ЖУРНАЛ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И МАТЕМАТИЧЕСКОЙ ФИЗИКИ, 2021, том 61, № 5, с. 896–910

ИНФОРМАТИКА

УДК 519.65

ОБЗОР МЕТОДОВ ВИЗУАЛИЗАЦИИ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ[1)]

© 2021 г. С. А. Матвеев[1,2,*], И. В. Оселедец[1,2,], Е. С. Пономарев[1,***], А. В. Чертков[1]**

[1] 121205 Москва, Большой бульвар, 30, стр. 1, Сколковский институт науки и технологий, Россия

[2] 119333 Москва, ул. Губкина, 8, Институт вычислительной математики им. Г.И. Марчука Российской академии наук, Россия

**e-mail: s.matveev@skoltech.ru*

***e-mail: evgenii.ponomarev@skoltech.ru*

****e-mail: andrei.chertkov@skolkovotech.ru*

Поступила в редакцию 24.11.2020 г.
Переработанный вариант 24.11.2020 г.
Принята к публикации 11.12.2020 г.

Современные алгоритмы, основанные на искусственных нейронных сетях, крайне полезны при решении множества сложных задач компьютерного зрения, робастного управления, анализа звука и текстов на естественном языке в приложениях обработки данных, робототехники и т.д. Однако для успешного внедрения нейросетевого подхода в критически значимые системы, например, в медицине или в судебной практике, необходима понятная человеку интерпретация внутренней архитектуры и процесса принятия решений сетью. В последние годы особую распространенность для создания интерпретируемых моделей глубокого обучения приобрели методы анализа, основанные на различных техниках визуализации, применяемых к графу вычислений, профилю функции потерь, к параметрам отдельных слоев сети и даже к отдельным нейронам. В данном обзоре систематизируются существующие математические методы анализа и объяснения поведения соответствующих алгоритмов и приводятся постановки соответствующих задач вычислительной математики. Исследование и визуализация глубоких нейронных сетей являются новыми, малоизученными, и в то же время бурно развивающимися областями. Рассмотренные методы позволяют заглянуть вглубь и лучше понять работу нейросетевых алгоритмов. Библ. 57. Фиг. 5. Табл. 2.

Ключевые слова: искусственная нейронная сеть, интеллектуальный анализ данных, машинное обучение, глубокое обучение, визуализация искусственной нейронной сети.

DOI: 10.31857/S0044466921050148

1. ВВЕДЕНИЕ

Современные вычислительные технологии и алгоритмы все чаще используют нейронные сети. Искусственные нейронные сети (далее ИНС) и глубокое обучение (Deep Learning, далее DL) [1] стали практически незаменимыми в приложениях анализа больших объемов данных, машинного зрения, автоматической обработки естественного языка и др. На сегодняшний день ИНС уже находят применение в автономных роботизированных системах на производстве, в автоматизированных биомедицинских системах, в системах автономного вождения автомобилей и в широком спектре иных робототехнических приложений (см., например, [2]–[4]). Однако для дальнейшего развития нейросетевого подхода и возможности полноценного использования ИНС в критически значимых практических областях, например, в медицине, в судебной или финансовой системах, где цена ошибки очень высока, необходима возможность создания интерпретируемых DL моделей (Explainable Deep Learning, далее EDL) [5]–[7]. На сегодняшний день ИНС в подобных приложениях используются преимущественно лишь в качестве систем поддержки принятия решений, т.е. конечное решение, которое принимается с учетом мнения ИНС, остается все-таки за человеком – специалистом в соответствующей области знания.

[1)]Работа выполнена при финансовой поддержке Минобрнауки РФ (проект № 075-15-2020-801).

Таким образом, основной целью данной работы является обзор математических методов и технологий анализа работы широкого класса вычислительных алгоритмов – искусственных нейронных сетей.

В широком смысле EDL может рассматриваться как понятное человеку объяснение, почему конкретное решение было принято конкретной искусственной нейросетевой моделью. Подобное объяснение может быть полезным в трех важных направлениях.

1. *Понимание модели*, связанное с нахождением зависимостей между конкретной реализацией ИНС и механизмами ее внутреннего функционирования с одной стороны, и даваемыми ИНС предсказаниями с другой стороны.

2. *Отладка модели*, связанная с поиском дефектов структуры ИНС или артефактов обучения при возникновении проблемы со сходимостью процесса обучения или наличия не оптимального режима функционирования.

3. *Улучшение модели*, связанное с динамическим внесением модификаций в ИНС на основе экспертных оценок и конкретных знаний из моделируемой предметной области.

В современной литературе часто вводятся два термина: “explainability” и “interpretability”. Также может отдельно рассматриваться вопрос, связанный с тем, какому именно человеку объяснение понятно – объяснение модели, наглядное для специалиста в области машинного обучения (Machine Learning, далее ML), может оказаться совершенно не понятным финансисту, врачу и т.п. Мы не будем углубляться в эти вопросы (подробное обсуждение можно найти, например, в книге [8]), и далее в работе употребляем единый термин EDL, предполагая, что его смысл понятен из контекста.

Зрение является для человека основным инструментом изучения окружающего мира, в этой связи одним из наиболее перспективных подходов к разработке EDL моделей, в контексте всех трех обозначенных выше направлений, оказывается *визуализация*. Визуальное представление может строиться для графа вычислений или профиля функции потерь, а также для параметров отдельных слоев ИНС, и даже для отдельных нейронов.

Развитие методов визуализации ИНС может привести к появлению полезных инструментов для использования учеными в фундаментальных исследованиях в области наук о мозге. И, наоборот, современные исследования в науках о мозге могут послужить катализатором для соответствующих разработок в области ML и непосредственно визуализации ИНС. В частности, они позволяют поставить важный вопрос о формировании в ИНС аналогов памяти и специализации нейронов, присутствующих в естественных нейронных сетях [9]–[11]. Интересным направлением исследований также является анализ способов получения максимального отклика от заданных групп искусственных нейронов на определенные типы “раздражителей” [12] по аналогии с соответствующими результатами из нейронаук. Так, например, в работе [13] в рамках экспериментов с мышами исследователи обнаружили особые (“главные”) нейроны, ответственные за конкретный приобретенный поведенческий навык. Безусловно, современные ИНС имеют ряд существенных отличий от естественных нейронных сетей, однако применение схожих подходов для исследования естественных и искусственных нейронных сетей представляется уместным и перспективным.

Отметим, что систематический интегральный подход к задаче визуализации ИНС начал формироваться лишь в последние пять лет, однако на сегодняшний день в литературе уже представлен ряд обзоров, затрагивающих в той или иной степени эту тематику. После обсуждения перспективных методов и программного обеспечения для визуализации ИНС нами будет проведен краткий анализ этих работ. Актуальность предлагаемого обзора связана с необходимостью совместного рассмотрения новейших алгоритмов и программных решений в области визуализации глубоких нейросетевых структур, имеющих различные архитектуры и назначения, а также систематизации уже существующих обзоров по данной теме.

2. МЕТОДЫ ВИЗУАЛИЗАЦИИ

Исследование и визуализация глубоких ИНС являются новыми, малоизученными и в то же время бурно развивающимися областями, позволяющими заглянуть вглубь и лучше понять работу нейросетевых алгоритмов. На сегодняшний день существует большое разнообразие методов визуализации, относящихся к различным сущностям: архитектура сети, процесс обучения, функционал потерь, поведение отдельных слоев и отдельных нейронов. В данном разделе мы рассмотрим наиболее распространенные на сегодняшний день подходы, а в последующих разде-

лах мы обсудим соответствующее программное обеспечение и ряд обзорных статей, более подробно затрагивающих определенные группы методов.

2.1. Максимизация активации

Разберем постановку задачи и методы, связанные с обнаружением и изучением стимулов, активирующих конкретные единичные нейроны или их малые группы [13]. Под стимулами в приложениях компьютерного зрения чаще всего понимаются изображения, а соответствующая задача ставится как поиск такого из них, которое бы максимизировало реакцию анализируемого нейрона, и известна в литературе под общим названием *максимизация активации* [14] (Activation Maximization, далее AM). Отметим, что в последнее время методы, решающие задачу AM, стали активно применяться в задаче визуализации ИНС.

В случае изучения зрения живых существ стимулом является видимое изображение, а активацией – электрический сигнал, снимаемый с нейрона в мозге. Одной из первых соответствующих работ биологов была работа, в которой было обнаружено, что отдельный нейрон в мозге кошки сильнее всего реагирует на изображение наклонных линий [15]. Подобный подход может быть применен также и для ИНС.

Простейшим способом проанализировать, на что реагирует тот или иной нейрон, является поиск изображения (например, из тренировочной выборки), которое максимизирует функцию активации исследуемого нейрона. Однако в таком подходе кроется ряд недостатков. Во-первых, в этом случае требуется произвести поиск по всей обучающей выборке для всех интересующих исследователя нейронов, что ведет к значительным затратам вычислительных ресурсов. Во-вторых, выборка может не содержать изображение, которое бы максимизировало выход исследуемого нейрона, так как пространство изображений обычно сильно больше размера выборки. И, наоборот, нейрон может активироваться совершенно разными изображениями приблизительно одинаково сильно, что усложняет интерпретируемость. Обычно в таких случаях рассматривают 9 наиболее сильных стимулов, но и они могут быть разрознены. И, наконец, для случая настоящих изображений, часто возникает неоднозначность – какие именно визуальные особенности заставляют нейрон реагировать. Например, если нейрон активируется изображением птицы на ветке дерева, то не понятно, важна ли здесь птица или ветка.

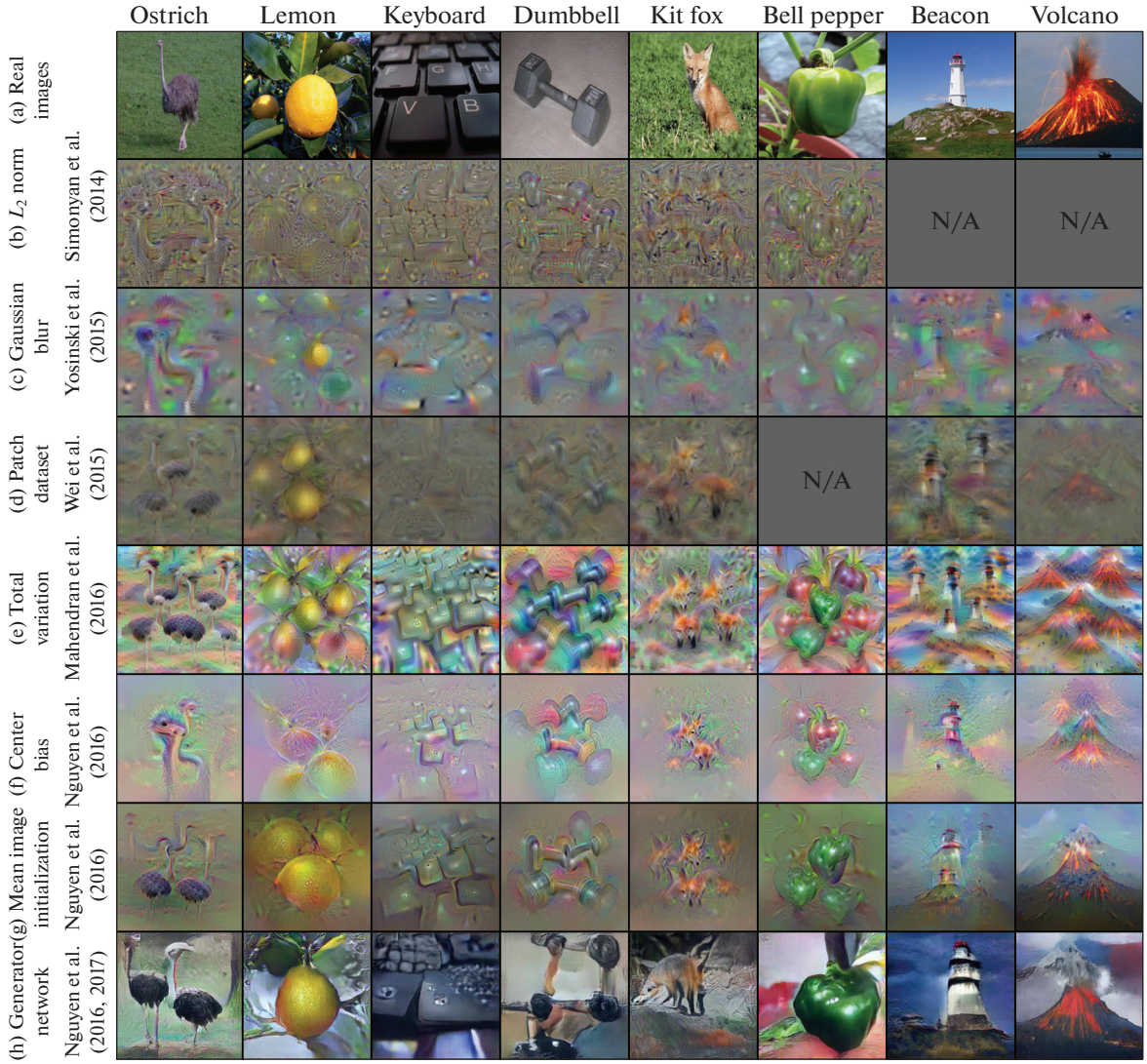
Для компенсации части обозначенных выше недостатков возможен синтез изображений. Такой подход позволяет отделять стимулы друг от друга и обладает большей репрезентативностью. Так, можно реконструировать стимулы без необходимости доступа к обучающей выборке целевой модели, которая может быть недоступна на практике, также можно контролировать количество объектов и их вид на изображении, например, создавать изображения только птиц или только веток. Отметим, что современные методы в основном используют именно синтетические изображения.

Рассмотрим задачу классификации изображений. Пусть Θ – параметры классификатора, отображающего картинку из C каналов размера $H \times W$ точек $x \in \mathbb{R}^{H \times W \times C}$, в распределение вероятностей по выходным классам. Тогда мы можем сформулировать задачу максимизации активации нейрона с индексом i из слоя l как задачу нахождения входного изображения x, которое максимизирует функцию активации $a_i^l(\Theta, x)$. Таким образом, оптимизационная задача запишется следующим образом:

$$x^* = \arg\max_x a_i^l(\Theta, x). \tag{1}$$

Поставленную задачу назовем задачей *максимизации активации* (AM) или *визуализации признаков* (Feature Visualization). Подобная постановка задачи (1) впервые была предложена в работе [14] и в дальнейшем широко использовалась для визуализации ИНС [17]. Однако ее прямое решение зачастую ведет к неинтерпретируемым входным изображениям, состоящим преимущественно из высокочастотного шума [18], поэтому исследователями были предложены модификации – добавление регуляризации и ограничений в задачу для получения более интерпретируемых результатов.

2.1.1. Регуляризация в задаче максимизации активации. Для того, чтобы решить задачу в пространстве “реалистичных изображений”, могут применяться регуляризация по L^2 норме [18], размытие с гауссовым ядром [19], создание датасета с кусочками реальных изображений (patch dataset) [20], ограничения на вариацию функции (total variation, TV) [21], использование цен-



Фиг. 1. Примеры изображений, максимизирующих активации соответствующих классов, изображения взяты из обзора методов AM [16].

трального смещения (center bias) [22], инициализация средним изображением (mean image initialization) [22] и другие.

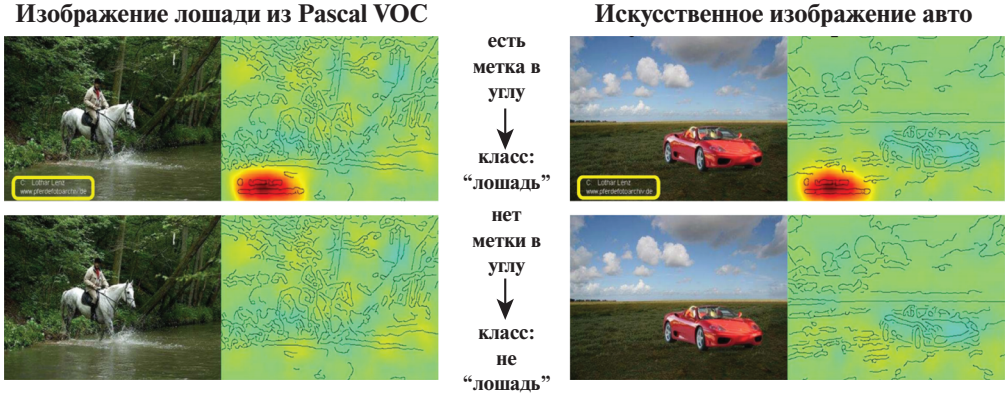
Таким образом, после добавления регуляризационного члена $R(x)$ задача максимизации активации имеет вид

$$x^* = \arg\max_x (a(x) - R(x)). \tag{2}$$

2.1.2. Генеративные сети в задаче максимизации активации. Генеративные сети [23] позволяют создавать изображения из переменных латентного пространства (Latent Space). Эти изображения можно использовать для поиска реалистичных изображений, максимизирующих активацию. Такой метод рассмотрен в работе [17], где генератор G ищет код в скрытом подпространстве $h \in R^{4096}$ такой, что изображение $G(h)$ максимизирует активацию ИНС $a(G(h))$. Соответственно задача AM из формы, записанной в уравнении 2, преобразуется в следующую:

$$h^* = \arg\max_h (a(G(h)) - R(h)). \tag{3}$$

Отметим, что подход с использованием генеративных сетей действительно позволяет во многих случаях находить реалистичные входные изображения. Найденные решения задачи AM для ряда методов представлены на фиг. 1.



Фиг. 2. Пример атрибуции тепловых карт из работы SpRAy [24].

2.1.3. Области применения метода максимизации активации. Метод AM допускает широкий спектр возможных приложений в области DL и EDL (см., например, [16]), включая следующие:

- визуализация выходных параметров для новых задач;
- визуализация внутренних параметров ИНС;
- синтезирование изображений, активирующих несколько нейронов;
- наблюдение эволюции нейрона во время обучения;
- синтезирование видео;
- использование максимизации активации как инструмента отладки;
- синтезирование изображений на основе описания;
- синтезирование изображений на основе маски семантической сегментации;
- синтезирование preferred stimuli для реального, биологического мозга.

2.2. Атрибуция

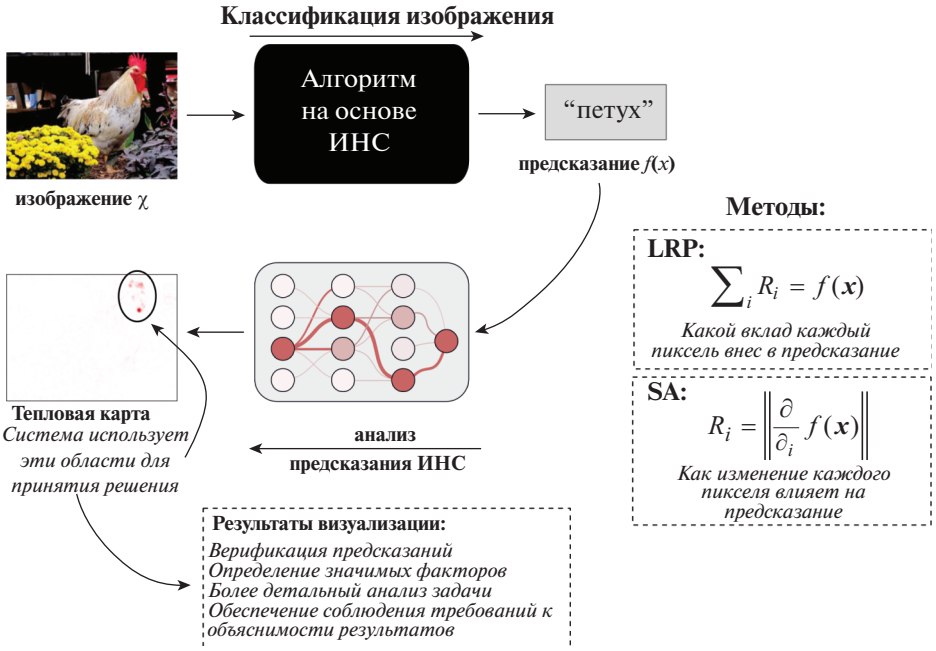
Атрибуция (Attribution, Heatmapping, Spectral Relevance Analysis) изучает, какая часть входного тензора ИНС, чаще всего какая область изображения, отвечает за активацию определенного нейрона сети. Так, например, для задачи детектирования объектов под атрибуцией обычно понимается построение тепловой карты вклада в детектирование данного объекта каждой точки на входном изображении (см. пример на фиг. 2).

Построение подобных тепловых карт проливает свет на процесс принятия решения алгоритмом компьютерного зрения. Так, в работе SpRAy [24] приводятся примеры, когда ИНС на наборе данных PASCAL VOC [25] обучилась находить подпись внизу картинки. По стечению обстоятельств подписи присутствовали на многих картинках с лошадьми, в итоге подобные примеры в литературе обрели название “Умный Ганс” (Clever Hans) в честь знаменитой лошади начала прошлого века.

Классическим методом построения тепловой карты атрибуции является аддитивный метод – Layer-wise Relevance Propagation (далее LRP) [26]. Пусть $x = (x_1, x_2, \ldots, x_d)$- входной вектор, а как $f(x)$ – выход ИНС. Тогда метод LRP заключается в нахождении такого вектора $R = (R_1, R_2, \ldots, R_d)$, имеющего такую же длину, как и входной вектор x, что верно следующее:

$$\sum_{p=1}^{d} R_p = f(x).$$

Схожий вид визуализации ИНС – анализ чувствительности (Sensitivity Analysis) [27], [28]. В рамках данного метода предполагается решение задачи сопоставления входному изображению тепловой карты, отображающей для каждого пикселя меру того, насколько изменится выход заданного нейрона при изменении значения в данной точке изображения. В отличие от методов анализа чувствительности, в методах атрибуции и разобранном выше методе LRP нас интересует само значение выхода, а не его локальное изменение (см. рис. 3 из работы [27]). Пусть индексы i



Фиг. 3. Описание процесса анализа и визуализации работы ANN на основе метода атрибуции LRP и на основе метода анализа чувствительности. Видно различие в данных двух задачах. Основано на статье [27].

и j кодируют номер нейронов на двух последовательных слоях, тогда $a_j = \sigma\left(\sum_i (a_i w_{ij} + b)\right)$ – активация на j слое. Здесь $\sum_i$ означает суммирование по всем нейронам i-го слоя, а $\sum_j$ – суммирование по всем нейронам j-го слоя. В этом случае метод распространения ошибки (propagation of LRP) можно записать в виде

$$R_i = \sum_j \frac{z_{ij}}{\sum_i z_{ij}} R_j, \tag{4}$$

где z_{ij} – вклад нейрона i в активацию a_j. Чаще всего этот вклад зависит от активации a_i и веса w_{ij}. Последовательно применяя данный метод обратного распространения ошибки, начиная с выхода ИНС и продолжая до входного изображения, можно получить веса для каждой компоненты входного ветора-картинки и отобразить результат в виде тепловой карты. Пример такой тепловой карты можно видеть на фиг. 3. Как можно видеть, подобная визуализация понятна и легко поддается анализу.

Кратко отметим также другие алгоритмы, решающие задачу атрибуции.

• DeepLIFT [29] – в рамках данного метода сравнивается активация каждого нейрона с некоторым “референсным” значением и присваивается значение “вклада” в активацию исследуемого выхода на основе их разницы.

• Guided Backpropagation (Guided BackProp) [30] – метод, основанный на методе обратного распространения ошибки (backpropagation) с той разницей, что отрицательные значения заменяются на 0, а также не используется информация от нейронов с отрицательным выходным значением.

• Integrated Gradients [31] – метод, основанный на интегрировании всех градиентов (из стандартного метода обратного распространения ошибки) вдоль “пути” от входного значения до выходного.

• Smooth Grad [32] – данный подход представляет модернизацию методов атрибуции, основанных на подсчете градиента, при помощи добавления сглаживания.

• Class Activation Mapping (CAM) [33] – метод, основанный на применении слоев глобального среднего (global average pooling, GAP) в сверточных нейронных сетях (Convolutional Neural Networks, далее CNN) для построения тепловой карты значимости пикселей входного изображения.

• Gradient-Weighted Class Activation Mapping (Grad-CAM) [34] – метод, основанный на идее CAM с добавлением информации о градиенте для потока информации, связанного с активацией нейрона предсказанного класса.

• Score-Weighted Class Activation Mapping (Score-CAM) [35] – модернизация метода Grad-CAM – вместо информации о градиентах используется специально введенная мера "увеличения уверенности", наподобие той, что используется в DeepLIFT.

• SHAPley Additive exPlanations (SHAP) [36] – метод, расширяющий идею аддитивной атрибуции признаков при помощи теоретико-игрового анализа. Может применяться для анализа важности признаков в широком спектре ML задач.

• Saliency Map (SM) [18] – метод, вычисляющий локальную чувствительность на основе частных производных. Позволяет, например, оценивать, для каких входных пикселей возмущения влияют на изменение финальной категории для изображения. Данный метод применим для достаточно общих типов архитектур нейронных сетей с дифференцируемыми входами.

• Deconvolution (DeConv) [37] – метод, основанный на построении CNN g с выходами в виде другой CNN f. Сеть g конструируется так, чтобы "обратить" операции, выполняемые исходной сетью f. Например, для операции свертки применяются транспонированные версии исходных фильтров с некоторыми оговорками. При этом деконволюционная сеть использует ReLU в качестве активационной функции каждый раз, тем самым приравнивая к нулю все возникающие отрицательные значения.

2.3. Визуализация функционала потерь

В наиболее общей форме ИНС может быть представлена как векторная функция $f(x,\theta)$, где x – это входной вектор, θ – набор параметров сети, а значение функции f – это соответствующее предсказание ИНС. Подстройка параметров θ производится на обучающем наборе данных (x_i, y_i), $i = 1,2,\ldots,N$, посредством минимизации функционала потерь:

$$L(\theta) = \frac{1}{N}\sum_{i=1}^{N} l(x_i, y_i, \theta),$$

где функционал потерь отдельного обучающего примера может, например, иметь простейшую квадратичную форму:

$$l(\theta) = \|f(x,\theta) - y\|^2.$$

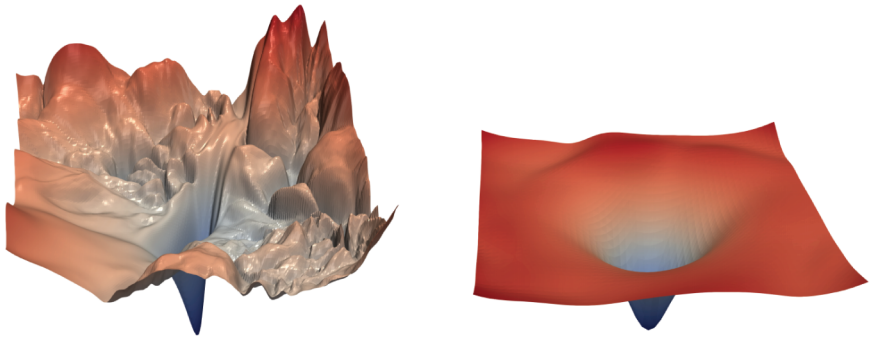
В общем случае функционал потерь является невыпуклой функцией, зависящей от огромного числа переменных (параметров ИНС), а его минимизация представляет сложнейшую вычислительную задачу [38]. Классический стохастический метод градиентного спуска позволяет с минимальными вычислительными затратами осуществлять итерационный процесс поиска локального минимума функционала потерь. Принципиальную важность здесь представляет форма функционала потерь, определяемая выбором гиперпараметров ИНС. Чем более гладким оказывается функционал потерь, тем быстрее будет происходить обучение и тем лучше окажется обобщающая способность ИНС.

Перспективным средством анализа профиля функционала потерь может быть визуализация. Удачная визуализация позволяет оценить общие характерные черты выбранного функционала, а также их изменение при соответствующей модификации гиперпараметров сети (см. пример на фиг. 4). Однако здесь возникают сложности, характерные для задачи представления существенно многомерных функций на одно- и двумерных графиках.

В литературе наибольшее распространение получили два способа визуализации функционала потерь – графики одномерной линейной интерполяции [40], [41] и контурные графики по случайным направлениям [42], [43]. В первом случае выбираются два различающихся значения вектора параметров ИНС: θ и θ', между которыми производится интерполяция функционала потерь по линейной формуле:

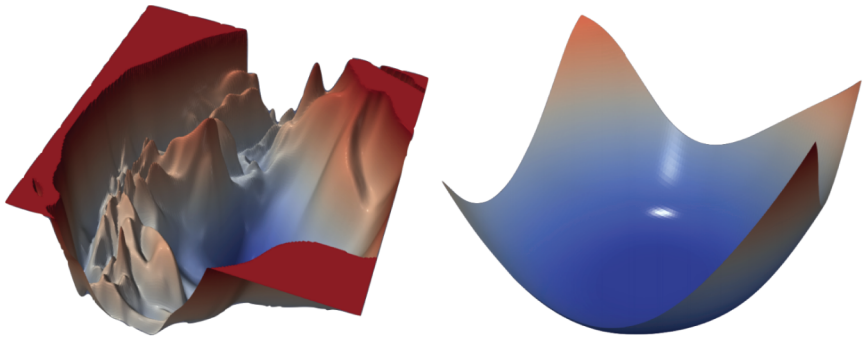
$$\phi(\alpha) = L(\theta(\alpha)), \quad \theta(\alpha) = (1-\alpha)\theta + \alpha\theta', \quad 0 \le \alpha \le 1,$$

(а) без skip connections (б) со skip connections



Фиг. 4. Профиль функционала потерь для ИНС типа ResNet-56, построенный в работе [39] с использованием оригинальной схемы нормализации.

(а) ResNet-110 без skip connections (б) DenseNet, 121 слой



Фиг. 5. Профиль функционала потерь для ИНС типа ResNet-110 и DenseNet на наборе данных CIFAR-10, построенный в работе [39] с использованием оригинальной схемы нормализации.

с последующим построением графика одномерной функции $\phi(\alpha)$. Во втором случае выбираются некоторая центральная точка θ^* и два вектора направлений δ и η, а затем строится одномерная функция:

$$\phi(\alpha) = L(\theta^* + \alpha\delta),$$

либо двумерная функция:

$$\phi(\alpha, \beta) = L(\theta^* + \alpha\delta + \beta\eta).$$

На фиг. 4 и 5 приводятся результаты по визуализации функционала потерь в окрестности обнаруженного при обучении локального минимума, полученные в работе [39] на основе обобщения метода контурных графиков. Как можно видеть, данный метод представляет эффективный инструмент для оценки итогового качества обучения ИНС и сравнения эффекта от выбора гиперпараметров и различных эвристик обучения.

3. ПРОГРАММНЫЕ РЕАЛИЗАЦИИ МЕТОДОВ ВИЗУАЛИЗАЦИИ

Рассмотрим наиболее популярные программные реализации алгоритмов визуализации ИНС. Для выявления релевантных программных продуктов авторами осуществлялись различные формы предметных запросов в поисковой системе Google, а также проводился поиск непосредственно по тегам репозиториев на Github, при этом в результаты поиска мы не добавляли программные продукты, с момента последнего обновления которых прошло более 12 мес. В итоге нами было отобрано 16 программных решений (библиотек), которые представлены в табл. 1. Для каждой библиотеки мы указываем поддерживаемый фреймворк ML (в рамках библиотеки осуществляется визуализация моделей, построенных с использованием только соответствующего

Таблица 1. Программные продукты для визуализации ИНС

Название программы	Фреймворк ML	Звезды Github	Последнее обновление
OpenAI Microscope	–	закрытый код	веб-интерфейс
SHAP	TensorFlow, PyTorch	11000	2020, ноябрь
Playground-TensorFlow	TensorFlow	9500	2020, апрель
TensorboardX	PyTorch	6700	2020, июль
Tensorboard	TensorFlow	5100	2020, ноябрь
PyTorch-CNN-visualizations	PyTorch	4900	2020, сентябрь
Lucid	TensorFlow	4000	2020, ноябрь
Keras-vis	TensorFlow (Keras)	2800	2020, апрель
Captum	PyTorch	1900	2020, ноябрь
PyTorch-grad-cam	PyTorch	1700	2020, апрель
Hiddenlayer	TensorFlow, PyTorch	1300	2020, апрель
TF-explain	TensorFlow	740	2020, июль
INNvestigate	TensorFlow	726	2020, октябрь
Saliency	TensorFlow	600	2020, октябрь
FlashTorch	PyTorch	507	2020, май
TCAV	TensorFlow	414	2020, июль

фреймворка; при этом, как можно видеть из полученных результатов, выбор в итоге осуществляется между двумя популярными фреймворками – TensorFlow [44] и PyTorch [45]), количество звезд на Guthub и дату (год и месяц) последнего обновления репозитория проекта. Для простоты и компактности мы приводим округленное количество звезд на Guthub в качестве меры интереса научного сообщества к соответствующему программному продукту. Безусловно, для более точной оценки необходим также учет даты создания репозитория, отдельное ранжирование программных продуктов, использующих различные фреймворки ML и т.д. Однако подобное рассмотрение выходит за рамки задач данного обзора. Отметим, что последняя актуализация данных осуществлялась нами в начале декабря 2020 г.

Программный продукт OpenAI Microscope (https://openai.com/blog/microscope) позволяет осуществлять визуализацию (признаков отдельных слоев и набора обучающих данных) для восьми популярных в задачах машинного зрения архитектур CNN (AlexNet, AlexNet (Places), Inception v1, Inception v1 (Places), VGG 19, Inception v3, Inception v4 и ResNet v2 50) на специализированном интерактивном веб-сайте. Данный продукт имеет закрытый исходный код (известно лишь, что визуализации подготавливались с использованием библиотеки Lucid, которая будет рассмотрена ниже) и, на наш взгляд, может быть использован лишь в учебных или иллюстративных целях.

Популярная библиотека SHAP (https://github.com/slundberg/SHAP) (SHAPley Additive exPlanations) реализует универсальный подход [36], основанный на теоретико-игровом анализе и расширяющий идею аддитивной атрибуции признаков, для интерпретации и последующей визуализации широкого класса моделей ML. Помимо оригинального алгоритма SHAP, в данной библиотеке реализованы также методы DeepLIFT и Smooth-Grad.

Интерактивная браузерная среда Playground-Tensor Flow (https://github.com/tensorflow/playground) предполагает использование в образовательных целях для визуализации процесса построения и обучения ИНС. Пользователь имеет возможность выбрать ряд гиперпараметров полносвязной сети (количество слоев, количество нейронов в слое, тип функции активации и параметры регуляризации), тип и степень зашумленности набора данных, а также используемые входные признаки из ограниченного набора возможных вариантов. После запуска процесса обучения сети производится интерактивная демонстрация эволюции весов связей нейронов и точности предсказания.

Библиотека Tensorboard (https://github.com/tensorflow/Tensorboard) для TensorFlow и ее адаптация TensorboardX (https://github.com/lanpa/TensorboardX) для PyTorch представляют популярный инструмент визуализации архитектуры ИНС и процесса обучения в режиме реального времени.

Библиотека PyTorch-CNN-visualizations (https://github.com/utkuozbulak/PyTorch-CNN-visualizations) предназначена для визуализации глубоких CNN, построенных и обученных с использованием фреймворка PyTorch. В рамках данной библиотеки реализован широкий спектр алгоритмов EDL, включая SM, CAM, Grad-CAM, Score-CAM, Guided BackProp, DeConv, Deep-dream, Smooth-Grad и др.

Библиотека Lucid (https://github.com/tensorflow/Lucid) содержит обширную подборку EDL методов, включая различные методы визуализации признаков, сетки активации, метод пространственной атрибуции и др. Отметим, что для работы библиотеки необходим фреймворк TensorFlow, причем поддержка современной 2-й версии пока что отсутствует.

Отметим также библиотеку Keras-vis (https://github.com/raghakot/Keras-vis), которая предоставляет набор инструментов для визуализации сверточных и полносвязных слоев ИНС на основе методов AM, SM и CAM.

Активно развивающаяся библиотека Captum (https://github.com/pytorch/Captum) реализует широкий набор методов атрибуции для интерпретации нейросетевых моделей, построенных с PyTorch, включая такие методы, как SM, DLIFT, SHAP, Grad-CAM, Guided BackProp, DeConv, Integrated gradients и многие другие. Отметим, что возможна также установка интерактивного браузерного интерфейса “Captum Insights” для визуализации результатов.

Библиотека PyTorch-grad-cam (https://github.com/jacobgil/PyTorch-grad-cam) реализует метод атрибуции Grad-CAM для моделей, созданных с PyTorch. Отметим, что существует версия данной библиотеки для TensorFlow, однако она несколько лет не обновлялась и в этой связи не включена в наш перечень.

Библиотека Hiddenlayer (https://github.com/waleedka/Hiddenlayer) реализует функционал, близкий к продукту Tensorboard и предоставляет набор инструментов для визуализации как графа ИНС с возможностью кастомизации, так и процесса ее обучения, включая эволюцию функционала стоимости, весов и активаций нейронов слоев сети и т.п. Особо отметим, что данная библиотека может использоваться и для моделей, обученных с TensorFlow, и с PyTorch.

Библиотека TF-explain (https://github.com/sicara/TF-explain) реализует ряд методов визуализации для моделей, построенных с использованием TensorFlow, включая Grad-CAM, Integrated gradients, Smooth-Grad, а также ряд базовых методов визуализации активаций и градиентов.

Библиотека iNNvestigate (https://github.com/albermax/innvestigate) содержит реализации ряда методов визуализации ИНС, включая Smooth-Grad, Guided BackProp, DeConv, LRP и Integrated gradients, DLIFT. Для работы библиотеки необходим фреймворк TensorFlow первой версии.

Библиотека Saliency (https://github.com/PAIR-code/saliency) реализует множество методов из класса SM, включая Smooth-Grad, Guided BackProp, Integrated gradients, Grad-CAM и XRAI [46] и осуществляет визуализацию карт значимости для моделей, обученных с использованием TensorFlow.

Библиотека FlashTorch (https://github.com/MisaOgura/FlashTorch) позволяет осуществлять визуализацию ИНС, созданных с использованием фреймворка PyTorch, посредством методов AM и SM. Отметим, что данная небольшая библиотека имеет простой понятный интерфейс и подробные инструкции в различных форматах (текст, видео, демонстрационные примеры).

Специализированная библиотека TCAV (https://github.com/tensorflow/TCAV) (Testing with Concept Activation Vectors, см. также работу [47]) позволяет выявить наиболее важные сложные признаки (не значения отдельных пикселей, ацвет, пол, раса и т.п.), влияющие на выбор класса при предсказании ИНС. Данная библиотека работает с уже обученными TensorFlow моделями и требует для своего функционирования набор примеров, демонстрирующих сложные признаки. В качестве вывода отображается график, иллюстрирующий степень важности выбранных сложных признаков для рассматриваемого варианта предсказания сети (например, на сколько “полосатость” и “зигзагообразность” влияют на выбор сетью класса “зебра”).

4. АНАЛИЗ СУЩЕСТВУЮЩИХ ОБЗОРОВ

На сегодняшний день в литературе представлен ограниченный набор обзоров, затрагивающих в той или иной степени рассматриваемую нами тематику визуализации ИНС. В табл. 2 приведены выявленные авторами релевантные обзорные работы, а ниже кратко приведено обсуждение каждой из этих работ с указанием характерных особенностей и степени соответствия теме.

Таблица 2. Обзоры по теме визуализации ИНС

Ссылка	Год	Заголовок	Журнал/Книга
[48]	2017	Towards better analysis of machine learning models: A visual analytics perspective	Visual Informatics
[49]	2017	Visualizations of deep neural networks in computer vision: A survey	Transparent Data Mining for Big and Small Data
[50]	2018	A user-based taxonomy for deep learning visualization	Visual Informatics
[51]	2018	Visual interpretability for deep learning: a survey	Frontiers of Informat. Technology & Electronic Engineering
[52]	2018	How convolutional neural network see the world – A survey of convolutional neural network visualization methods	ArXiv Preprint
[53]	2018	Visual analytics for explainable deep learning	IEEE Computer Graphics and Applications
[54]	2018	Visual analytics in deep learning: An interrogative survey for the next frontiers	IEEE Transactions on Visualization and Computer Graphics
[55]	2018	A task-and-technique centered survey on visual analytics for deep learning model engineering	Computers & Graphics
[16]	2019	Understanding neural networks via feature visualization: A survey	Explainable AI: Interpreting, Explaining and Visualizing Deep Learning
[56]	2020	A survey of visual analytics techniques for machine learning	Computational Visual Media
[57]	2020	A survey of surveys on the use of visualization for interpreting machine learning models	Informat. Visualization

В статье [48] приводится краткий обзор методов и перспектив визуализации нейросетевых структур. Данная работа не может рассматриваться как полноценный обзор, однако она представляет определенный интерес для исследователей и в этой связи включена в перечень.

В работе [49] впервые, на наш взгляд, предлагается системный подход к задаче визуализации ИНС, формулируется собственная многоуровневая схема классификации, включающая цель и метод визуализации, архитектуру ИНС и область ее применения в ML, а также набор данных, на котором обучалась ИНС. Для каждого из этих пяти критериев вводится набор возможных значений, что позволяет авторам эффективно типизировать все отобранные научные работы, а также провести соответствующий сравнительный анализ популярности различных направлений и т.п. Данная работа безусловно обладает высокой научной ценностью, однако представленные в ней методы визуализации и перечень обсуждаемых публикаций на сегодняшний день являются не вполне актуальными.

В обзоре [50] формулируется система классификации методов визуализации ИНС с точки зрения конечного заинтересованного лица (“начинающие”, “практики”, “разработчики”, “эксперты”). В зависимости от типа заинтересованного лица уточняется специализация инструмента визуализации и далее в рамках этой специализации осуществляется краткое обсуждение нескольких практических реализаций подобных систем. В данной работе содержится описание ряда практически значимых программных продуктов, однако малый объем работы и отсутствие в ней описания алгоритмов и методов визуализации ИНС не позволяют рассматривать ее как полноценный масштабный обзор.

В обзоре [51] рассматриваются общие вопросы EDL в контексте CNN и обсуждаются различные методы для интерпретации нейросетевых моделей, включая их представление в форме графов и решающих деревьев. Таким образом, данный обзор не вполне соответствует рассматриваемой нами тематике, хотя и содержит ряд важных положений по общей задаче построения интерпретируемых моделей.

В обзоре [52], посвященном визуализации непосредственно CNN, формулируется ряд методов, включая AM, обращение сети (Network Inversion), деконволюционные нейронные сети (DeConv) и метод “рассечения” сети (Network Dissection). Для каждого метода обсуждается структура, алгоритм, соответствующие операции и результаты из научных публикаций, при этом

особый акцент делается на EDL. Основной вывод, к которому приходят на основе проведенного анализа – это иерархический характер организации CNN (каждый последующий слой отвечает за распознавание все более сложных признаков), имеющий определенную аналогию с механизмом действия зрительной коры человека. Отметим, что данная работа отличается высоким уровнем описания математических постановок методов визуализации. Однако рассмотрение в этом обзоре сосредоточено на одном типе нейросетевых архитектур, также с момента публикации данного обзора появился ряд новых релевантных публикаций.

Работа [53] может лишь условно рассматриваться как обзорная. В данной публикации обсуждаются общие аспекты EDL и особая роль визуализации в этой глобальной задаче. В работе приводится перечень основных задач EDL и визуализации ИНС, а также демонстрируются некоторые программные реализации систем визуализации ИНС. Таким образом, в данной публикации отражен ряд важных задач и прикладных направлений визуализации, однако отсутствует детальный обзор методов визуализации и соответствующих результатов.

В обзоре [54] предлагается оригинальный подход к классификации работ по теме визуализации ИНС, в основе которого лежит идея интегрального рассмотрения связанных вопросов: *почему* проводится визуализация (*why*); *кто* хочет осуществлять визуализацию (*who*); *что* визуализируется (*what*); *как* реализуется визуализация (*how*); *когда*, т.е. на каком этапе работы модели ML, производится визуализация (*when*); *где* используется визуализация (*where*). Для каждого из представленных вопросов в работе предлагаются варианты ответов с соответствующими подробными комментариями. Построенная таким образом система позволила осуществить аккуратную визуальную классификацию анализируемых ими научных работ. На наш взгляд, данный обзор является одним из наиболее обширных и в то же время глубоких из представленных в табл. 2. Однако предложенная система классификации в определенных аспектах представляется несколько искусственной (в отличие, например, от классификации, предложенной в работе [49]). Также данный обзор не охватывает новые научные результаты, полученные за 2019 и 2020 г.

В обзоре [55] для формализации задачи классификации методов и публикаций по визуализации ИНС вводится система из трех категорий, соответствующих цели визуализации: понимание нейросетевой архитектуры; улучшение процесса тренировки сети; выявление значимых входных признаков. На наш взгляд, данная классификация не является универсальной, однако в этом обзоре обсуждается ряд практических реализаций систем визуализации.

В работе [16] рассматриваются возможные реализации и применения одного конкретного метода визуализации ИНС – метода AM, строятся различные формулировки данного метода и описывается ряд современных работ, в которых используется метод AM для визуализации ИНС. Отметим, что в данной работе также приводятся рассуждения, вскрывающие связь максимизации активации с задачей анализа активности головного мозга в нейронауках. Поскольку авторы сосредоточились на обсуждении лишь одного метода, то данная работа не может рассматриваться как полноценный обзор по методам визуализации ИНС.

В работе [56] выполнен обширный обзор публикаций по теме методов визуализации в ML. Для организации анализируемых научных работ предложена классификация по этапу процесса подготовки и обучения модели ML (соответствует вопросу *when* в контексте рассмотренного выше обзора [54]) и далее по цели визуализации (соответствует вопросу *why* в обзоре [54]). Соответственно рассматривается визуализация: до построения модели (для улучшения качества исходных данных или качества входных признаков); в процессе построения модели (для понимания модели или для диагностики модели, или для “ручного управления” моделью); после построения модели (для понимания статического или динамического распределения результатов работы модели). Авторы приводят кривые тренда публикационной активности по соответствующим направлениям классификации, согласно которым наибольший рост соответствует направлениям диагностики и “ручного управления” моделью, при этом направление, связанное с пониманием модели, характеризуется восходящим по годам трендом с незначительным снижением активности в 2020 г. Проведенный в работе анализ публикаций позволяет судить об общих направлениях развития направления визуализации в области ML, однако в этой работе не приводятся описания алгоритмов и технические подробности реализации соответствующих подходов и систем визуализации.

Особо отметим работу [57], позиционируемую как “обзор обзоров” в области визуализации для EDL. Авторы данной работы детально описывают использованную ими интеллектуальную стратегию поиска научных работ, в результате которой был построен список из 18 публикаций, являющихся обзорами. Затем авторы проводят анализ близости и релевантности отобранных ра-

бот в контексте совпадения внешних ссылок и др., на основе которого производится их последующее краткое обсуждение. Отметим, что в данном "метаобзоре" предложены интересная методология поиска и оценки близости научных работ, однако в нем отсутствует подробное обсуждение содержания работ и методов. Также отобранные авторами научные обзоры соответствуют не только лишь теме визуализации ИНС, но и более общим областям, связанным с предиктивной аналитикой и визуализацией больших объемов данных.

5. ЗАКЛЮЧЕНИЕ

В работе приведен обзор современных методов визуализации искусственных нейронных сетей, включающий методы максимизации активации, атрибуции и визуализации функционала потерь. Кроме непосредственного обзора ключевых алгоритмов для каждой подзадачи визуализации, в работе построен подробный перечень релевантных программных пакетов с практическими реализациями алгоритмов и рассмотрены уже представленные в литературе обзорные работы.

Как следует из проведенного анализа алгоритмов, программного обеспечения и обзорных работ, научное направление, связанное с визуализацией искусственных нейронных сетей, на сегодняшний день является актуальным и бурно развивающимся. При этом существует ряд потенциальных новых приложений данной методологии в современных задачах по исследованию естественных нейронных сетей и формированию в них памяти и специализации нейронов.

СПИСОК ЛИТЕРАТУРЫ

1. *LeCun Y., Bengio Y., Hinton G.* Deep learning // Nature. 2015. V. 521. Issue 7553. P. 436–444.
2. *Shahid N., Rappon T., Berta W.* Applications of artificial neural networks in health care organizational decision-making: A scoping review // PLoS ONE. 2019. V. 14. Issue 2.
3. *Nassif A.B., Shahin I., Attili I., Azzeh M., Shaalan K.* Speech recognition using deep neural networks: A systematic review // IEEE Access. 2019. V. 7. P. 19143–19165.
4. *Alkinani H.H., Al-Hameedi A.T.T., Dunn-Norman S., Flori R.E., Alsaba M.T., Amer A.S.* Applications of artificial neural networks in the petroleun industry: A review // SPE Middle East Oil and Gas Show and Conference, Society of Petroleum Engineers. 2019.
5. *Tjoa E., Guan C.* A survey on explainable artificial intelligence (xai): Toward medical xai // Proc. of the IEEE Transactions on Neural Networks and Learning Systems. 2020.
6. *Xu F., Uszkoreit H., Du Y., Fan W., Zhao D., Zhu J.* Explainable ai: A brief survey on history, research areas, approaches and challenges // CCF Internat. Conference on Natural Language Proc. and Chinese Comput. 2019. P. 563–574.
7. *Samek W., Montavon G., Vedaldi A., Hansen L.K., Müller K.-R.* Explainable AI: interpreting, explaining and visualizing deep learning // Nature. 2019. V. 11700.
8. *Lipton Z.C.* The mythos of model interpretability // Queue. 2018. V. 16. Issue 3. P. 31–57.
9. *Cowan N.* The many faces of working memory and short-term storage // Psychonomic Bulletin Review. 2017. V. 24. Issue 4. P. 1158–1170.
10. *Anokhin K., Ivashkina O., Toropova K., Gruzdeva A., Rogozhnikova O.B., Plushnin V., Fedotov I.* Neuronal encoding of object-type and object-place memories in hippocampus and neocortex of young and old mice // The FASEB Journal. 2020. V. 34. Issue S1. P. 1–1.
11. *Zhigulina P., Ushakov V., Kartashov S., Malakhov D., Orlov V., Novikov K., Korotkova A., Anokhin K., Nourkova V.* The architecture of neural networks for enhanced autobiographical memory access: a functional mri study // Procedia Computer Science. 2020. V. 169. P. 787–794.
12. *Tiunova A.A., Komissarova N.V., Anokhin K.V.* Mapping the neural substrates of recent and remote visual imprinting memory in the chick brain // Frontiers in Physiology. 2019. V. 10. P. 351–351.
13. *Marshel J.H., Kim Y.S., Machado T.A., Quirin S., Benson B., Kadmon J., Raja C., Chibukhchyan A., Ramakrishnan C., Inoue M.* Cortical layer-specific critical dynamics triggering perception // Science. 2019. V. 365. Issue 6453.
14. *Erhan D., Bengio Y., Courville A., Vincent P.* Visualizing higher-layer features of a deep network, Technical Report // ICML 2009 Workshop on Learning Feature Hierarchies, Montréal, Canada. 2009.
15. *Hubel D., Wiesel T.* Receptive fiedls of single neurones in the cat's striate cortex // J. of Physiology. 1959. V. 148.
16. *Nguyen A., Yosinski J., Clune J.* Understanding neural networks via feature visualization: A survey // Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. 2019. P. 55–76.
17. *Nguyen A., Dosovitskiy A., Yosinski J., Brox T., Clune J.* Synthesizing the preferred inputs for neurons in neural networks via deep generator networks // Advances in Neural Informat. Processing Systems. 2016. P. 3395–3403.

18. *Simonyan K., Vedaldi A., Zisserman A.* Deep inside convolutional networks: Visualising image classification models and saliency maps // Workshop at Internat. Conference on Learning Representations. 2014.
19. *Yosinski J., Clune J., Nguyen A., Fuchs T., Lipson H.* Understanding Neural Networks Through Deep Visualization // Deep Learning workshop at ICML 2015. 2015.
20. *Wei D., Zhou B., Torrabla A., Freeman W.* Understanding Intra-Class Knowledge Inside CNN // arXiv:1507.02379, 2015, url: https://arxiv.org/abs/1507.02379
21. *Mahendran A., Vedaldi A.* Visualizing deep convolutional neural networks using natural pre-images // Internat. Journal of Computer Vision. 2016. V. 120. Issue 3. P. 233–255.
22. *Nguyen A., Yosinski J., Clune J.* Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned By Each Neuron in Deep Neural Networks // arXiv:1602.03616, 2016, url: https://arxiv.org/abs/1602.03616
23. *Goodfellow I.J., Pouget-Abadie J., Mirza M., Xu B., Warde-Farley D., Ozair S., Courville A., Bengio Y.* Generative adversarial nets // Proc. of the 27th Internat. Conference on Neural Informat. Proc. Systems. 2014. V. 2. P. 2672–2680.
24. *Lapuschkin S., Wäldchen S., Binder A., Montavon G., Samek W., Müller K.-R.* Unmasking clever hans predictors and assessing what machines really learn // Nature Communications. 2019. V. 10. Issue 3.
25. *Everingham M., Eslami S.M.A., Van Gool L., Williams C.K.I., Winn J., Zisserman A.* The pascal visual object classes challenge: A retrospective // Internat. Journal of Computer Vision. 2015. V. 111. Issue 1. P. 98–136.
26. *Lapuschkin S., Binder A., Montavon G., Klauschen F., Müller K.-R., Samek W.* On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation // PLoS ONE. 2015. V. 10.
27. *Samek W., Wiegand T., Müller K.-R.* Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models // ITU Journal: ICT Discoveries. 2019. V. 1. P. 39–48.
28. *Samek W., Binder A., Montavon G., Lapushckin S., Müller K.-R.* Evaluating the visualization of what a deep neural network has learned // IEEE Transactions on Neural Networks and Learning Systems. 2017. V. 28. Issue 11. P. 2660–2673.
29. *Shrikumar A., Greenside P., Kundaje A.* Learning important features through propagating activation differences // Proc. of the 34th Internat. Conference on Machine Learning, PLMR. 2017. P. 3145–3153.
30. *Springenberg J.T., Dosovitskiy A., Brox T., Riedmiller R.* Striving for simplicity: The all convolutional net // arXiv:1412.6806, 2014, url: https://arxiv.org/abs/1412.6806
31. *Sundararajan M., Taly A., Yan Q.* Axiomatic attribution for deep networks // Proc. of the Internat. Conference on Machine Learning, ICML. 2017. P. 3319–3328.
32. *Smilkov D., Thorat N., Kim B., Viégas F., Wattenberg M.* Smoothgrad: removing noise by adding noise // Workshop on Visualization for Deep Learning, ICML. 2017.
33. *Zhou B., Khosla A., Lapedriza A., Oliva A., Torralba A.* Learning deep features for discriminative localization // Proc. of the IEEE Conference on Computer Vision and Pattern Recognition. 2016. P. 2921–2929.
34. *Selvaraju R.R., Cogswell M., Das A., Vedantam R., Parikh D., Batra D.* Grad-cam: Visual explanations from deep networks via gradient-based localization // Proc. of the IEEE Internat. Conference on Computer Vision. 2017. P. 618–626.
35. *Wang H., Wang Z., Du M., Yang F., Zhang Z., Ding S., Mardziel P., Hu X.* Score-cam: Score-weighted visual explanations for convolutional neural networks // Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2020. P. 24–25.
36. *Lundberg S.M., Lee S.-I.* A unified approach to interpreting model predictions // Advances in Neural Informat. Processing Systems. 2017. P. 4765–4774.
37. *Zeiler M.D., Fergus R.* Visualizing and understanding convolutional networks // European Conference on Computer Vision. Springer. 2014. P. 818–833.
38. *Choromanska A., Henaff M., Mathieu M., Arous G.B., LeCun Y.* The loss surfaces of multilayer networks // Artificial Intelligence and Statistics. 2015. P. 192–204.
39. *Li H., Xu Z., Taylor G., Studer C., Goldstein T.* Visualizing the loss landscape of neural nets // Advances in Neural Informat. Processing Systems. 2018. P. 6389–6399.
40. *Dinh L., Pascanu R., Bengio S., Bengio Y.* Sharp minima can generalize for deep nets // Proc. of the 34th Internat. Conference on Machine Learning, PMLR. 2017. P. 1019–1028.
41. *Keskar N.S., Mudigere D., Nocedal J., Smelyanskiy M., Tang P.T.P.* On large-batch training for deep learning: Generalization gap and sharp minima // 5th Internat. Conference on Learning Representations, ICLR. 2017.
42. *Goodfellow I.J., Vinyals O., Saxe A.M.* Qualitatively characterizing neural network optimization problems // Internat. Conference on Learning Representations. 2015.
43. *Im D.J., Tao M., Branson K.* An empirical analysis of deep network loss surfaces // arXiv:1612.04010, 2016, https://arxiv.org/abs/1612.04010
44. *Abadi M., Agarwal A., Barham P., Brevdo E., Chen Z., Citro C., Corrado G.S., Davis A., Dean J., Devin M., Ghemawat S., Goodfellow I., Harp A., Irving G., Isard M., Jia Y., Jozefowicz R., Kaiser L., Kudlur M., Levenberg J., Mané D., Monga R., Moore S., Murray D., Olah C., Schuster M., Shlens J., Steiner B., Sutskever I., Talwar K., Tucker P., Vanhoucke V., Vasudevan V., Viégas F., Vinyals O., Warden P., Wattenberg M., Wicke M., Yu Y., Zheng X.*

TensorFlow: Large-scale machine learning on heterogeneous systems // arXiv:1603.04467, 2016, url: https://arxiv.org/abs/1603.04467

45. *Paszke A., Gross S., Massa F., Lerer A., Bradbury J., Chanan G., Killeen T., Lin Z., Gimelshein N., Antiga L., Desmaison A., Kopf A., Yang E., DeVito Z., Raison M., Tejani A., Chilamkurthy S., Steiner B., Fang L., Bai J., Chintala S.* Pytorch: An imperative style, high-performance deep learning library // Advances in Neural Informat. Processing Systems. 2019. V. 32. P. 8024–8035.
46. *Kapishnikov A., Bolukbasi T., Viégas F., Terry M.* Xrai: Better attributions through regions // Proc. of the IEEE Internat. Conference on Computer Vision. 2019. P. 4948–4957.
47. *Kim B., Wattenberg M., Gilmer J., Cai C., Wexler J., Viegas F., et al.* Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV) // Internat. conference on machine learning, PMLR. 2018. P. 2668–2677.
48. *Liu S., Wang X., Liu M., Zhu J.* Towards better analysis of machine learning models: A visual analytics perspective // Visual Informatics. 2017. V. 1. Issue 1. P. 48–56.
49. *Seifert C., Aamir A., Balagopalan A., Jain D., Sharma A., Grottel S., Gumhold S.* Visualizations of deep neural networks in computer vision: A survey // Transparent Data Mining for Big and Small Data. 2017. P. 123–144.
50. *Yu R., Shi L.* A user-based taxonomy for deep learning visualization // Visual Informatics. 2018. V. 2. Issue 3. P. 147–154.
51. *Zhang Q.-S., Zhu S.-C.* Visual interpretability for deep learning: a survey // Frontiers of Informat. Technology Electronic Engineering. 2018. V. 19. Issue 1. P. 27–39.
52. *Qin Z., Yu F., Liu C., Chen X.* How convolutional neural network see the world-a survey of convolutional neural network visualization methods // Mathematical Foundations of Computing. 2018. V. 1. Issue 2. P. 149–180.
53. *Choo J., Liu S.* Visual analytics for explainable deep learning // IEEE computer graphics and applications. 2018. V. 38. Issue 4. P. 84–92.
54. *Hohman F., Kahng M., Pienta R., Chau D.H.* Visual analytics in deep learning: An interrogative survey for the next frontiers // IEEE transactions on visualization and computer graphics. 2018. V. 25. Issue 8. P. 2674–2693.
55. *Garcia R., Telea A.C., da Silva B.C., Tørresen J., Comba J.L.D.* A task-and-techniquecentered survey on visual analytics for deep learning model engineering // Computers Graphics. 2018. V. 77. P. 30–49.
56. *Yuan J., Chen C., Yang W., Liu M., Xia J., Liu S.* A survey of visual analytics techniques for machine learning // Computational Visual Media. 2020.
57. *Chatzimparmpas A., Martins R.M., Jusufi I., Kerren A.* A survey of surveys on the use of visualization for interpreting machine learning models // Informat. Visualization. 2020. P. 6572–6583.