

Оценка качества семян пшеницы с использованием сверточной нейронной сети

А.В. Ковалев¹, А.С. Исаева²

*¹Научно-конструкторское бюро моделирующих и управляющих систем (НКБ
МИУС) ЮФУ*

*²Дизайн-центр микроэлектронной компонентной базы для систем искусственного
интеллекта ЮФУ*

Аннотация: В статье предлагается способ анализа и оценки качества семян пшеницы посредством классификации их изображений с использованием искусственного интеллекта, а именно сверточной нейронной сети. Предложена архитектура глубокой сверточной нейронной сети, создана база изображений для обучения и тестирования предложенной нейронной сети, обучение и тестирование нейронной сети проводилось в Tensorflow. Сделаны выводы об эффективности классификации изображений и предложены области использования предложенного способа анализа и оценки качества семян пшеницы.

Ключевые слова: сверточные нейронные сети, искусственный интеллект, классификация изображений.

Архитектура сверточной нейронной сети

Сверточная нейронная (convolutional neuron network, CNN) сеть является одним из самых популярных сейчас алгоритмов глубокого обучения, позволяющим классифицировать и отличать изображения. На основе данного алгоритма предлагаются автоматизированные системы анализа, классификации и преобразования изображений в различных областях знаний [1, 2]. При этом, сверточные нейронные сети требуют гораздо меньшей обработки исходных изображений [3]. Структура нейронной сети подразумевает наличие различного типа слоев: входного слоя, нескольких скрытых слоев и выходного слоя. Охарактеризуем используемые виды слоев:

- Входной слой. В этом слое каждому нейрону соответствует отдельная точка изображения с учетом количества цветовых каналов.
- Сверточный слой. Осуществляет двухмерную свертку значений из предыдущего уровня, используя набор ядер свертки (фильтров). Обычно в

каждом слое содержится определенный набор различных ядер свертки, что необходимо для выделения нескольких признаков в одном слое. Значения коэффициентов фильтров перед обучением устанавливаются случайным образом или с использованием специальных алгоритмов (обычно алгоритмов Ксавье [4] или Хе [5]), что важно для возможности достижения глобального минимума в процессе обучения. Далее каждое из ядер свертки обучается для выделения определенных признаков из массива данных, полученных из предыдущего слоя. На выходе сверточного слоя, в случае отсутствия дополнения краевыми точками, размерность массива данных уменьшается в соответствии со следующим правилом: $(n - k + 1)$, где n – является размерностью массива данных на входе, k – размерность используемого ядра свертки.

– Слой субдискретизации. В данном слое происходит уменьшение размерности данных (обобщение) путем объединения значений входной матрицы данных в рамках определенной области в одно значение. Этот слой позволяет увеличить устойчивость нейронной сети к изменению масштаба изображений, а также уменьшает число требуемых вычислений в последующих слоях нейронной сети. Существует несколько возможных вариантов субдискретизации: по максимуму [6, 7] (самый популярный вариант в настоящее время), по среднему значению [8] и промежуточные варианты [9]. В данной архитектуре применена субдискретизация по максимальному значению.

– Полносвязный слой. Нейроны в данном слое связаны с каждым из предыдущего и осуществляют конечное разделение на классы.

Для анализа и оценки качества семян пшеницы посредством классификации их изображений предложена архитектура сети, представленная на рис. 1.

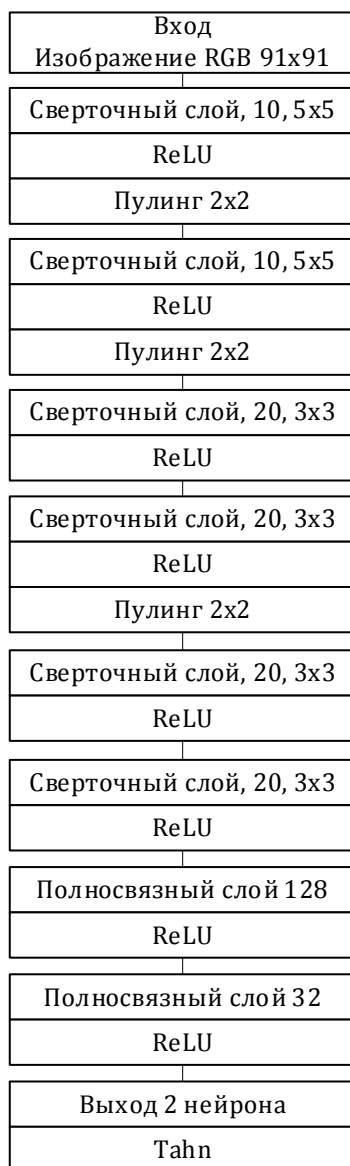


Рис. 1. – Архитектура сверточной нейронной сети

Предложенная архитектура сверточной нейронной сети состоит из следующих блоков:

1. Вход, на который подаются RGB-изображения размерностью 91x91;
2. Сверточный слой, содержащий 10 фильтров размером 5x5. Так как дополнения нулями не происходит, получаемая карта свертки имеет размерность 10x87x87, далее идет слой активации ReLU;
3. Слой подвыборки (пулинга) с окном 2x2 и шагом 2. Выходная размерность слоя составила 10x43x43;

4. Сверточный слой, содержащий 10 фильтров размером 5×5 . Дополнения нулями не происходит, получаемая карта свертки имеет размерность $10 \times 39 \times 39$, далее идет слой активации ReLU;

5. Слой пулинга с окном 2×2 и шагом 2. Выходная размерность слоя составила $10 \times 19 \times 19$;

6. Сверточный слой, содержащий 20 фильтров размером 3×3 . Дополнения нулями не происходит, получаемая карта свертки имеет размерность $10 \times 39 \times 39$, далее идет слой активации ReLU;

7. Сверточный слой, содержащий 20 фильтров размером 3×3 . Дополнения нулями не происходит, получаемая карта свертки имеет размерность $20 \times 15 \times 15$, далее идет слой активации ReLU;

8. Слой пулинга с окном 2×2 и шагом 2. Выходная размерность слоя составила $20 \times 7 \times 7$;

9. Сверточный слой, содержащий 20 фильтров размером 3×3 . Дополнения нулями не происходит, получаемая карта свертки имеет размерность $20 \times 5 \times 5$, далее идет слой активации ReLU;

10. Сверточный слой, содержащий 20 фильтров размером 3×3 . Дополнения нулями не происходит, получаемая карта свертки имеет размерность $20 \times 3 \times 3$, далее идет слой активации ReLU;

11. Полносвязный слой с числом нейронов 128, слой активации ReLU;

12. Полносвязный слой с числом нейронов 32, слой активации ReLU;

13. Выходной слой с двумя нейронами, соответствующими двум выбранным классам, слой активации Tanh.

Полученная архитектура имеет 43 312 весовых коэффициентов.

Параметры модели и обучения

Одним из важнейших параметров нейросети является функция активации нейронов. В данной работе используется два вида функции

активации: в большинстве слоев функция «линейный выпрямитель» ReLU (rectified linear unit) и гиперболический тангенс Tanh для выходного слоя. Выбор функции активации ReLU в качестве основной был сделан, исходя из следующих соображений: данная функция ускоряет обучение нейронной сети [10], приводит к лучшим решениям [11], чем обычные сигмоидоподобные функции, при этом в отличие от гиперболического тангенса и сигмоиды, имеет меньшую вычислительную сложность. Гиперболический тангенс Tanh, выбранный в качестве функции активации выбран из-за подходящего диапазона значений $(-1, 1)$ и хороших результатов в применении в нейронных сетях [12]. Инициализация весов в данной нейронной сети выполнена с применением метода Хе. Для решения проблемы переобучения применена регуляризация L2 [13] со значением коэффициента регуляризации равно $\lambda=10^{-4}$. В качестве метода градиентной оптимизации стохастических целевых функций, применяемого в качестве обучающего нейронную сеть, используется алгоритм Adam, основанный на адаптивных оценках моментов более низкого порядка [14].

Обучение и тестирование сверточной нейронной сети

Для обучения и тестирования разработанной архитектуры сверточной нейронной сети был создан набор из фотографий семян пшеницы разрешением 91×91 пикселей, разделенный на два класса: «хорошие семена» и «плохие семена». Изображения для данной предварительной выборки получены путем фотографирования отобранных вручную семян с использованием бытовой камеры. Пример характерных изображения семян из каждого из классифицируемых наборов данных показаны на рис. 2 и 3.



Рис. 2. – Пример изображений из обучающей выборки, класс
«хорошие семена»



Рис. 3. – Пример изображений из обучающей выборки, класс
«плохие семена»

В обоих классах были выделены следующие подклассы: обучающий подкласс, состоящий из 400 изображений, верификационный подкласс, состоящий из 68 изображений и проверочный, состоящий из 50 изображений. Весь набор данных, таким образом, состоит из 800 тренировочных изображений, 136 верификационных изображений и 100 тестовых изображений. Все используемые данные для каждого этапа обучения берутся

в случайном порядке, за исключением тестовых данных, которые сохраняют исходную последовательность для корректного последующего сравнения результатов обучения нейронной сети. Обучение и тестирование нейронной сети проводилось в Tensorflow версии 2.1, установленной в системе Ubuntu 18.04 LTS. Для обучения нейронной сети применялась система с графическим процессором Nvidia 2070 Super, что позволило осуществлять обучение нейронной сети за 25-27 секунд. Графики точности нейронной сети и функции потерь в процессе обучения показаны на рис. 4.

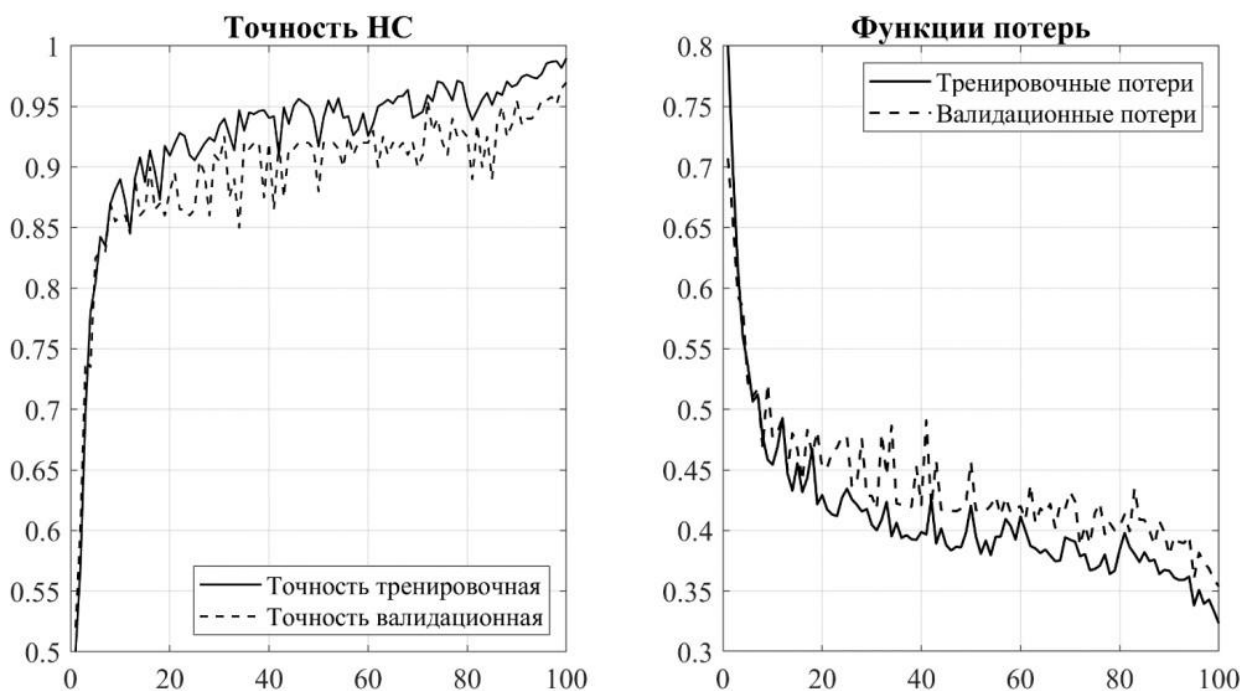


Рис. 4. – Процесс обучения нейронной сети

Результаты обучения (100 этапов по 8 изображений) показаны в таблице 1.

Таблица № 1

Результаты обучения сверточной нейронной сети

Тренировочные потери	Валидационные потери	Тренировочная точность	Валидационная точность	Тестовая точность
0.3233	0.3540	0.9900	0.9600	0.9700

Исходя из результатов, показанных в таблице 1, точность классификации на основе предложенной архитектуры глубокой сверточной нейронной сети на тестовой выборке достигла 97%. Однако, следует учесть тот факт, что обучающая и тестовая выборки были ограничены в объеме (800 и 100 соответственно). Более обширная выборка повысит точность классификации изображений. Имплементация разработанного способа оценки качества семян в реальном проекте, например, при разработке автоматизированного сепаратора семян, позволит достаточно быстро увеличить обучающую выборку и увеличить точность классификации изображений.

Научные исследования выполнены в рамках проекта «Создание высокотехнологичного производства аппаратно-программных комплексов обработки сельхозсырья на основе СВЧ излучения» (Соглашение с Минобрнауки РФ № 075-11-2019-083 от 20.12.2019 г., Договор ЮФУ № 18 от 20.09.2019 г., номер работы в ЮФУ № ХД/19-25-РТ).

Литература

1. Соловьев Р.А., Тельпухов Д.В., Кустов А.Г. Автоматическая сегментация спутниковых снимков на базе модифицированной сверточной нейронной сети UNET // Инженерный вестник Дона, 2017, №4. URL: ivdon.ru/ru/magazine/archive/n4y2017/4433.
2. Игнатьев А.В., Гилка В.В., Матыцына Д.А. Автоматическое распознавание типа застройки для системы экологического мониторинга // Инженерный вестник Дона, 2020, №1. URL: ivdon.ru/ru/magazine/archive/n1y2020/6266.
3. Tian Y. Artificial Intelligence Image Recognition Method Based on Convolutional Neural Network Algorithm // IEEE Access, 2020, pp. 1-14.

4. Glorot X., Bengio Y. Understanding the difficulty of training deep feedforward neural networks // Proceedings of the thirteenth international conference on artificial intelligence and statistics, 2010, pp. 249-256.

5. He K. et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification // Proceedings of the IEEE international conference on computer vision, 2015, pp. 1026-1034.

6. LeCun Y. et al. Backpropagation applied to handwritten zip code recognition // Neural computation, 1989, Vol. 1, No. 4, pp. 541-551.

7. Boureau Y. L. et al. Learning mid-level features for recognition // 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. – IEEE, 2010, pp. 2559-2566.

8. Scherer D., Müller A., Behnke S. Evaluation of pooling operations in convolutional architectures for object recognition // International conference on artificial neural networks, 2010, pp. 92-101.

9. Boureau Y. L., Ponce J., LeCun Y. A theoretical analysis of feature pooling in visual recognition // Proceedings of the 27th international conference on machine learning (ICML-10), 2010, pp. 111-118.

10. Krizhevsky A., Sutskever I., Hinton G. E. Imagenet classification with deep convolutional neural networks // Advances in neural information processing systems, 2012, pp. 1097-1105.

11. Zeiler M. D. et al. On rectified linear units for speech processing // 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. – IEEE, 2013, pp. 3517-3521.

12. Marra S., Iachino M. A., Morabito F. C. Tanh-like activation function implementation for high-performance digital neural systems // Ph. D. Research in Microelectronics and Electronics, 2006, pp. 237-240.

13. Cortes C., Mohri M., Rostamizadeh A. L2 regularization for learning kernels // arXiv preprint arXiv:1205.2653, 2012.

14. Kingma D. P., Adam Ba J. A method for stochastic optimization // arXiv preprint arXiv:1412.6980, 2014.

References

1. Solov'ev R.A., Tel'puhov D.V., Kustov A.G. Inzhenernyj vestnik Dona, 2017, №4. URL: ivdon.ru/ru/magazine/archive/n4y2017/4433.
2. Ignat'ev A.V., Gilka V.V., Matytsyna D.A. Inzhenernyj vestnik Dona, 2020, №1. URL: ivdon.ru/ru/magazine/archive/n1y2020/6266.
3. Tian Y. IEEE Access, 2020, pp. 1-14.
4. Glorot X., Bengio Y. Proceedings of the thirteenth international conference on artificial intelligence and statistics, 2010, pp. 249-256.
5. He K. et al. Proceedings of the IEEE international conference on computer vision, 2015, pp. 1026-1034.
6. LeCun Y. et al. Neural computation, 1989, Vol. 1, No. 4, pp. 541-551.
7. Boureau Y. L. et al. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. – IEEE, 2010, pp. 2559-2566.
8. Scherer D., Müller A., Behnke S. International conference on artificial neural networks, 2010, pp. 92-101.
9. Boureau Y. L., Ponce J., LeCun Y. Proceedings of the 27th international conference on machine learning (ICML-10), 2010, pp. 111-118.
10. Krizhevsky A., Sutskever I., Hinton G. E. Advances in neural information processing systems, 2012, pp. 1097-1105.
11. Zeiler M. D. et al. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. – IEEE, 2013, pp. 3517-3521.
12. Marra S., Iachino M. A., Morabito F. C. Ph. D. Research in Microelectronics and Electronics, 2006, pp. 237-240.
13. Cortes C., Mohri M., Rostamizadeh A. arXiv preprint arXiv:1205.2653, 2012.
14. Kingma D. P., Adam Ba J. arXiv preprint arXiv:1412.6980, 2014.