

УДК 004.855.5

ВЫБОР ФУНКЦИИ АКТИВАЦИИ И ФУНКЦИИ ПОТЕРЬ ДЛЯ ЗАДАЧИ КЛАССИФИКАЦИИ АКТИВНОСТЕЙ ЧЕЛОВЕКА



В.В. Венгеренко

Мл. науч. сотр. ОИПИ НАН
Беларуси, аспирант ОИПИ
НАН Беларуси
vengerenkovadim1@gmail.com



М.М. Лукашевич

Доцент кафедры
информационных систем
управления БГУ, ст. науч.
сотр. лаборатории №222
ОИПИ НАН Беларуси,
кандидат технических
наук, доцент
LukashevichMM@bsu.by

В.В. Венгеренко

Окончил Белорусский государственный университет информатики и радиоэлектроники. Область научных интересов связана с разработкой методов и алгоритмов машинного обучения для решения задач анализа данных и компьютерного зрения.

М.М. Лукашевич

Является доцентом кафедры информационных систем управления БГУ, ст. науч. сотр. лаборатории №222 ОИПИ. Область научных интересов связана с разработкой методов и алгоритмов машинного обучения для решения задач анализа данных и компьютерного зрения.

Аннотация. Цель работы – исследовать влияние выбора функции активации и функции потерь на показатели нейросетевых моделей при решении задачи классификации активностей человека.

Представлены результаты экспериментов по оценке показателей нейросетевых моделей, которые классифицируют активности человека и при этом используют различные комбинации функции активации и функции потерь. В экспериментальных исследованиях использовалась выборка из набора данных HMDB51 для 10 классов. Для рассматриваемой нейросетевой архитектуры была выявлена комбинация гиперпараметров, позволяющая достичь максимальной точности классификации.

Ключевые слова: нейронная сеть, функция активации, функция потерь, точность предсказаний, классификация.

Введение. Актуальность темы исследования заключается в том, что классификация активностей людей повышает уровень безопасности посредством приложений в видеонаблюдении реального времени, позволяя обнаруживать потенциально опасные ситуации, а также способствует достижениям в таких областях, как здравоохранение и робототехника.

Задача исследования состоит в оценке показателей нейросетевых моделей, которые классифицируют активности человека и при этом используют различные комбинации функции активации и функции потерь.

Распознавание активностей человека является актуальной задачей в компьютерном зрении и особенно полезно в системах видеонаблюдения. Можно выделить два подхода для распознавания активностей человека: на основе акселерометров и гироскопов и на основе визуальных данных (изображения и видео). Во втором подходе используется компьютерное

зрение, позволяющее определять сложные контекстно-зависимые активности. В настоящее время подход на основе нейронных сетей дает наиболее высокие результаты. Среди архитектур следует выделить архитектуру трансформера и сверточные нейронные сети [1–5], на которых строятся современные решения.

Для рассматриваемой задачи используются следующие наборы данных: HMDB51 [6], UCF101 [7], Kinetics [8], ActivityNet [9], Charades [10], THUMOS [11], HACs [12], MOD20 [13], Sports-1M [14], NAA500 [15].

Перечисленные наборы содержат видео различной длительности из таких источников, как YouTube, архивы Прелингера (the Prelinger archives), Google видео и так далее.

Среди перечисленных наборов MOD20 содержит наименьшее количество классов (20) и экземпляров (2324 видео). В наборе Kinetics-700 700 классов, что является наибольшим значением данного показателя для перечисленных наборов. Из крупных наборов данных стоит отметить HACs, в котором содержится 1,55 млн клипов длительностью 2 с, полученных из 504 тыс. видео, и 140 тыс. сегментов, полученных из 50 тыс. видео.

Примерами классов могут послужить такие классы UCF101, как “Typing” (набор на клавиатуре), “Playing Guitar” (игра на гитаре), “Rope Climbing” (лазание по канату), “Walking with a Dog” (прогулка с собакой), “Biking” (езда на велосипеде) [7].

Экспериментальный набор данных. Для проведения экспериментов была выбрана часть набора данных HMDB51, который в исходном виде включает в себя 6766 видео и 51 класс [6]. Экспериментальный набор содержит 500 видео из первых 10 классов (“brush_hair”, “cartwheel”, “catch”, “chew”, “clap”, “climb”, “climb_stairs”, “dive”, “draw_sword”, “dribble”) набора HMDB51. Данный экспериментальный набор делится на обучающую, валидационную и тестовую выборки в соотношении 60/20/20 соответственно. При этом применяется стратификация по классам. Каждая выборка заключается в 1 пакет данных (batch). Как видно из рисунка 1, данные в экспериментальном наборе и выборках сбалансированы по классам.

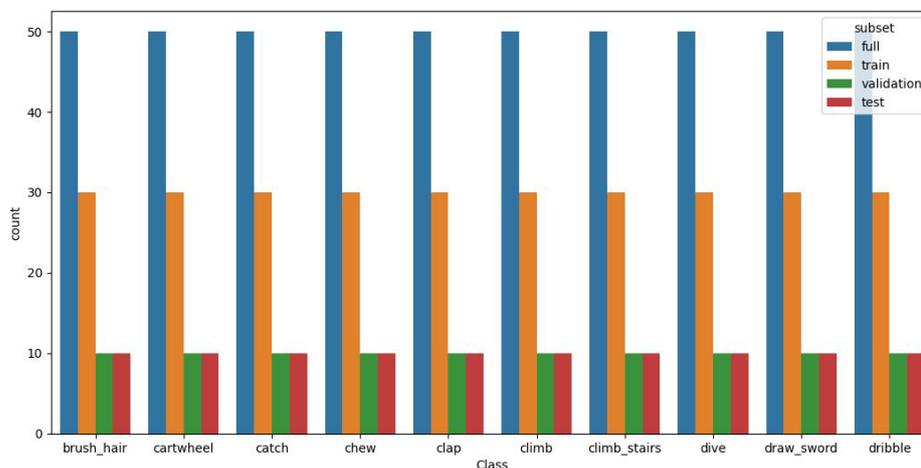


Рисунок 1. Распределение экспериментальных данных по классам и выборкам

Каждое видео из экспериментального набора содержит 40 кадров размером 224×224 пикселей. В случае, если длительность видео из исходного набора данных превышает установленное ограничение в 40 кадров, то последовательность кадров этой длины выбирается из середины видео. Иначе вся последовательность кадров из исходного видео зацикливается до нужной длины.

Результаты исследований. Эксперименты проводились с использованием языка программирования Python и его пакетов, из которых стоит отметить tensorflow, numpy, cv2, scikit-learn, pandas, seaborn, matplotlib. Для выполнения вычислений с помощью библиотеки

TensorFlow использовался графический процессор (Graphical Processing Unit, GPU) NVIDIA GeForce MX150.

Общая нейросетевая архитектура для проведения экспериментов по примеру в [16] представлена на рисунке 2.

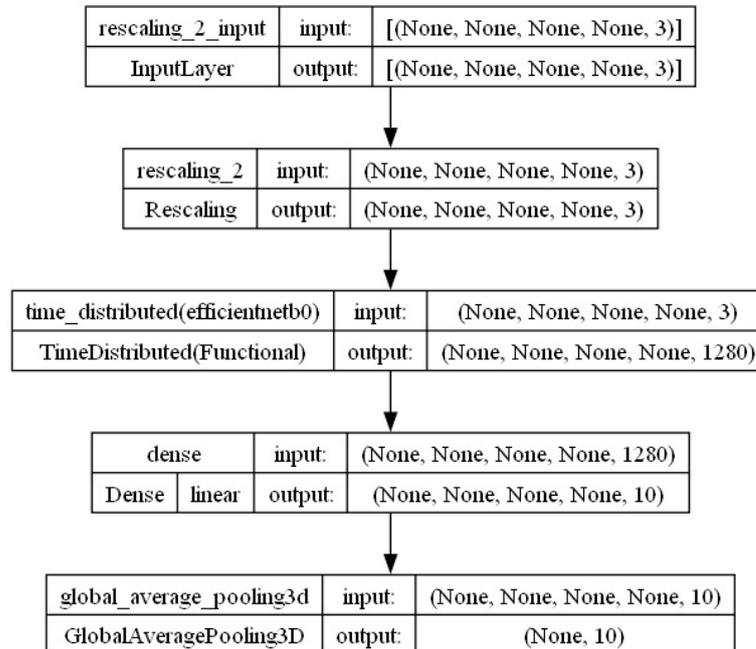


Рисунок 2. Общая архитектура нейронной сети для проведения экспериментов

Для предварительной обработки входных данных используется слой *Rescaling* с параметром *scale* со значением 255, что позволяет отмасштабировать входные значения из исходного диапазона [0, 1] в новый диапазон [0, 255].

Слой *TimeDistributed* является оберткой над слоем, передаваемым в качестве параметра, то есть в данном случае над предварительно обученной на наборе данных ImageNet (1000 классов) моделью EfficientNetB0, возвращаемой соответствующей функцией библиотеки TensorFlow. Это позволяет независимо применить оборачиваемый слой к каждому временному срезу входных данных, используя одинаковый набор весовых коэффициентов. Таким образом, в рассматриваемой задаче модель EfficientNetB0 независимо задействуется для каждого кадра отдельного видео из поступающего на нее набора данных.

Использование предварительно обученной модели EfficientNetB0 в составе экспериментальных нейросетевых моделей позволяет ускорить достижение высоких показателей при решении поставленной задачи. Параметры *include_top* и *trainable* данной модели принимают значение *False* для исключения раннего полносвязного слоя из этой модели и обеспечения неизменности значений весовых коэффициентов ее оставшихся слоев на этапе обучения экспериментальных моделей.

Предпоследний полносвязный слой *Dense* предназначен для соотнесения признаков, извлеченных на предыдущих слоях для каждого кадра отдельного видео, с 10 классами, обозначенными в решаемой задаче.

Последний слой *GlobalAveragePooling3D* выполняет операцию усреднения данных с предыдущего слоя по всем кадрам отдельного видео таким образом, что это позволяет соотнести анализируемое видео с 10 классами, обозначенными в решаемой задаче.

Все экспериментальные модели обучаются в течение 15 эпох с применением оптимизатора *Adam* с параметрами по умолчанию.

Точность предсказаний рассчитывается следующим образом [17]:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}, \quad (1)$$

где TP – True Positives (истинно положительные), количество положительных наблюдений, которые модель правильно предсказала как положительные;

TN – True Negatives (истинно отрицательные), количество отрицательных наблюдений, которые модель правильно предсказала как отрицательные;

FP – False Positives (ложноположительные или ошибка 1-го рода), количество отрицательных наблюдений, которые модель неверно предсказала как положительные;

FN – False Negatives (ложноотрицательные или ошибка 2-го рода), количество положительных наблюдений, которые модель неверно предсказала как отрицательные.

В TensorFlow для оценки точности предсказаний присутствуют такие метрики, как *SparseCategoricalAccuracy* (SCA) [18] и *CategoricalAccuracy* (CA) [19]. В первом случае истинная принадлежность экземпляра определенному классу должна определяться как число, соответствующее этому классу, например, его порядковый номер. Во втором – как one-hot вектор, длина которого равна количеству классов в задаче, при этом в случае истинной принадлежности рассматриваемого экземпляра классу с порядковым номером i метка класса в i -й позиции данного вектора принимает значение 1, иначе – 0.

В экспериментах использовались функции активации [20] и функции потерь [21; 22], представленные в таблицах 1 и 2 соответственно.

Таблица 1. Функции активации, используемые в экспериментах

Название функции активации	Формула
<i>Softmax</i>	$\frac{e^x}{\sum_{i=0}^{K-1} e^x}$
<i>Swish</i> (или <i>SiLU</i> – Sigmoid-weighted Linear Unit)	$x \cdot \frac{1}{1 + e^{-x}}$
<i>Hard Swish</i> (или <i>Hard SiLU</i>)	$\begin{cases} 0, & x < -3, \\ \frac{x \cdot (x + 3)}{6}, & -3 \leq x \leq 3, \\ x, & x > 3 \end{cases}$
<i>Mish</i>	$x \cdot \tanh(\log(e^x + 1))$
<i>GELU</i> (Gaussian Error Linear Unit)	$0,5 \cdot x \cdot \left(1 + \frac{x}{\sqrt{2}} \int_0^{\frac{x}{\sqrt{2}}} e^{-t^2} dt\right)$
<i>CELU</i> (Continuously-differentiable Exponential Linear Unit)	$\begin{cases} \alpha \cdot (e^{\frac{x}{\alpha}} - 1), & x < 0, \\ x, & x \geq 0 \end{cases}$

Таблица 2. Функции потерь, используемые в экспериментах

Название функции потерь	Формула
<i>CategoricalCrossentropy</i> (категориальная перекрестная энтропия, <i>CCE</i>)	$-\sum_{i=0}^{C-1} y_{true\ i} \cdot \log(y_{pred\ i})$
<i>SparseCategoricalCrossentropy</i> (разреженная <i>CCE</i> , <i>SCCE</i>)	$-\log(y_{pred\ y_{true}})$
<i>CategoricalFocalCrossentropy</i> (категориальная фокальная перекрестная энтропия, <i>CFCE</i>)	$-\alpha \cdot \sum_{i=0}^{C-1} (1 - y_{pred\ i})^\gamma \cdot y_{true\ i} \cdot \log(y_{pred\ i})$

<p><i>CategoricalHinge</i> (категориальная граница, <i>CH</i>)</p> <p style="text-align: right;">верхняя</p>	$\max \left(1 + \max \left((1 - y_{true}) \odot y_{pred} \right) - \sum_{i=0}^{C-1} (y_{true} \odot y_{pred})_i, 0 \right)$
<p><i>KLDivergence</i> (Kullback- Leibler Divergence, <i>KLD</i>; расстояние Кульбака- Лейблера)</p>	$KLD(y_{true}, y_{pred}) = \sum_{i=0}^{C-1} y_{true\ i} \cdot \log \left(\frac{y_{true\ i}}{y_{pred\ i}} \right)$

В первой части экспериментов, посвященной исследованию влияния выбора применяемой между сверточными слоями EfficientNetB0 функции активации на показатели экспериментальных моделей, для оценки точности предсказаний использовалась метрика *SCA* и функция потерь *SCCE* (разреженная категориальная перекрестная энтропия) с параметром *from_logits* со значением *True* (в таком случае к предсказаниям модели перед непосредственным расчетом значения *SCCE* будет применяться функция *Softmax*). На предпоследнем полносвязном слое используется линейная (*Linear*) функция активации, пропускающая входные данные без изменений [20].

Функция активации задается параметром *activation* функции `tensorflow.keras.applications.EfficientNetB0`.

Результат применения функции активации *Swish* между сверточными слоями EfficientNetB0 представлен на рисунке 3.

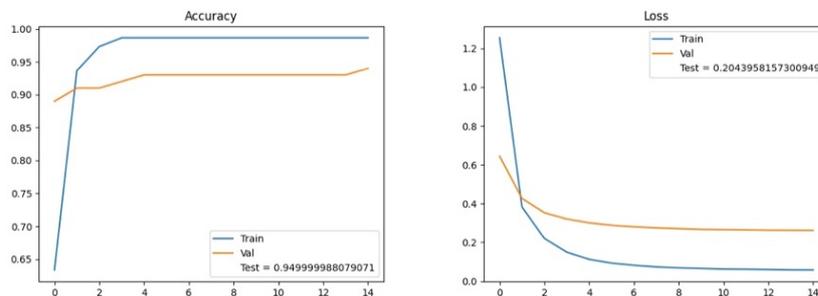


Рисунок 3. Графики точности (слева) и функции потерь (справа) при применении между сверточными слоями EfficientNetB0 функции активации *Swish*

Результат применения функции активации *Hard Swish* между сверточными слоями EfficientNetB0 представлен на рисунке 4.

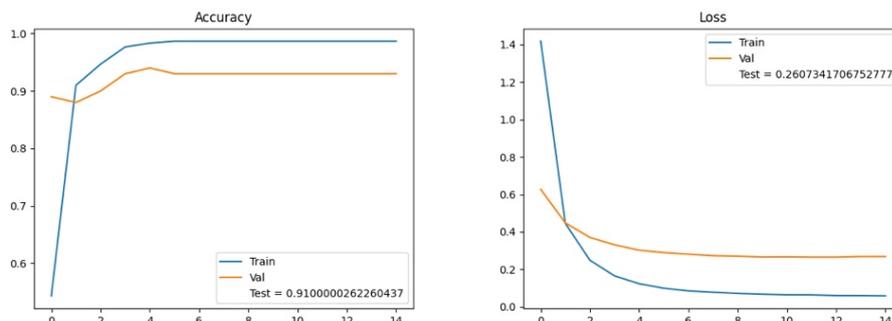


Рисунок 4. Графики точности (слева) и функции потерь (справа) при применении между сверточными слоями EfficientNetB0 функции активации *Hard Swish*

Результат применения функции активации *Mish* между сверточными слоями EfficientNetV0 представлен на рисунке 5.

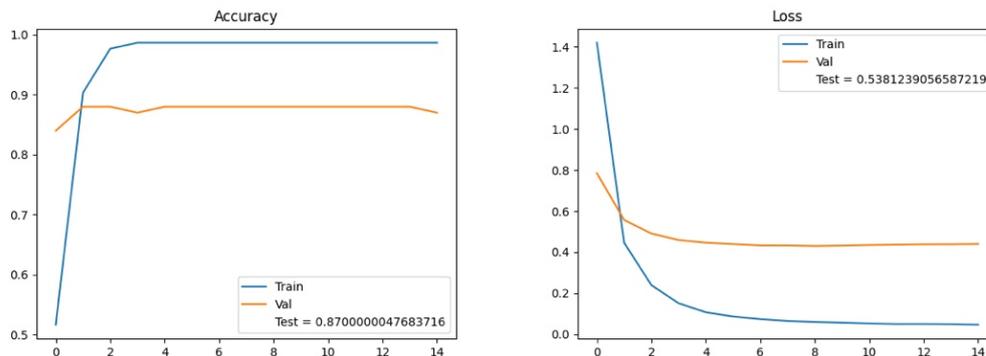


Рисунок 5. Графики точности (слева) и функции потерь (справа) при применении между сверточными слоями EfficientNetV0 функции активации *Mish*

Результат применения функции активации *GELU* между сверточными слоями EfficientNetV0 представлен на рисунке 6.

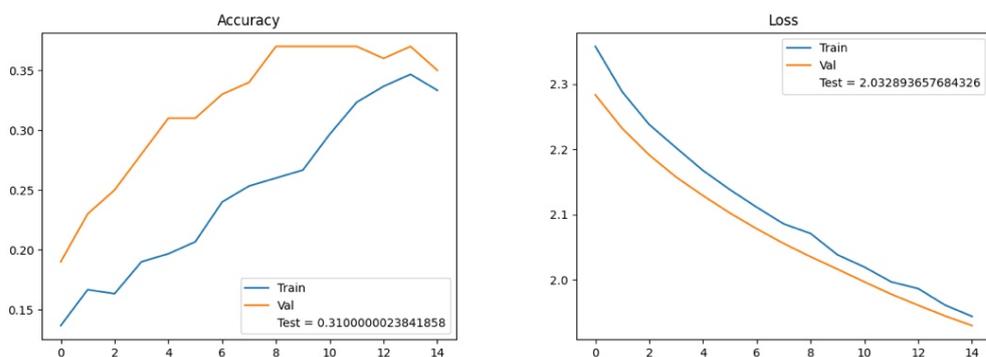


Рисунок 6. Графики точности (слева) и функции потерь (справа) при применении между сверточными слоями EfficientNetV0 функции активации *GELU*

Результат применения функции активации *CELU* (параметр α равен 0,5) между сверточными слоями EfficientNetV0 представлен на рисунке 7.

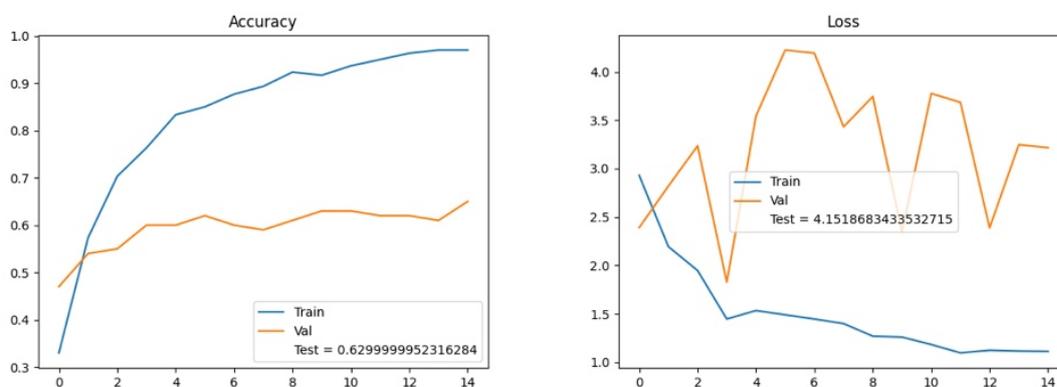


Рисунок 7. Графики точности (слева) и функции потерь (справа) при применении между сверточными слоями EfficientNetV0 функции активации *CELU* (параметр α равен 0,5)

Во второй части экспериментов, посвященной исследованию влияния выбора функции потерь на показатели экспериментальных моделей, на предпоследнем слое EfficientNetV0 используется функция активации *Softmax*, а параметр *from_logits* при его наличии у функции потерь принимает значение *False*. Между сверточными слоями EfficientNetV0 применяется функция активации *Swish*, поскольку в первой части экспериментов она позволила добиться лучших показателей.

Результат выбора *SCCE* в качестве функции потерь представлен на рисунке 8. Метрикой оценки точности в данном случае выступает *SCA*.

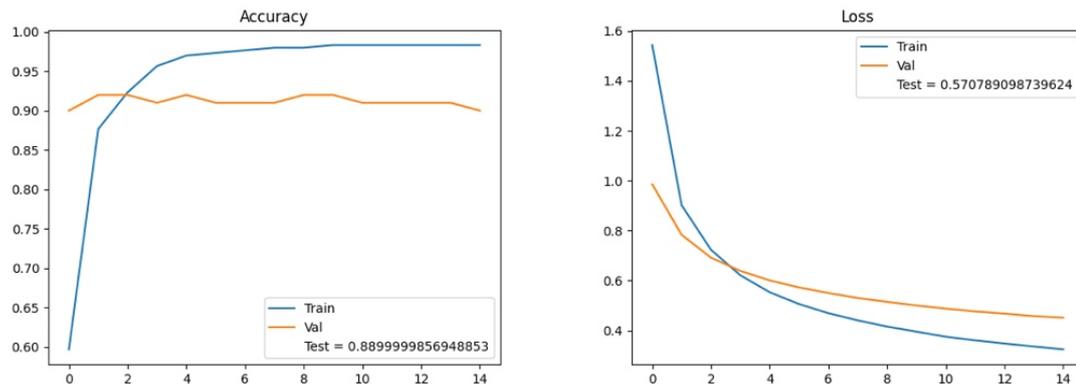


Рисунок 8. Графики точности (слева) и функции потерь *SCCE* (справа)

В оставшихся экспериментах этой части метрикой оценки точности выступает *CA*, а y_{true} задается как one-hot вектор.

Результат выбора в качестве функции потерь *CCE* представлен на рисунке 9.

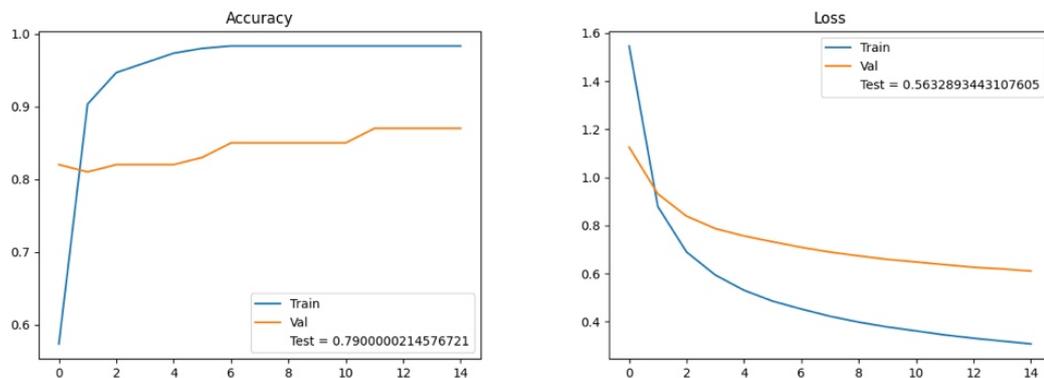


Рисунок 9. Графики точности (слева) и функции потерь *CCE* (справа)

Результат выбора в качестве функции потерь *CFCE* с параметрами α со значением 0,25 и γ со значением 2 представлен на рисунке 10.

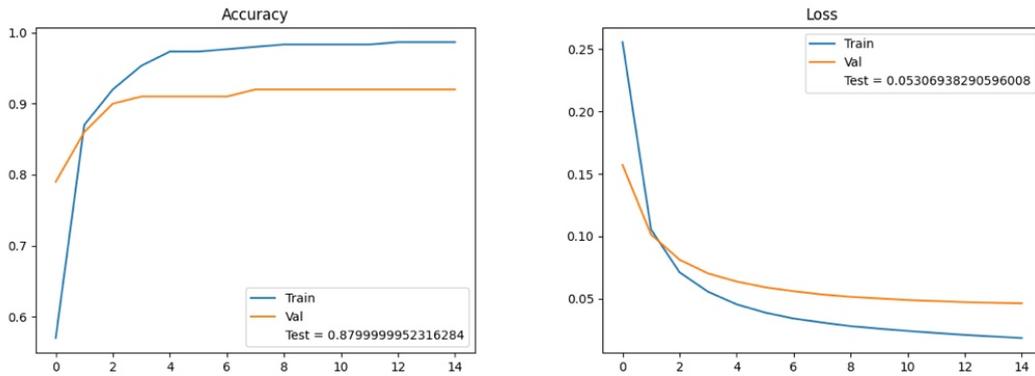


Рисунок 10. Графики точности (слева) и функции потерь *CFCE* (справа), α со значением 0,25 и γ со значением 2
 Результат выбора в качестве функции потерь *CH* представлен на рисунке 11.

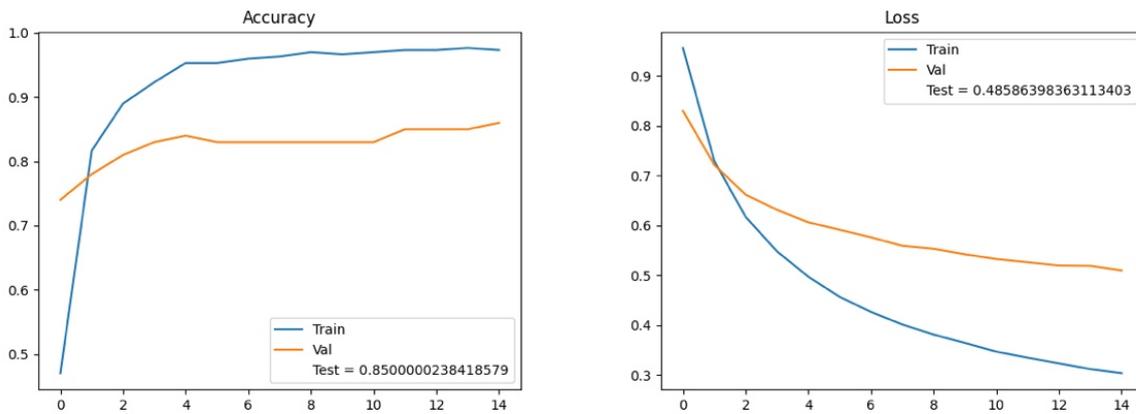


Рисунок 11. Графики точности (слева) и функции потерь *CH* (справа)
 Результат выбора в качестве функции потерь *KLD* представлен на рисунке 12.

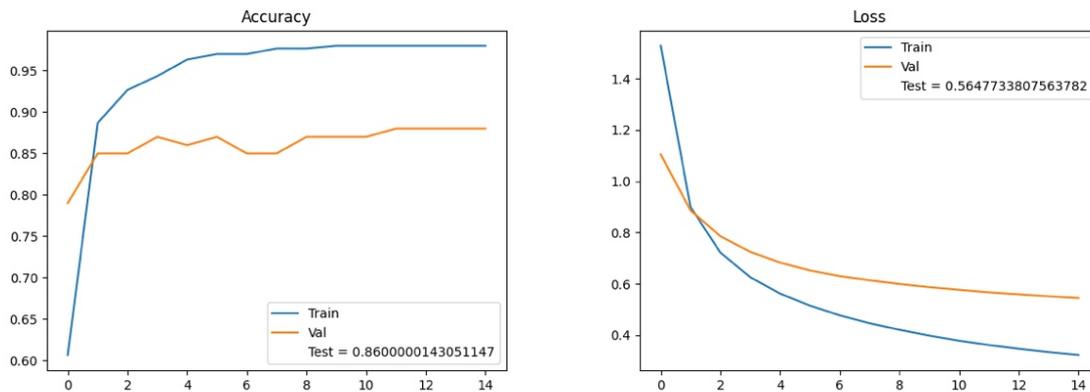


Рисунок 12. Графики точности (слева) и функции потерь *KLD* (справа)
 В TensorFlow для *KLD* значения, содержащиеся в *y_{true}* и *y_{pred}*, ограничиваются до диапазона [0, 1] при их несоответствии этому диапазону.

Результаты экспериментов сведены в таблицу 3.
 Таблица 3. Результаты экспериментов

Функция активации между сверточными слоями EfficientNetB0	Функция потерь	Метрика оценки точности предсказаний, функция активации на предпоследнем полносвязном слое экспериментальной модели	Точность предсказаний на тестовой выборке	Значение функции потерь на тестовой выборке
<i>Swish</i>	<i>SCCE</i> (<i>from_logits=True</i>)	<i>SCA, Linear</i>	0,95	0,20
<i>Hard Swish</i>			0,91	0,26
<i>Mish</i>			0,87	0,54
<i>GELU</i>			0,31	2,03
<i>CELU</i> ($\alpha = 0,5$)			0,63	4,15
<i>Swish</i>	<i>SCCE</i> (<i>from_logits=False</i>)	<i>SCA, Softmax</i>	0,89	0,57
	<i>CCE</i> (<i>from_logits=False</i>)	<i>CA, Softmax</i>	0,79	0,56
	<i>CFCE</i> ($\alpha = 0,25, \gamma = 2$, <i>from_logits=False</i>)		0,88	0,05
	<i>CH</i>		0,85	0,49
	<i>KLD</i>		0,86	0,56

Таким образом, по результатам всех экспериментов совместное использование функции активации *Swish* между сверточными слоями EfficientNetB0, функции потерь *SCCE* с параметром *from_logits* со значением *True*, метрики оценки точности предсказаний *SCA* и линейной функции активации на предпоследнем полносвязном слое позволяет достичь лучших показателей для данной задачи и нейросетевой архитектуры.

Заключение. Выбор функции активации *Swish* и функции потерь *SCCE* позволил достичь наилучших показателей как в первой, так и во второй части экспериментов. При этом для данной архитектуры применять функцию *Softmax* предпочтительнее не на предпоследнем полносвязном слое, а после прохождения данными всех слоев модели, что обеспечивается установкой значения *True* для параметра *from_logits* функции потерь *SCCE*. Выбор функций активации *GELU* и *CELU* с параметром α со значением 0,5 в первой части экспериментов значительно ухудшил показатели соответствующих экспериментальных моделей. Стоит отметить незначительное влияние выбора функции потерь на точность предсказаний во второй части экспериментов.

Список литературы

- [1] Weon I.-S., Lee S.-G., Ryu J.-K. Object Recognition Based Interpolation With 3D LIDAR and Vision for Autonomous Driving of an Intelligent Vehicle. *IEEE Access*. 2020;8:65599-65608. DOI:10.1109/ACCESS.2020.2982681.
- [2] Eltzner B., Huckemann S. F. A SMEARY CENTRAL LIMIT THEOREM FOR MANIFOLDS WITH APPLICATION TO HIGH-DIMENSIONAL SPHERES. *The Annals of Statistics*. 2019;47(6):3360-3381. DOI:10.1214/18-AOS1781.
- [3] Simonyan K., Zisserman A. Two-Stream Convolutional Networks for Action Recognition in Videos. *Proceedings of the 28th International Conference on Neural Information Processing Systems*. 2014;1:568-576.
- [4] Yedder H. B., Cardoen B., Hamarneh G. Deep learning for biomedical image reconstruction: a survey. *Artificial Intelligence Review*. 2021;54:215-251. DOI:10.1007/s10462-020-09861-2.
- [5] Wang L., Xiong Y., Wang Z., Qiao Y., Lin D., Tang X., Van Gool L. Temporal segment networks: Towards good practices for deep action recognition. *Computer Vision – ECCV 2016*. 2016;9912:20-36. DOI:10.1007/978-3-319-46484-8_2.
- [6] Serre Lab » HMDB: a large human motion database [Electronic resource]. – Mode of access: <https://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/>. – Date of access: 30.01.2025.

- [7] CRCV | Center for Research in Computer Vision at the University of Central Florida [Electronic resource]. – Mode of access: <https://www.crcv.ucf.edu/data/UCF101.php>. – Date of access: 30.01.2025.
- [8] GitHub - cvdfoundation/kinetics-dataset [Electronic resource]. – Mode of access: <https://github.com/cvdfoundation/kinetics-dataset>. – Date of access: 30.01.2025.
- [9] Activity Net [Electronic resource]. – Mode of access: <http://activity-net.org/>. – Date of access: 03.02.2025.
- [10] Charades Challenge [Electronic resource]. – Mode of access: <http://vuchallenge.org/charades.html>. – Date of access: 03.02.2025.
- [11] THUMOS Challenge 2015 [Electronic resource]. – Mode of access: <http://www.thumos.info/home.html>. – Date of access: 05.02.2025.
- [12] HACs Dataset [Electronic resource]. – Mode of access: <http://hacs.csail.mit.edu/>. – Date of access: 07.02.2025.
- [13] MOD20 Dataset [Electronic resource]. – Mode of access: <https://asankagp.github.io/mod20/>. – Date of access: 08.02.2025.
- [14] Large-scale Video Classification with Convolutional Neural Networks (CVPR 2014) [Electronic resource]. – Mode of access: <https://cs.stanford.edu/people/karpathy/deepvideo/>. – Date of access: 08.02.2025.
- [15] HAA500 Project Website [Electronic resource]. – Mode of access: <https://www.cse.ust.hk/haa/>. – Date of access: 10.02.2025.
- [16] Load video data | TensorFlow Core [Electronic resource]. – Mode of access: https://www.tensorflow.org/tutorials/load_data/video. – Date of access: 11.02.2025.
- [17] Classification Metrics using Sklearn - GeeksforGeeks [Electronic resource]. – Mode of access: <https://www.geeksforgeeks.org/sklearn-classification-metrics/>. – Date of access: 11.02.2025.
- [18] tf.keras.metrics.SparseCategoricalAccuracy | TensorFlow v2.16.1 [Electronic resource]. – Mode of access: https://www.tensorflow.org/api_docs/python/tf/keras/metrics/SparseCategoricalAccuracy. – Date of access: 14.02.2025.
- [19] tf.keras.metrics.CategoricalAccuracy | TensorFlow v2.16.1 [Electronic resource]. – Mode of access: https://www.tensorflow.org/api_docs/python/tf/keras/metrics/CategoricalAccuracy. – Date of access: 15.02.2025.
- [20] Layer activation functions [Electronic resource]. – Mode of access: <https://keras.io/api/layers/activations/>. – Date of access: 15.02.2025.
- [21] Probabilistic losses [Electronic resource]. – Mode of access: https://keras.io/api/losses/probabilistic_losses/. – Date of access: 16.02.2025.
- [22] Hinge losses for "maximum-margin" classification [Electronic resource]. – Mode of access: https://keras.io/api/losses/hinge_losses/. – Date of access: 17.02.2025.

Авторский вклад

Венгеренко Вадим Владимирович – проведение экспериментов, анализ полученных результатов.

Лукашевич Марина Михайловна – руководство исследованием, постановка задачи исследования, формирование структуры статьи.

SELECTION OF ACTIVATION FUNCTION AND LOSS FUNCTION FOR THE PROBLEM OF HUMAN ACTIVITIES CLASSIFICATION

V.V. Vengerenko

*Junior Researcher at the UIIP
NASB, Postgraduate Student at the
UIIP NASB*

M.M. Lukashevich

*Associate Professor of the
Department of Information
Management Systems of BSU,
Senior Researcher of the Laboratory
No. 222 of the UIIP NASB
PhD of Technical sciences,
Associate Professor*

Abstract. The purpose of the research is to investigate the influence of the selection of activation function and loss function on the performance of neural network models when solving the problem of classifying human activities.

The results of experiments on assessing the performance of neural network models that classify human activities and use various combinations of activation function and loss function are presented. The experimental studies used a sample from the HMDB51 dataset for 10 classes. A combination of hyperparameters was identified for the considered neural network architecture that allows achieving maximum classification accuracy.

Keywords: neural network, activation function, loss function, predictions accuracy, classification.