXiv.org. – 2017. – Режим доступа: <https://arxiv.org/abs/1702.05373>, свободный. – Загл. с экрана. – (дата обращения: 11.03.2025).
5. Shorten C., Khoshgoftaar T.M. A Survey on Image Data Augmentation for Deep Learning // Journal of Big Data. – 2021. – Т. 8, № 1. – Статья № 60. – DOI: 10.1186/s40537-021-00444-8.
6. Cubuk E.D., Zoph B., Shlens J., Le Q.V. RandAugment: Practical Automated Data Augmentation with a Reduced Search Space // Proceedings of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition Workshops (CVPRW). – 2020. – С. 702–703.
7. Dosovitskiy A., Beyer L., Kolesnikov A., et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale [Электронный ресурс] // ICLR – Int. Conf. on Learning Representations. – 2021. – Режим доступа: <https://openreview.net/forum?id=YicbFdNTTy>, свободный. – Загл. с экрана. – (дата обращения: 11.03.2025).

© Н. И. Егоренко, М. П. Егоренко, 2025