УДК 51 **Физико-математические науки**

Кристина Сергеевна Качагина, *магистратрант*
Оренбургский государственный педагогический университет, г. Оренбург
Сафарова А. Д., *научный руководитель, канд. пед. наук, доц.*
Оренбургский государственный педагогический университет, г. Оренбург

НЕЙРОННЫЕ СЕТИ - ПЕРСПЕКТИВЫ РАЗВИТИЯ

Аннотация: В данной статье рассматривается проблема, проведенная в целях распространения использования нейросетей, показания их перспектив и уязвимостей, которые следует исправить. Данное направление IT довольно многообещающее и может использоваться для автоматизации практически всех процессов производства, что делает его практически универсальным инструментом. Конечно, пока это недешевое удовольствие, но в процессе развития и расширения этой области, снизится и цена на подобный агрегат. Доказательством может служить ситуация с мобильными телефонами, которые при появлении стоили чуть больше зарплаты среднестатистического гражданина, а теперь позволить себе их может каждый.

Ключевые слова: нейронные сети, математика нейросетей, применение нейросетей.

Annotation: This article discusses the problem conducted in order to spread the use of neural networks, indications of their prospects and vulnerabilities that should be fixed. This area of IT is quite promising and can be used to automate almost all production processes, which makes it an almost universal tool. Of course, while this is not cheap, but in the process of development and expansion of this area, the price of such a unit will also decrease. The proof can be the situation with mobile phones, which when they appeared cost a little more than the average citizen's salary, and now everyone can afford them.

Keywords: neural networks, mathematics of neural networks, application of neural networks.

В данной статье я попытаюсь осветить тему нейросетей с точки зрения человека непосвященного, простым языком, без использования завуалированных определений, а не «массив нейронов образует перцептрон, работающий по известной, зарекомендовавшей себя схеме».

Так для чего же нужны нейросети?

Нейросеть – это обучаемая система нейронов, построенная на основе нервных сетей, простирающихся по всему телу внутри каждого из нас. Она действует не только в соответствии с заданным алгоритмом и формулами, но и на основании прошлого опыта. То есть, она запоминает свои предыдущие действия и после их анализа учитывает прошлые ошибки, повышая точность выполнения команд.

Чтобы ничего не усложнять, договоримся, что нейрон – это просто воображаемый чёрный объект, обладающий некоторым количеством входных отверстий и одним выходным. Как выходной сигнал формируется из кучи входных – определяет внутренний алгоритм нейрона, прописанный пользователем.

Нейросети появились относительно недавно, но тем не менее, уже широко используются в различных сферах нашей жизни. В охранных организациях их внедряют в камеры видеонаблюдения и сканирующие аппараты. Нейросеть обрабатывает область мира, попадающую в объектив камеры/аппарата и выдает пользователю требующиеся данные, начиная от внешности людей, попавших в поле видимости, заканчивая номерами машин, проезжающих мимо. На заводах нейросети управляют работой тех или иных механизмов в огромных масштабах, что позволяет уменьшить количество рабочей силы, а значит – сэкономить деньги.

Однако, не все так хорошо, как кажется. Внедрение подобных технологий часто несет за собой массовые недовольства со стороны населения по

понятным причинам. Нейросеть неидеальна, а это значит, что в ней имеется ряд уязвимостей, который будет использоваться для ее выведения из строя. Одной из таких причин является погрешность измерений/неточность вычислений, которой любая нейросеть обладает. В данной работе я бы хотел разобраться в причинах и следствиях этой проблемы и найти способ ее решения [2].

Делать это я буду на примере ImageRes 50v2 - одной из самых передовых сетей для классификации изображений, натренированной на датасете (наборе определенных данных,в нашем случае – картинок) ImageNet.

Нейросети предлагается картинка, и сеть должна определить, что на ней находится посредством сканирования и сравнения с другими подобными.

Обратимся к примеру, написанному на языке pythonv3(на который прежде были поставлены библиотеки с GitHub):

Рисунок 1. Персидский кот

```
importnumpyas np
fromkeras.preprocessingimport image
fromkeras.applicationsimport resnet_v2
## Загружаемпредобученнуюмодель
```

```
model= resnet_v2.ResNet50V2(include_top=True,
weights='imagenet',
input_shape=(224,224,3))
## Загружаемизображение
img=image.load_img("kitten.jpg#26759185",target_size=(224,224))
input_image=image.img_to_array(img,'channels_last')
## Перевод изображения из формата [0; 255] в [-1; 1]
input_image=(input_image/255-0.5)*2
## Делаем из изображения массив с изображением (batch)
input_image=np.expand_dims(input_image, axis=0)
## Прогонимчерезнейроннуюсеть
predictions=model.predict(input_image)
## Переведем ответ нейронной сети (вектор) в категорию
predicted_classes= resnet_v2.decode_predictions(predictions, top=1)
imagenet_id, name, confidence =predicted_classes[0][0]
print("Яна {:.4}% уверен, чтоэто - {}!".format(confidence *100, name))
```

В результате программа скажет нам, что это кот.

```
$ python3 test.py
UsingTensorFlow backend.
Яна99.51%уверен,чтоэто-Persian_cat!
```

Высокая точность вычислений обусловлена заранее загруженными библиотеками и высоким качеством изображения. Подобную нейросеть можно вывести из строя просто понизив качество изображения, и так как все нейронные сети — это большие математические функции, чтобы найти ложное изображение, разумно использовать их же.

Так что наша задача сводится к оптимизации методом градиентного спуска.

Математика такой атаки до неприличия проста: мы выворачиваем процесс обучения нейронной сети наизнанку. Вместо фиксированных входных данных (тренировочного датасета) и обучающейся сети тут мы имеем меняющиеся, «обучающиеся» входные данные и фиксированную сеть.

Как и для обучения нейронной сети, нам нужно два параметра: функция потерь (способ подсчитать ошибку) и градиент (мы используем производную нейронной сети) [2].

И градиент (вектор, указывающий на точку наибольшего возрастания функции), и ошибку мы можем посчитать, используя алгоритм обратного распространения (метод вычисления градиента), просто приняв все веса нейронной сети правильными (и, соответственно, неизменными), а вход — ошибочным и подлежащим исправлению.

Виды атак на нейросеть:

Поиски ошибочно распознаваемого примера можно поделить на два разных вида:

- ненаправленную атаку, когда ищется любой подходящий пример (необязательно имеющий смысл),
- специально направленную атаку, цель которой — создать минимально измененный ошибочный пример, глядя на который человек распознает объект без проблем, а программа — нет.

Самый большой минус ненаправленной атаки — это полное отсутствие у результата какой-либо смысловой нагрузки для человека. Но преимущество такой атаки — возможность легко применять в реальном мире. Например, через видеокамеры устройств IoT или систем безопасности: достаточно распечатать результат на бумажке и поднести к объективу [3].

В заключение хотелось бы сказать, что данная работа была проведена в целях распространения использования нейросетей, показания их перспектив и уязвимостей, которые следует исправить. Данное направление IT довольно многообещающее и может использоваться для автоматизации практически всех процессов производства, что делает его практически универсальным

инструментом. Конечно, пока это недешевое удовольствие, но в процессе развития и расширения этой области, снизится и цена на подобный агрегат. Доказательством может служить ситуация с мобильными телефонами, которые при появлении стоили чуть больше зарплаты среднестатистического гражданина, а теперь позволить себе их может каждый.

Библиографический список:

1. Безопасность, разработка DevOps: сайт. – URL: http://www.xakep.ru (дата обращения: 29.04.2020). – Текст: электронный.

2. Википедия-свободная энциклопедия: сайт. – URL: http://www.Wikipedia.org (дата обращения: 29.04.2020). – Текст: электронный.

3. Онлайн - школа: сайт. – URL: http://www.skillfactory.ru (дата обращения: 29.04.2020). – Текст: электронный.