МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «КубГУ»)

Факультет компьютерных технологий и прикладной математики
Кафедра анализа данных и искусственного интеллекта

КУРСОВАЯ РАБОТА

СОВРЕМЕННЫЕ НЕЙРОСЕТЕВЫЕ ТЕХНОЛОГИИ РАСПОЗНАВАНИЯ ОБРАЗОВ

Работу выполнил _____ М.И. Ковалев

(подпись)

Направление подготовки 09.03.03 Прикладная информатика

Направленность Прикладная информатика в экономике

Научный руководитель

д-р. техн. наук, зав. каф. _____ А.В. Коваленко

(подпись)

Нормоконтролер

канд. физ.-мат. наук, доц. _____ Г.В. Калайдина

(подпись)

Краснодар
2022

РЕФЕРАТ

Курсовая работа 26 с., 17 рис., 8 источников.

НЕЙРОННАЯ СЕТЬ, РАСПОЗНОВАНИЕ ОБРАЗОВ, МНОГОСЛОЙНЫЙ ПЕРЦЕПТРОН, КЛАССИФИКАЦИЯ, KERAS, TENSORFLOW

Объектом исследования данной курсовой работы являются нейронные сети распознавания образов.

Цель работы: изучение современных нейросетевых технологии распознавания образов, а также разработка собственной нейросетевой модели.

В результате выпускной квалификационной работы были изложены основные понятия, связанные с данной темой и рассмотрены различные нейронные сети. Были изучены основные принципы работы и создана нейронная сеть.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Распознавание визуальных образов – одна из самых популярных задач, в решении которых используются нейросетевые технологии. Сегодня создаются сети, в которых машины способны успешно распознавать символы на бумаге и банковских картах, подписи на официальных документах, детектировать объекты и т.д. Эти функции позволяют существенно облегчить труд человека, а также повысить надежность и точность различных рабочих процессов за счет отсутствия возможности допущения ошибки из-за человеческого фактора.

Актуальность нейронных сетей обусловлена широким применением подобных технологий и, вследствие этого, большой популярностью на рынке программ, распознающих образы.

Целью курсовой работы является изучение современных нейросетевых технологии распознавания образов, а также разработка собственной нейросетевой модели.

Данная цель предопределила решение следующих задач:

– изучение различных нейронных сетей, правил их функционирования,

– теоретическое исследование алгоритмов обучения нейронных сетей,

– построение нейронной сети «WardrobeDet», которая сможет распознать различные детали гардероба.

Курсовая работа состоит из трех глав, введения, заключения, списка использованной литературы, содержащего 8 наименований.

В первой главе подробно описаны основные понятия нейронных сетей, виды нейронных сетей, обучение нейронных сетей.

Во второй главе описаны основные программные средства для разработки нейронной сети «WardrobeDet», дана краткая характеристика языка программирования Python и описаны основные библиотеки.

В третей главе описано создание нейронной сети «WardrobeDet» и её тестирование по загруженному пользователем изображению.

В заключение подведен основной итог работы.

1 Нейронные сети

Нейронная сеть — это метод в искусственном интеллекте, который учит компьютеры обрабатывать данные таким же способом, как и человеческий мозг. Это тип процесса машинного обучения, называемый глубоким обучением, который использует взаимосвязанные узлы или нейроны в слоистой структуре, напоминающей человеческий мозг. Он создает адаптивную систему, с помощью которой компьютеры учатся на своих ошибках и постоянно совершенствуются. Таким образом, искусственные нейронные сети пытаются решать сложные задачи, такие как резюмирование документов или распознавание лиц, с более высокой точностью [1].

Основные принципы работы нейронных сетей были описаны еще в 1943 году У. Мак-Каллоком и У. Питтсом. В 1957 году нейрофизиолог Ф. Розенблатт разработал первую нейронную сеть, а в 2010 году большие объемы данных для обучения открыли возможность использовать нейронные сети для машинного обучения.

На данный момент нейронные сети используются в многочисленных областях машинного обучения и решают проблемы различной сложности.

Работа нейронной сети сравнима с действиями человека: сталкиваясь с незнакомым предметом, он узнает его свойства и делает выводы. Аналогичные процессы происходят в узлах нейронных сетей, когда, решая определенную задачу, они используют полученный опыт для дальнейшего обучения.

Подобная естественному аналогу искусственная нейросеть состоит из нейронов и синапсов.

Нейрон — это единица, которая получает информацию и производит над ней определенные вычисления. Он является простейшей структурной единицей любой нейросети. Как правило, нейроны упорядочиваются в слои, которые в конечном счете формируют сеть [2].

Все нейроны работают примерно одинаково. Однако существуют некоторые частные случаи нейронов, выполняющих специфические функции.

Основные типы слоев нейронов:

– входной (input) — слой нейронов, получающий информацию,

– скрытый (hidden) — некоторое количество слоев, обрабатывающих информацию,

– выходной (output) — слой нейронов, представляющий результаты вычислений,

Синапс — это связь, соединяющая выход одного нейрона со входом другого. Проходящий через него сигнал может усиливаться или ослабевать.

Параметром синапса является вес — коэффициент, из-за которого передаваемая информация из одного нейрона другому может изменяться.

Рисунок 1 схематично показывает расположение слоев, которые соединены связями.

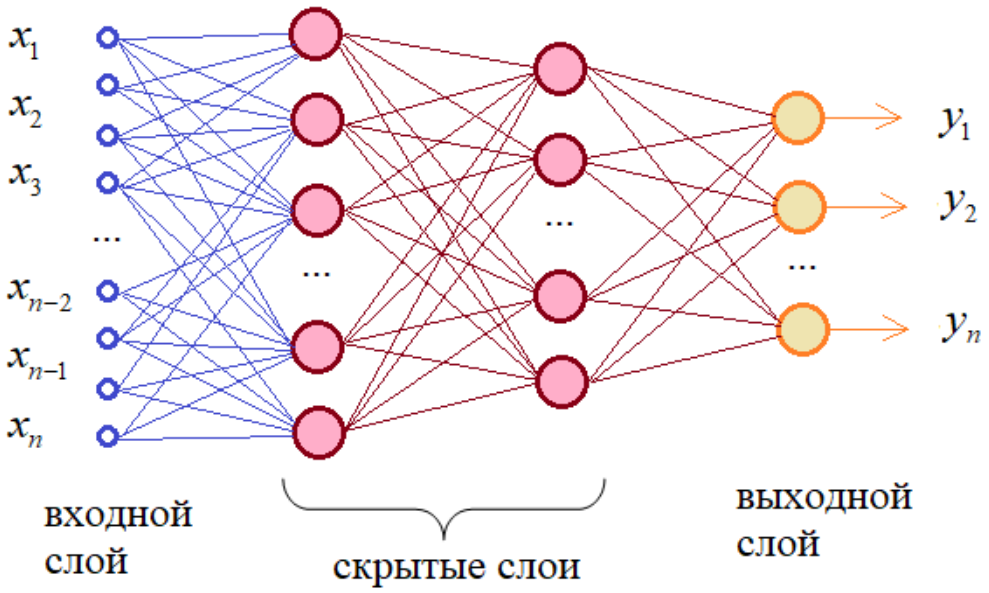

Рисунок 1. Схема слоев нейронной сети [3].

На входной слой нейронов поступает некая информация, которая по синапсам переходит на следующий слой. При этом каждый синапс обладает

собственным коэффициентным весом, а любой следующий нейрон в новом слое может иметь несколько входов. Информация передается дальше до тех пор, пока не дойдет до конечного выхода.

Например, алгоритм распознавания рукописного текста должен иметь возможность справляться с огромным разнообразием способов представления данных. Каждую цифру от 0 до 9 можно записать множеством способов: размер и точная форма каждого символа могут сильно отличаться в зависимости от того, кто пишет и в каких обстоятельствах [2].

Входному слою подаются значения, представляющие пиксели, которые составляют изображение рукописной цифры. Выходной слой, в свою очередь, предсказывает, какой символ изображен на картинке.

Модель узнает, какие связи между нейронами важны для успешного прогнозирования во время обучения. На каждом этапе тренировки сеть использует математическую функцию, чтобы определить, насколько точным был ее последний прогноз по сравнению с ожидаемым.

Эта функция генерирует серию значений ошибок, которые могут использоваться системой для расчета того, как модель должна обновлять значение весов, прикрепленных к каждой ссылке, с конечной целью повышения точности прогнозов сети.

В течение многих тренировочных циклов и с помощью периодической ручной настройки параметров сеть будет продолжать генерировать все более точные прогнозы, пока не достигнет максимального значения. На этом этапе, например, когда рукописные цифры можно распознать с точностью более 95%, можно сказать, что нейросеть обучена.

1.1 Виды нейронных сетей

Однослойная нейронная сеть — сеть, в которой сигналы от входного слоя сразу подаются на выходной слой, который и преобразует сигнал и сразу же выдает ответ [4].

Как видно из схемы (рисунок 2) однослойной нейронной сети, представленной справа, сигналы поступают на входной слой (который не считается за слой нейронной сети), а затем сигналы распределяются на выходной слой обычных нейронов. На каждом ребре от нейрона входного слоя к нейрону выходного слоя написано число – вес соответствующей связи.

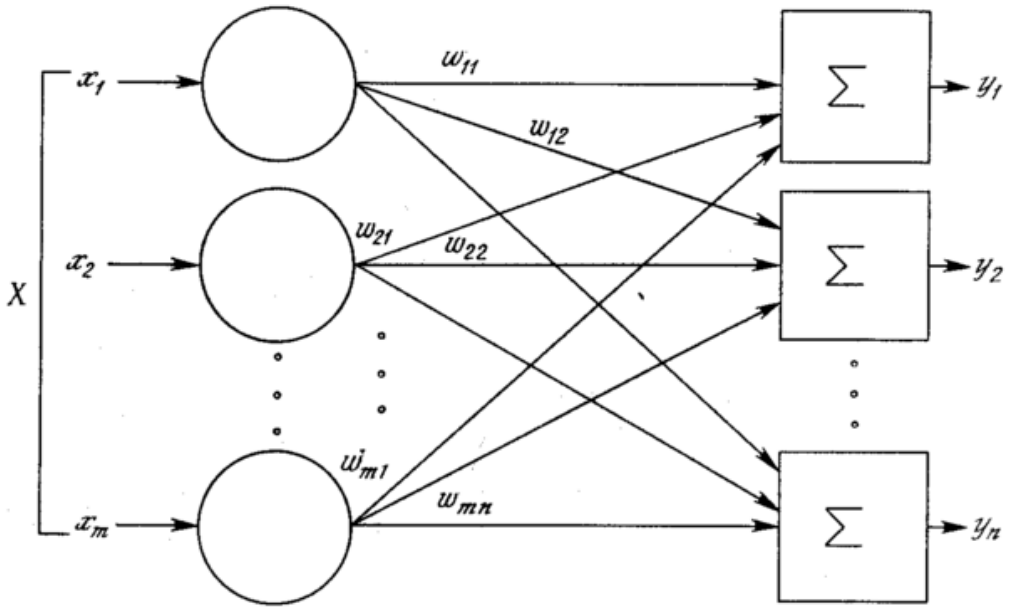

Рисунок 2. Схема однослойной нейронной сети [4].

Многослойная нейронная сеть — нейронная сеть, состоящая из входного, выходного и расположенного между ними одного (нескольких) скрытых слоев нейронов (рисунок 3) [4].

Помимо входного и выходного слоев эти нейронные сети содержат промежуточные, скрытые слои. Такие сети обладают гораздо большими

возможностями, чем однослойные нейронные сети, однако методы обучения нейронов скрытого слоя были разработаны относительно недавно.

Работу скрытых слоев нейронов можно сравнить с работой большого завода. Продукт (выходной сигнал) на заводе собирается по стадиям на станках. После каждого станка получается какой-то промежуточный результат. Скрытые слои тоже преобразуют входные сигналы в некоторые промежуточные результаты.

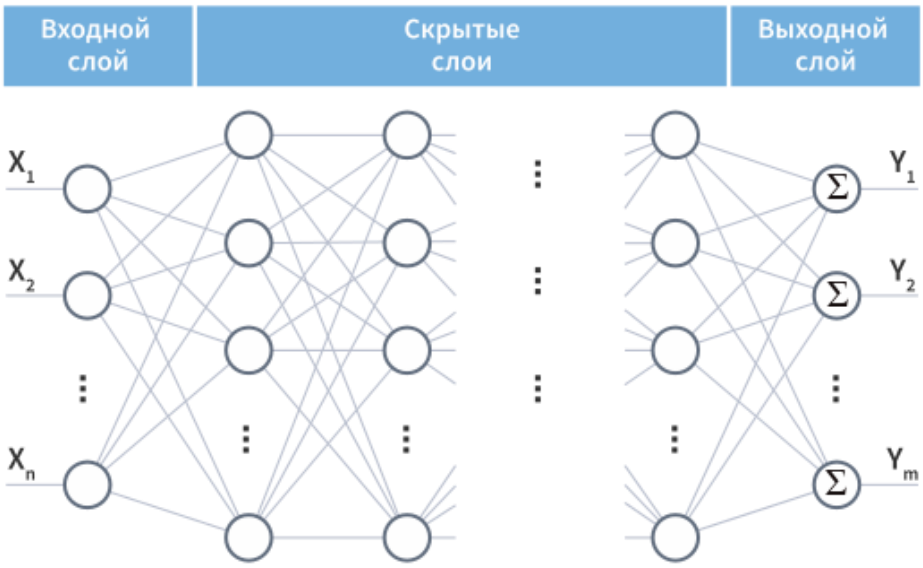

Рисунок 3. Схема многослойной нейронной сети [4].

Сети прямого распространения — искусственные нейронные сети, в которых сигнал распространяется строго от входного слоя к выходному. В обратном направлении сигнал не распространяется.

Все сети, описанные выше, являлись сетями прямого распространения, как следует из определения. Такие сети широко используются и вполне успешно решают определенный класс задач: прогнозирование, кластеризация и распознавание.

Однако сигнал в нейронных сетях может идти и в обратную сторону.

Сети с обратными связями — искусственные нейронные сети, в которых выход нейрона может вновь подаваться на его вход, как показано на рисунке 4. В более общем случае это означает возможность распространения сигнала от выходов к входам.

В сетях прямого распространения выход сети определяется входным сигналом и весовыми коэффициентами при искусственных нейронах. В сетях с обратными связями выходы нейронов могут возвращаться на входы. Это означает, что выход какого-нибудь нейрона определяется не только его весами и входным сигналом, но еще и предыдущими выходами (так как они снова вернулись на входы).

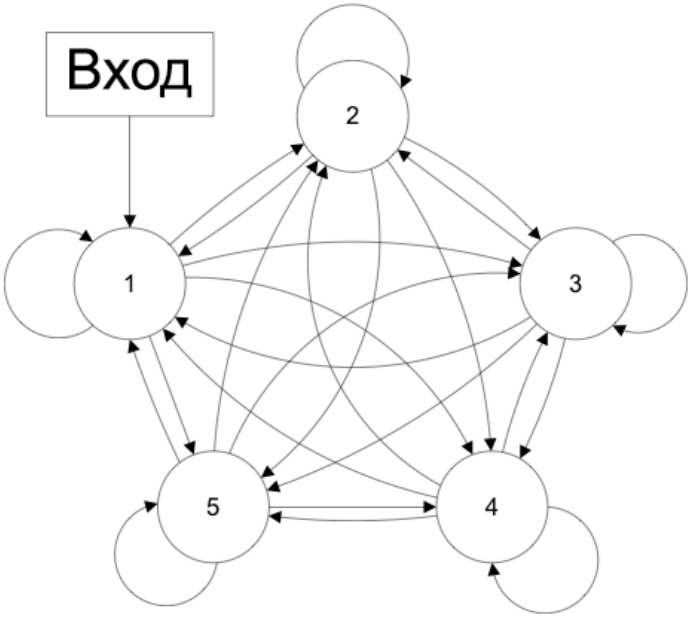

Рисунок 4. Схема сети с обратными связями [4].

1.2 Обучение нейронной сети

Распознавание изображений с помощью нейронных сетей возможно только посредством специального обучения, представляющего собой процесс, направленный на настройку параметров НС [5].

Есть несколько способов обучить нейросеть:

1) Машинное обучение с учителем. При обучении НС для распознавания образов с учителем имеется выборка с истинными ответами на вопрос, что изображено на картинке – метками классов. Нейросети подаются на вход эти изображения, после чего вычисляется ошибка, сравнивающая выходные значения с истинными метками классов. В зависимости от степени и характера несоответствия предсказания НС, её веса корректируются, ответы НС подстраиваются под истинные ответы, пока ошибка не станет минимальной.

2) Обучение без учителя. В этом случае у обучающей выборки нет меток классов, и перед НС стоит задача найти заранее не известные ответы. Нейронная сеть пытается самостоятельно найти закономерности в данных, извлекая полезные признаки и анализируя их. Например, кластеризация — наиболее распространенная задача для обучения без учителя. Алгоритм подбирает похожие данные, находя общие признаки, и группируют их вместе. В обучении без учителя сложно вычислить точность алгоритма, так как в данных отсутствуют «правильные ответы» или метки. Но размеченные данные бывает сложно или слишком дорого получить. В таких случаях, предоставляя модели свободу действий для поиска зависимостей, можно получить определённый результат.

3) Обучение с частичным привлечением учителя. Обучающая выборка содержит как размеченные, так и неразмеченные данные. Этот метод особенно полезен, когда разметить все объекты – трудоемкая задача. Тем не менее, нейронная сеть может извлечь информацию из небольшой доли размеченных данных и улучшить точность предсказаний по сравнению с моделью, обучающейся исключительно на неразмеченных данных.

4) Обучение с подкреплением. Обучение с подкреплением (reinforcement learning) действует по принципу получения обратной связи - награды за определённые действия.

2 Технологии разработки

2.1 Язык программирования Python

Python – один из самых популярных языков программирования, с помощью которого можно решать самые разные задачи. Именно поэтому он так распространен и для создания нейронных сетей [6].

Язык позволяет разрабатывать сложные алгоритмы за короткое время. Его отличают простота, лаконичность и выразительность. Помимо этого, он позволяет производить быстрые вычисления.

Нейронные сети – преимущественно небольшие программы, но при этом существует необходимость часто изменять их, подбирая наилучшую архитектуру, предобработку данных и другие параметры. Именно на Python есть библиотеки для более простого написания нейронных сетей.

2.2 Основные библиотеки

Keras – это библиотека для языка программирования Python, которая предназначена для глубокого машинного обучения. Она позволяет быстрее создавать и настраивать модели – схемы, по которым распространяется и подсчитывается информация при обучении. Но сложных математических вычислений Keras не выполняет и используется как надстройка над другими библиотеками [7].

Keras с версии 2.3 – это надстройка над библиотекой TensorFlow, которая нужна для машинного обучения. TensorFlow выполняет все низкоуровневые вычисления и преобразования и служит своеобразным движком, математическим ядром. Keras же управляет моделями, по которым проходят вычисления.

До версии 2.3 Keras мог использовать в качестве движка вычислительные различные библиотеки. Но в новых версиях поддержка прекратилась, теперь библиотека работает только с TensorFlow.

Keras создавалась как гибкая модульная библиотека, которую легко настраивать и модифицировать. Она бесплатная, у нее открытый исходный код, который может посмотреть любой человек.

Keras имеет узкую специализацию. Это инструмент для специалистов по машинному обучению, которые работают с языком Python: именно его чаще всего используют благодаря удобству математических вычислений. Keras применяют разработчики, которые создают, настраивают и тестируют системы машинного обучения и искусственного интеллекта, в первую очередь нейронные сети.

2.3 Используемый DataSet

Для создания и обучения сети «WardrobeDet» будет использован набор данных, который называется Fashion MNIST. Он содержит 70 тыс. изображений следующих деталей гардероба: футболки, брюки, туфли, сумки, свитера, пальто, рубашки, кроссовки, сандали и батильоны. Пример изображений, входящий в DataSet Fashion MNIST представлен на рисунке 5 [8].

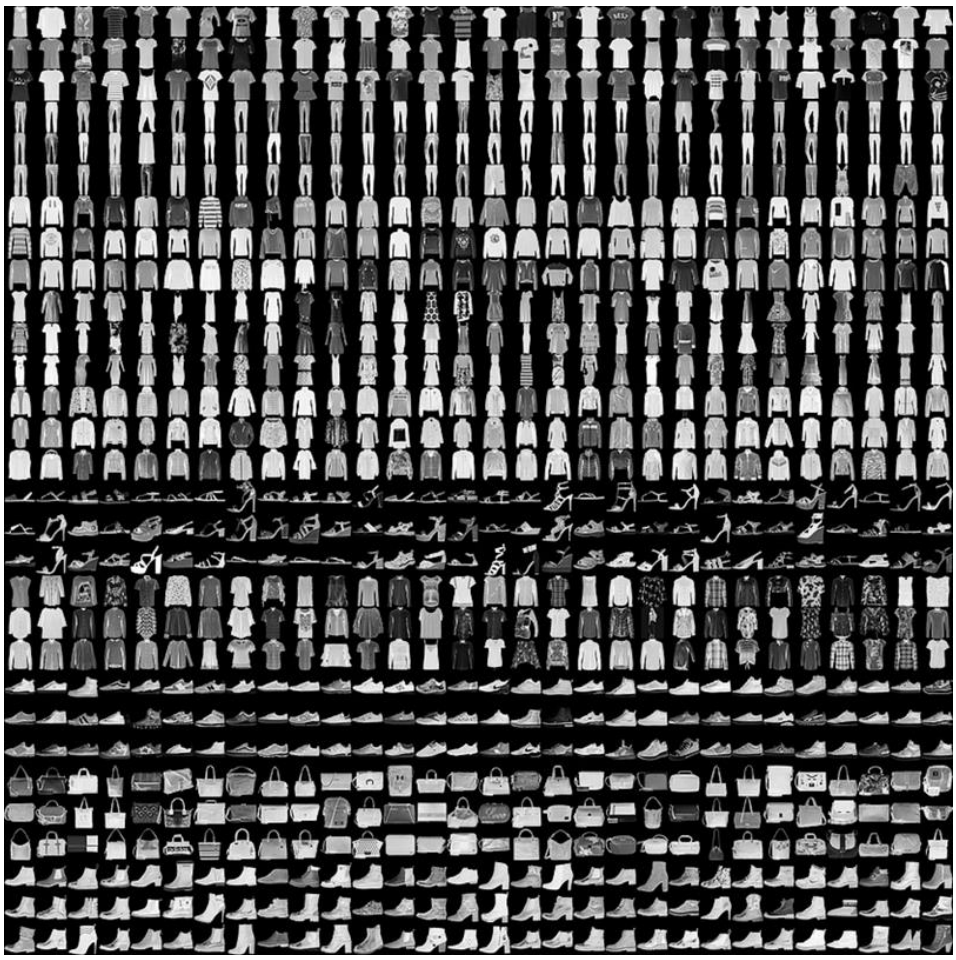

Рисунок 5. DataSet Fashion MNIST [8].

3 Разработка нейронной сети «WardrobeDet»

3.1 Создание «WardrobeDet»

Для начала подключаем нужные нам библиотеки и модули. Numpy и Matplotlib нужны для работы с массивами и визуализации изображения, модуль для работы с Fashion MNIST уже присутствует в Keras. Далее импортируем модель Sequential, которая является аналогом многослойного персептрона в Python. Подключаем тип слоев Dense, который означает, что наши слои будут полносвязными.

Так как набор данных Fashion MNIST уже импортирован, загружаем данный набор. Обучающая выборка составила 60 тыс. элементов, тестовая – 10 тыс.

Перед тем как создать нейронную сеть нужно провести предварительную обработку данных. Была произведена нормализация данных с целью улучшения алгоритма оптимизации, который используется в обучении нейронных сетей. Делим интенсивность каждого пикселя изображения на 255, как показано на рисунке ниже (рисунок 6), чтобы данные на входе в нейронную сеть находились в диапазоне от 0 до 1.

```
x_train = x_train / 255
x_test = x_test / 255
```

```
plt.figure()
plt.imshow(x_train[55555])
plt.colorbar()
plt.grid(False)
```

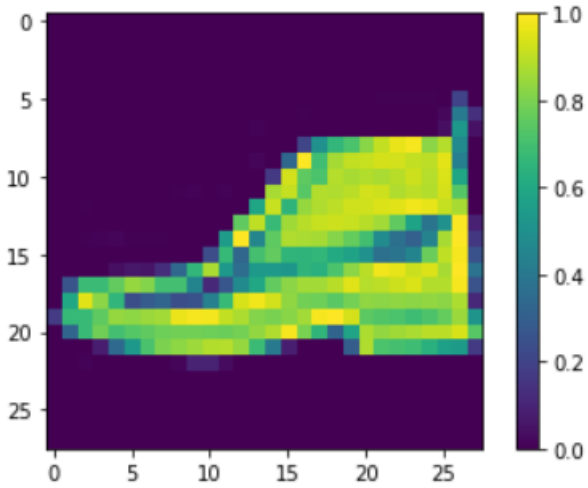

Рисунок 6. Тренировочное изображение после обработки.

Данные уже подготовлены, можно приступать к созданию нейронной сети. В нейросетях основным строительным блоком является слой, и основная часть глубокого обучения состоит в объединении простых слоев. В Keros нейронные сети, как и в целом машинное обучение называются моделями. Архитектура нейронной сети «WardrobeDet» предоставлена на рисунке 7.

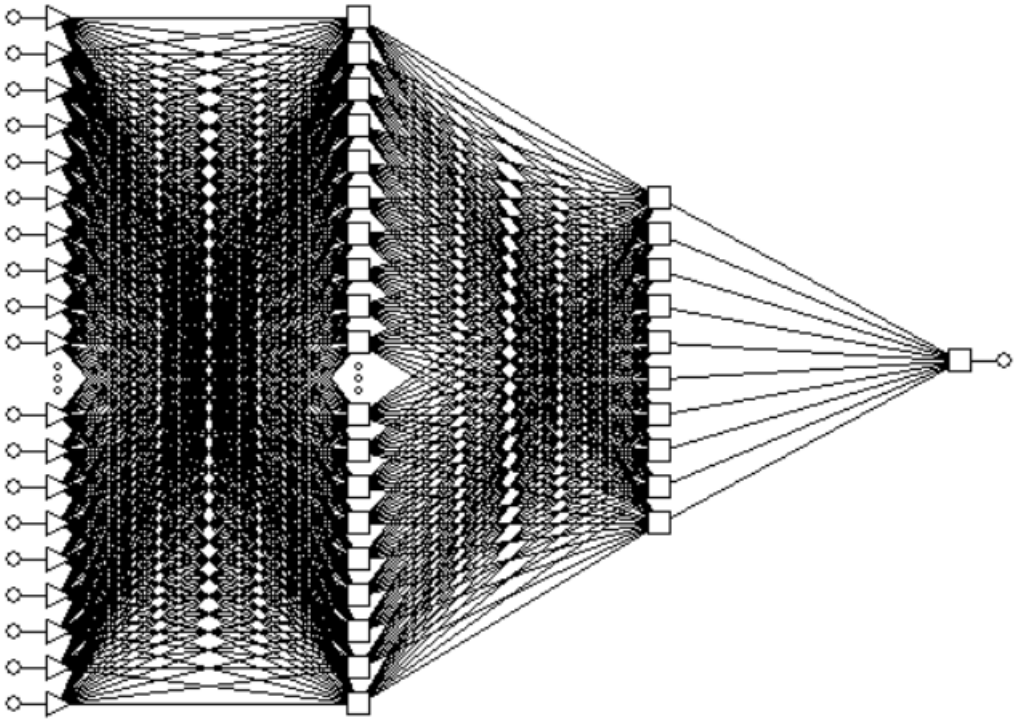

Рисунок 7. Архитектура сети «WardrobeDet»

Она содержит 2 скрытых слоя с 128 и 10 элементами на каждом слое, а также один входной слой с 784 элементами, каждый из которых является пикселем, подаваемого на вход изображения, и один выходной слой с одним элементом. Исполняемый код, отвечающий за создание нейронной сети «WardrobeDet», представлен на рисунке 8.

```
model = keras.Sequential([
    keras.layers.Flatten(input_shape=(28,28,1)),
    keras.layers.Dense(128, activation="relu"),
    keras.layers.Dense(10, activation="softmax")
])
```

Рисунок 8. Создание модели нейронной сети.

Функция активации для первого скрытого слоя relu (рисунок 9), она показала хорошую эффективность в подобных нейронных сетях.

$$f(x) = \max(0, x) = \begin{cases} x_i \; if \; x_i > 0 \\ 0 \; if \; x_i < 0 \end{cases}$$

Рисунок 9. Функция relu.

В качестве функции активации второго слоя была выбрана функция SoftMax, благодаря которой второй слой возвращает массив из 10 вероятностных оценок, сумма которых равна единице. Каждый элемент массива будет содержать оценку, которую указывает вероятность того, что текущее изображение принадлежит одному из 10 классов.

Отметим, что если перед нами задача классификации, то есть определение признака, к которому относится наблюдение, критерием выбора в таком случае является принцип Максимального правдоподобия, т.е. нейронная сеть относит исследуемый случай к той группе, сигнал которой по модулю больше.

Перед обучением модели необходимо сделать настройки, то есть скомпилировать модель, как показано на рисунке ниже (рисунок 10). При компиляции модели указываются параметры обучения, используется оптимизатор SGD. Данный оптимизатор очень популярен для решения подобных задач распознавания изображения с помощью нейронных сетей. В модели будет использоваться категориально-перекрестная энтропия, данная функция ошибки хорошо работает с задачей классификации, когда классов больше двух. Последний используемый параметр качества accuracy, то есть доля правильных ответов.

```
model.compile(optimizer=tf.keras.optimizers.SGD(),loss='sparse_categorical_crossentropy', metrics=['accurasy'])
```

```
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 flatten_1 (Flatten)          (None, 784)               0

 dense_2 (Dense)              (None, 128)               100480

 dense_3 (Dense)              (None, 10)                1290

=================================================================
Total params: 101,770
Trainable params: 101,770
Non-trainable params: 0
```

Рисунок 10. Компиляция модели.

Нейронная сеть обучалась на 10 эпохах алгоритмом градиентного спуска. Обучение выполнялось с помощью функции fit (рисунок 11), в данную функцию передается обучающая выборка, ответы и количество эпох (10).

```
model.fit(x_train, y_train, epochs=10)
```

```
Epoch 8/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.3850 - accuracy: 0.8678
Epoch 9/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.3774 - accuracy: 0.8697
Epoch 10/10
1875/1875 [==============================] - 4s 2ms/step - loss: 0.3691 - accuracy: 0.8726
<tensorflow.python.keras.callbacks.History at 0x7f31cbc40c50>
```

Рисунок 11. Обучение модели.

В конце каждой строки каждой эпохи указывается функция ошибки через параметр loss и точность предсказаний accuracy. С каждой новой эпохой значение ошибки снижается, а точность повышается. С последней эпохой заканчивается обучение сети и её точность составляет приблизительно 90%.

Далее «WardrobeDet» была проверена на тестовой выборке (рисунок 12).

```
test_loss, test_acc = model.evaluate(x_test, y_test)
print('Test accuracy:', test_acc)
```

```
313/313 [==============================] - 0s 1ms/step - loss: 0.4203 - accuracy: 0.8523
Test accuracy: 0.8522999882698059
```

Рисунок 12. Проверка точности предсказания.

Качество предсказания немного ниже, чем было, но все равно достаточно высокое.

После завершения обучения можно использовать нейронную сеть, чтобы предсказывать, что изображено на рисунках. Для этого используется метод predict у данной модели, чтобы предсказывать на тех изображениях, на которых модель обучалась (рисунок 13).

```
predictions = model.predict(x_train)
```

```
predictions[0]
```

```
array([7.6671562e-08, 4.0058370e-09, 1.7837683e-08, 1.9779975e-09,
       4.9734346e-09, 6.5499445e-04, 5.9100438e-07, 1.2859391e-03,
       4.1288513e-05, 9.9801707e-01], dtype=float32)
```

```
np.argmax(predictions[0])
```

```
9
```

```
y_train[0]
```

```
9
```

Рисунок 13. Предсказание тренировочного изображения.

Чтобы не искать какое из выходных данных максимальное, используется функция argmax, она выдает максимальное значение. По итогу вывелся индекс максимального значения (9). Реальный ответ по данному изображению совпал с предсказанием модели, значит модель работает.

3.3 Апробация и тестирование

Теперь используем нейронную сеть для распознавания моделей одежды на собственных изображениях.

Для начала загружаем собственные картинки на платформу google colab (в данном примере загружаем футболку, как показано на рисунке 14).

```
img_path = 'tshirt.jpg'
```

```
Image(img_path, width=150, height=150)
```

Рисунок 14. Изображение футболки.

Далее загружаем картинку в память с помощью инструментов Keros из набора для работы с изображением, указываем размер и модель цветов. Хоть наша картинка большого размера, и она цветная, нам нет необходимости предварительно преобразовывать картинку к нужному и размеру и цвету, за нас делает все Keras.

Далее преобразуем картинку в массив, меняем форму массива в плоский вектор, инвертируем и нормализуем изображение. И после этого запускаем распознавание (рисунок 15).

```
prediction = np.argmax(prediction)
print("Номер класса:", prediction)
print("Название класса:", classes[prediction])

Номер класса: 0
Название класса: футболка
```

Рисунок 15. Результат распознавания футболки.

Те же действия повторяем с изображением сумки (рисунок 16 и 17).

```
img_path = 'bag.jpg'

Image(img_path, width=150, height=150)
```

Рисунок 16. Изображение сумки.

```
prediction = np.argmax(prediction)
print("Номер класса:", prediction)
print("Название класса:", classes[prediction])

Номер класса: 8
Название класса: сумка
```

Рисунок 17. Результат распознавания сумки.

Распознавание в обоих случаях было выполнено правильно. Из этого следует вывод, что созданная нейронная сеть «WardrobeDet» работает корректно для данных, не входящих в обучающую и тестовую выборку.

ЗАКЛЮЧЕНИЕ

В курсовой работе была рассмотрена задача распознавания образов при помощи нейросетевых технологий.

Была осуществлена разработка нейронной сети «WardrobeDet», которая классифицирует детали гардероба по загруженному пользователем изображению.

В итоге разработки удалось добиться высокой точности предсказаний типа детали гардероба.

Стоит подчеркнуть актуальность нейросетевого подхода в распознавании образов. Из-за большого числа данных, содержащихся в отдельно взятом изображении, затруднительно использовать не только нечеткими продукционные системы, но и другие способы решения задач такого класса. Для таких задач выгоднее всего использовать нейросетевой подход, ведь именно при нем мы лишены необходимости в создании множества отдельных правил для классификации образа.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Эндрю Тарсак, Грокаем глубокое обучение / Эндрю Тарсак. СПБ.: Питер, 2021 – 352 с. – ISBN 978-5-4461-1334-7
2. Из чего состоит нейросеть? URL: https://forklog.com/cryptorium/ai/chto-takoe-nejronnaya-set (дата обращения 20.12.2022)
3. Структура и принцип работы полносвязных нейронных сетей URL: https://proproprogs.ru/neural_network/struktura-i-princip-raboty-polnosvyaznyh-neyronnyh-setey (дата обращения 20.12.2022)
4. Нейронные сети, перцептрон – викиконспекты. URL: https://neerc.ifmo.ru/wiki/index.php?title=Нейронные_сети,_перцептрон (дата обращения 20.12.2022)
5. Нейронные сети: распознавание образов и изображений с помощью ИИ. URL: https://center2m.ru/ai-recognition (дата обращения 20.12.2022)
6. Нейронные сети на Python: как всё устроено. URL: https://gb.ru/blog/nejronnye-seti-python/(дата обращения 20.12.2022)
7. Распознавание изображений на Python с помощью TensorFlow и Keras URL: https://evileg.com/ru/post/619/#header_TensorFlow_/_Keras (дата обращения 20.12.2022)
8. Fashion MNIST URL: https://www.kaggle.com/datasets/zalando-research/fashionmnist (дата обращения 20.12.2022)